

Martin Kvam Klefstad

Gjennomgang av friksjonsverdier på bruer under vinterforhold

En vitenskapelig tilnærming for dimensjonering
av minste horisontalkurveradius på norske
vegbruer

Masteroppgave i Bygg- og miljøteknikk

Veileder: Kelly Pitera

Juni 2021



Foto: Statens Vegvesen

Martin Kvam Klefstad

Gjennomgang av friksjonsverdier på bruer under vinterforhold

En vitenskapelig tilnærming for dimensjonering av
minste horisontalkurveradius på norske vegbruer

Masteroppgave i Bygg- og miljøteknikk
Veileder: Kelly Pitera
Medveileder: Arek Zielinkiewicz
Juni 2021

Norges teknisk-naturvitenskapelige universitet
Fakultet for ingeniørvitenskap
Institutt for bygg- og miljøteknikk



Norwegian University of
Science and Technology

Sammendrag

Denne masteroppgaven undersøkte historiske friksjonsmålinger for å evaluere totalfriksjonsverdier på bruer under vinterforhold. Kunnskap om friksjonens oppførsel er viktig ettersom at sidekreftene som kan tas opp gjennom friksjon i kurvebevegelser reduseres når vegene blir glatte. Uventede fall i friksjon kan komme overraskende på sjåfører slik at de ikke rekker å justere hastigheten for å imøtekomme friksjonsforholdene før de er inne i kurva, og dermed øker risikoen for ulykker. Vegbruer har dessverre gode forhold for rimdannelse som tillater at tynne lag med rim legger seg på bruene uten at resten av vegen påvirkes. Norske vegbruer i dag utformes derfor med strengere krav til minste horisontalkurveradius for å redusere sidekreftene som kjøretøyene opplever i kurver, og reduserer dermed behovet for sidefrikksjon i kurver. Likevel er lite kjent om hvilke friksjonsverdier som faktisk opptrer på bruer under vinterforhold.

Oppgaven tok for seg historiske friksjonsmålinger fra 2018 til 2021, målt med kontinuerlige friksjonsmålere, Road analyzer and Recorder versjon 5 og versjon 6, utført av profesjonelle sjåfører hos Statens Vegvesen. Målinger med lufttemperaturer eller vegoverflatetemperaturer lavere enn 1°C ble knyttet til vegbruer og vinterdriftsklasser i Nasjonal VegDataBank ved bruk av databankens API og programmet QGIS. Det ble laget buffersoner rundt bruene og rundt bruenes endepunkter i QGIS for å finne målinger som var mindre enn 100 meter unna bruene i luftavstand. Deretter ble målingene som var utført på bruer analysert mot målingene som var utført på vegstrekningene utenfor bruene.

Analysene vurderte totalfrikksjon avhengig av henholdsvis vinterdriftsklasse, temperatur og luftfuktighet. Førstnevnte inkluderte densitetsfordelinger, 5-persentiler, 15-persentiler, gjennomsnittsverdier og standardavvik. De to resterende avhengighetene ble analysert visuelt gjennom spredningsplott, inkludert trendlinjer for luftfuktighet. I tillegg ble det utført sensitivitetsanalyser av variabler som inngår i dimensjoneringen av minste horisontalkurveradius. Dette inkluderte sidefrikksjonsandeler, sikkerhetsfaktorer for friksjon, og overhøyder. Til slutt ble totalfrikksjonsverdiene fra de historiske målingene vurdert opp mot dagens krav til 50 % økning av minste horisontalkurveradius for å vurdere hvor godt dagens krav fungerer.

Resultatene viste at totalfrikksjonen generelt sett var signifikant høyere på bru enn utenfor bru. Dette gjaldt for 5-persentil, 15-persentil og gjennomsnittlige verdier. Blant vinterdriftsklassene ble det samme observert for vinterdriftsklasse E, mens vinterdriftsklasse B, høy, hadde signifikant lavere totalfrikksjon for 15-persentil og gjennomsnittsverdi. Vinterdriftsklasser med barvegsstrategi hadde både høyere verdier og lavere standardavvik enn vinterdriftsklasser med vintervegsstrategi. Lavest 5-persentilverdi på bru blant vinterdriftsklassene ble registrert for klasse C, med totalfrikksjon lik 0,18. Vegoverflatetemperaturen var stort sett lik på bruene som utenfor bruene. Målinger med høy luftfuktighet hadde en tendens til å ha lavere totalfrikksjon under vinterforholdene. Sensitivitetsanalysen viste at overhøyden var den mest kritiske variabelen når den ble kombinert med lave totalfrikksjonsverdier. Ved plausible avvik i overhøyde kombinert med rim kan behovet for horisontalkurveradius bli mer enn det dobbelte av den minste horisontalkurveradiusen for vanlig veg dersom man skal ivareta sikkerheten ved normalhastigheter.

Behov for minste horisontalkurveradius kan beregnes ved å koble vinterdriftsklasser opp mot vegdimensjoneringsklassene og legge de registrerte 5-persentilene fra de historiske målingene til grunn. En slik tilnærming stemmer ikke så verst overens med dagens krav om 50 % økning av minste horisontalkurveradius. I dette tilfellet kan både vegklasse H5 og H3 redusere kravet til 25 % økning. Likevel er det tvilsomt at denne fremgangsmåten vil redusere ulykkesrisiko så mye som man ønsker. Som oftest er friksjonsforholdene bedre på bruene enn utenfor, og dermed vil trafikantene som oftest justere hastigheten til de gjeldene forholdene lenge før de når bruene. Hovedutfordringen knyttes til tilfeller med rim på bruer uten at forholdene blir dårligere på vegene rundt. For å håndtere dette bør totalfriksjonsverdien til rim på 0,15 legges til grunn for dimensjoneringen av minste horisontalkurveradius på bru. Dette medfører å øke kravet til ca 70—75 % økning av minste horisontalkurveradius for de fleste vegklassene. Vegklasse H5 kan bruke dagens krav til 50 % økning, mens vegklasse H3 kan redusere kravet til 40 % økning.

De historiske målingene som ligger til grunn for resultatene i denne oppgaven har flere utfordringer med å representere norske vinterforhold på bruer bredt nok. Målingene dekker kun tre fulle vintersesonger, mer enn 75 % av målingene er utført i Trøndelag fylke, og mer enn 85 % av alle målingene ble gjennomført mellom klokken 8 og 20. Til gjengjeld dekker disse målingene over 4800 delstrekninger på bruer, og over 19000 delstrekninger utenfor bruer. Dessverre stemmer ikke alle disse delstrekningene overens med faktiske bruer som har blitt undersøkt. I noen tilfeller har bruer som krysser på tvers over den målte vegstrekningen blitt definert som bru på selve målestrekningen. I motsatt tilfelle har også noen bruer som faktisk lå på målestrekningen ikke blitt plukket opp. Metoden med de historiske målingene som ble brukt i denne oppgaven er også lite egnet til å fange opp tilfeller med rimfrost på bruer.

Abstract

This thesis studied historical friction measurements for the purpose of evaluating total friction values on and off bridges during winter conditions. Building an understanding of the friction's behavior is important as the amount of side forces that can be counteracted in horizontal curves is dependent on the available side friction, subsequently increasing the risk of accidents on slippery roads. Unexpected reductions of friction may catch drivers off guard, not giving them time to reduce their speed to match the conditions before they enter a curve. Unfortunately, road bridges have some of the best conditions for the formation of hoar frost, allowing tiny layers of hoar frost to form on bridges without affecting nearby roads. Norwegian road bridges are therefore designed with stricter requirements to the minimal horizontal curve radius in order to reduce side forces in curves, thus reducing the need for side friction in curves. Despite these requirements, little is known about which friction values actually occur on bridges during winter conditions.

The thesis examined historical friction measurements from 2018 to 2021 measured by continuous friction measurers, Road analyzer and Recorder version 5 and version 6, performed by professional drivers at the Norwegian Public Roads Administration. Measurements with air temperatures or road surface temperatures lower than 1°C were connected to road bridges and winter maintenance classes in the National Road DataBase using the database's API and QGIS. To find measurements less than 100 meter away from the bridges, buffer zones were made around the bridges and around the bridges' endpoints in QGIS. Afterwards, the measurements done on bridges were analyzed against the measurements done outside of the bridges on both sides of the bridges.

The analyses consisted of total friction analyzed against winter maintenance classes, temperature and humidity, respectively. The first analysis considered density distributions, 5-percentiles, 15-percentiles, averages and standard deviations. The rest of the analyses were done visually through scatter plots, including trend lines for humidity. Additionally, sensitivity analyses were done for variables affecting the minimum horizontal curve radius. This included shares of side friction, safety factors for friction, and superelevation. Finally, the total friction values from the historical measurements were evaluated against today's requirements for 50 % increase of the minimal horizontal curve radius in order to evaluate how well this requirement works.

The results showed that the total friction, generally speaking, was significantly higher on the bridges as opposed to off the bridges. This was true for 5-percentile, 15-percentile and average values. When total friction was examined for winter maintenance classes, class E had significantly higher values for both of the percentiles and the average, whereas class B, high, had significantly lower values for the 15-percentile and the average value. Winter maintenance classes with strategies allowing no ice nor snow had higher total friction values and less standard deviation than the classes with winter road strategies. The lowest 5-percentile value among the different winter maintenance classes on bridges was registered on class C with a total friction of 0,18. Road surface temperatures were mostly constant, both on and off the bridges. Measurements with higher humidity tended to have lower total friction values during winter conditions. The sensitivity analyses showed that the superelevation was the most critical factor when com-

bined with low total friction. Plausible deviations in superelevation combined with hoar frost would require more than twice the minimum horizontal curve radius in order to upkeep safe driving at normal limit.

The need for minimum horizontal curve radius can be calculated by tying the different winter maintenance classes to the road dimensioning classes, and using the registered 5-percentiles for total friction from the historical measurements. This approach coincides fairly well with the current requirements of 50 % increasement of the minimum horizontal curve radius. The requirements could even be lowered to 25 % for the road classes H5 and H3. However, the friction conditions tend to be better on the bridges as opposed to off the bridges, meaning that if the friction conditions are bad, the drivers are likely to adjust their speed way before they ever reach the bridges. The main problem comes from cases where hoar frost only forms on the bridges, leaving the connecting roads with decent friction conditions. To accommodate these cases, the total friction value of hoar frost of 0,15 should be used as the dimensioning total friction value when calculating the minimum horizontal radius on bridges. This requires about 70-75 % increasements of the minimum horizontal radii for most of the road classes, but road class H5 can stay unchanged with 50 % increasement, and road class H3 can be reduced to 40 % increasement.

The historical measurements, which make up the basis for the results in this thesis, have some challenges with representing Norwegian winter conditions on bridges broadly enough. There were only measurements from three full winter seasons, more than 75 % of the measurements were done in the county of Trøndelag, and more than 85 % of the measurements took place between 8 am and 8 pm. Despite these limitations, more than 4800 measurements were carried out on bridges, with an additionaly 19000 measurements done off the bridges. Unfortunately, not all these measurements coincide with actual bridges that were measured. In some cases, if there was a bridge crossing the road where the actual measurements took place, the measurements were wrongfully attributed to the bridge. In the opposite case, a few bridges were left out despite having measurements done on them. Additionally, using these historical measurements fail to pick up any significant number of bridges with hoar frost formation.

Forord

Denne oppgaven leveres i forbindelse med emnet TBA4940 Veg, masteroppgave, og er en videreføring av oppgaven med samme tittel som ble levert i TBA4541 Veg, fordypningsprosjekt. Oppgaven markerer slutten på fem år med integrert master på studiet Bygg- og miljøteknikk ved NTNU. Oppgaven er skrevet i samarbeid med Statens Vegvesen som har bidratt med kompetanse og forslag til gjennomføring.

En stor takk til Kelly Pitera for god veiledning og oppfølging gjennom hele prosessen fra prosjektoppgave til masteroppgave. Kelly har bidratt med å svare på små og store spørsmål og hjulpet med å rette et kritisk blikk på det som ble gjort.

Takk til alle i Statens Vegvesen som har bidratt med å gjøre oppgaven mulig. Arek Zielinkiewicz spilte en viktig rolle tidlig i samarbeidet og var essensiell i utformingen av prosjektoppgaven. Silje Monika Hjelle Strand har tatt seg av alt av formalia knyttet til oppgaven. Bård Nonstad har bidratt med uunnværlig kunnskap om friksjonsmålinger og kom med forslaget om å benytte ViaTechs friksjonsdatabase til å løse oppgaven. Jan Kristian Jensen stod for den manglende brikken i puslespillet og har bidratt mye mer enn det han klarer selv. Jans forslag om å benytte QGIS og hans ressurser på nett gjorde at friksjonsmålingene kunne knyttes til NVDB.

Det ligger mye arbeid bak alle masteroppgaver, og det er ikke alltid man vet hvor man skal starte eller hvor man skal ende opp. I den forbindelse vil jeg si takk til Alex Klein-Paste og Janne Siren Fjærestad som stilte opp i sonderingsfasen og hjalp til med å sette realistiske forventninger, særlig om rimdannelse.

Jeg vil også gi hedre vennene mine Didrik Frimann Barosso Koren og Andreas Nistad. Ikke bare har de mye av æren for at jeg har trivdes så godt som student i Trondheim, men de har også bidratt direkte til oppgaven. Takk til Didrik som har fungert som sparringspartner for oppgavens programmering, og takk til Andreas for korrekturlesing av oppgaven.

Ikke minst går en stor takk til Marianne Elise Lia som har gjort dagene med arbeid mye mer spennende og innholdsrike enn jeg noen gang hadde sett for meg.

Mine tanker går ut til alle som har mistet noen i trafikken. Jeg håper dette arbeidet kan være et skritt i riktig retning for å gjøre vegene våre tryggere.

Trondheim, juni 2021

Martin Kvam Klefstad

Martin Kvam Klefstad

Innhold

Sammendrag	i
Abstract	iii
Forord	v
1 Introduksjon	1
1.1 Motivasjon	1
1.2 Bakgrunn	2
1.3 Problemstilling og forskningsspørsmål	4
1.4 Avgrensninger	4
2 Litteraturgjennomgang	7
2.1 Ulykker	7
2.1.1 Nullvisjonen	7
2.1.2 Introduksjon til risiko	7
2.1.3 Håndtering av risiko	8
2.1.4 Ulykkesrisiko på bruer	11
2.2 Kurvebevegelser	12
2.2.1 Kjøretøyets bevegelse gjennom kurver	12
2.2.2 Avvik fra ideelle kurvebevegelser	16
2.2.3 Avvik i overhøyde	18
2.3 Friksjon	20
2.3.1 Hva er friksjon?	20
2.3.2 Sidefriksjon på bar veg	21
2.3.3 Friksjonsmålinger	26
2.4 Glatthet	28
2.4.1 Grader av glatt	28
2.4.2 Rimfrost	29
2.4.3 Motvirke glatthet	30
3 Metode	33
3.1 Reproduksjon med vedlagte koder	33
3.2 Oversikt	34
3.3 Del 1: Fremstilling av filer	36
3.3.1 Valg av data	36
3.3.2 Filtrering av data	37

3.3.3	GIS	38
3.3.4	Rensing av datafiler	41
3.4	Del 2: Analyse av datafiler	42
3.4.1	Utvalgte datasett	42
3.4.2	Vinterdriftsklasser	43
3.4.3	Temperatur	44
3.4.4	Luftfuktighet	45
3.5	Del 3: Sensitivitetsanalyse	45
3.5.1	Totalfriksjon og sidefriksjon	45
3.5.2	Sikkerhetsfaktor for friksjon	46
3.5.3	Overhøyde	46
3.6	Del 4: Vurdering av dagens krav	47
4	Resultater	49
4.1	Friksjon på og utenfor bruer under vinterforhold	50
4.1.1	Utvalgte datasett	52
4.1.2	Friksjon avhengig av vinterdriftsklasse	58
4.1.3	Friksjon avhengig av temperatur	68
4.1.4	Friksjon avhengig av luftfuktighet	70
4.2	Sensitivitet til variabler som påvirker dimensjoneringen av minste horisontalkurvatur	71
4.2.1	Totalfriksjon og sidefriksjon	71
4.2.2	Sikkerhetsfaktor for friksjon	74
4.2.3	Overhøyde	77
4.3	Hvor godt treffer dagens krav om 50 % økning av minste horisontalkurveradius?	78
4.3.1	Friksjonens behov for horisontalkurveradius	78
4.3.2	Scenarioer	81
5	Diskusjon	83
5.1	Hvordan varierer friksjonsforholdene på vegstrekninger på og utenfor bruer under vinterforhold?	83
5.2	Hvor sensitivt er behovet for minste horisontalkurvatur etter endringer i parametrene friksjon, sikkerhetsfaktor og overhøyde?	85
5.3	Hvor godt dekker dagens standarder for friksjon og sikkerhetsfaktorer de observerte friksjonsforholdene på bruer under vinterforhold?	87
5.4	Hvilke totalfriksjonsverdier bør ligge til grunn for dimensjonering av minste horisontalkurvatur på norske vegbruer?	88
5.5	Svakheter og alternativer	90
5.5.1	Datagrunnlagets gyldighet	90
5.5.2	Utførelse av metoden	91
5.5.3	Alternative metoder	92
6	Avslutning	95
6.1	Konklusjon	95
6.2	Anbefaling	96
6.3	Videre arbeid	96
	Referanser	99

Vedlegg	103
A: Datafiler med friksjonsmålinger	103
A1: Original fil	103
A2: Behandlet fil	111
B: Koder	113
B1: Filtrering av datagrunnlag	113
B2: QGIS main	116
B3: QGIS hjelpefunksjoner	118
B4: Rensing av filer	127
B5: Lage matrise	128
B6: Analyse	130
B7: Gaulabrua	176
B8: Sensitivitet	182

Kapittel 1

Introduksjon

1.1 Motivasjon

Norske vegbruer er mer utsatte for glatte føreforhold enn resten av vegene, og de har forhøyet ulykkesrisiko under perioder med rimdannelse [1]. Bruer har lav termisk masse som gjør at de holder dårlig på varme, samtidig som de er ekstra utsatt for vind også på undersiden av vegbanen. Uten jordvarme som begrenser varmetapet, og med vindens varmetransport, kan vegoverflaten på bruer bli raskere nedkjølt enn vegoverflaten til den tilhørende vegen. Bruer spenner ofte over vann, noe som kan bidra til å øke luftfuktigheten lokalt. Kombinasjonen av høy luftfuktighet og en vegbane under frysepunktet kan resultere i rimdannelse på bruens vegoverflate uten at den øvrige vegen dekkes av rim. Derfor kan man oppleve helt andre friksjonsforhold på brua enn vegen for øvrig. Disse lokale forskjellene kan komme overraskende på bilister som ikke rekker å tilpasse farten etter forholdene.



brobildet1 (Foto: Faksimile Havarikommisjonen)

Farlig bro kostet et menneskeliv

En person omkom og tre ble skadet i ulykke på denne broen. Myndighetene visste veien var for glatt, men gjorde ingenting.

Får kritikk etter dødsulykke

Statens vegvesen ble varslet om glatt vei, men tok det ikke videre. En omkom og seks ble skadet i ulykken.



Ulykken skjedde da tre vogntog kjørte i rekke inn på Brekke bru i Flåm for tre år siden. Vogntog nummer to mistet veggrepet, fikk slæss og kom over i motgående kjørefelt. To personbiler som kom i motsatt retning hadde ikke mulighet til å stoppe, og kolliderte med vogntoget. Det siste vogntoget som kom bakerst (t.v.) fikk stoppet i tide. Foto: FOTO.B7.NO

Figur 1.1: TV2s [2] og Aftenspostens [3] oppslag etter dødsulykken på Brekke Bru i Flåm.

Dersom ulykkesrisikoen ikke håndteres tilstrekkelig ender man for ofte opp med en tragedie. Dødsulykken på Brekke bru i Flåm i 2007 viste for alvor hvor ille det kunne gå, og som vist i Figur 1.1 vakte ulykken sterke reaksjoner i media i etterkant. Det var nemlig flere bilister som hadde varslet om glatte kjøreforhold til Vegtraffikksentralen flere timer før ulykken, men

ingenting ble gjort for å utbedre forholdene. Hverken entreprenør, byggherre eller trafikanter ble varslet om de glatte forholdene [2].

Som følge av ulykken ble kravene i håndbok N100 *Veg- og gateutforming* oppdatert etter Statens havarikommisjon for transport sine anbefalinger til tiltak. I dagens versjon av N100 står følgende om bruer i temakapittel D.10: "Linjeføringen inntil og over brua vurderes både med hensyn til trafiksikkerhet og estetikk. Minimumskravet til horisontalkurveradius skal økes med 50 % over brua. Dette kravet gjelder ikke for gang-og/eller sykkelbruer" [4]. I praksis innebærer dette å regne ut minste horisontalkurveradius etter Formel 1.1 [5] og deretter øke resultatet med 50 %.

$$R_{h,min} = \frac{V^2}{127 * (e_{maks} + f_k)} \quad (1.1)$$

$R_{h,min}$ = Minste horisontalkurveradius [m]

V = Fartsgrense med fartstillegg og fartsprofiltillegg [km/t]

e_{maks} = Største tillate overhøyde

f_k = Sidefriksjonskoeffisient med sikkerhetsfaktor

127 = Konstant som inneholder tallverdiene til gravitasjonskonstanen og kvadratet av omregningsaktoren fra m/s til km/t

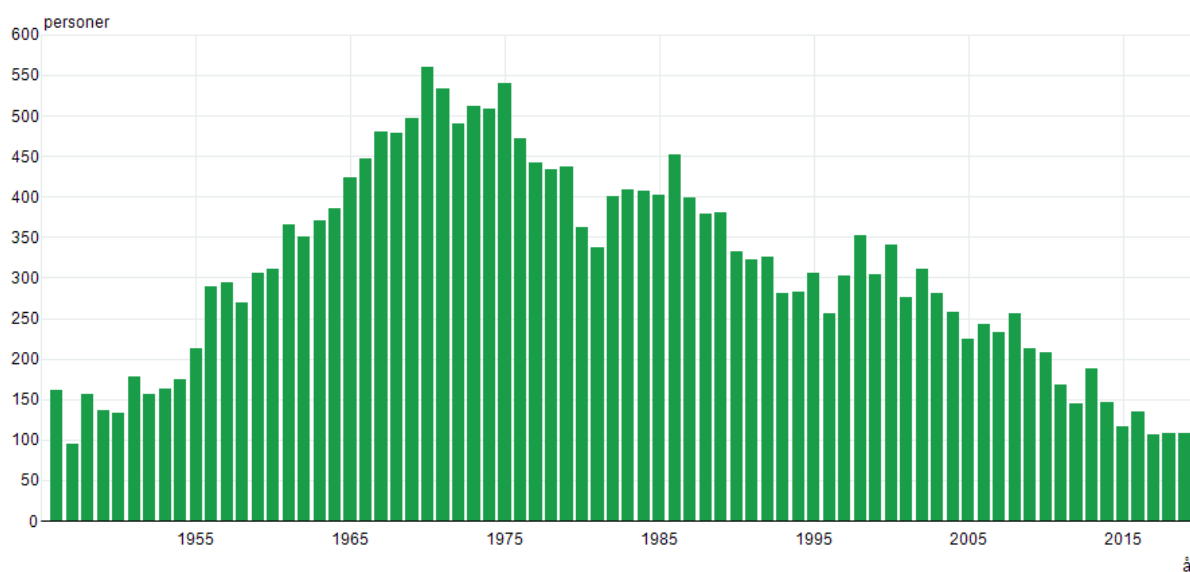
Elvik påpeker at slike endringer i standardene ofte gjøres uten grundige evalueringer [6]. Denne oppgaven streber dermed etter å tilføre forskningsbasert kunnskap om friksjonsforholdene som opptrer på norske vegbruer under vinterforhold. Målet med oppgaven er å bidra til å danne beslutningsgrunnlag for dimensjoneringskrav til bruer.

1.2 Bakgrunn

Antallet årlig drepte i trafikken kan reduseres ved hjelp av systematisk arbeid og sikkerhetsøkende tiltak. Figur 1.2 viser Statistisk Sentralbyrå sin oversikt over antallet drepte i vegtrafikkulykker i Norge hvert år siden 1946 [7]. Til tross for en kontinuerlig økning i trafikkmengden på landsbasis har man klart å redusere antallet drepte i trafikken siden dette ble et ordentlig fokusområde på 70-tallet. Tiltak for å redusere hastighet og sikrere biler får mye av æren for den reduserte ulykkesrisikoen, men det har også foregått mye forskning på veginfrastrukturen.

Friksjonsforhold er et område som påvirker ulykkesrisiko. Friksjon beskriver hvor stor kraft som kan overføres fra et kjøretøy til vegbanen, og den er nødvendig under nedbremsing og for å holde kjøretøyene på vegen under kjøring i kuver [5]. Sidefriksjon inngår direkte som en parameter ved dimensjonering av horisontalkurver som vist i Formel 1.1. For å si noe om friksjonsforhold og ulykkesrisiko er det gjennomgått ulykker registrert hos Vegkart. I perioden 01.01.2010 til 26.08.2019 var det registrert 26'902 bilulykker hos Vegkart [8]. 7'151 av disse bilulykkene, eller 26,6 %, ble kategorisert med føreforholdene snø, delvis snø, isbelagt, eller glatt ellers. Standarden for dimensjonering av veginfrastruktur i dag forutsetter friksjonsforhold tilsvarende våt, men ikke glatt vegbane [5]. Det vil si at over en fjerdedel av ulykkene på norske veger fant sted under friksjonsforhold som vegens fartsgrense ikke var dimensjonert for. Av disse ulykkene var 2'712 klassifisert som ulykker i kurver, tilsvarende 37,9 % av alle ulykkene

Personer drept i vegtrafikkulykker



Kilde: Statistisk sentralbyrå

Figur 1.2: Statistikk over antall drept i vegtrafikkulykker hvert år siden 1946 [7].

som fant sted under vinterforhold. Til sammenligning var 7'348 av ulykkene uavhengig av føreforhold forbundet med kurver, altså kun 27,3 %. Ikke overraskende er andelen bilulykker i kurver betydelig større under forhold med lav friksjon. I et større perspektiv kan man se at 10,1 % av alle ulykkene i perioden 2010 til 2019 fant sted på vinterføre i kurver. Dette er med andre ord en kategori det er verdt å se nærmere på for å redusere antallet ulykker og antallet drept i trafikken.

Friksjonsforhold på bruer under vinterforhold bør gis ekstra oppmerksomhet. Ser man på samme ulykkesperiode for bilulykker som i forrige avsnitt finner man at 295 av ulykkene er registrert på bru. Vinterforhold er registrert på 88 av ulykkene, altså 29,8 %. 37 av disse ulykkene er direkte knyttet opp til ulykker i kurver, tilsvarende 42,0 %. Det vil si at bruer har et noe høyere ulykkesmønster enn øvrige vegstrekninger. Selv om relativt få ulykker skjer på bru, så er andelen ulykker på bru nokså høy tatt i betraktning hvor liten andel bruer utgjør av vegnettet. Det finnes nemlig ca 95'200 kilometer veg i Norge [9], og bare 18'000 bruer som forvaltes av Statens Vegvesen [10]. Med tanke på at kun 110 bruer har en lengde større enn 400 meter, hvor den lengste er 1892 meter, og dermed kun utgjør en brøkdel av vegnettet, så kan man forstå hvorfor det stilles strengere krav til linjeføringen av bruer enn resten av vegnettet i vegstandarden [4].

Hensikten til dagens løsning med økte krav for minste horisontalkurveradius på bru var god, men utførelsen manglet forankring i vitenskapelige undersøkelser. En fast økning på 50 % utover normalkravene for minste horisontalkurveradius betyr at man er 50 % mindre avhengig av overhøyde og sidefriksjon i kurver slik Formel 1.1 viser, men denne økningen rammer vegklassene ulikt. For vegklassen H02 med minste horisontalkurveradius på 125 meter vil 50 % økning resultere i en minste horisontalkurveradius på 187,5 meter. For vegklasse H3, som er dagens strengeste vegklasse, vil 50 % økning endre minste horisontalkurveradius fra 800 meter til 1200 meter. Dette setter begrensninger for linjeføringen og gjør det vanskeligere å tilpasse vegstrekningen til terrenget. Slike begrensninger kan blant annet føre til økte anleggskostnader

og negative konsekvenser for natur og miljø. Derfor vil et større antall fravikssøknader frempro-
voseres, og med begrenset faglig tyngde til å argumentere mot fravikssøknadene viser erfaringer
hos Statens Vegvesen at denne typen fravikssøknader ofte blir godkjent. Hensikten med å øke
horisontalkurveradiusen er å øke sikkerheten under glatte friksjonsforhold, men når det stadig
blir innvilget fravik står man i samme situasjon som før hvor vinterforholdene ikke tas til hen-
syn i dimensjoneringen. Nettopp dette påpekte Statens havarikommisjon for transport eksplisitt
at ikke måtte skje [1]. En ny metode bør utformes for å finne veiledende verdier for minste
horisontalkurveradius, gjerne krav som er dynamiske og avhengige av dimensjoneringsklassen,
samt gi kravene faglig tyngde. Med en god forståelse for kravenes betydning for trafikksikkerhet
kan man behandle de fravikssøknadene som likevel sendes inn på en bedre måte. I denne oppga-
ven foreslås det en tilnæringsmåte hvor man dimensjonerer minste horisontalkurveradius ut
fra opptredende vinterfriksjonsforhold på og ved bruer.

1.3 Problemstilling og forskningsspørsmål

Oppgaven tar for seg problemstillingen **hvilke totalfriksjonsverdier bør ligge til grunn for dimensjonering av minste horisontalkurveradius på norske vegbruer?** For å spesifisere og besvare problemstillingen er følgende forskningsspørsmål utformet:

- Hvordan varierer friksjonsforholdene på vegstrekninger på og utenfor bruer under vinterforhold?
- Hvor sensitivt er behovet for minste horisontalkurvatur etter endringer i parametrene friksjon, sikkerhetsfaktor og overhøyde?
- Hvor godt dekker dagens standarder for friksjon og sikkerhetsfaktorer de observerte friksjonsforholdene på bruer under vinterforhold?

Til å besvare oppgaven brukes historiske friksjonsmålinger fra ViaTechs friksjonsdatabase [11] som har blitt utført av profesjonelle sjåførere ved Statens Vegvesen. De historiske målingene vil bli knyttet til informasjon om bruer og vinterdriftsklasser fra NVDB ved bruk av GIS-verktøy før de blir analysert med ulike fordelinger og statistiske analyser.

Videre vil oppgaven ta for seg avgrensninger. Dette etterfølges av et litteraturstudium som presenterer ulykker og risiko, kurvebevegelser, og friksjon både generelt og spesifikt under vinterforhold. Deretter gjennomgås metoden som er valgt for å løse problemstillingen. Etterpå presenteres resultater for henholdsvis variasjon i friksjon på og utenfor bruer, sensitivitet og virkningsgrad av dagens krav. Nest sist diskuteres resultatene i lys av forskningsspørsmålene, samt svakheter ved oppgaven. Avslutningsvis legges konklusjonen frem sammen med anbefalinger til tiltak og videre forskning.

1.4 Avgrensninger

For å begrense omfanget gjøres det avgrensninger på enkelte punkter som direkte eller indirekte påvirker kraftlikevekten når et kjøretøy kjører gjennom en kurve.

Det første oppgaven ikke tar tilstrekkelig hensyn til er variasjoner mellom teoretisk og reelt opptredende overhøyde. Når man dimensjonerer for minste horisontalkurveradius påvirker

overhøyden direkte hvor stor horisontalkurveradiusen må være for å ta opp sidekrefter. Litteraturstudiet vil gå inn på tidligere forskning som har vist at det oppstår betydelige avvik mellom teoretisk overhøyde og reelt opptredende overhøyde etter som en vegstrekning eldres. I denne oppgaven brukes den teoretisk høyeste overhøyden som vegklassen kan bygges med, men for å oppnå bedre sikkerhet burde fornuftige verdier for lavere, reelle overhøyde benyttes. Oppgaven tar i utgangspunktet ikke for seg undersøkelser på hvordan disse verdiene burde tallfestes, men demonstrer utfallet av lavere opptredende overhøyder gjennom sensitivitetsanalyser.

Det andre oppgaven ikke undersøker grundig nok er hvor stor sidefriksjonen reelt sett er. Tilsvarende som for overhøyde påvirker sidefriksjonen motstandsevnen mot sidevegskrefter. Oppgaven bruker målinger for totalfriksjon og antar at dagens metoder for beregning av sidefriksjon som andel av totalfriksjon er korrekt. Som litteraturstudiet vil utforske nærmere så er verdiene som brukes for dette i Norge ofte annerledes enn verdiene som brukes i andre land. Siden verdiene for sidefriksjonsandelene indirekte påvirker dimensjoneringen av minste horisontalkurveradius burde også dette fastslås med større nøyaktighet. Tilsvarende som for overhøyden vil det også bli benyttet sensitivitetsanalyser for å vise hvilke konsekvenser ulike sidefriksjonsandeler vil medføre.

Til slutt må manglende bidrag fra ytre krefter slik som vind trekkes frem. Kryssvind vil bidra til å skape sidekrefter som må tas opp gjennom overhøyde og sidefriksjon. Bruer er typisk mer utsatt for vind enn det øvrige vegnettet, og sterke vindkast kan gi betydningsfulle sidekrefter særlig på større kjøretøyer med store sideflater. Oppgaven kommer ikke til å ta hensyn til ytre sidekrefter.

Kapittel 2

Litteraturgjennomgang

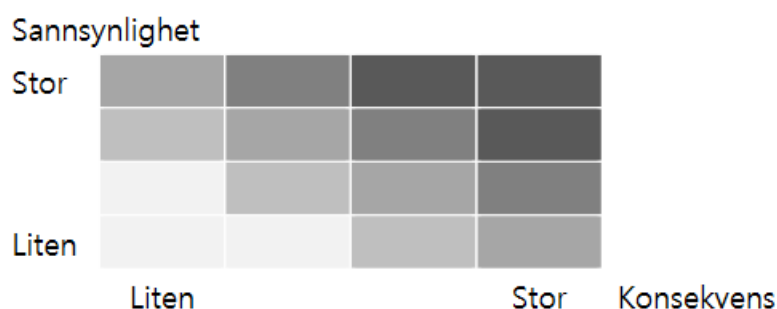
2.1 Ulykker

2.1.1 Nullvisjonen

I nasjonal transportplan 2022-2033 dedikeres et eget kapittel til nullvisjonen [12]. I stortingsmeldingen skrives det følgende: “Regjeringen har høye ambisjoner for transportsikkerheten, og legger visjonen om et transportsystem der ingen blir drept eller hardt skadd, nullvisjonen, til grunn”. Nullvisjonen baserer seg på tre grunnpillarer; etikk, vitenskapelighet og ansvar. Førstnevnte er selvforklarende med at hvert liv er unikt og uerstattelig. Fra et vitenskapelig standpunkt kjenner man menneskets begrensninger. Derfor skal vegtrafikksystemet beskytte trafikantene mot alvorlige konsekvenser av normale feilhandlinger. Alle som deltar i trafikken, enten direkte eller indirekte, har et delt ansvar for å ivareta sikkerheten. Stortingsmeldingen påpeker også at transportsystemet skal ha innebygde barrierer for å hindre ulykker eller begrense konsekvensen. På denne måten legger nullvisjonen føringen for alt arbeid med utformingen av veger.

2.1.2 Introduksjon til risiko

En vanlig tolkning av risiko er å betegne den som en kombinasjon av sannsynlighet og konsekvens [13]. I sannsynlighetsbegrepet inngår frekvens eller hyppighet, sett i sammenheng med eksponering. De forventede risikoene kan fremstilles som en risikomatrix. Håndbok V120 benytter risikomatrixen i Figur 2.1 for å illustrere hvordan risikoen vurderes for ulike kombinasjoner av sannsynlighet og konsekvens [5]. Et alternativt syn på risiko er knyttet til usikkerheten rundt utfallet til en hendelse [14]. I disse hendelsene er det noe av menneskelig verdi som står på spill. En slik definisjon kan være nyttig når flere faktorer spiller inn på hendelsen, og særlig dersom det er snakk om alvorlige ulykker med lav sannsynlighet. Et viktig moment med risiko er at risiko ikke er utelukkende negativt. Uten å ta risiko får man heller ingen gevinst. Nullvisjonen kan enkelt realiseres ved å fjerne all risiko, men dette innebærer også å stoppe med transport.



Figur 2.1: Håndbok V120s risikomatrise [5].

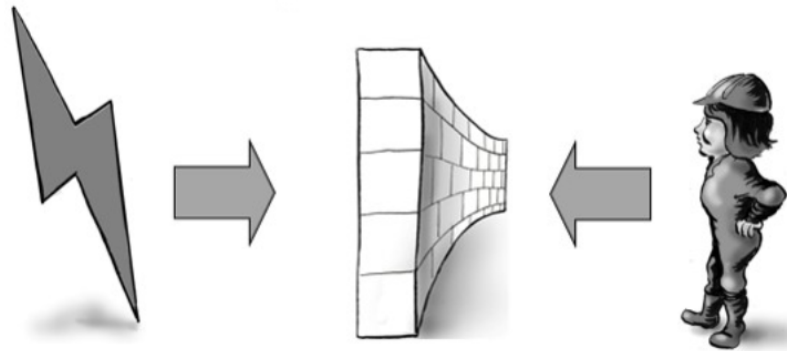
Uansett hvilken definisjon man legger til grunn så forblir risiko en subjektiv verdi. Hvor stor risikoen oppfattes til å være av én person vil kunne variere betraktelig fra en annen persons oppfatning. Dette fenomenet omtales som risikopersepsjon og vil styre hvordan enkeltpersoner reagerer på risikofulle hendelser. Det viser seg også at sjåførere vurderer sannsynligheten for at de selv utsettes for en ulykke som lavere enn at andre sjåførere opplever ulykker [15]. Hvordan risikoen kommuniseres til sjåførene vil også være med på å styre atferden deres. Eksempelvis vil snø i vegbanen kunne oppfattes på lang avstand og effektivt kommunisere at det er en farekilde tilstede. Tynne hinner av is er vanskeligere å oppdage og gjør at sjåførene ikke nødvendigvis klarer å anslå den reelle risikoen før de selv utsettes for den.

2.1.3 Håndtering av risiko

Håndtering av risiko i transportsammenheng kan sees som synonymt med å hindre ulykker. For å komme med tiltak må man også kjenne til hvordan ulykkene kan oppstå. I ulykkesteorien finnes det utallige modeller som forklarer hva som leder til ulykker på generelt basis. Her vil tre teorier kort presenteres slik de beskrives av Kjellén og Albrechtsen [13]. Den første modellen er utviklet av *The International Loss Control Institute* og går under navnet ILCI modellen. Da beskrives ulykker som en domino-effekt hvor en årsak leder til den neste før det til slutt bunnur ut i en ulykke. For å hindre ulykker trenger man bare å fjerne én årsak. Den neste modellen tilhører en samlebetegnelse som kalles prosessmodeller. Her skilles det mellom hendelser som fører til at man mister kontroll og underliggende årsaker som bidrar til kontrolltapet. For å hindre ulykker må man ta grep i de påvirkende faktorene og de underliggende årsakene. Siste modell beskriver ulykker som energi på avveie, og går under det passende navnet energimodellen. Ulykker oppstår når for mye energi, gjerne i form av bevegelse, løper løpsk og skader mennesker. For å hindre ulykker må man kontrollere energioverførselen mellom energikildene og menneskene.

Energimodellen er godt egnet til bruk i transportsituasjoner da den stemmer godt overens med regjeringens krav om innebygde barrierer slik det ble påpekt i Seksjon 2.1.1. Haddon la frem 10 strategier han kallte for barrierefunksjoner for å forebygge skade i 1980 [16]. Barrierefunksjonene er generelle tiltak som realiseres ved å innføre spesifikke tiltak. De spesifikke tiltakene kalles barriere-elementer, og flere ulike barriere-elementer kan realisere de samme barrierefunksjonene. Barrierefunksjonene deles inn i 3 hovedkategorier etter hvordan de griper inn for å hindre at energien fører til skade. Man kan enten rette tiltak mot selve energikilden, prøve å hindre energien i å nå mennesker, eller rette tiltakene mot menneskene. Alle barrierenefunksjonene fordelt i riktig kategori er vist i Figur 2.2. Det er sjeldent at alle barrierefunksjonene kan

realiseres samtidig. For en transportsituasjon vil det naturligvis være en hastighet som skaper kinetisk energi, noe som utelukker den første barrierefunksjonen på lista som ønsker å hindre at det er energi tilstede. Alternativt kan det være aktuelt å begrense mengden energi ved å redusere hastigheten. For vinterforhold vil den mest aktuelle barrierefunksjonen ofte være å forhindre ukontrollert utløsning av energi ved å sikre nødvendig friksjon mellom kjøretøyene og vegbanen.



<p>Relatert til energikilden:</p> <ol style="list-style-type: none"> 1. Forhindre oppbygging av energi 2. Modifisere egenskapene til energien 3. Begrense mengden energi 4. Forhindre ukontrollert utløsning av energi 5. Endre hastighet eller fordeling av energi-overførsel 	<p>Relatert til å skille energikilde fra menneske:</p> <ol style="list-style-type: none"> 6. Separere i tid eller rom 7. Separere med fysiske barrierer 	<p>Relatert til mennesket:</p> <ol style="list-style-type: none"> 8. Gjør mennesket mer motstandsdyktig mot energioverføring 9. Begrense utvikling av skade 10. Stabilisere, reparere og rehabilitere
---	---	--

Figur 2.2: Haddons 10 strategier for å forebygge skader [16]

Ikke alle barriere-elementene er like gode til å realisere barrierefunksjonene. Et barriere-element kan enten være aktivt eller passivt i måten den griper inn i energioverførselen [13]. Aktive barriere-elementer skiller seg fra de passive ved at de krever et kontrollsystem som utløser dem. Et klassisk eksempel på et aktivt barriere-element i transportsammenheng er airbag. Kontrollsystemet kan også være et menneske som observerer et avvik og reagerer deretter, for eksempel ved å trykke på bremsen. De passive barriere-elementene er i utgangspunktet alltid til stede og klare for å påvirke energioverførselen. Ser man tilbake til situasjonen med vinterforhold kan aktive barriere-elementer for eksempel være å redusere hastigheten til kjøretøyene eller å brøyte og salte vegene. Et passivt barriere-element kan være veger med store horisontalkurveradier slik at sidefriksjonsbehovet blir mindre. Ved å bruke flere barrierer samtidig gir man forsvar i dybden og øker sannsynligheten for at alt går bra selv om noen barrierer svikter [13].

Jones-Lee og Aven presenterer to ulike prinsipper for å avgjøre hvordan tiltak bør prioriteres [17]. Kost-nytte-prinsippet baserer seg på at nytten av et tiltak skal overstige kostnadene, og fortrinnsvis skal ikke tiltaket gjennomføres dersom det medfører mer kostnad enn nytte. Prinsippet brukes også til å prioritere mellom ulike prosjekter når det er begrenset med økonomiske midler til å gjennomføre alle prosjektene. Det andre prinsippet er ALARP-prinsippet, *As Low*

As Reasonably Practicable, som sier at man skal tilstrebe å redusere risikoen så mye som mulig så lenge kostnaden ikke blir vesentlig større en nytten. Begge prinsippene medgir at det finnes en grense for når man har gjort nok, men grensen er mye tydeligere definert for kost-nytte-prinsippet. Ved å følge det første prinsippet vil det ofte være akseptabelt å ikke gjennomføre tiltak, mens det andre prinsippet krever som oftest at risikoen reduseres noe. For trafikksikkerhetstiltak anses nytten som oftest å være reduserte skadestandarder. Tryggere veginfrastruktur vil ha færre ulykker, og avhengig av ulykkesalvorlighet vil samfunnet spares for store kostnader dersom de unngås. Dersom en person blir skadet vil vedkomme trenge hjelp fra helsevesen og behandling, samtidig som nedsatt funksjonsevne vil redusere hvor mye nytte personen kan bringe til samfunnet i fremtiden. I tillegg til ulempene knyttet til personen som skades vil også pårørende bli påvirket og bringe redusert nytte til samfunnet. For å tallfeste disse skadestandardene har Transportøkonomisk Institutt utviklet følgende verdier avhengig av alvorlighetsgrad [18]:

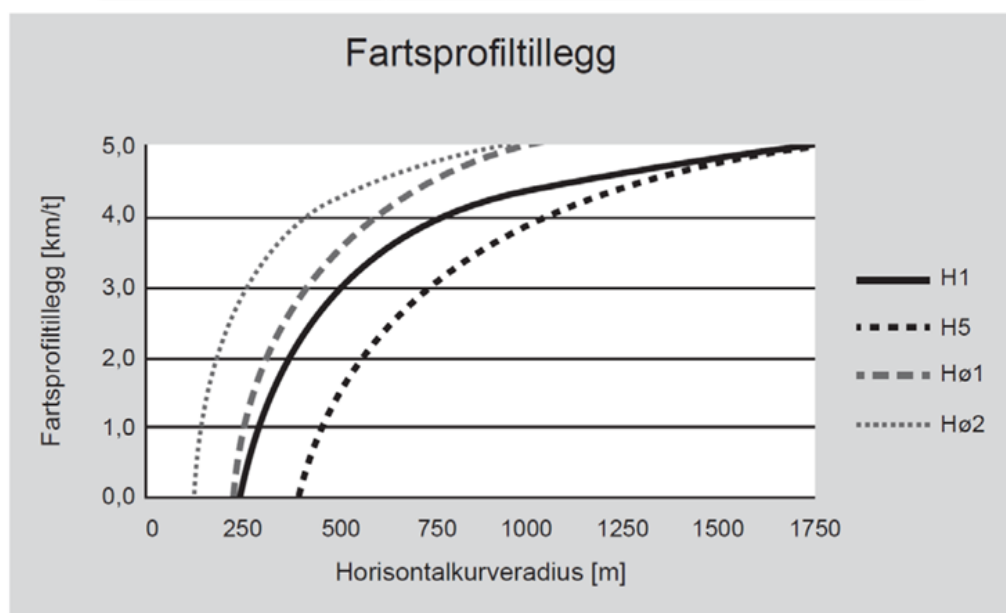
- Dødsfall: 42'181'011 kr
- Hardt skadde 10'830'382 kr
- Lettere skadde 719'312 kr

Jones-Lee og Aven argumenterer samtidig for at det finnes 4 situasjoner hvor kost-nytte prinsippet kan følge ALARP-prinsippet. Den første situasjonen gjelder tilfeller hvor nytten ikke representeres godt nok og det kan være ekstra bekymringer i samfunnet. Den andre situasjonen gjelder tilfeller med høy usikkerhet, mens den tredje situasjonen gjelder tilfeller hvor risikoen er veldig høy i utgangspunktet. Den fjerde og siste situasjonen er en respons på tilfeller hvor kostnadene overdrives for å overskygge nytten i håp om å slippe tiltak. Selv om sikkerhetstiltak på norske veger i utgangspunktet skal følge kost-nytte-prinsippet, viser det seg at tiltakene ofte er konsekvente med en uformell bruk av ALARP-prinsippet [19].

Risikohåndtering starter allerede før prosjekteringsfasen. For ulike trafikkmengder og fartsgrenser benyttes ulike vegklasser til dimensjonering av veger. Alle vegklassene baserer seg på de samme grunnparameterne, men strengere vegklasser har høyere krav til sikkerhet for enkelte av de dimensjonerende verdiene. I tråd med risikomatriksen som ble introdusert i Figur 2.1 vurderes risikoen til å være høyere for de strengeste vegklassene, noe som krever større innsats for å kontrollere risikoen. I praksis gjøres dette gjennom fartstillegg, sikkerhetsfaktorer for friksjon, og fartsprofiltillegg [5]. Hvilke verdier som benyttes for de ulike vegklassene er oppsummert i Figur 2.3. De to første tiltakenes primærfunksjoner er å tillate høyere hastigheter enn fartsgrensen tilsier gjennom kurvene, eller sikre trygg ferdsel selv om friksjonsverdien oppleves lavere enn det man forventer på en bar veg. Fartsprofiltillegget fungerer på samme måte som fartstillegg, men er mer enn respons på at trafikantenes hastighet øker av naturlige årsaker etter hvert som horisontalkurveradiene øker og blir rettere. Målet med sikkerhetstiltakene er at veggeometrien skal tillate trygg ferdsel med litt slingringsmann for personbiler som kjører i dagslys på en våt, men ellers ren og isfri vegbane [5].

ÅDT	0-4 000		4 000 – 6 000		6 000 – 12 000		>12 000
Fartsgrense	60	80	60	80	60	90	110
Nasjonale hovedveger		H1		H1		H5	H3
Øvrige hovedveger	Hø2	Hø1	Hø2		Hø2		

Fartstillegg = 0	Sikkerhetsfaktor -friksjon 1,0
Fartstillegg = 5	Sikkerhetsfaktor -friksjon 1,0
Fartstillegg = 5	Sikkerhetsfaktor -friksjon 1,1
Fartstillegg = 10	Sikkerhetsfaktor -friksjon 1,1



Figur 2.3: Håndbok V120s metoder for å håndtere risiko for ulike vegklasser [5].

2.1.4 Ulykkesrisiko på bruer

Elvik et al. tar til ordet for å se på bruer som egne elementer under ulykkesanalyser [20]. Begrunnelsen for dette er at bruer skiller seg mye fra det øvrige vegnettet, for eksempel ved at bruer utsettes for mer vind, blir fortere glattere, og har begrenset med plass til å gjøre unnamanøvere. Modeller som ikke skiller mellom bruer og veger kan ha vansker for å oppdage forskjeller på sikkerhet mellom bruer og veger. Dette kommer av at bruene utgjør mye kortere distanser enn vegene. Artikkelen benytter seg av en negativ binomisk regresjonsmodell for å identifisere faktorer som kan knyttes til ulykker på bruer. Analysen inkluderer ulykker som fant sted på bruene og ulykker som fant sted på tilfartene 50 meter ut fra brua. Fra analysen ble det funnet at bruer med høyere trafikkmengder hadde flere ulykker, men ulykkesfrekvensen var lavere per million kjøretøykilometer. Tilsvarende hadde lengre bruer flere ulykker, men lavere ulykkesfrekvens per million kjøretøykilometer. Fra denne analysen konkluderes det med at trafikkmengde var den viktigste faktoren for ulykkesrisiko, og at korte bruer har høyere risiko for ulykker enn lange bruer. Artikkelen gikk også gjennom ulykkesgranskninger publisert av Statens havari-kommisjon for transport, hvor vegelementene ble klassifisert etter horisontalkurvatur. Skillet gikk mellom rette segmenter, segmenter med kurver, og segmenter med krappe kurver lavere

enn 150 meter radius. Omtrent halvparten av ulykkene som ble undersøkt fant sted på et segment som var kurvet, enten på tilfarten av brua eller på selve brua. Den vanligste ulykkestypen var utforkjøring før selve brua. Artikkelen foreslår tiltak for å forbedre overflatefriksjon eller implementering av riktig autovern.

Sagberg et al. undersøkte eneulykker på bruer for å vurdere ulykkesrisikoen på inngangssoner, utgangssoner og midtsoner [21]. Inngangssonene defineres som de første 50 meterne et kjøretøy bruker før de når brua, og tilsvarende med utgangssoner som de siste 50 meterne etter brua har blitt kjørt over. Midtsonene deles inn i flere mindre soner avhengig av bruens lengde. Undersøkelsen inkluderte alle eneulykker med personskafer som fant sted i perioden 2010 til 2016 på bruer, inkluderte inngangssoner. Resultatene viste at det var signifikant flere ulykker på inngangssonene og utgangssonene enn de midtstilte sonene, og at ulykkesrisikoen var størst for inngangssonene. Ser man på de midtstilte sonene var det flest ulykker på den første sonen på brua etter inngangssonen. Inngangssonene til korte bruer hadde signifikant flere ulykker enn vegenettet forøvrig.

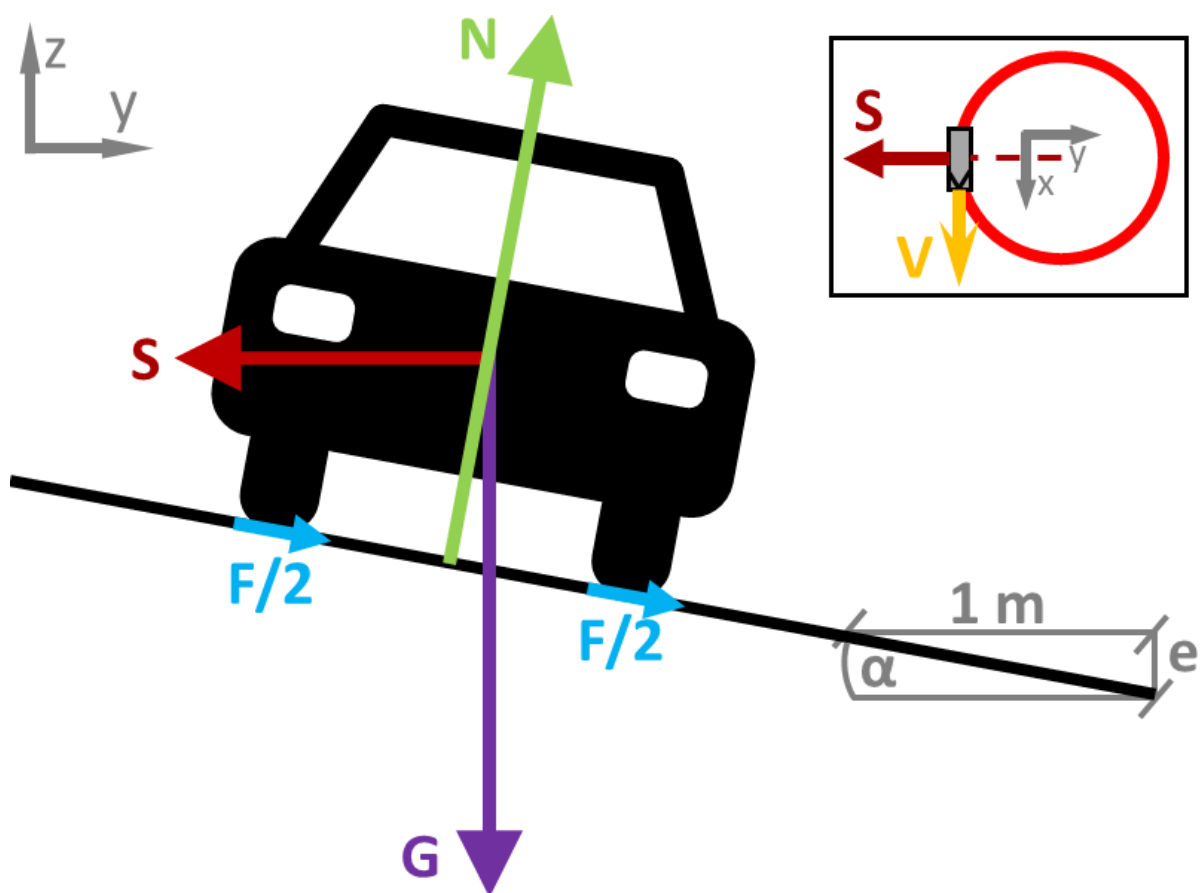
Artiklene underbygger dermed påstanden om at bruer bør vurderes annerledes for risiko enn øvrige veger, slik som denne oppgaven også ønsker å gjøre. I tillegg gjør artiklene det tydelig at strekningene rett før brua skiller seg ut. I oppgaven vil det bli undersøkt om dette kan ha en sammenheng med de opplevde friksjonsforholdene.

2.2 Kurvebevegelser

2.2.1 Kjøretøyets bevegelse gjennom kurver

Tidligere NTNU Professor Hovd beskrev horisontalkurvaturen for studenter i emnet TBA 4201 *Veg og miljø* gjennom et notat fra 2014 [22]. Hovds beskrivelse av sirkelkurven er lagt til grunn for alt som presenteres i denne seksjonen.

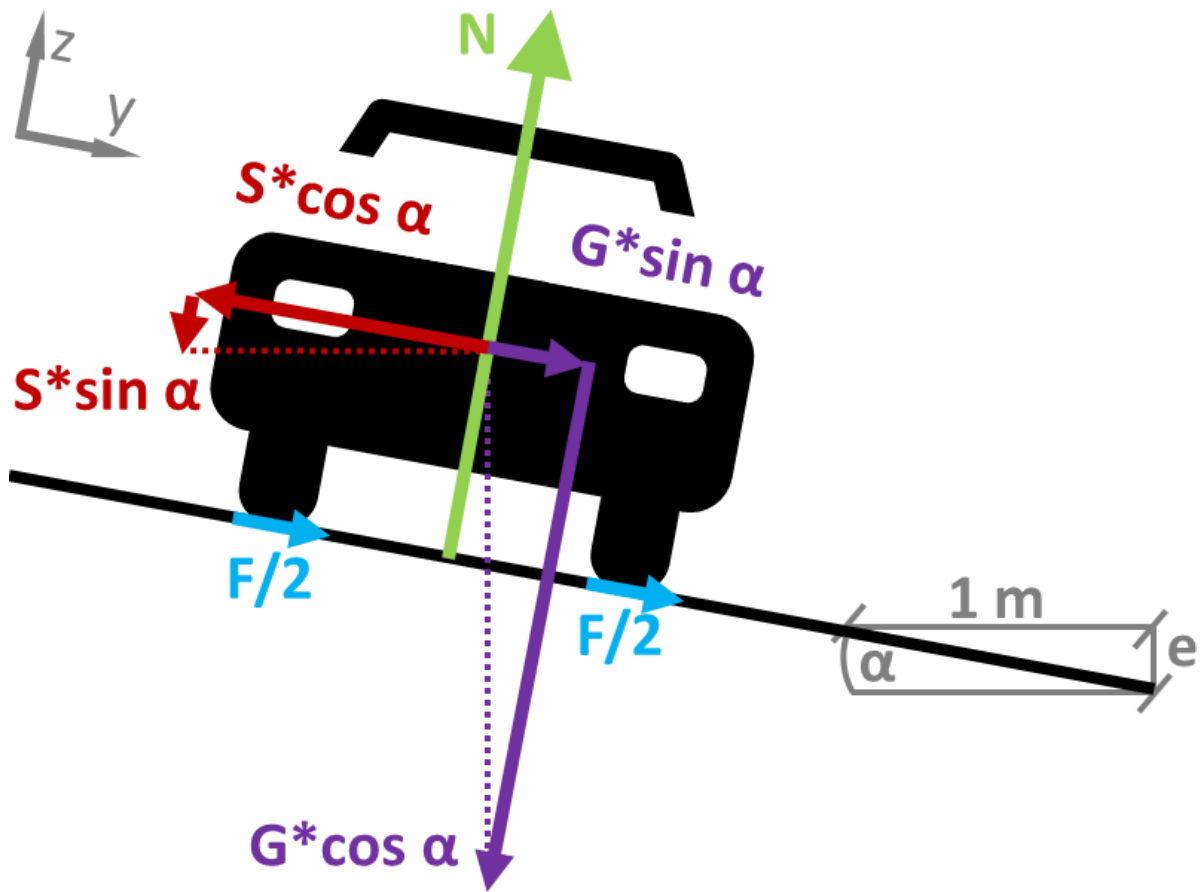
Kreftene som virker på kjøretøy i kurver er illustrert i Figur 2.4. Øverst i høyre hjørne vises det hvordan hastigheten skaper sentrifugalkrefter i sirkelbevegelser for kjøretøyer sett ovenfra. Resten av figuren viser kreftene for kjøretøyer sett forfra. Størrelsene til normalkreftene og gravitasjonskreftene vil avhenge av kjøretøyenes masser, mens sentrifugalkreftene i tillegg vil påvirkes av kjøretøyenes hastigheter og kurvenes horisontalradier. Friksjonskreftene styres av størrelsene til normalkreftene og friksjonskoeffisienter, og sidefriksjonskreftene vil motvirke bevegelser på tvers av kjøreretningen. Friksjonskraften fungerer som en respons på bevegelser og arbeider kontinuerlig for at objektene skal stå i ro. Slik figuren er tegnet opp vil sentrifugalkreftene arbeide for å dytte kjøretøyet ut av kurven, mens friksjonskreftene påfører en motkraft som dytter kjøretøyet inn i kurven. Ved å gi vegen en overhøyde, en helning på tvers av kjøreretningen, vil normalkreftene bidra til å skape krefter inn mot kurven som reduserer behovet for friksjonskrefter. For lave hastigheter kan sentrifugalkreftene bli lavere enn det horisontale bidraget fra normalkreftene. I slike tilfeller vil friksjonskreftene arbeide i motsatt retning av pilene på figuren for å hindre bevegelser inn mot kurven.



Figur 2.4: Krefter som virker på et kjøretøy i en kurve.

- $N = \text{Normalkraft}$
- $S = \text{Sentrifugalkraft}$
- $F = \text{Friksjonskraft}$
- $G = \text{Gravitasjonskraft}$
- $V = \text{Hastighet}$

Normalkreftene og friksjonskreftene kan beskrives med utgangspunkt i sentrifugalkreftene og gravitasjonskreftene. Ved å rotere referanseplanet til å matche helningen til vegen kan sentrifugalkraften og gravitasjonskraften til et kjøretøy dekomponeres som vist i Figur 2.5. Deretter kan normalkreftene og friksjonskreftene beskrives med de dekomponerte vektorene til sentrifugalkreftene og gravitasjonskreftene etter henholdsvis Formel 2.1 og Formel 2.2. Sentrifugalkreftene og gravitasjonskreftene som inngår i de to foregående formlene kan igjen beskrives etter henholdsvis Formel 2.3 og Formel 2.4.



Figur 2.5: Krefter i planet til vegens helning på tvers av kjøreretning som virker på et kjøretøy i en kurve.

$$N = S * \sin \alpha + G * \cos \alpha \quad (2.1)$$

$$F = S * \cos \alpha - G * \sin \alpha \quad (2.2)$$

$$S = \frac{m * V^2}{R_h} \quad (2.3)$$

$$G = m * g \quad (2.4)$$

I Figur 2.5 er friksjonskraften tegnet opp som det nødvendige behovet for å motvirke sentrifugalkraftene. Behovet for sidefriksjon er noe mindre enn den dekomponerte sentrifugalkraften siden gravitasjonen bidrar litt til sidekrefter inn mot kurven. For at kraftlikevekten skal opprettholdes kan ikke det nødvendige behovet for sidefriksjon overstige den tilgjengelige sidefriksjonen som kan hentes fra normalkraften kombinert med sidefriksjonskoeffisienten. I Formel 2.5 er den beskrevne situasjonen satt opp som en ulikhet.

$$F \leq f_k * N \quad (2.5)$$

Videre i Formel 2.6 er de forrige formlene satt inn i ulikheten, før sidefriksjonskoeffisienten isoleres på venste side i Formel 2.7.

$$\left(\frac{m * V^2}{R_h}\right) * \cos \alpha - m * g * \sin \alpha \leq f_k * \left[\left(\frac{m * V^2}{R_h}\right) * \sin \alpha + m * g * \cos \alpha\right] \quad (2.6)$$

$$f_k \geq \frac{\left(\frac{m * V^2}{R_h}\right) * \cos \alpha - m * g * \sin \alpha}{\left(\frac{m * V^2}{R_h}\right) * \sin \alpha + m * g * \cos \alpha} \quad (2.7)$$

I overgangen mellom Formel 2.7 og Formel 2.8 benyttes det at $\tan \alpha$, altså tangens av vinkelen på helningen til veggen, er det samme som motstående katet delt på hosliggende katet. Som man ser i Figur 2.5 tilsvarer dette én gang overhøyde per meter, og dermed blir tangensuttrykket redusert til kun overhøyden, e . I tillegg sløyfes massen fra uttrykket da den inngår i alle ledd.

$$f_k \geq \frac{\left(\frac{V^2}{R_h * g}\right) - e}{\left(\frac{V^2 * e}{R_h * g}\right) + 1} \quad (2.8)$$

Uttrykket $(V^2 * e) / (R_h * g)$ i Formel 2.8 vil bli forsvinnende lite når det settes inn fornuftige tallverdier for variablene. Ved å forenkle uttrykket med antagelsen om at denne delen blir 0 gjør det at hele nevneren i brøken kan sløyfes. Dette medfører at formelen som beregner hvor stor sidefriksjonskoeffisient som kreves i en kurve kan skrives med SI-enheter på formen til Formel 2.9. Tilsvarende beregning kan gjøres for å anslå minste nødvendige sidefriksjonskoeffisient for å hindre glidning inn mot senteret i en kurve. For slike tilfeller ender man opp med motsatt fortegn på leddene i Formel 2.9 siden friksjonskreftene virker i motsatt retning.

$$f_k \geq \left(\frac{V^2}{R_h * g}\right) - e \quad (2.9)$$

I stedet for å utelukkende bruke SI-enheter som Formel 2.9, byttes enheten til hastighet fra m/s til km/t . Siden man regner med kvadratet av hastigheten må også omregningsfaktoren opphøyes i andre. Gravitasjonskonstanten er slått sammen med omregningsfaktoren til hastigheten for å gi én konstant i uttrykket. Dette resulterer i konstanten 127 som illustrert i Formel 2.10

$$f_k \geq \left(\frac{V^2}{R_h * 9,81 * 3,6^2}\right) - e = \left(\frac{V^2}{127 * R_h}\right) - e \quad (2.10)$$

Den endelige formen for uttrykket, som også er formelen som brukes i håndbok V120 [5], vises i Formel 2.11. Horisontalkurveradiusen er isolert i uttrykket, og det er satt likhetstegn for å beregne den minste mulige verdien som tillates. Derav legges det til suffikser for horisontalkurveradius og overhøyde for å indikere at det er snakk om henholdsvis minimums- og maksimumsverdier.

$$R_{h,min} = \frac{V^2}{127 * (e_{maks} + f_k)} \quad (2.11)$$

$R_{h,min}$ = Minste horisontalkurveradius [m]

V = Fartsgrense med fartstillegg og fartsprofiltillegg [km/t]

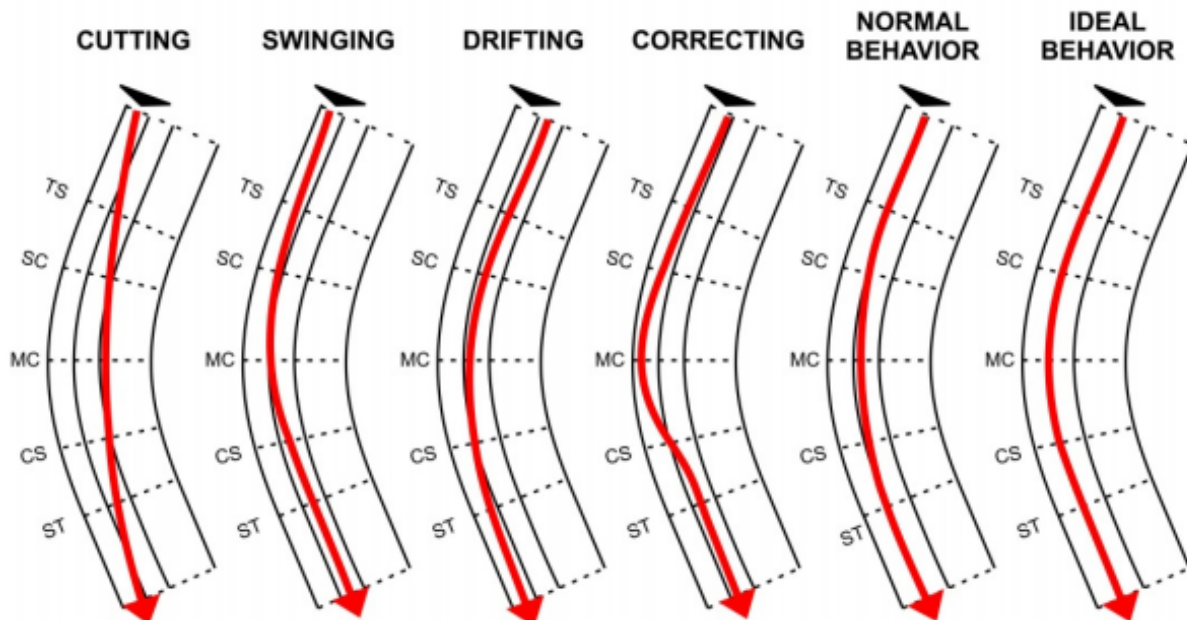
e_{maks} = Største tillate overhøyde

f_k = Sidefriksjonskoeffisient med sikkerhetsfaktor

127 = Konstant som inneholder tallverdiene til gravitasjonskonstanen og kvadratet av omregningsaktoren fra m/s til km/t

2.2.2 Avvik fra ideelle kurvebevegelser

Formlene som ligger til grunn for dimensjonering av minste horisontalkurveradius baserer seg på ideell ferdsel gjennom kurvene. Basert på hvordan en sjåfør velger å følge kurven kan det oppstå store avvik fra den tiltenkte atferden som kurvene faktisk dimensjoneres for.



Figur 2.6: Ulike bevegelser gjennom kurver basert på føreratferd [23].

Spacek undersøkte hvordan bilister valgte å kjøre gjennom forskjellige kurver i 2005 [23]. Bortimot 2000 sjåførers atferd gjennom kurver ble undersøkt ved å sette opp søyler langs kurven som målte avstand til kjøretøyene. Undersøkelsen antok at kjøretøyene enten var personbiler med 1,80 meters bredde, tunge kjøretøy med 2,50 meters bredde, eller motorsykler med 0,80 meters bredde. Gjennom undersøkelsen ble det definert seks typer bevegelser gjennom kurvene; kutting, svinging, drifting, korreksjon, normal atferd og ideell atferd. Hver av de ulike kurvebevegelsene er illustrert i Figur 2.6.

Spacek beskriver kurvebevegelsene på følgende måter. Kutting er en bevisst bevegelse hvor man kjører tett mot innerkurven for å balansere sentrifugalakselerasjon. Svinging er en asymmetrisk bevegelse mellom starten og slutten på kurven. Bevegelsen starter til høyre i kjørefeltet

og glir gradvis mot venstre gjennom kurven. Drifting fungerer tilsvarende som svinging bare at bevegelsen starter til venstre i kjørefeltet og ender på høyre side ved slutten av kurven. Korreksjon er en bevegelse hvor man gradvis drifter ut fra midten av kjørefeltet før man gjør en korreksjon i siste halvdel av kurven. Korreksjon gir en bevegelse med form som en S. Denne kjøreatferden antas å ofte være ubevisst siden kurvens lengde eller kurvatur undervurderes av sjåføren. Normal atferd defineres som en symmetrisk bevegelse med litt slingringsmonn og små kuttinger uten at senterlinja berøres. Ideell atferd defineres som symmetrisk bevegelse langs midten av kjørefeltet. I tillegg til de seks definerte kurvebevegelsene inkluderes også en kategori for øvrige bevegelser som ikke passer inn i de forhåndsdefinerte kategoriene. Resultatene fra undersøkelsen med fordelinger av aktuelle kurvebevegelser i ulike kurver er oppsummert i Figur 2.7. De ugunstige kurvebevegelsene; korreksjon, kutting, svinging og drifting, utgjorde i snitt 36 % av kurvebevegelsene, med variasjoner fra 16 % til 53 % avhenging av kurvene sjåførene kjørte gjennom. Kutting utgjorde den klart største andelen av de ugunstige kurvebevegelsene.

Track type (%)	Left-hand curves no.					Right-hand curves no.				
	1 <i>R</i> =65 m	3 <i>R</i> =70 m	4 <i>R</i> =79 m	4 N ^a <i>R</i> =79 m	6 <i>R</i> =143 m	2 <i>R</i> =65 m	5 <i>R</i> =115 m	5 N ^a <i>R</i> =115 m	7 <i>R</i> =220 m	8 ^b <i>R</i> =195 m
Ideal behavior (I)	0.5	1.0	0.4	0.6	0.0	0.0	0.9	0.8	1.1	0.2
Normal behavior (N)	9.8	20.0	34.8	22.5	7.0	8.9	48.2	36.3	33.3	18.1
Correcting (K)	5.4	0.0	2.1	0.6	2.8	1.2	3.1	4.9	0.2	11.5
Cutting (C)	33.9	48.0	31.1	27.2	18.6	28.7	8.1	16.7	12.2	44.9
Swinging (S)	9.3	0.0	7.3	4.7	4.4	5.9	6.4	4.9	0.0	0.0
Drifting (D)	1.8	5.0	0.7	0.6	8.0	4.9	2.7	5.9	3.6	3.2
Remaining track paths	39.3	26.0	23.6	43.8	59.2	50.4	30.6	30.5	49.6	22.1

^aMeasurements by night.

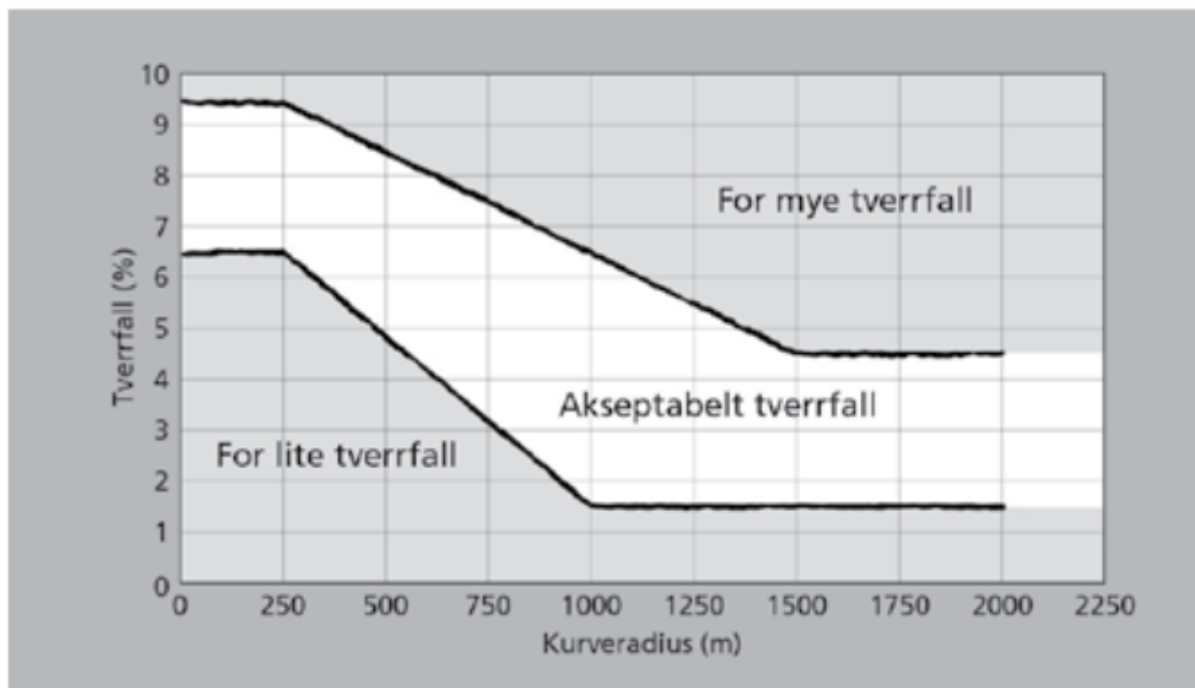
^bCurve with increased accident frequency (% of track types in both driving directions).

Figur 2.7: Fordelinger av kurvebevegelser gjennom ulike kurver [23].

Artikkelen påpeker at kutting og svinging ofte ble avbrutt i løpet av kurvebevegelsen, enten som følge av motgående kjøretøy eller feilberegnete kurvelengder. Endringer i disse kurvebevegelsene identifiseres som farlige siden unnamanøvere vil skape høye sentrifugalkrefter. Korreksjon opptrådte sjeldent, men denne kurvebevegelsen skapte høyest sentrifugalkrefter.

2.2.3 Avvik i overhøyde

Overhøyden til en veg, også kjent som tverrfallet, har en tendens til å avvike fra de dimensjonerende verdiene. Dette kan eksempelvis komme av feil under bygging eller naturlig slitasje fra kjørtøyene og omgivelsene. Fra figurene i Seksjon 2.2.1 er det vist hvordan overhøyden bidrar til å redusere behovet for sidefriksjon. Dersom overhøyden ikke følger de dimensjonerende verdiene vil det oppstå andre behov for sidefriksjon. Standarden for drift og vedlikehold gir krav til overhøyde ved forskjellige horisontalradier som utløser vedlikehold av vegen [24]. Så lenge overhøyden er mellom kravene som er gjengitt i Figur 2.8 er det stor sannsynlighet for at overhøyden aksepteres uten at det kompenseres for endrede sidefriksjonsbehov.



Figur 2.8: Krav til overhøyde ved ulike horisontalkurveradier [24].

Bogdashova undersøkte variasjoner i overhøyde på en lengre vegstrekning som en del av masteroppgaven hennes ved NTNU i 2012 [25]. Oppgaven tok for E6 i Sør-Trøndelag og strakte seg fra Oppdal i Sør til Malvik i nord. På denne strekningen undersøkte hun blant annet hyppigheten av overhøyder på strekningen som var lavere eller høyere enn overhøydekravene i Figur 2.8. Figur 2.9 viser resultatene hennes over hvor ofte overhøyden avviker fra overhøydekravene, fordelt etter horisontalkurveradier. Avvikene oppgis i prosent da enheten til overhøyde er i prosent. Eksempelvis vil 2% avvik fra vedlikeholdskravet på 6,5 % overhøyde ved horisontalkurveradius lik 250 meter gi en overhøyde på 4,5 %.

Resultatet til Bogdashova kan brukes til å gi en pekepinn på hvor lav overhøyden kan bli for dimensjonerende minstehorisontalkurveradier for ulike vegklasser. En lavere overhøyde enn forventet vil være mer kritisk enn høyere overhøyder enn forventet da dette vil kreve at en større andel av sidekreftene i en kurvebevegelse må tas opp av sidefriksjon. Av intervallene hun presenterer i sine resultater er de utvalgte horisontalkurveradiene i Tabell 2.1 særlig interessante da de dekker de ulike vegklassenes minste horisontalkurveradius for både standardverdi og 50 % økning av standardverdi. Deretter oppsummeres det hvor ofte disse vegklassene har overhøyder

Horisontal radius	Nødvendig justering av tverrfall, %											Andel av total kjørefeltslengde, %
	<(-9)	[-9-(-7))	[-7-(-5))	[-5- (-2))	[-2-0)	0	(0-2]	(2-5]	(5-7]	(7-9]	>9	
0	0,3	1,4	5,0	12,3	16,4	48,9	11,5	3,9	0,1	0,0	0,0	49,1
(0-150]	0,9	3,1	4,0	11,2	11,2	21,4	9,8	15,2	9,4	6,3	7,6	2,4
(150-200]	2,3	4,1	3,4	10,9	17,7	24,8	13,2	10,5	5,3	2,6	5,3	2,9
(200-250]	0,4	0,4	3,3	11,6	11,2	26,9	16,5	16,9	5,4	6,2	1,2	2,6
(250-300]	1,0	1,0	2,4	14,7	10,9	26,3	18,4	10,2	7,8	3,8	3,4	3,1
(300-350]	3,0	1,7	3,7	10,3	11,0	28,7	15,3	13,3	6,0	4,0	3,0	3,2
(350-400]	0,4	0,0	2,2	13,6	7,9	36,2	16,5	14,3	5,4	2,2	1,4	3,0
(400-450]	1,7	0,0	1,7	2,5	14,8	48,9	10,1	8,4	6,8	3,8	1,3	2,5
(450-500]	2,4	3,4	1,9	9,7	10,2	40,8	11,2	9,2	6,8	2,9	1,5	2,2
(500-550]	1,9	1,9	2,8	7,5	6,1	45,1	16,4	8,9	4,7	1,9	2,8	2,3
(550-600]	0,6	0,6	1,9	3,8	6,3	48,7	19,6	8,9	3,2	3,2	3,2	1,7
(600-700]	1,7	1,3	0,4	3,8	17,6	45,0	12,2	13,0	4,2	0,4	0,4	2,6
(700-800]	1,4	1,4	1,4	9,0	8,0	56,1	12,3	7,5	2,8	0,0	0,0	2,3
(800-900]	0,0	0,0	0,0	3,2	9,5	71,4	9,5	3,2	0,8	0,0	2,4	1,4
(900-1000]	0,0	0,0	0,9	0,0	3,6	63,4	11,6	11,6	7,1	1,8	0,0	1,2
>1000	0,2	0,2	2,5	5,2	8,8	60,4	14,3	7,0	1,1	0,2	0,0	17,5
Sum	0,7	1,2	3,7	9,9	13,3	47,5	12,8	6,9	2,1	1,0	0,8	100,0

Figur 2.9: Natalia Bogdashovas resultat for hyppigheten av ulike avvik i overhøyde for ulike horisontalkurveradier [25].

som er minimum 2 % lavere enn vedlikeholdskravene til overhøyden i Tabell 2.2. Disse resultatene viser at overhøyden blir mer enn 2 % lavere i mer enn 15 % av tilfellene for H1, Hø1 og Hø2, uavhengig om standardverdien deres for minste horisontalkurveradius økes med 50 % eller ei. Dermed gir det mening å forvente betydelige avvik i overhøyde for disse driftsklassene, også utover vedlikeholdskravene som ble presentert tidligere. Skal man dimensjonere etter de forholdene som gjelder for opp til 85 % av vegene reiser det spørsmålet om ikke disse avvikene bør inkluderes i dimensjoneringen for vegklassene H1, Hø1 og Hø2. For H5 og H3 ser man også at overhøyden kan bli lav, men med 50 % økning i minste horisontalkurveradius skjer det betydelig sjeldnere.

Horisontal- kurveradius [m]	Variasjon i tverrfall [%]				Total	n [%]
	<(-9)	[-9-(-7))	[-7-(-5))	[-5- (-2))		
0-150	0,9	3,1	4	11,2	19,2	2,4
150-200	2,3	4,1	3,4	10,9	20,7	2,9
200-250	0,4	0,4	3,3	11,6	15,7	2,6
300-350	3	1,7	3,7	10,3	18,7	3,2
350-400	0,4	0	2,2	13,6	16,2	3
550-600	0,6	0,6	1,9	3,8	6,9	2,3
700-800	1,4	1,4	1,4	9	13,2	2,3
>1000	0,2	0,2	2,5	5,2	8,1	17,5

Tabell 2.1: Utvalg av Bogdashovas resultat som viser hvor ofte overhøyden var minimum 2 % lavere enn vedlikeholdskravene til overhøyde. n viser andelen av den totale strekningen som hadde de gitte horisontalkurveradiene.

Horisontal- kurveradius [m]	Vegklasse		<(-2) [%]
	Standard	50 % økning	
0-150	Hø1		19,2
150-200		Hø1	20,7
200-250	H1 og Hø2		15,7
300-350		Hø2	18,7
350-400	H5	H1	16,2
550-600		H5	6,9
700-800	H3		13,2
>1000		H3	8,1

Tabell 2.2: Sammenheng mellom vegklassenes minste horisontalkurveradius, både som standardverdi og med 50 % økning i minste horisontalkurveradius, og hvor ofte overhøyden var minimum 2 % lavere enn vedlikeholdskravene til overhøyde.

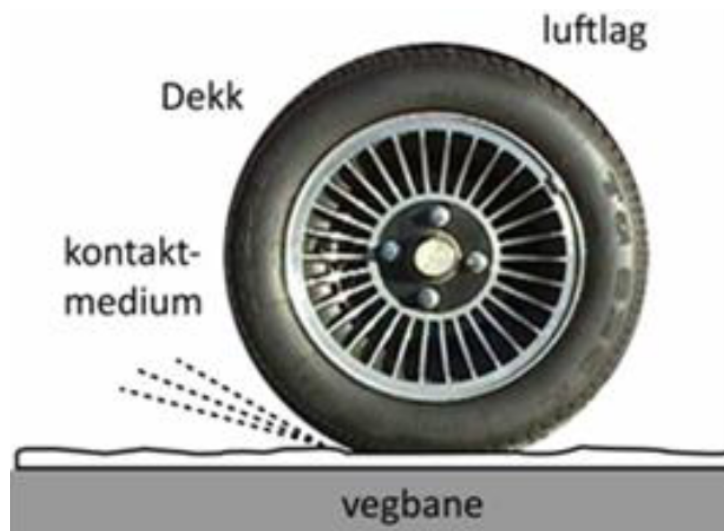
2.3 Friksjon

2.3.1 Hva er friksjon?

Generell kunnskap om friksjon er godt beskrevet i Statens Vegvesens lærebok for drift og vedlikehold av veger [24]. Friksjon er en koeffisient μ som beskriver forholdet mellom normalkraft og friksjonskraft som vist i Formel 2.12.

$$\mu = \frac{F_t}{F_n} \quad (2.12)$$

Gjennom friksjonskoeffisienten kan man beskrive hvor stor kraft man kan overføre mellom et kjøretøy og vegbanen. Høye verdier for friksjonskoeffisienter betyr at mye av kreftene kan overføres til vegbanen, noe som er ønskelig for å kunne styre kjøretøyet og unngå å skli. Siden friksjon kun beskriver forhold mellom komponenter vil den opplevde friksjonen variere mye avhengig av hvilke komponenter som settes sammen. Dette beskrives av tribosystemet som dekker de fire hovedkomponentene; dekk, vegbane, kontaktmedium og omgivelser vist i Figur 2.10. Hvilke friksjonskoeffisienter man opplever vil avhenge av tribosystemet. Eksempelvis vil kontaktmediet interferere med gummidekkets adhesjon til overflaten og gummidekkets deformasjon rundt teksturen i vegbanen. Med andre ord er ikke friksjon en iboende egenskap til en komponent, men hvordan ulike komponenter samspiller. Dermed kan man ikke si at en overflate har en gitt friksjonsverdi, man kan kun måle motstanden overflaten gir mot de andre komponentene i tribosystemet.



Figur 2.10: Illustrasjon over tribosystemet for hjul-vegbane [24].

Slipp S beskriver hvor mye et dekk sklir og defineres som hastighetsforskjellen til et kjøretøy og ytterste del av et hjul som er i kontakt med vegbanen. Dette er uttrykt i Formel 2.13. Når man skal undersøke friksjon er man interessert i hvor mye dekket sklir. Dette fremstilles som slipprate λ og uttrykkes som forholdet mellom slipp og kjøretøyets hastighet som fremstilt i Formel 2.14. Fra formelen ser man at lave verdier for slipprate tyder på liten glidning, mens høye verdier tilsvarer mer fastlåste dekk med større glidning. Erfaringer tilsier at den optimale bremskraften finner sted ved slipprater på rundt 20 %.

$$S = V_{\text{kjøretøy}} - V_{\text{slitebane}} \quad (2.13)$$

$$\lambda = \frac{S}{V_{\text{kjøretøy}}} = \frac{V_{\text{kjøretøy}} - V_{\text{slitebane}}}{V_{\text{kjøretøy}}} = 1 - \frac{V_{\text{slitebane}}}{V_{\text{kjøretøy}}} \quad (2.14)$$

S = Slipp

λ = Slipprate

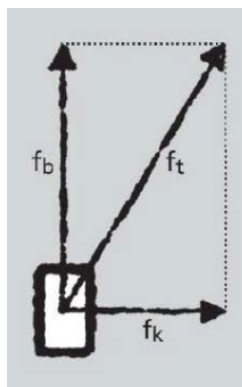
$V_{\text{kjøretøy}}$ = Hastighet til kjøretøyet

$V_{\text{slitebane}}$ = Hastighet til ytterkant av dekket

2.3.2 Sidefriksjon på bar veg

I dimensjoneringen av minste horisontalkurve radius som ble beskrevet i Seksjon 2.2.1 trengs et mål på sidefriksjonskoeffisienten. Ulike land bruker ulike tilnærminger til hvordan sidefriksjonskoeffisienten tallgis. Ofte gjøres dette med hensyn på sidekreftene som friksjonskoeffisienten kan motvirke, og det brukes tallverdier som skal hindre ubehag i kurvene. I Norge gjøres vurderingen på bakgrunn av friksjonsmålinger som gir totalfriksjonsverdier. Hvordan

disse målingene utføres beskrives i nærmere detalj i Seksjon 2.3.3. For å imøtekomme forutsetningene i håndbok V120 om våt, men ren og isfri kjørebane [5] foregår friksjonsmålingene på veger som matcher håndbokens forutsetninger. Deretter benyttes 5-persentilen til verdiene av målingene for å dekke friksjonsforhold som ikke overskrides i mer enn 5 % av tilfellene med våt og isfri kjørebane.



Figur 2.11: Sammenheng mellom totalfriksjon f_t , sidefriksjon f_k og bremsefriksjon f_b [5].

Som vist i Figur 2.11 består totalfriksjonen f_t av både sidefriksjon f_k og bremsefriksjon f_b . Bremsefriksjonen er den tilgjengelige friksjonskoeffisienten i kjøreretning som styrer fartsendringen og dermed bremsedistansen. Sidefriksjonen beskrives av friksjonskoeffisienten som tar opp sidekrefter og hindrer kjøretøyene fra å skli utover eller innover i en kurvebevegelse. Andelen sidefriksjonen utgjør av totalfriksjonen avhenger av hastigheten kjøretøyet beveger seg med. Håndbok V120 [5] legger til grunn fordelingene i Figur 2.12 for ulike fartsgrenser når man skal avgjøre hvor stor andel sidefriksjonen utgjør av totalfriksjonen. I tillegg legges det til sikkerhetsfaktorer for friksjonen dersom man dimensjonerer for de strengeste vegklassene. Bremsefriksjonen kan forøvrig beregnes ved bruk av Pytagoras læresetning når totalfriksjon, og dermed sidefriksjon, er kjent.

Fartsgrense [km/t]							
40	50	60	70	80	90	100	110
39 %	39 %	37 %	37 %	34 %	30 %	26 %	20 %

Figur 2.12: Sidefriksjonens andel av totalfriksjonen ved ulike fartsgrenser [5]

Masteroppgaven til Andersen og Wasmuth vurderte blant annet friksjon på bar veg som en del av FoU-programmet Vegutforming i regi av Statens Vegvesen [26]. De var ute etter å vurdere sidefriksjonsverdiene som brukes til dimensjonering av horisontalkurveradier og sammenlignet norske sidefriksjonsverdier med måledata og standardverdier fra andre land. Disse sammenligningene av sidefriksjonsverdier oppsummerte dem i Figur 2.13. Verdiene presentert i tabellen fra USA er hentet fra American Association of State and Transportation Officials, også kjent som AASHTO, sine retningslinjer fra 2011. Av nysgjerrighet ble verdiene fra 2011 [27] sammenlignet med en tidligere utgave fra 2001 [28] på eget initiativ, noe som avslørte at det hadde skjedd endringer som vist i Tabell 2.3.

Dimensjonerende verdier for sidefriksjon				
Fartsgrense [km/t]	110	100	90	80
85%-fraktil	0,073	0,098	0,120	0,145
90%-fraktil	0,069	0,092	0,114	0,137
95 %-fraktil	0,061	0,083	0,102	0,124
95 %-fraktil_Sv. ford.	0,093	0,103	0,114	0,127
Norge 18	0,075	0,098	0,119	0,157
Norge 14_H3-H9/NA	0,063	0,070	0,100	0,120
Norge 14_H2				0,150
Norge 14_Hv.Ø				0,170
Sverige	0,097	0,107	0,118	0,130
Sverige Vdim_S	0,088	0,097	0,107	0,118
Danmark	0,097	0,107	0,118	0,130
Storbritannia	0,060	0,060	0,060	0,060
Tyskland	0,101	0,107	0,114	0,120
USA	0,110	0,120	0,130	0,140
Australia	0,120	0,120	0,130	0,160
Australia Vdim_A	0,110	0,120	0,120	0,130

Figur 2.13: Andersen og Wasmuths resultater over sidefriksjonsverdier ved ulike fartsgrenser for ulike land sammenlignet med målte verdier.

Hastighet [km/t]	AASHTOs anbefalte verdier for sidefriksjon	
	Verdier fra 2001	Verdier fra 2011
40	0,17	0,23
50	0,16	0,19
60	0,15	0,17
70	0,14	0,15
80	0,14	0,14
90	0,13	0,13
100	0,12	0,12
110	0,11	0,11

Tabell 2.3: AASHTOs anbefalinger til dimensjonerende sidefriksjonsverdier fra 2001 og 2011.

Etter at det ble oppdaget endringer i de amerikanske sidefriksjonsverdiene ble det utløst tilleggsundersøkelser utover Andersen og Wasmuths oppgave for å verifisere at de andre landenes sidefriksjonsverdier ikke hadde endret seg med nyere utgaver av standarder og veiledningsdokument. Det ble sammenlignet norske sidefriksjoner brukt av Statens Vegvesen med publiserte verdier fra 2019 [5] kontra verdier under revisjon fra 2018, svenske verdier fra 2020 [29] kontra 2018, danske verdier fra oktober 2018 [30] kontra april 2018, og australske verdier fra 2021 [31] kontra 2016. Ingen av de nyere versjonene viser til andre sidefriksjonsverdier enn de som allerede er presentert i Figur 2.13, men det er hentet ut flere verdier for hastigheter mellom 40 km/t og 70 km/t som presenteres i Tabell 2.5. De tyske og britiske verdiene er ikke gjennomgått på ny da tyske standardene er krevende å oppdrive på engelsk, mens de britiske sidefriksjonsverdiene kun er beregninger ettersom britene ikke bruker sidefriksjon som en dimensjonerende verdi.

For å øke innsikten i sidefriksjonsverdier på det Nord-Amerikanske kontinentet er det også hentet ut sidefriksjonsverdier fra Canada [32], samt verdier fra delstaten Montana [33] for å vurdere opp mot AASHTO. Både Canada og Montana anses som særlig interessante da de ofte opplever tilsvarende vinterutfordringer som Norge. Montanas verdier ble kun oppgitt med amerikanske enheter, og er derfor konvertert til SI enheter slik Tabell 2.4 viser for å gi en fornuftig sammenligning mot de andre landene. Konverteringen innebar noen subjektive vurderinger, særlig da to hastigheter i kilometer per time kunne rundes av til samme verdi. I disse tilfellene ble gjennomsnittsverdien for sidefriksjonene benyttet. Ikke overraskende er verdiene fra både Canada og Montana veldig like AASHTOs anbefalinger. Begge har hakket lavere verdier for fartsgrensen på 110 km/t, men Canada skiller seg ut fra sine Nord-Amerikanske motparter med vesentlig lavere friksjonsverdier for lave hastigheter. Canada stiller faktisk med de laveste friksjonsverdiene blant alle de undersøkte landene for hastigheter fra 60 km/t og lavere.

Oppgitt hastighet [mp/h]	Konvertert hastighet [km/t]	Avrundet hastighet [km/t]	Anbefalt sidefriksjon	Hastighet [km/t]	Tilnærmet sidefriksjon
20	32,2	30	0,27	40	0,23
25	40,2	40	0,23	50	0,20
30	48,3	50	0,20	60	0,17
35	56,3	60	0,18	70	0,15
40	64,4	60	0,16	80	0,14
45	72,4	70	0,15	90	0,13
50	80,5	80	0,14	100	0,115
55	88,5	90	0,13	110	0,10
60	96,6	100	0,12		
65	104,6	100	0,11		
70	112,7	110	0,10		
75	120,7	120	0,9		
80	128,7	130	0,8		

Tabell 2.4: Konvertering av Montanas anbefalte sidefriksjonsverdier fra miles per time til kilometer per time. Venstre tabell viser de originale hastighetene konvertert og avrundet til nærmeste 10-er med anbefalte sidefriksjonsverdier. Høyre tabell viser hastigheter i kilometer per time med en tilnærmet sidefriksjonsverdi som bruker gjennomsnittsverdien til sidefriksjonene dersom det var flere like avrundede hastigheter.

En oversikt over sidefriksjonsverdier til Norge, Sverige, Danmark, USA, Montana, Canada, og Australia for hastigheter mellom 40 km/t og 110 km/t er presentert i Tabell 2.5. Til sammenligning med de andre landene ender Norge opp med nokså optimistiske friksjonsverdier for de laveste hastighetene, men skilte med de laveste verdiene for den høyeste hastigheten.

Hastighet [km/t]	Dimensjonerende sidefriksjonsverdi						
	Norge	Sverige	Danmark	USA	Montana	Canada	Australia
40	0,249	0,191	0,191	0,23	0,23	0,17	0,30
50	0,224	0,173	0,173	0,19	0,20	0,16	0,30
60	0,195	0,157	0,157	0,17	0,17	0,15	0,24
70	0,182	0,143	0,143	0,15	0,15	0,15	0,19
80	0,157	0,130	0,130	0,14	0,14	0,14	0,16
90	0,131	0,118	0,118	0,13	0,13	0,13	0,13
100	0,108	0,107	0,107	0,12	0,115	0,12	0,12
110	0,079	0,097	0,097	0,11	0,10	0,10	0,12

Tabell 2.5: Sidefriksjonsverdier for ulike land ved ulike hastigheter.

Sverige og Danmark benytter seg av de samme sidefriksjonsverdiene, men de legger annerledes totalfriksjonsverdier til grunn. For begge landene er sidefriksjonen gitt av Formel 2.15 [29][30], men der Sverige benytter en lignende formel for å beregne totalfriksjon uttrykt med Formel 2.16 [29] baserer Danmark seg på en konstant totalfriksjon lik 0,377 for alle hastigheter [30].

$$f = 0,28 * EXP(-0,0096 * V) \quad (2.15)$$

$$f = 0,62 * EXP(-0,0063 * V) \quad (2.16)$$

Sammenlignet med Norge benytter de andre skandinaviske landene alltid en lavere totalfriksjon, men ofte er de norske sidefriksjonsandelene mellom verdiene til Sverige og Danmark. Sammenhenger mellom totalfriksjon og sidefriksjon for de skandinaviske landene er gitt i Tabell 2.6. Norge skiller seg vesentlig ut med en veldig lav sidefriksjonsandel når hastigheten settes til 110 km/t. I tillegg brukes en ekstra sikkerhetsfaktor for friksjon når hastigheten blir 90 km/t eller høyere. Denne sikkerhetsfaktoren er ikke inkludert i noen av grafene i denne seksjonen. På denne måten blir Norge mer konservativ med tanke på sidefriksjon for høye hastigheter, men da bremsefriksjonen utgjør resten av totalfriksjonen blir bremsefriksjonen oftere høyere enn de andre landene. Bremselengdene holdes likevel under kontroll ved bruk av fartstillegg som forutsetter en høyere hastighet enn det fartsgrensen tilsier.

Hastighet [km/t]	Totalfriksjon			Sidefriksjonsandel [%]		
	Norge	Sverige	Danmark	Norge	Sverige	Danmark
40	0,638	0,482	0,377	39	40	51
50	0,575	0,452	0,377	39	38	46
60	0,528	0,425	0,377	37	37	42
70	0,491	0,399	0,377	37	36	38
80	0,461	0,375	0,377	34	35	34
90	0,437	0,352	0,377	30	34	31
100	0,416	0,330	0,377	26	32	28
110	0,397	0,310	0,377	20	31	26

Tabell 2.6: Totalfriksjoner og sidefriksjonsandeler for de skandinaviske landene.

2.3.3 Friksjonsmålinger

Læreboken for drift og vedlikehold av veger beskriver tre hovedmåter for å måle friksjon på vegoverflater i dag; bremsetest, retardasjonsmålere og kontinuerlige friksjonsmålere [24]. Den minst krevende metoden er å gjennomføre en bremsetest. Ved fullt utslag på bremsen kan man måle bremselengden som kreves for å nå en full stopp. Bremsetest kan utføres med vanlige biler uten spesialisert utstyr. Ved å utføre testen på strekninger uten stigning trenger man kun å kjenne startfarten og ha muligheten for kunne måle bremselengden for å beregne den gjennomsnittlige friksjonsverdien for strekningen. Beregningene baserer seg på formelen for bremselengde gitt av Formel 2.17 fra håndbok V120 [5]. For antagelsen om vegstrekning uten stigning kan bremsefriksjonskoeffisienten isoleres som vist i Formel 2.18. Denne metoden er derimot ikke særlig nøyaktig dersom man bruker ordinære kjøretøy da de avviker fra standardiserte krav til blant annet vekt, dekk, og bremsesystem. I tillegg risikerer man å skape trafikkfarlige situasjoner, og man får ikke målt friksjon i kurver.

$$L_b = \frac{1}{2} * \frac{\left(\frac{V}{3,6}\right)^2}{g * (f_b + s)} \quad (2.17)$$

$$f_b = \frac{V^2}{254,3 * L_b} \quad (2.18)$$

f_b = Bremsefriksjonskoeffisient

V = Hastighet før brems [km/t]

L_b = Bremselengde [m]

254,3 = Konstant som inneholder 2 ganger

tallverdiene til gravitasjonskonstanen

og kvadratet av omregningsaktoren

fra m/s til km/t

Bruk av retardasjonsmålere til å beregne friksjon er mer fleksibelt enn bremsetest da man ikke trenger å bremse til full stopp. Så lenge man har mulighet til å gjennomføre en kraftig nedbremsing kan retardasjonsmåleren automatisk registrere starthastighet, sluttastighet og tidsbruk, og

dermed beregne friksjon etter Formel 2.19. Metoden benyttes av driftspersonell i dag, men man får ikke de mest nøyaktige resultatene. Målingene avhenger blant annet av bilens dekk, vektfordeling og bremsesystem. For å sikre reproduserbarhet må man tilstrebe å holde disse faktorene så like som mulig fra måling til måling, og man bør presse like hardt på bremsen hver gang. For å redusere usikkerhet i målingene er det nødvendig med årlig kalibrering av retardasjonsmålerne.

$$f_b = \frac{V_1 - V_2}{g * t} \quad (2.19)$$

f_b = Bremsfriksjonskoeffisient

V_1 = Hastighet før brems [m/s]

V_2 = Hastighet etter brems [m/s]

t = Varighet av brems [s]

g = Gravitasjonskonstanten [m/s^2]



Figur 2.14: Illustrasjon over de kontinuerlige friksjonsmålerne OSCAR (til venstre) og RoAR versjon 5 (til høyre) [24].

Den siste metoden som beskrives er friksjonsmålinger ved hjelp av kontinuerlige målere. Disse kan måle over lange strekninger og gir gjennomsnittlige totalfriksjonskoeffisienter for korte intervaller, typisk hver 20 meter. Målingene utføres ved å bruke et standardisert dekk etter kravene til ASTM E524 som konfigureres til å ha konstant 20% slipp-rate for vintermålinger. Fra Seksjon 2.3.1 kjenner man igjen denne slipp-raten som den optimale slipp-raten for å få optimal bremskraft. Friksjonsverdien beregnes ved å måle dreiemomentet som påføres målehjulet mens det bremses for å opprettholde ønsket slipp-rate. Som nevnt i Seksjon 2.3.1 er ikke friksjon en iboende egenskap til et materiale, men en respons i et system som avhenger av tribosystemet. For å forholde seg til dette og få standardiserte verdier har Statens Vegvesen valgt å definere alle målinger som utføres med friksjonsmåleren OSCAR (Optimum Surface Contamination Analyzer and Recorder) som sanne representasjoner av friksjonsforholdene. OSCAR fungerer som referanse for de avanserte friksjonstillhengerne RoAR (Road Analyzer and Recorder), slik at RoAR-ene kalibreres mot OSCAR før hver målesesong. OSCAR og RoAR beskrives nærmere i håndbok R211 Feltundersøkelser [34], og de er illustrert i Figur 2.14. Forskjellige RoAR-er

dekker ulike deler av landet, og brukes som referansemåler når andre vinterfriksjonsmålere skal kalibreres. Versjon 5 av RoAR har som standard ett målehjul som bremses ved hjelp av hydraulikk, mens versjon 6 av RoAR kan leveres med to målehjul som bremses elektronisk for å dekke begge hjulsporene. Utover friksjonskoeffisienter gir RoAR også lufttemperatur og overflatetemperatur, med mulighet for å inkludere relativ luftfuktighet og duggpunkttemperatur.

2.4 Glatthet

2.4.1 Grader av glatt

Artikkelen til Norrman et al. beskriver en metode for å bruke data fra vegværinformasjonsystem til å vurdere ulike typer glatthet og vurdere om vinterdrift forebygger vegtrafikkulykker [35]. Artikkelen viser til tidligere forskning som har funnet ut at hyppigheten av vegtrafikkulykker øker ved starten av snøfall, men stabiliserer seg over tid etter hvert som sjåførene venner seg til de nye vegforholdene. Vegforholdene kan variere mye, selv over korte distanser. Det er dermed behov for meteorologisk data av høy kvalitet dersom man ønsker å si noe om de atmosfæriske forholdene for en bestemt lokasjon. Forskningsgruppen utviklet et system for automatisk klassifisering av ulike typer glatthet i forbindelse med ulykkesregistrering. Klassifiseringssystemet hans er vist i Figur 2.15, og tar inn parametre for nedbør, lufttemperatur, vegtemperatur, duggpunkttemperatur, relativt luftfuktighet og vindhastighet. Fra den sentrale databasen kan man blant annet hente ut varigheten for de ulike forholdene. I Figur 2.16 vises antallet timer det ble registrert med de forskjellige typene glatthet per måned i nordre deler av Sverige.

Type of slipperiness	Precipitation	Variables
(1) Precipitation (rain/sleet) on a frozen road surface	Yes	$P, T_{\text{air}}, T_{\text{road}}$
(2) Precipitation (snow) on a frozen road surface	Yes	$P, T_{\text{air}}, T_{\text{road}}$
(3) Precipitation (snow/sleet) on a warm road surface	Yes	$P, T_{\text{air}}, T_{\text{road}}$
(4) Snowfall together with hoarfrost	Yes	$P, T_{\text{air}}, T_{\text{road}}, T_{\text{dew}}$
(5) Hoarfrost and low visibility	No	$T_{\text{road}}, T_{\text{dew}}, Rh$
(6) Freezing dew followed by hoarfrost	No	$T_{\text{road}}, T_{\text{dew}}$
(7) Strong formation of hoarfrost	No	$T_{\text{road}}, T_{\text{dew}}, Rh, U$
(8) Weak formation of hoarfrost	No	$T_{\text{road}}, T_{\text{dew}}, Rh, U$
(9) Drifting snow	Yes	P, T_{air}, Rh, U
(10) Watercover which freezes	Yes	$P, T_{\text{air}}, T_{\text{dew}}, T_{\text{road}}, Rh$

Figur 2.15: System for automatisk klassifisering av ulike typer glatthet [35].

Type	1991 11	1991 12	1992 01	1992 02	1992 03	1992 04	1993 11	1993 12	1994 01	1994 02	1994 03	1994 04	1995 11	1995 12	1996 01	1996 02	1996 03	1996 04
1	0	4	3	3	2	0	7	16	13	8	9	0	2	9	10	4	1	0
2	0	0	8	3	0	0	15	15	3	72	25	0	16	25	21	45	11	0
3	0	0	20	2	1	0	3	0	0	8	0	0	7	0	0	0	1	1
4	0	0	0	1	0	0	0	2	4	1	0	0	0	7	0	0	6	0
5	13	71	48	13	0	1	2	14	10	1	10	0	0	0	0	0	24	23
6	1	4	4	21	4	0	0	9	9	0	3	0	0	0	0	5	3	0
7	1	37	33	11	2	0	1	7	3	4	0	0	16	37	51	12	25	2
8	3	11	10	1	0	0	8	42	43	26	9	0	7	13	1	3	6	0
9	0	0	0	0	0	0	15	0	0	6	0	0	6	1	0	23	0	0
10	0	2	1	6	0	0	2	10	10	2	0	0	0	0	0	0	0	0

Figur 2.16: Antall timer med ulike typer glatthet per måned i nordre deler av Sverige [35].

Artikkelen beskriver kort de forskjellige typene glatthet og hvilke forhold som tilrettelegger for dem. Når det kommer til rimfrost skiller det mellom rim som følge av høy luftfuktighet (5), rim som følge av dugg (6), og rim som følge av nedadgående vanndamp uten turbulens (7) eller med turbulens (8).

Studien viser at i de fleste tilfeller hvor det skjedde ulykker under en av de 10 typene av glatthet så var det også fult vintervedlikehold på vegen. Med andre ord var vintervedlikeholdet som oftest ikke tilstrekkelig nok for å forebygge ulykkene. Som mulige tiltak for å hindre ulykker på glatte veger trekker artikkelen frem mulighetene for å informere sjåførene om de glatte forholdene, eller å midlertidig redusere fartsgrensen til å samsvare med de opptredende forholdene.

I et temahefte tilknyttet standarden for drift og vedlikehold benyttes bestemte typer føreforhold med tilhørende erfaringsverdier [36]. Føreforholdenes friksjonsverdier er oppsummert i Tabell 2.7. Tabellen viser hvordan snø og slaps gir gunstigere friksjonsforhold en rene is- og rimforhold. Dette henger sammen med ujevnhetene i snøen og mykheten dens som tillater deformasjoner. Is er derimot ofte både jevn og hard, noe som gir lite rom for å overføre krefter fra dekket gjennom deformasjon. Rundt frysepunktet for både snø, slaps og is vil molekylene være veldig aktive og skape et vannlignende lag [37], noe som gjør det vanskelig å skape adhesjon til dekket.

Føreforhold på vegen	Friksjonsverdi
Tørr/bar	0,70
Fuktig/bar	0,65
Våt/bar	0,50 - 0,60
Hard snø	0,35 - 0,40
Is/snø	0,30
Is/slaps	0,25
Tynn is	0,20
Rimfrost	0,15

Tabell 2.7: Erfaringsverdier for friksjon for ulike føreforhold på veger.

2.4.2 Rimfrost

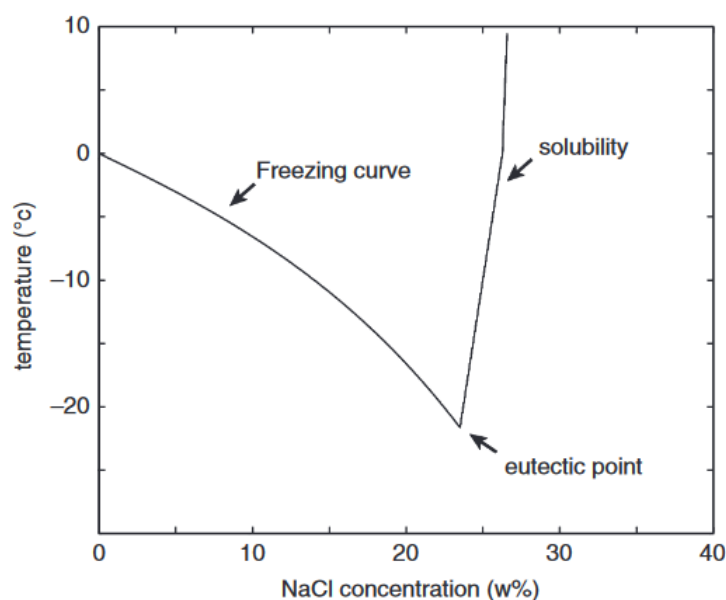
Karlson forsøkte å forutse rimdannelse ved hjelp av vegværinformasjonsystem i 2001 [38]. I studien beskrives dannelsen av rimfrost og sammenhengen mellom variabler. Teoretisk sett dannes rim når overflatetemperaturen er lavere enn både duggpunkttemperaturen og 0°C , samtidig som fuktig luft beveger seg mot vegoverflaten. Da kan vannmolekylene gå direkte fra gass og over til fast form slik at fuktigheten deponeres som rim på vegbanen. Studien viste at flere variabler var med på å mengden rim som ble avsatt på vegoverflaten. Høye vindhastigheter, stor differanse mellom duggpunkttemperatur og overflatetemperatur, særlig over lengre tid, samt lav lufttemperatur vil gi større mengder med rim. Likevel var det ingen sammenheng mellom mengden avsatt rim og rimens friksjonskoeffisient. Derimot vil større differanser mellom duggpunkttemperatur og overflatetemperatur, og lange perioder med denne differansen redusere friksjonsverdien. Karlson mener at dette muligens kan forklares av strukturen til rimfrosten, uten at hun går nærmere inn på dette i artikkelen. Antagelsen fremstår som fornuftig da strukturen til rimfrosten kan enten vokse med mange små krystaller i tynne søyler eller som en mer

sammenhengende og jevn gitterstruktur. Førstnevnte gir skjøre krystaller som knuses lett av bildekk, mens sistnevnte fungerer som en jevn og glatt isflate.

Ou et al. undersøkte risiko for rimdannelse og klimaendringer i Sverige i 2019 [39]. I undersøkelsen ble det funnet at dager med rimdannelse i Midt-Sverige hadde høyere lufttemperatur og lavere overflatetemperatur enn for dager uten rimdannelse. I Nord-Sverige var begge parametrene varmere for dager med rimdannelse enn for dagene uten rimdannelse. Dette begrunner forfatteren med at i nord styres rimdannelsen av varmluftsadveksjon, mens sentralt i Sverige spiller også nedkjøling i form av utstråling en rolle.

2.4.3 Motvirke glatthet

Fjærestad fullførte sin doktorgrad om rimdannelse på saltede vegoverflater ved NTNU i 2020. Gjennom doktorarbeidet undersøkte hun også ulike vintervedlikeholdstrategier. Her går hun inn på generelle strategier for vegnettets som anti-ising og dynamisk varsling, samt spesifikke tiltak rettet mot bruk som termiske metoder og høyfriksjonsdekker [40].



Figur 2.17: Fasediagram for natriumklorid i vann [41].

Gjennom anti-ising ved bruk av vegsalt eller andre kjemikalier kan man redusere risikoen for at en veg blir glatt. Formålet med tilsettelsen av disse stoffene er å redusere frysepunktstemperaturen til isen slik at vann ikke fryser til is før det når temperaturer på godt under 0°C. Dermed unngås dannelsen av is som ville gitt glatte kjøreforhold. Likevel er det krevende å få til anti-ising som gir god effekt. Ved å gå utover Fjærestads avhandling kan Klein-Paste og Dalens forskning på anti-ising fra 2018 brukes til forklare viktigheten av vanninnhold for å velge riktig mengde vegsalt [41]. Som Figur 2.17 viser beskrives senkingen av frysepunkt via et fase-diagram hvor konsentrasjonen av kjemikallet, her natriumklorid, avgjør hvor mye frysepunktet kan senkes. Konsentrasjonen av salt må dermed være høy nok slik at frysepunktet til vannet senkes under den gjeldende temperaturen. Begynner løsningen å fryse vil ikke saltet fra løsningen tas opp i krystallstrukturen til isen. Dermed blir det mindre vann per salt og konsentrasjonen øker. Ved økende konsentrasjoner vil etter hvert det eutektiske punktet bli nådd, noe som betyr at saltet ikke lenger løses opp i vannet. Dette betyr at man ikke kan påføre salt i store mengder

vann siden konsentrasjonen blir veldig liten, og at store mengder salt fort blir overflødig siden konsentrasjonen ikke øker etter at det eutektiske punktet er nådd. For å oppnå god anti-ising må vegsalt påføres etter at store vannmengder har forsvunnet fra veien, og påføres på en slik måte at trafikken ikke blåser det bort. Fjærestad påpeker også viktigheten av tidspunktet saltet legges ut på veien, og legger til at det ikke må komme for sent slik at veien allerede har rukket å bli glatt. Å finne riktig tidspunkt er også krevende med tanke på at vintervedlikeholdsmannskapet må rekke innom hele ruten før det blir glatt. En tilsvarende problemstilling er at saltet må gi en langvarig nok effekt slik at det ikke blir glatt innen vintervedlikeholdsmannskapet har fullført ruten og kan komme tilbake med nye tiltak. Disse tidskrisiske utfordringene kan løses på bruer ved bruk av automatiske anti-isingssystemer som sprayer kjemikalier etter kriterier fra sanntids vær- og overflatemålinger.

For varslingssystemer viser Fjærestad blant annet til Veneziano et al. som undersøkte fartsendringer som følge av dynamiske varslingssystem for glatte forhold i 2013 [42]. Målet var å redusere antallet ulykker i glatte kurver på motorvegen som krysser Fredonyer fjellpass i California. Ved bruk av vegsensorer som detekterte isete forhold ble det aktivert dynamiske varselskilt som informerte sjåførerene om de gjeldende forholdene gjorde det glatt på veien. Studien viste gjennom kontrollforsøk at bilistene reduserte hastigheten når de ble varslet om glatte forhold fra de dynamiske varselskiltene.

For de termiske metodene trekker Fjærestad fram Zhang et al. som i 2009 forsøkte å evaluere metoder for å fjerne snø og is fra bruer [43]. Artikkelen undersøker kjemiske og termiske metoder og vurderer kostnader og miljøeffekter. De termiske metodene baserer seg på å smelte is og snø, noe som krever store energimengder. Dette kan for eksempel gjøres med elektrisitet, infrarød stråling, geotermiske varmpumper eller radiobølger. Uavhengig av hvilket alternativ man velger blir det veldig dyrt med tanke på kapitalkostnader, energiforbruk og driftskostnader. Artikkelen konkluderer med at vegsalting generelt sett er å foretrekke fremfor termiske metoder. De termiske metodene burde kun brukes for spesielle steder, for eksempel overgangen mellom bru og veg. Utover Fjærestads oppgave kan man finne enda et alternativ som krysser grensene mellom tradisjonell anti-ising og termiske metoder, nemlig Liston et al. sin bruk av fase-endrende materialer fra 2016 [44]. De fase-endrende materialene lagrer latent varme fra eksterne kilder før varmen slippes ut igjen gjennom fase-endringer. For bruk på bruer kan de fase-endrende materialene justeres for å slippe ut varme rett over 0°C og dermed bidra til å unngå ising. Studien viser positive resultater og åpner for at konseptet kan fungere, men det er enda tidlig i forskningsstadiet og bruksområdet er foreløpig begrenset til bruer med vegger bestående av betong.

For høyfriksjonsdekker bruker Fjærestad artikkelen til Dave et al. som sin primære kilde. De undersøkte i 2017 hvor effektivt det var med høyfriksjonsdekker for å redusere ulykker på bruer [45]. Høyfriksjonsdekkene bestod av fine aggregater blandet inn i slitelaget som skulle bidra til å hindre skrensing. For å sammenligne virkeevnen til høyfriksjonsdekkene ble skrenstallet til hvert av dekkene målt. Artikkelen påpeker at økning i skrenstallet ikke nødvendigvis sammenfaller med redusert sannsynlighet for ulykker. Artikkelen viser også til tidligere forskning som fant at skrenstallet til et asfaltdekke reduseres kort tid etter bygging. Degraderingen av overflatedekke skyldes hovedsaklig abrasjon fra vegtrafikk og ploging fra vintervedlikehold. Artikkelen konkluderer med at høyfriksjonsdekker ikke bidrar til signifikant bedring av ulykkesrisiko.

Kapittel 3

Metode

Hovedparten av metoden gikk ut på å programmere i Python. Store deler av oppgaven har dermed gått med på å lære programmering. Alle kodene som ble skrevet for oppgaven er oppsummert i Vedlegg B og viser nøyaktig hva som ble gjort. Siden metoden kan reproduseres ved bruk av disse vedleggene så er metoden skrevet på et mer overordnet nivå. Fokuset er lagt på forklaringer og illustrasjoner fremfor å gjengi kodesnutter. Spesifikke funksjoner fremheves der det ble ansett som hensiktsmessig. En grundigere gjennomgang av hvilke forutsetninger som kreves for å kjøre koden uten nevneverdige modifikasjoner presenteres i neste seksjon. Deretter presenteres en oversikt over hva som ble gjort i de ulike analysene.

Resten av metodekapitlet er delt opp i ulike deler. Første del beskriver datagrunnlaget og hvordan filene ble behandlet for å bli klare til analyse. Andre del beskriver analyser av disse filene. Tredje del er en sensitivitetsanalyse av øvrige variabler som brukes i dimensjonering av horisontalkurveradius. Fjerde del vurderer resultatene fra analysene opp mot dagens krav til 50 % økning av minste horisontalkurveradius på bru.

3.1 Reproduksjon med vedlagte koder

Alle resultatene som ble produsert i oppgaven kan reproduseres ved bruk av kodene i Vedlegg B. I tillegg kreves noen koder som er utviklet av Statens Vegvesen for å koble QGIS til NVDB sin API [46]. I oppgaven ble det benyttet Python versjon 3.9 og QGIS versjon 3.16, og kodene skal være kompatible med disse versjonene. For å gjenskape analysen eller utføre egne analyser trengs kun matrisefila, *BruMatrise.npy*. Derimot kreves det en bestemt mappestruktur for å gjenskape en tilsvarende matrise med målepunkter på og utenfor bru med utgangspunkt i filer fra Viatechs friksjonsdatabase. Uten denne mappestrukturen klarer ikke kodene å finne filene som skal behandles. Mappestrukturer som ligger til grunn for kodene er vist i Figur 3.1. På starten av flere av kodene blir det spurt etter sti, altså hvor hovedmappa *Programmering* ligger lagret på datamaskinen. Oppdatering av stiene og bruk av samme mappestruktur vil gjøre at kodene kan kjøres slik de ble brukt i oppgaven. Koden *NVDB.py* i QGIS-mappa må kjøres fra QGIS sin innebygde Python-konsoll via editoren, de øvrige kodene i QGIS-mappa blir brukt gjennom dette programmet.

```

>Programmering
  >Friksjonsmålinger
    >QGIS_input
      >Input.txt
    >QGIS_output
      >Filer...
  >Renset
    >Filer...
  >Rådata
    >Mappe1 (valgfritt navn)
      >Filer...
    >Mappe2 (valgfritt navn)
      >Filer...
    >(Valgfritt antall mapper)
  >Python
    >BruMatrise.npy
    >Analyse.py
    >Gaulabura.py
    >Grovsiling.py
    >LageMatrise.py
    >Rensing.py
    >Sensitivitet.py
  >QGIS
    >Custom
      >NVDB.py
      >Hjelpfunksjoner.py
    >nvdbapi-V3-master
      >Koder fra Statens Vegvesen

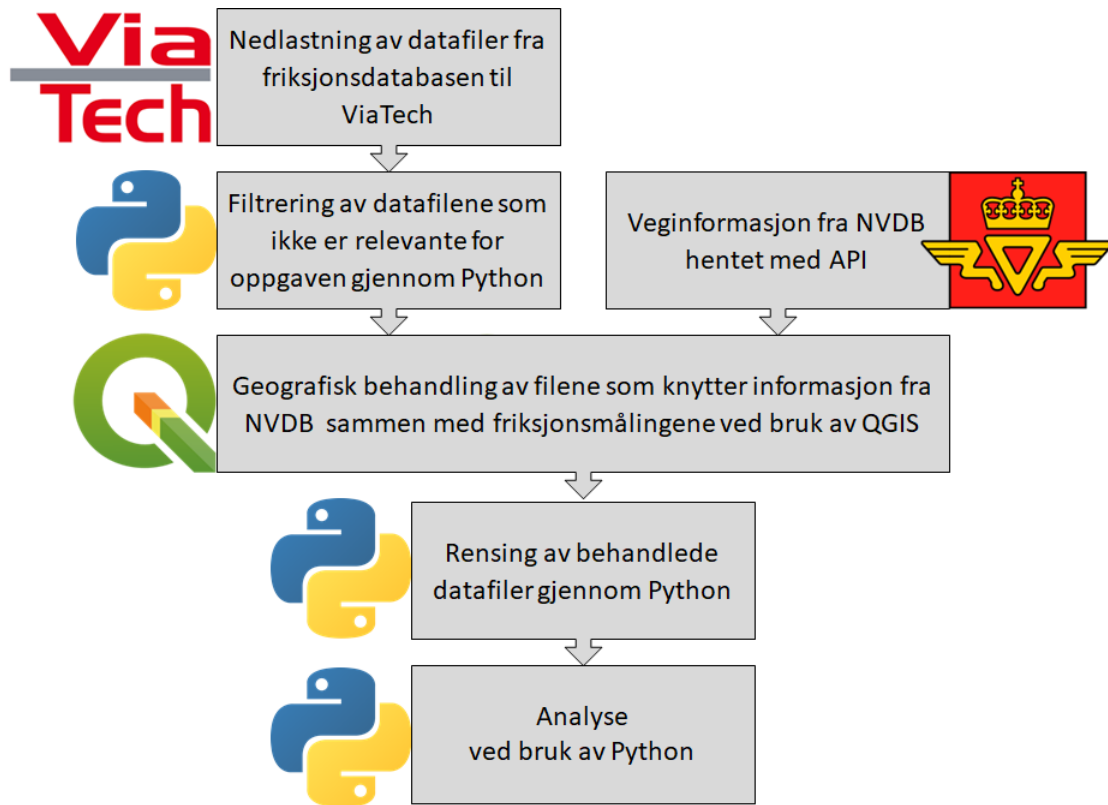
```

Figur 3.1: Oversikt for mappestrukturen som brukes for kodene i oppgaven.

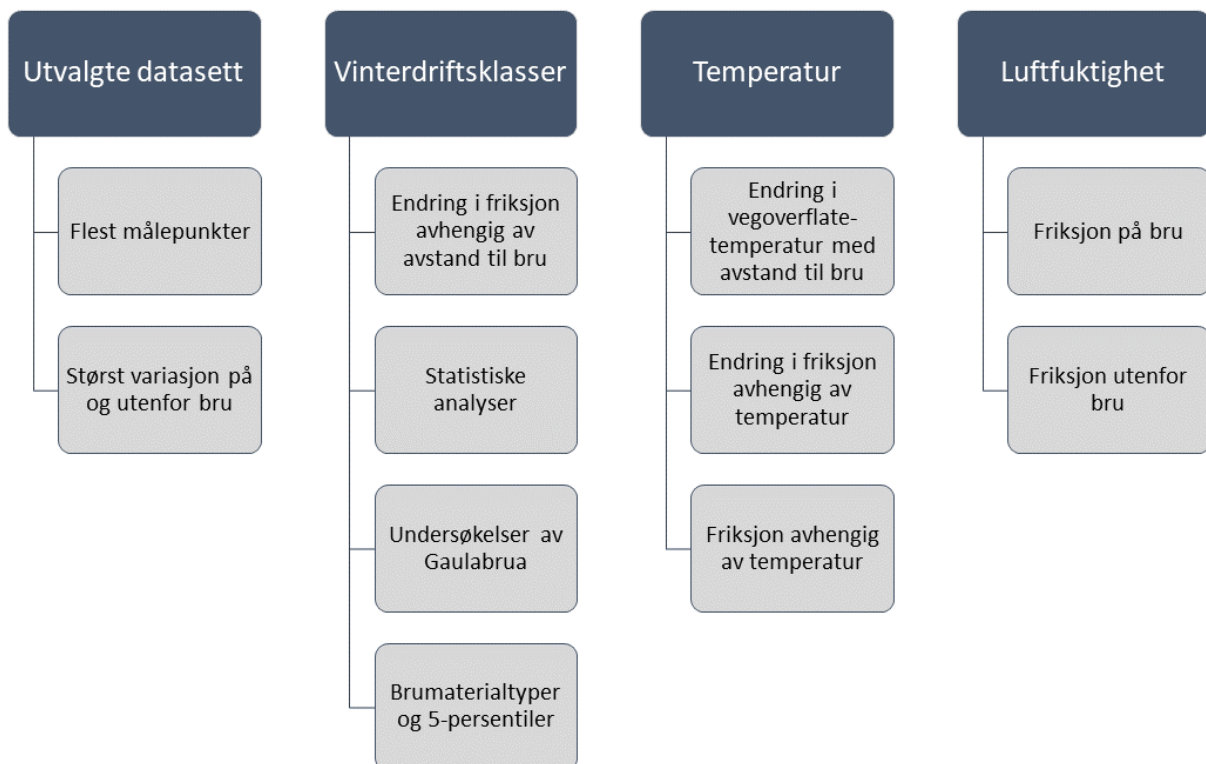
3.2 Oversikt

Metoden var i stor grad fire-delt. Første del omhandlet hvordan datagrunnlaget ble behandlet slik at det kunne benyttes til å svare på problemstillingen. Den andre delen handlet om analyser av disse filene. Tredje del var en sensitivitetsanalyse. Fjerde og siste del vurderte resultatene fra de forrige delene opp mot dagens krav til minste horisontalkurveradius på bru.

Første del av metoden baserte seg på å knytte historiske friksjonsmålinger opp mot informasjon fra NVDB, *Nasjonal VegDataBank*, ved hjelp av GIS-verktøy, *Geografiske Informasjons-System*. Et skjematisk oppsett for denne delen av metoden er vist i Figur 3.2. Friksjons-målingene ble hentet fra ViaTechs friksjonsdatabase i form av filer på DAT-filformat. Filene ble filtrert gjennom selvutviklede koder i Python før aktuelle filer gikk videre til behandling i QGIS. Fra QGIS ble det sendt spørringer til NVDBs API for å hente informasjon om bruer, vinterdrifts-klasser og horisontalkurvatur. Målepunkter i nærheten av bruene fra NVDB arvet relevant veg-informasjon og ble skrevet ut i nye filer ved bruk av QGIS. De nye filene ble renset for feil og unyttig informasjon i Python før all relevant informasjon ble samlet i én matrise. Den endelige matrisen, heretter omtalt som *Brumatriisen*, ble deretter analysert i Python ved bruk av egenprodusert kode.



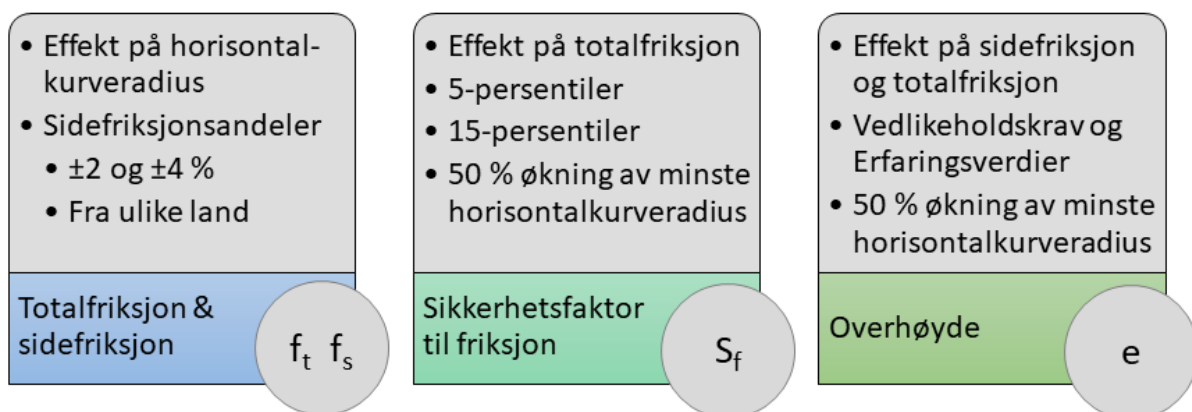
Figur 3.2: Del 1: Fremstilling av filer. Figuren viser en skjematisk oversikt over hovedstegene med relevante verktøy for hvert steg.



Figur 3.3: Del 2: Analyse av datafiler. Hovedtema med tilhørende undertema.

Andre del av metoden startet med å klargjøre Brumatriisen for analyse. Brumatriisen ble splittet opp i mindre datasett på matriseform med målepunkter tilhørende én bru. Etterpå ble det hentet ut informasjon om målepunktene i datasettene hvor det ble skilt mellom målepunkter som lå plassert på bruer og målepunkter som lå utenfor bruer. Deretter ble analysene utført som illustrert i Figur 3.3. Datasettene med flest målepunkter og datasettene med størst variasjon i friksjon på og utenfor bru ble undersøkt hver for seg. Neste analyse studerte vinterdriftsklasser. Det ble undersøkt endringer i friksjon med avstand til bru, statistikk for friksjonsverdiene, egne analyser av Gaulabrua, og sammenheng mellom lav friksjon og brumateriale. Neste del av analysene handlet om temperatur. Det ble sett på endringer i vegoverflatetemperatur med økende avstand til bru, endring i friksjonsverdi avhengig av henholdsvis lufttemperatur og vegoverflatetemperatur, før de faktiske friksjonsverdiene ved ulike temperaturer ble gjennomgått. Til slutt ble det analysert luftfuktighet. Dette ble gjort både for punkter på bru og for punkter utenfor bru.

Tredje del av metoden var en sensitivitetsanalyse. I disse analysene ble resultatene fra del 2 satt i kontekst med endringer i parametrene til variablene som inngår i dimensjonering av horisontalkurveradius. Først ble det sett på ulike sidefriksjonsandeler og hvordan disse påvirket behovet for horisontalkurveradius ved ulike totalfriksjonsverdier. Deretter ble sidefriksjoner fra ulike land sammenlignet. Neste analyse tok for seg behovet for sikkerhetsfaktorer ved ulike totalfriksjoner dersom man dimensjonerte horisontalkurveradius for lave friksjonsverdier uten å endre friksjonen, men økte sikkerhetsfaktoren. Til slutt ble det undersøkt behov for sidefriksjon og totalfriksjon ved ulike reduksjoner i overhøyde, men uendret minste horisontalkurveradius.



Figur 3.4: Del 3: Sensitivitetsanalyse. Hvilke effekter som ble undersøkt og hvilke verdier som ble brukt.

Fjerde og siste del av metoden vurderte dagens krav til 50 % økning av minste horisontalkurveradius opp mot resultatene fra del 2 og 3.

3.3 Del 1: Fremstilling av filer

3.3.1 Valg av data

Første steg i metoden var å få tak i data som inneholdt friksjonsverdier fra vegnettet. Det ble besluttet å bruke ViaTechs friksjonsdatabase over historiske friksjonsmålinger. Bruken av denne databasen eliminerte behovet for tidkrevende datainnsamling og ga de eksisterende målingene nye bruksområder. Blant alle de tilgjengelige datafilene ble filene som var utført med RoAR,

samme målesystem som ble beskrevet i Seksjon 2.3.3, valgt ut. Denne beslutningen ville gi målinger av tilsvarende kvalitet, kun ett steg unna målesystemet OSCAR som per definisjon måler helt korrekte friksjonsverdier i Norge. Øvrige målesystemer skal i utgangspunktet ha samme kvalitet til tross for at de kalibreres opp mot RoAR-ene og dermed blir enda ett steg lenger unna OSCAR. Likevel ble de valgt bort for å begrense datagrunnlaget og holde kompleksiteten i oppgaven under kontroll. Utvelgingen av filer ble i praksis gjort inne på nettsiden som holder Viatechs friksjonsdatabase ved å søke opp målinger filtrert med bestemte systemnavn. For å finne filer med målinger utført av RoAR versjon 5 og 6 ble systemnavnene i Tabell 3.1 benyttet inne på nettsiden. Det ble ikke funnet målinger utført av et målesystem med systemnavn 4R51. Deretter ble filene som var knyttet til systemnavnene lastet ned.

RoAR	Systemnavn i friksjonsdatabase				
Versjon 5	1R51	2R51	3R51	-	5R51
Versjon 6	1R61	2R61	3R61	4R61	5R61

Tabell 3.1: Systemnavn i Viatechs friksjonsdatabase for ulike målebiler med RoAR versjon 5 og 6

Filene kom i DAT-filformat og inneholdt 28 linjer med generell informasjon om målingene, også kalt metadata, og 15 kategorier med informasjon for hvert målepunkt. Som oftest var det ikke registrert data inne i alle de 15 kategoriene. Et eksempel på en slik datafil ligger vedlagt oppgaven i Vedlegg A1. Blant metadataen var informasjon om tidspunkt for første målepunkt, og informasjon om hvilken veg i hvilket fylke målingen var utført på, mest relevant. De mest interessante kategoriene til målepunktene var for denne oppgaven totalfriksjonsverdi, lufttemperatur, vegoverflatetemperatur, luftfuktighet i de få tilfellene det var registrert verdier, og lengde- og breddegrader til målepunktene.

3.3.2 Filtrering av data

Ikke alle de nedlastede datafilene som inneholdt friksjonsmålinger var like godt egnet for en oppgave om vinterforhold. En filtreringsprosess ville bli nødvendig på et tidspunkt for å begrense målepunktene til å kun bestå av interessante data. Samtidig var det flere filer som ikke hadde registrerte datapunkter og dermed ikke kunne kobles opp mot bru-informasjon eller brukes i analyser. Enkelte filer som inneholdt datapunkter hadde også blitt utført på steder utenfor vegnettet, slik som gang- og sykkelveg og flyplasser. Sistnevnte ville resulterte i feilmeldinger dersom det ble forsøkt å hente data fra NVDB for å koble sammen med friksjonsfilene. En filtreringsprosess i forkant av sammenkoblingen mot NVDB ble derfor utviklet for å fjerne filer som ville gitt feilmeldinger, samt redusere mengden data som måtte kjøres gjennom den tidkrevende prosessen med å knytte filene sammen med data fra NVDB. Det ble satt opp 6 testkriterier, og dersom filene bestod alle testene ble de tatt med videre. En komplett oversikt over koden som ble laget for formålet er lagt ved oppgaven som Vedlegg B1.

Første kriterium var at filen inneholdt friksjonsmålinger. Andre kriterium var at målingene strekte seg over 300 meter. Tredje kriterium var at det skulle finnes målepunkter i filen som hadde lufttemperatur eller vegoverflatetemperatur som var lavere enn 1°C. Fjerde kriterium var at det fantes målepunkter med ulike lengde- og breddegrader. Femte kriterium var at friksjonsverdiene ikke var konstante for alle målepunktene. Sjette og siste kriterium var at målingene var utført på en europaveg, riksveg, fylkesveg eller kommunalveg som var registrert i NVDB. Til slutt

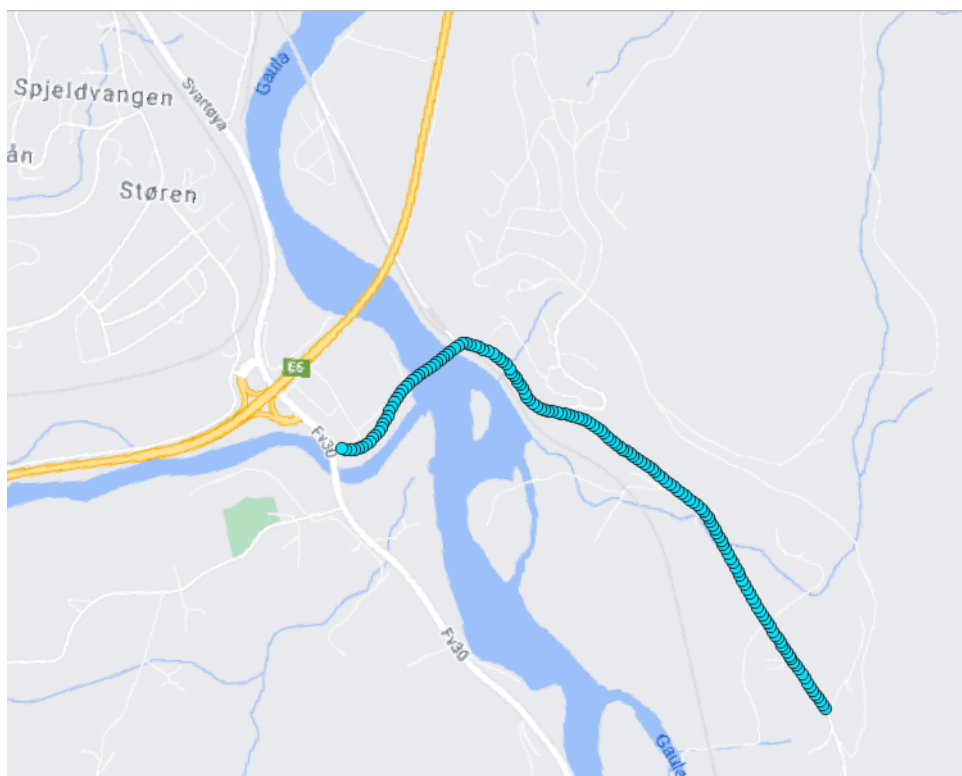
ble det laget en oversikt over filene som bestod alle kriteriene og som dermed kunne være både interessante for oppgaven og kunne knyttes til bruinformasjon fra NVDB.

3.3.3 GIS

QGIS ble brukt som oppgavens GIS-verktøy for å koble filene fra ViaTech til relevant informasjon fra NVDB om bruer, vinterdriftsklasse og horisontalkurvatur. QGIS ble benyttet på grunnlag av de innebygde funksjonene som gjorde sammenkoblingen av filene med informasjonen fra NVDB mulig, og fordi Statens Vegvesen hadde utviklet enkle måter å sende spørringer til NVDB-API-et fra dette programmet. Alle funksjonene som ble brukt i QGIS ble kjørt gjennom den integerte Python-konsollen med selvkomponerte koder. Hovedskriptet som ble brukt i oppgaven er gitt i Vedlegg B2 med alle hjelpefunksjonene gitt i Vedlegg B3.

Innlasting av filer

Første steg i QGIS var å legge inn en friksjonsfil. DAT-filformatet til filene var i praksis CSV-filer med 29 linjer metadata på toppen av filen. Ved å be QGIS hoppe over de første 28 linjene kunne filene leses som om de var på CSV-format. Målepunktene ble lastet inn i et koordinatsystem som fulgte WGS 84-referansesystemet med lengdegrader for x-verdier og breddegrader for y-verdier. Hvordan måleseriene representert av målepunkter så ut i QGIS er illustrert i Figur 3.5. I figuren er det lagt inn et bakgrunnskart fra Google Maps for visuelle grunner, kartet var ikke inkludert i metoden. I tillegg til å legge inn filen slik at målepunktene ble klare til behandling, så ble navnet på vegen som målepunktene lå på tilegnet en variabel. Denne variabelen ble brukt i neste steg for innlasting av informasjon fra NVDB.

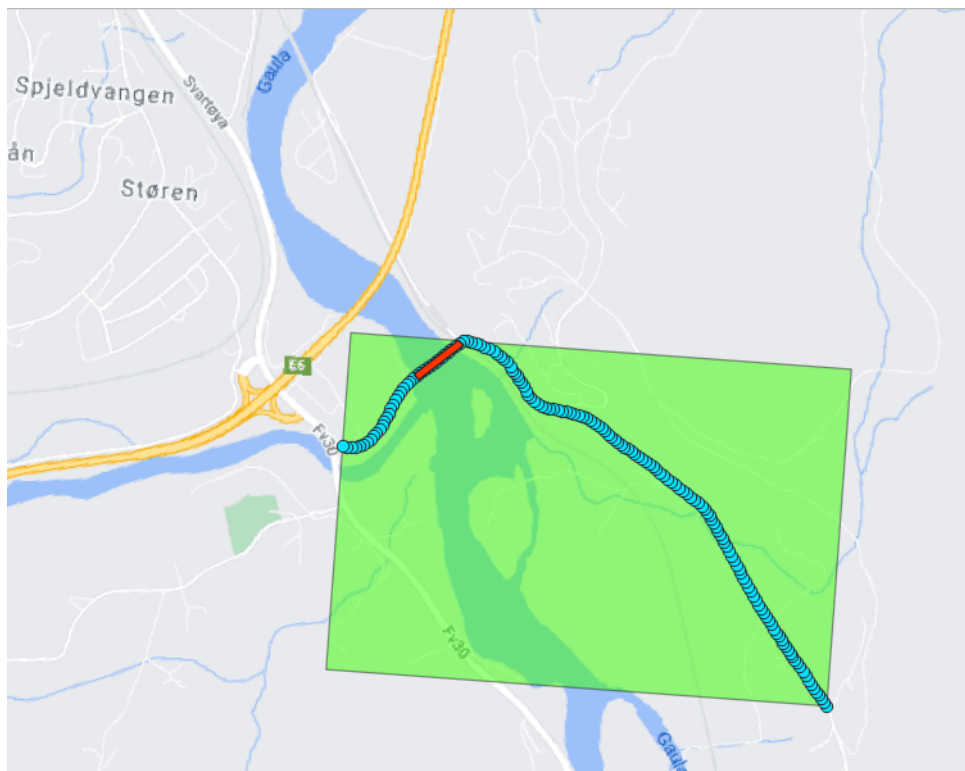


Figur 3.5: Eksempel på målepunkter i en måleserie. Skjerdump fra QGIS.

NVDB-API

Informasjon fra NVDB ble lastet inn i QGIS ved å gjøre spørringer mot NVDB-API-et. I praksis ble en funksjon, utviklet av Statens Vegvesen, brukt for å sende spørringer til NVDB-API versjon 3 for å laste inn ønsket informasjon i QGIS. Funksjonen ble lastet ned fra GitHub [46] og brukt som en svart boks. Spørringene til API-et ble spisset så mye som mulig. Dette ble gjort for å begrense belastningen på API-et og for å begrense antall elementer som måtte behandles inne i QGIS.

Spørringene til API-et fungerte ved å spesifisere ønsket fagdata etter oppbygningen til datakatalogen til NVDB [47], før det ble satt på filter for fagdataen. I resten av avsnittet er nummeret på hver fagdata og egenskap fra datakatalogen satt i parentes. For de aktuelle fagdataene ble det brukt generelle filter på alle, og spesifikke filter for hver datatype. For fagdata om bruer (60) ble det brukt to spesifikke filter. Status (11317) ble satt til trafikkert (19262) og brukategori (1263) ble satt til vegbru (7304). For vinterdriftsklasser (810) ble det filtrert bort driftsklasser (9260) som var GsA (12909), GsB (12910), eller Sideanlegg (15876), slik at kun vinterdriftsklassene på bilveger stod igjen. For fagdata om horisontalkurvatur (639) ble det ikke benyttet spesifikke filter.



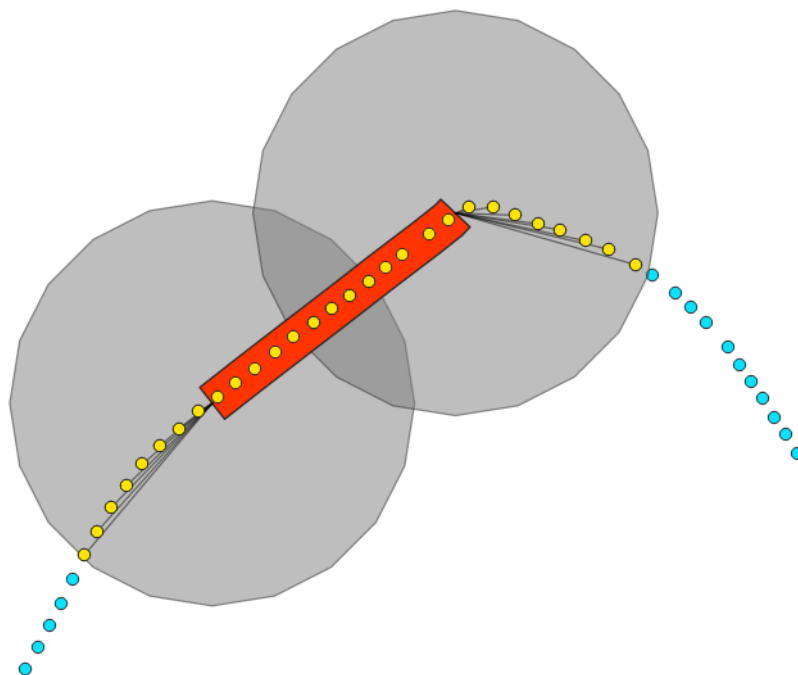
Figur 3.6: Omriss av målepunktene sammen brua Folstad hentet via NVDB-API-et. Skjerdump fra QGIS.

For alle spørringene ble det brukt to generelle filter, nærmere bestemt vegreferanse og kartutsnitt. Vegreferansen til vegen som målepunktene hadde blitt utført på var allerede lagt inn i metadataen til filene fra friksjonsdatabasen. Denne informasjonen ble hentet ut ved å lese linje 18 i hver fil. Dette filteret gjorde at kun ble spurt etter fagdata som var knyttet til den aktuelle vegen. Det siste generelle filteret var det mest omfattende. I spørringer til NVDB-API-et var det

mulig å begrense søkeområdet til bestemte kartutsnitt. Derfor ble det generert et firkantet omriss rundt målepunktene som vist i Figur 3.7. Ved å hente ut koordinatene til hvert hjørne kunne det ønskede kartutsnittet som skulle undersøkes i NVDB defineres. Omrisset hadde tilsvarende koordinater som de øvrige målepunktene i WGS 84-referansesystemet, og måtte derfor bli omprojisert til UTM 33-referansesystemet før det kunne bli tolket av NVDB. Både genereringen og omprojiseringen av omrisset ble gjort ved bruk av innebygde funksjoner i QGIS, mens koordinatene til hjørnepunktene ble funnet med den generelle `.extent()` funksjonen. Med alle fagdataene definert og filtrert ble spørringene sendt til NVDB og ønsket data ble returnert. I Figur 3.6 er brua Folstad som ble returnert av API-et fremhevet i rødt. Også denne figuren inkluderer Google Maps av illustrative hensyn.

Behandling av datafiler

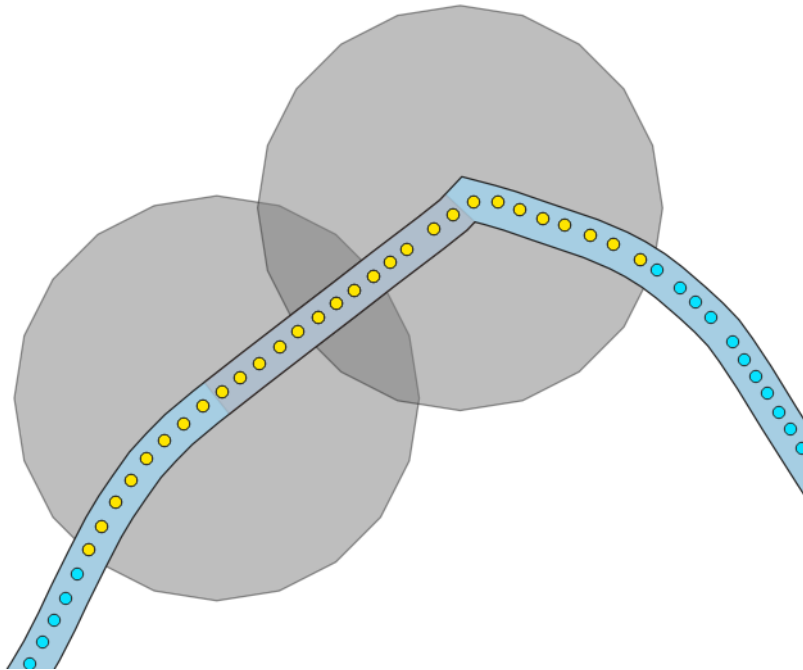
Behandlingen av målepunktene i datafilene baserte seg på at det var lastet inn elementer fra NVDB. I de tilfellene hvor det ikke fantes bruer i kartutsnittet på den aktuelle vegen ble behandlingen avsluttet før neste fil ble sendt gjennom samme prosess. For tilfeller der det var bruer tilstede, men ingen elementer med vinterdriftsklasse eller horisontalkurvatur, så ble tomme dummy-elementer lagt inn slik at koden kunne fortsette å kjøre. I disse tilfellene ble verdiene satt til NULL. Med alle elementene fra NVDB tilstede i QGIS ble elementene gitt en utstrekning i form av QGIS sin buffer-funksjon. Ettersom elementene i utgangspunktet bestod av linjer trengtes noe slingringsmonn for å avgjøre om målepunktene var på linjene eller ei.



Figur 3.7: Utvelging av målepunkter på bru og målepunkter utenfor bru med avstand til bruas endepunkter. Skjerdump fra QGIS.

Det måtte avgjøres hvilke målepunkter som lå på bruer og hvilke som lå utenfor bruer innenfor en akseptabel distanse. En visuell guide vises i Figur 3.7 for å illustrere hvordan det ble gått frem. Bruene ble gitt bredder på 10 meter i hver retning fra senterlinjene, illustrert med rødt i figuren. Punkter innenfor det røde bufferet ble definert som målepunkter på bru. Disse ble til-

egnet en avstand til bru lik 0 meter. For å finne punkter utenfor bruene ble det laget sirkulære buffere med radius lik 100 meter rundt endepunktene til bruene, markert med grått i figuren. For punktene i bufferene utenfor bruene ble det kalkulert avstand til nærmeste bru-endepunkt i luftlinje ved bruk av QGIS-funksjonen *Distance to nearest hub, line to hub*. Disse avstandene vises som streker fra de gule punktene i figuren. For alle punktene på og utenfor bru, markert i gult i figuren, ble avstanden lagt inn som en attributt. I tillegg ble generell informasjon om bruene som lå lagret i NVDB lagt inn. Dette inkluderte ID til bruene, byggverkstype, materialtype, brulengde, og byggeår. Sistnevnte var ofte ikke registret i NVDB.



Figur 3.8: Arving av vinterdriftsklasse, markert i lyseblått, for målepunkter på og utenfor bru, markert i gult: Skjerdump fra QGIS.

I tillegg ble det lagt til informasjon om vinterdriftsklasse og horisontalkurveradius på tilsvarende måte som for bruer. Linjene fra NVDB ble gitt bredder på 10 meter for å skape overlapp mellom målepunktene og bufferne. Figur 3.8 viser hvordan de gule punktene som representerer målepunkter på og utenfor bru krysset bufferet til vinterdriftsklasse markert i lyseblått. Disse målepunktene arvet vinterdriftsklassene på de aktuelle vegstrekningene. Prosessen ble deretter gjentatt for horisontalkurvatur for å arve horisontalradius.

Alle elementene lå lagret i minnet til programmet som midlertidige lag. Etter hver fil var ferdigbehandlet ble de midlertidige lagene fjernet for å gjøre klart til neste fil. Filene ble kjørt rett etter hverandre i en for-løkke.

3.3.4 Rensing av datafiler

Før analysen kunne starte måtte resultatene fra behandlingen i QGIS renses. Koden som ble brukt til rensingen ligger i Vedlegg B4. I Vedlegg A2 ligger også et eksempel på hvordan de endelige filene så ut etter behandlingen i QGIS og rensingen i Python. Rensingen gikk ut på å fjerne duplikater av målepunkter og sortere dem i kronologisk rekkefølge slik at første punkt

tilsvarte første målepunkt utenfor ei bru, og det siste målepunkt tilsvarte siste målepunkt utenfor ei bru på motsatt ende av brua. I tillegg ble enkelte filer stående som tomme etter behandlingen og måtte fjernes. Dette skjedde for tilfeller hvor målepunktene ikke ble gjort kontinuerlig og det fantes bruer på strekningene mellom målepunktene.

Som siste ledd i filtreringsprosessen, og for å gjøre den behandlede dataen mer tilgjengelig, ble alle relevante målepunkter samlet i en matrise og lagret som en separat fil. Filtreringen fjernet tilfeller hvor det ikke var informasjon om bruer knyttet til målepunktene, tilfeller med friksjonsverdier lik -1, målepunkter med avstander høyere enn 100 meter i luftlinje til nærmeste bruendepunkt, og tilfeller der hverken lufttemperaturen eller vegoverflaten var lavere enn 1°C. Koden som ble laget for å skape matrisen er gjengitt i Vedlegg B5.

3.4 Del 2: Analyse av datafiler

Denne analysen tok utgangspunkt i matrisefila som ble laget i forrige steg. Den fullstendige oversikten over hva som ble gjort finnes i vedlegg B6. Matrisefila ble laget med filnavn i første kolonne, bru-ID i neste, etterfulgt av resten av informasjonen i de andre kolonnene. Matrisen ble designet med målepunkter sortert etter filnavn, deretter sortert etter bru-ID. Matrisen ble delt opp i mindre biter for å skille målepunktene ut i mindre datasett, slik at hvert datasett bestod av alle målingene til éi bru fra samme måleserie. Denne oppdelingen ble gjort i to omganger. Først ble matrisen delt opp etter filer, og deretter ble den delt opp igjen etter sett med bruer, som illustrert i Figur 3.9. Datasett som ikke inneholdt både registrerte målepunkter på bru og utenfor bru ble forkastet. Videre i analysen ble de resterende datasettene benyttet. Det ble også laget to densitetsplott for alle de målte totalfriksjonsverdiene, ett for punkter på bru, og ett for punkter uavhengig av hvor de ble målt.

Fil	Bru	Målepunkt	...
Fil1	Bru1	Målepunkt1.1	...
		Målepunkt1.2	...
		Målepunkt1.3	...
		Målepunkt1.4	...
		Målepunkt1.5	...
	Bru2	Målepunkt2.1	...
		Målepunkt2.2	...
		Målepunkt2.3	...
Fil2	Bru3	Målepunkt3.1	...
		Målepunkt3.2	...
		Målepunkt3.3	...
		Målepunkt3.4	...
...

Figur 3.9: Oppbygning til Brumatriksen.

3.4.1 Utvalgte datasett

Det ble laget to ulike utvalg med datasett for grundigere undersøkelser. Første utvalg bestod av datasettene med flest målepunkter. Det andre utvalget bestod av datasettene hvor det var størst

variasjon mellom friksjonsmålinger på bru og friksjonsmålinger utenfor bru.

Det første datasettet ble identifisert ved å måle størrelsene på matrisene, og plukke ut de største matrisene som dermed inneholdt flest punkter. Utvalget ble begrenset til 6 datasett for å matche utvalg 2, og fordi denne mengden ga et greit innblikk i datasettene uten å bli for overveldende. For hvert av de utvalgte datasettene ble det laget spredningsplott med friksjonsverdi på y-aksen og nummeret på målepunktet på x-aksen. Plottene følger første måling før bruene, over bruene og til enden av de 100 meter lange buffersonene. Deretter ble bruene til hvert datasett søkt opp på vegkart.no for å finne spesifikk informasjon om bruene og områdene de var plassert ved.

Det andre utvalget startet med å undersøke hvor mange punkter som hadde visse differanser mellom laveste friksjonsverdi målt utenfor bru og laveste verdi målt på bru. Det ble laget en tabell med antallet datasett hvor differansen var 0 til 0,05, 0,05 til 0,1, 0,1 til 0,15, og differanser større enn 0,15. Alle disse differansene baserte seg på at friksjonen var lavest på bruene. Deretter ble det laget tilsvarende spredningsplott for alle settene med større differanse enn 0,15, men her ble også vegoverflatetemperaturen inkludert. Tilsvarende som for første sett ble disse bruene også søkt opp i vegkart.no for mer utdypende informasjon om bruene. I tillegg ble filene som datasettene kom fra søkt opp i ViaTechs Friksjonsdatabase for å utforske de fullstendige måleseriene, og for å avgjøre om datasettene skilte seg ut fra den naturlige variasjonen innad i måleseriene.

3.4.2 Vinterdriftsklasser

Analysen av vinterdriftsklassene startet med å fordele målepunktene fra de ulike datasettene etter vinterdriftsklasse. Målepunktene ble også fordelt etter plassering på bru og utenfor bru. For hvert datasett ble det også lagt til informasjon om forskjellen i friksjonsverdi til målepunktene utenfor bru relativt til målepunktene på bru. Siden målepunktene på bruene kunne variere ble det laget to referanseverdier til hvert sett; medianverdiene til målepunktene på bru, og minimumsverdiene til målepunktene på bru. Deretter ble det laget to spredningsplott hvor målepunktene utenfor bruene ble sammenlignet mot henholdsvis medianverdiene og minimumsverdiene til bruene med økende avstand. Dermed kunne endringen måles mellom settene, uavhengig om friksjonsverdiene i utgangspunktet var lave eller høye innad i settene. Disse plottene var uavhengig av vinterdriftsklasse for å si noe om datasettene på en helhetlig måte. Siden det ikke ble registrert store forskjeller mellom de to typene referanseverdier ble medianverdien brukt videre. Deretter ble det laget tilsvarende plot med endring i friksjon etter økende avstand til bruene, nå avhengig av vinterdriftsklasse.

Etterpå ble friksjonsverdiene til de ulike vinterdriftsklassene vurdert med statistiske metoder. Først ble det laget densitetsfordelinger til friksjonsverdiene avhengig av vinterdriftsklasse. Her ble det skilt mellom verdier på bru og verdier utenfor bru. Videre ble 5-persentilene, 15-persentilene og gjennomsnittsverdiene for de målte totalfriksjonene til hver vinterdriftsklasse funnet. Også her ble det skilt mellom friksjon på bru og utenfor, i tillegg til at det ble lagt inn friksjon uavhengig av plassering. De registrerte verdiene ble så vurdert for signifikans. Det ble regnet ut p-verdier for alle de nevnte verdiene hvor verdiene på bru ble vurdert opp mot verdiene utenfor bru. P-verdiene ble beregnet med tosidige, uavhengige Students t-tester. Friksjonsverdier med p-verdier lavere enn 0,05 ble vurdert som statistisk signifikante fra målingene utenfor bru. I tillegg ble standardavviket til hver vinterdriftsklasse vurdert.

I løpet av oppgaven ble Gaulabrua utenfor Støren identifisert som en ekstremitet sammenlignet med øvrige bruer med samme vinterdriftsklasse. Dette utløste tilleggsundersøkelser som ble programmert i en egen del separert fra den øvrige analysen. Fullstendig kode for Gaulabrua finnes i Vedlegg B7. Til denne analysen ble alle målepunkter som matchet Gaulabrua hentet ut, ikke bare datasett med punkter på og utenfor bru som tidligere. Tilsvarende ble gjort for alle andre bruer med samme vinterdriftsklasse som Gaulabrua, altså alle andre målepunkter med vinterdriftsklasse *B*, *høy*. Først ble det laget densitetsfordelinger som viste hvordan friksjonsverdiene til Gaula, uavhengig om de ble målt på eller utenfor bru, skilte seg ut fra målepunktene til de øvrige bruene. Deretter ble det laget densitetsfordelinger som viste ulikheten mellom målinger på bru og utenfor bru, med ett plott for Gaulabrua-punktene, og ett for de øvrige punktene. Tilsvarende som for friksjonsverdiene til vinterdriftsklassene i forrige avsnitt ble det beregnet 5-persentil, 15-persentil og gjennomsnitt til friksjonsverdiene. Dette ble gjort for målepunkter på bru, utenfor bru, og uavhengig av posisjon. Det ble utført vurdering av signifikante endringer mellom målinger på bru og utenfor bru, men denne gangen ble det benyttet ensidig uavhengig t-test for å avgjøre om endringen faktisk var mindre. Dette ble først gjort for Gaulabrua, deretter for de øvrige målepunktene med samme vinterdriftsklasse. P-verdier lavere enn 0,05 ble ansett som signifikante.

Siste analyse innenfor vinterdriftsklassene ble gjort for å undersøke brumaterialtyper. Hensikten var å avgjøre om noen materialtyper var oftere registrert med de laveste friksjonsverdiene enn ellers. Det ble benyttet målepunkter på bruer fra datasett som var fordelt i vinterdriftsklasser. Det ble tatt for seg en vinterdriftsklasse om gangen mens brumålepunktene sine friksjonsverdier ble sjekket opp mot 5-persentilverdien til brupunktens vinterdriftsklasse. Brumaterialtypen ble registrert hver gang friksjonen var lik eller lavere enn 5-persentilen. Deretter ble alle forekomstene som var i 5-persentilen summert for de ulike vinterdriftsklassene. Disse summene ble deretter sammenlignet med hvor ofte et målepunkt på bru var registrert med samme materialtype uavhengig av friksjonsverdi og vinterdriftsklasse. For å vurdere signifikans ble utvalget for 5-persentiler testet mot det generelle utvalget. Det ble benyttet binomisk fordeling og Z-skår. Estimatet på andel positive tilfeller for hver materialtype ble satt til den prosentvise forekomsten til hvert materiale i henholdsvis 5-persentil-utvalget og det generelle utvalget. Deretter ble det beregnet varians og Z-skår. Z-skårene ble brukt til ensidige tester for normalfordelinger for å få p-verdier. P-verdier lavere enn 0,05 ble ansett som signifikante.

3.4.3 Temperatur

Temperaturanalysen startet med å undersøke hvordan vegoverflatetemperaturen endret seg med økende avstand fra bruene. Tilsvarende som for vinterdriftsklassene ble det benyttet medianverdi til målepunktene på bru som referanseverdi for endringer i vegoverflatetemperatur utenfor bru. For temperaturendringene ble det tegnet opp et spredningsplott. Det ble også beregnet p-verdi fra uavhengig, tosidig t-test hvor alle punktene på bruer i datasettene ble vurdert opp mot punktene utenfor bru.

Neste analyse vurderte endringer i friksjonsverdi avhengig av temperaturer. Dette ble gjort for både lufttemperatur og vegoverflatetemperatur. Igjen ble referanseverdien satt til medianverdien på bru. Også her ble det tegnet opp spredningsplott for begge tilfellene.

Siste temperaturanalyse studerte målte friksjonstemperaturer avhengig av temperatur. Det ble laget spredningsplott for målt totalfriksjon, først avhengig av lufttemperatur, deretter avhengig av vegoverflatetemperatur.

3.4.4 Luftfuktighet

Få målepunkter hadde registrert luftfuktighet, men det ble laget spredningsplott for friksjonsverdier avhengig av luftfuktighet. Det ble skilt mellom målepunkter med høyere temperatur enn 0°C, og målepunkter med lavere temperatur enn 0°C. Det ble også skilt mellom punkter målt på bruer og punkter målt utenfor bruer, før de ble presentert i respektive spredningsplott. I begge tilfeller ble det lagett en trendlinje basert på lineær regresjon av målepunktene med lavere temperatur enn 0°C.

3.5 Del 3: Sensitivitetsanalyse

Siden hovedformålet til totalfriksjonsverdiene som ble undersøkt i denne oppgaven var å gi beslutningsstøtte til dimensjonerende verdier for minste horisontalkurveradius, så måtte også andre variabler som påvirker behovet for minste horisontalkurveradius omtales. Derfor ble det utført sensitivitetsanalyser for totalfriksjon og sidefriksjon, sikkerhetsfaktorer for friksjon, og overhøyde. For å begrense resultatkapittelet ble sensitivitetsanalysene begrenset til veger med vegklasse H1. Koden som ligger til grunn for sensitivitetsanalysene er gjengitt i Vedlegg B8. Dette vedlegget er ikke avhengig av øvrige filer eller andre koder, og det kan enkelt tilpasses andre vegklasser ved å endre på de dimensjonerende verdiene øverst i fila.

3.5.1 Totalfriksjon og sidefriksjon

Først ble behovet for horisontalkurveradius analysert med ulike kombinasjoner av totalfriksjon og sidefriksjonsandeler. Det ble tegnet opp grafer for beregnet horisontalkurveradius til veger med vegklasse H1 og totalfriksjonsverdier mellom 0 og 0,5. Totalfriksjonsverdiene ble omregnet til sidefriksjonsverdier ved bruk av omregningsfaktorer. Standardkravet for H1 veg lå på 34 % sidefriksjonsandel. I tillegg til standardkravet ble det tegnet opp grafer for sidefriksjonsandeler som var 2% og 4 % større for henholdsvis 36 % og 38 %. Tilsvarende grafer ble tegnet med 2% og 4 % lavere enn standardkravet for henholdsvis 32 % og 30 % sidefriksjonsandeler. For hver av grafene ble det plottet behov for horisontalkurveradius for bestemte totalfriksjonsverdier som ble identifisert i analysen av de historiske friksjonsmålingene. Verdiene med beskrivelser er gjengitt i Tabell 3.2.

Beskrivelse	Totalfriksjonsverdi
Erfaringsverdi til rim:	0,15
5-persentil for vinterdriftsklasse C:	0,18
5-persentil uavhengig av vinterdriftsklasse:	0,20
5-persentil for vinterdriftsklasse B, middels:	0,23
5-persentil for vinterdriftsklasse A:	0,30
Dimensjoneringskrav for H1-veg:	0,461

Tabell 3.2: Totalfriksjonsverdier basert på 5-persentilverdier som ble brukt i sensitivitetsanalyser.

Etterpå ble det laget en graf som viste hvilke horisontalkurveradier som krevdes for en H1-veg ved ulike sidefriksjonsverdier. I denne grafen ble det tegnet opp det vanlige behovet for horisontalkurveradius, samt 50 % økning av det vanlige behovet. Det ble tegnet inn punkter som benyttet sidefriksjonsverdier for de samme landene som ble undersøkt i Tabell 2.5. Dette inkluderte altså følgende land med følgende sidefriksjonsverdier: Norge = 0,157, Sverige = Danmark = 0,13, USA = Montana = Canada = 0,14, og Australia = 0,16.

3.5.2 Sikkerhetsfaktor for friksjon

Under sensitivitetsanalysen til sikkerhetsfaktorer ble det beregnet hvilke sikkerhetsfaktorer som krevdes for å gjøre om vegklasse H1 sin dimensjonerende verdi for totalfriksjon til ulike observerte totalfriksjonsverdier. Det ble vurdert tilfeller for 5-persentiler og 15 persentiler. I tillegg ble det analysert behov for sikkerhetsfaktorer dersom horisontalkurveradiusen var 50 % større enn den minste horisontalkurveradiusen til vegklassen.

Først ble det undersøkt bruk av 5-persentiler for de historiske målingene sine totalfriksjoner, oppsummert i Tabell 3.2. Deretter ble det undersøkt bruk av 15-persentiler fra de historiske friksjonsmålingenes totalverdier. Som et mål på dimensjonerende totalfriksjonsverdi for vegklasse H1 som baserte seg på 15-persentil, så ble den forrige utgaven av håndbok V120 [48] og fartsgrense lik 80 km/t benyttet. Hvilke tallverdier for 15-persentiler som ble benyttet er oppsummert i Tabell 3.3.

Beskrivelse	Totalfriksjonsverdi
Erfaringsverdi til rim:	0,15
15-persentil for vinterdriftsklasse C:	0,23
15-persentil uavhengig av vinterdriftsklasse:	0,27
15-persentil for vinterdriftsklasse B, middels:	0,33
15-persentil for vinterdriftsklasse A:	0,35
Dimensjoneringskrav for H1-veg etter forrige utgave av håndbok V120: [48]	0,55

Tabell 3.3: Totalfriksjonsverdier basert på 15-persentilverdier som ble brukt i sensitivitetsanalyser.

3.5.3 Overhøyde

Overhøyden startet med å finne vedlikeholds kravene for vegklasse H1 sin minste horisontalkurveradius og samme radius med 50 % økning. Det ble registrert grenser på henholdsvis 6,5 % og 5,7 % Deretter ble erfaringsverdier, som introdusert i Seksjon 2.2.3, definert som vedlikeholdskrav - 2 % for denne vegklassen, altså 4,5 % og 3,7 %.

Deretter ble det analysert hvor stor sidefriksjon som krevdes for å håndtere sidekreftene ved lave overhøydeverdier. Det ble også omregnet sidefriksjon til totalfriksjon etter sidefriksjonsandelen på 34 % for vegklasse H1. Samme analyse ble gjentatt med 50 % økning i minste horisontalkurveradius. Resultatene ble oppsummert i egne grafer med punkter for standardverdier, vedlikeholdskrav og erfaringsverdier.

3.6 Del 4: Vurdering av dagens krav

Siste del av oppgaven vurderte hvor godt dagens krav til 50 % økning av minste horisontalkurve radius fungerte til å håndtere de observerte totalfriksjonsverdiene fra de historiske friksjonsmålingene. I alle undersøkelsene ble minste horisontalkurve radius beregnet med vegklassenes standardiserte verdier for hastigheter, sidefriksjonsandeler og sikkerhetsfaktorer for friksjon.

Første undersøkelse beregnet minste horisontalkurve radius for alle vegklassene for hovedveger og øvrige hovedveger. Det ble beregnet minste horisontalkurve radius for hver av vegklassene, og i disse beregningene ble også dimensjonerende verdier for overhøyde benyttet. For hver vegklasse ble den minste nødvendige horisontalkurve radiusen beregnet med standardisert totalfriksjon. Disse verdiene ble etterpå rundet opp til håndbok N100s minsteverdier [4], og håndbokens verdier ble deretter brukt som sammenligningsgrunnlag mot de neste beregningene. Videre ble det beregnet behov for minste horisontalkurve radius for samme totalfriksjonen som i sensitivitet analysen, Tabell 3.2, ekskludert dimensjoneringskravet for H1 veg. De ulike behovene for minste horisontalkurve radius ble til slutt sammenlignet mot håndbokverdiene.

Siste undersøkelse beregnet behov for horisontalkurve radius og den prosentvise økningen fra håndbokverdiene etter ulike scenarioer. Første scenario ble definert som meget ugunstig med rimfrost tilsvarende totalfriksjon på 0,15, og avvik fra dimensjonerende verdier i overhøyde. For vegklasse H5 og H3 ble overhøyden satt lik vedlikeholdskravene til håndbokverdiene for minste horisontalkurve radius. Dette tilsvarte henholdsvis 5,5 % og 2,83 %. For de øvrige vegklassene ble det trukket fra 2 % fra vedlikeholdskravene da det ble forventet mer enn 2 % avvik fra vedlikeholdskravene i mer enn 15 % av tilfellene for disse vegklassene slik det ble presentert i Seksjon 2.2.3. For vegklasse H1, Hø1 og Hø3 tilsvarte dette en overhøyde på 4,5 %.

Det andre scenarioet ble definert som ugunstig med rimfrost, men standardiserte verdier for overhøyde. Dette tilsvarte totalfriksjoner lik 0,15 for alle vegklassene.

Tredje og siste scenario ble definert som noe ugunstig og benyttet 5-persentilverdier til vinterdriftsklasser som kunne forventes å finne på de ulike vegklassene. Det ble forventet vinterdriftsklasse *B*, *middels* for vegklassene H5 og H3, noe som tilsvarte totalfriksjon lik 0,23. For vegklassene Hø1 og Hø2 ble det forventet vinterdriftsklasse *C*, altså totalfriksjonsverdier lik 0,18. Vegklasse H1 var krevende å plassere i et system for vinterdriftsklasser som i utgangspunktet ikke avhenger av vegklasser. Til slutt ble det besluttet å bruke den generelle 5-persentilen til de historiske målingene, altså uavhengig av vinterdriftsklasser, med totalfriksjonsverdi lik 0,20. Dette scenarioet antok også standardiserte verdier for overhøyde.

Kapittel 4

Resultater

Resultatene i Seksjon 4.1 og 4.3 baserer seg på datasett med følgende forutsetninger:

- Hvert datasett inneholder målepunkter tilknyttet ei bru fra samme måleserie.
- Hvert datasett inneholder minst to målepunkter slik at følgende to krav oppfylles:
 1. Minst ett målepunkt er målt nøyaktig på brua.
 2. Minst ett målepunkt er målt på vegen utenfor brua, men avstanden til bruas nærmeste endepunkt overstiger ikke 100 meter i luftlinje.
- Alle målepunktene i datasettene oppfyller minst ett av følgende to temperaturkrav:
 1. Målt vegtemperatur er lavere enn 1°C.
 2. Målt lufttemperatur er lavere enn 1°C.

I disse seksjonene presenteres det verdier for 5-persentil og 15-persentil. Verdien til 5-persentilen representerer verdien som 95 % av resultatene vil være høyere enn, slik at verdien kun er lavere i 5 % av tilfellene. Dette tilsvarer persentilen håndbok V120 bruker for å avgjøre friksjonsverdien for rene og isfrie veger [5]. Tilsvarende for 15-persentilen vil verdiene være høyere i 85 % av tilfellene slik friksjonsverdiene ble håndtert i forrige utgave av V120 [48].

Før resultatene presenteres vil det gjennomgå hvor store mengder av datagrunnlaget som kunne brukes med de overnevnte forutsetningene. Datagrunnlaget bestod av 6809 filer med friksjonsmålinger hentet fra Viatechs friksjonsdatabase. Disse filene dekket alle tilgjengelige friksjonsmålinger utført med RoAR versjon 5 og 6 før påsken 2021. Fordelingene av filer per målebil med dato for første og siste registrert måling per 31.03.2021 er oppsummert i Tabell 4.1. Antallet filer i databasen øker kontinuerlig ettersom det gjøres nye målinger, og det er mulig å utvide datagrunnlaget ytterligere for fremtidige analyser ved å inkludere andre målebiler fra andre aktører.

Av dette datagrunnlaget var det 983 filer som hadde minimum ett målepunkt som oppfylte kravet om å ligge utenfor ei bru, og minimum ett målepunkt som oppfylte temperaturkravet. Disse filene dekket 3610 bruer, hvorav 2019 av disse bruene også oppfylte kravet om å ha minst ett målepunkt målt på selve brua. Noen bruer er målt flere ganger, men i ulike måleserier. Alle punktene i én måleserie som oppfyller de nevnte kravene utgjør ett datasett hver. Dette ga 2019 datasett med totalt 24219 målepunkter. Fylkene som disse målepunktene er målt i er oppsummert i Tabell 4.2. Forøvrig var 85,1 % av målepunktene målt mellom klokken 08:00 og 20:00.

Måleutstyr	Antall filer	Første måling	Siste måling*
1R51	661	18.04.2018	07.01.2019
1R61	413	12.12.2018	18.02.2021
2R51	238	28.05.2018	10.01.2019
2R61	474	18.12.2018	15.03.2021
3R51	201	29.05.2018	23.01.2019
3R61	460	23.01.2019	02.03.2021
4R61	3394	10.04.2018	10.02.2021
5R51	535	06.04.2018	13.09.2019
5R61	433	02.12.2019	12.03.2021

Tabell 4.1: Fordeling av datafiler på måleutstyr.

*Siste måling per 31.03.2021.

Fylke	Antall målepunkter	Andel [%]
Trøndelag	18539	76,5
Innlandet	3551	14,7
Nordland	825	3,4
Viken	522	2,2
Møre og Romsdal	460	1,9
Troms og Finnmark	212	0,9
Oslo	56	0,2
Agder	20	0,1
Vestfold og Telemark	19	0,1
Vestland	15	0,1
Rogaland	0	0,0

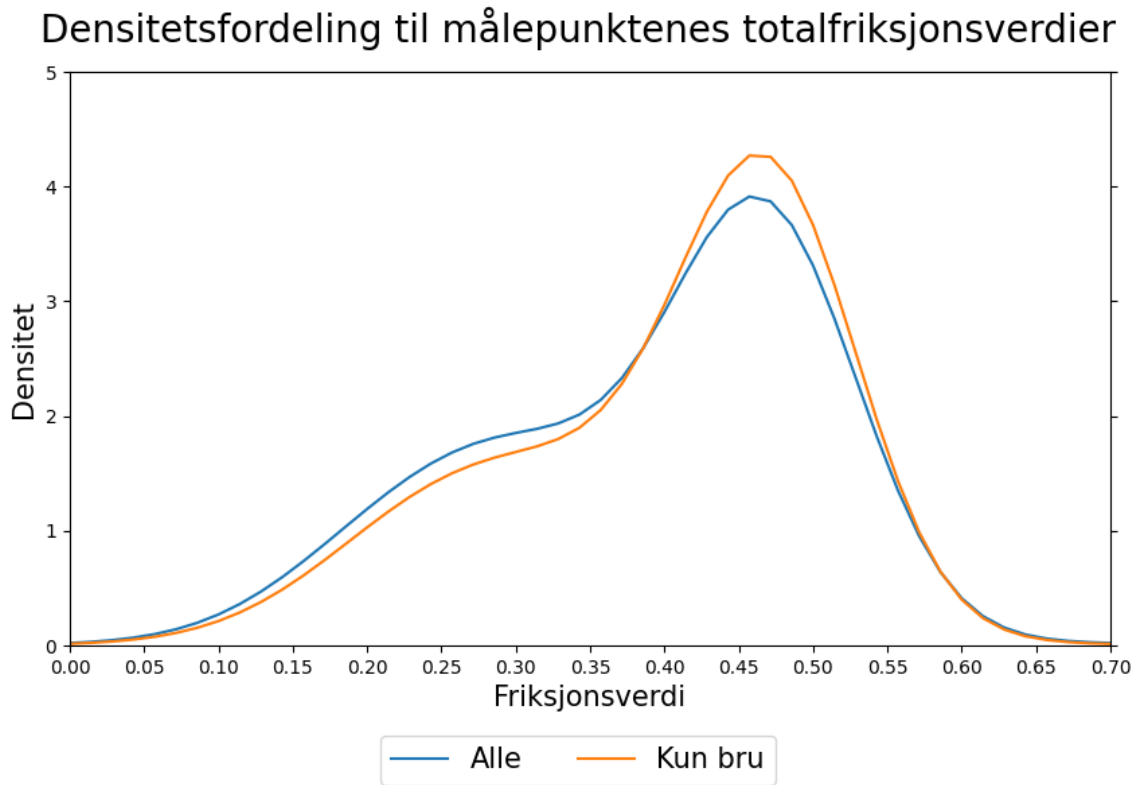
Tabell 4.2: Fordeling av målepunkter på fylker etter Norges fylkesinndeling fra 2020.

4.1 Friksjon på og utenfor bruer under vinterforhold

Hensikten med denne seksjonen er å vise hvordan friksjon og øvrige faktorer som kan påvirke friksjon opptrer på og utenfor bruer. Første underseksjon presenterer bestemte utvalg av datasettene i form av datasettene med flest målepunkter etterfulgt av datasettene med størst variasjon i målte friksjonsverdier på bru og utenfor bru. Neste underseksjon undersøker friksjon avhengig av vinterdriftsklasse. I første omgang ses det på endringer i friksjonsverdier med økende avstand utenfor bru sammenlignet med friksjonsverdiene på bru, før det i neste omgang dykkes dypere ned i målte friksjonsverdiene på og utenfor bru. Underseksjonen avsluttes med analyser over forekomstene av ulike brumaterialer for de laveste friksjonsverdiene til hver vinterdriftsklasse. Tredje underseksjon tar for seg friksjon avhengig av temperatur. Også her ses det på endring i friksjonsverdi utenfor bru sammenlignet med friksjonsverdiene på bruene, men her gjøres det med hensyn til lufttemperatur og overflatetemperatur. Det undersøkes også målte friksjonsverdier avhengig av luft- og vegoverflatetemperatur, samt en analyse over hvordan vegoverflatetemperaturen endres med økende avstand til bruene. Siste underseksjon analyserer friksjonsverdiene sett i sammenheng med luftfuktighet.

I forkant av de overnevnte analysene presenteres densitetsfordelingen til alle de målte totalfrik-

sjonsverdiene for å gi et overordnet innblik i datasettene. Figur 4.1 viser densitetsfordeling for alle punkter uavhengig av hvor de er målt, og densitetsfordeling for målepunkter kun målt på bru. Figuren viser at man kan forvente å støte på høyere friksjonsverdier litt oftere når man måler på ei bru, og at totalfriksjonsverdiene som oftest ligger rundt 0,45 til tross for vinterforholdene.



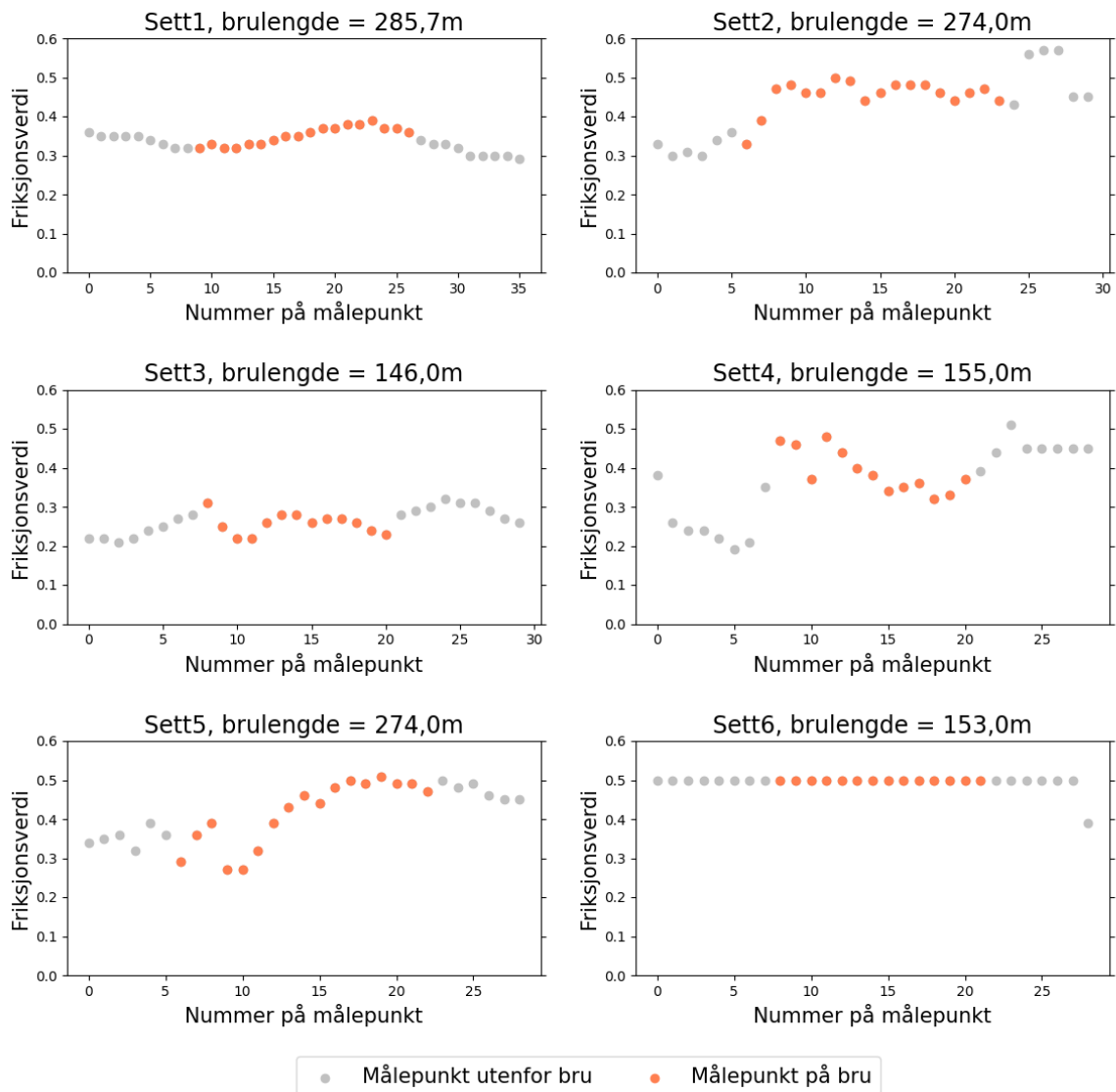
Figur 4.1: Fordelingen til totalfriksjonsverdiene til alle målepunktene i datasettene. Densiteten til målepunkter som kun er målt på bru vises med den oransje linjen. Densiteten til alle målepunktene, uavhengig av hvor de er målt, vises med den blå linjen.

4.1.1 Utvalgte datasett

Denne seksjonen tar for seg to utvalg med 6 ulike datasett hver valgt ut av de 2019 tilgjengelige datasettene. Første utvalg dekker de 6 datasettene som har flest målepunkter. Det andre utvalget består av de 6 datasettene hvor det størst forskjell mellom målte totalfriksjonsverdier utenfor bru og på bru.

Første utvalg illustrer bredden i datagrunnlaget ved å sette søkelyset på de seks datasettene med flest målepunkter. Utvalget er illustrert i Figur 4.2, og gir en idé om hvordan ulike datasett kan se ut. Det blir umiddelbart klart at totalfriksjonen ikke nødvendigvis er konstant over bruene, og at bruene gjerne kan ha høyere målte verdier enn målepunktene tatt utenfor bruene. Figurene illustrer godt hvordan friksjonen kan variere betraktelig fra et målepunkt til et annet, men også hvordan mange datasett holder relativt jevne friksjonsverdier. De 6 fremhevede settene er alle hentet fra ulike steder i Trøndelag. Sett 1 viser Verdalsbrua som ligger på E6 i en relativt slak kurve med radius lik ca 1300 over Verdalselva i Verdal. Sett 2 og 5 viser Bjørndalsbrua i Trondheim. Denne brua ligger på en rettlinje som forbinner Kolstad og Flatåsen med Tønstadkrysset og E6. Begge settene ble målt fra vest mot øst og viser lavere friksjonsverdier på vestsiden til tross for at sett 2 ble målt 11. februar og sett 5 ble målt 7. mars 2019. Sett 3 viser brua Gullberget mellom Stjørdal og Levanger som også ligger i kurve på E6. Denne brua skiller seg ut med en lav horisontalkurveveradius på kun 374 meter til tross for en fartsgrense på 80 km/t. Sett 4 viser brua Folstad utenfor Støren som krysser Gaula parallelt med E6. Selv om denne brua er tilnærmet rett så har inngangssonen en horisontalkurveveradius på fattige 205 meter i ene enden, mens man støter på 90 graders svinger i motsatt ende. Brua hjelpes noe med en lavere fartsgrense på kun 60 km/t. Friksjonsmålingene er målt fra nord-øst med 90 graders svingene mot sør-vest som ender i kurva. Sett 6 viser Sneppenbrua på E6 i Steinkjer. Også denne brua har en utsatt inngangssone i form av et kurveelement med 241 meter i horisontalkurveveradius, men brua ligger trygt plassert i sentrum av Steinkjer mellom to rundkjøringer som burde sikre lave hastigheter og god tilgang på vinterdrift.

Utvalg 1: Datasett med flest målepunkter og deres totalfriksjonsverdier



Figur 4.2: Illustrasjon over friksjonsverdier målt før bru, på bru, og etter bru. Målepunktene før og etter brua har en distanse til bruas endepunkter som varierer mellom 0 og 100 meter i luftlinje.

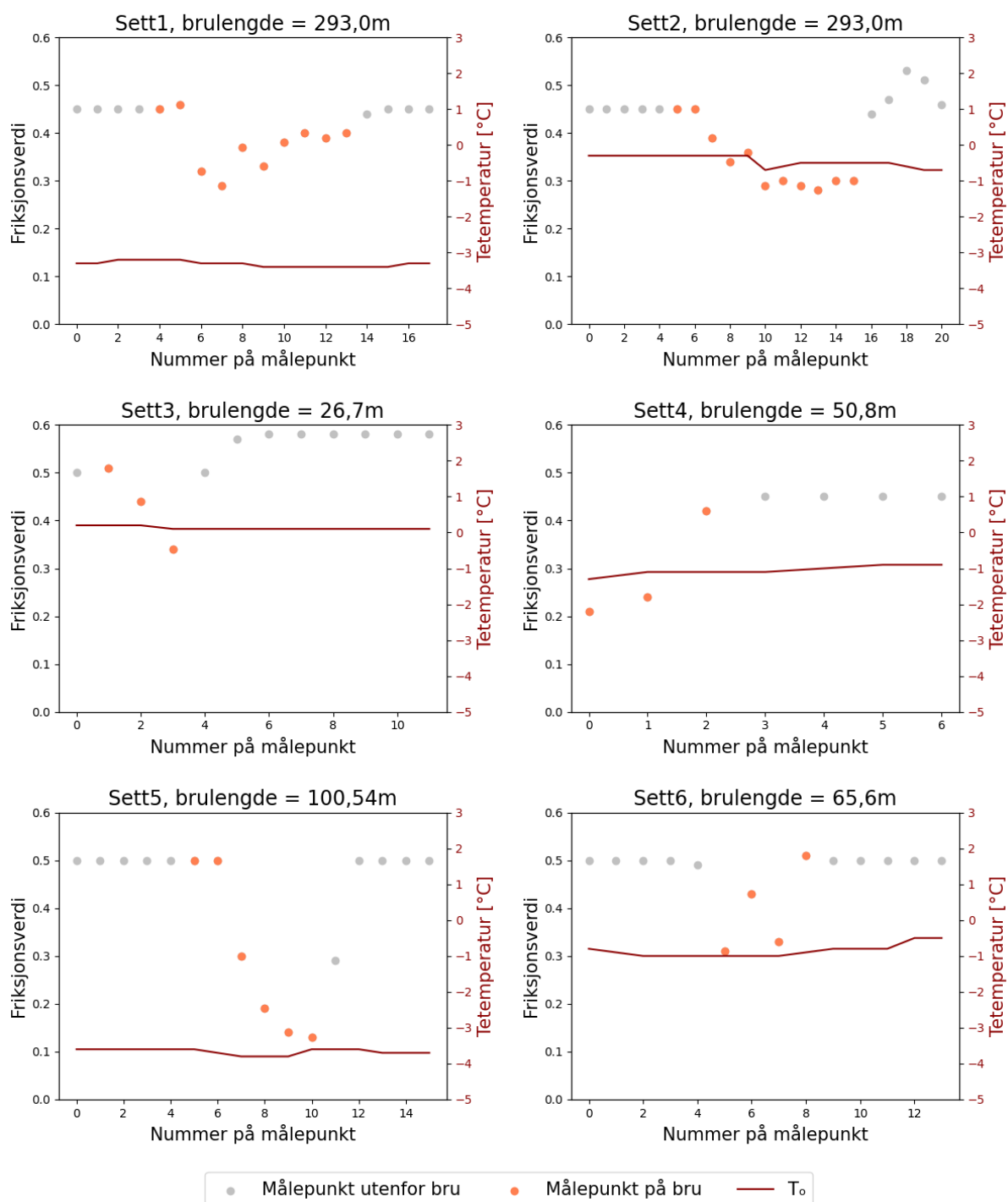
Det er få datasett hvor den laveste målte friksjonsverdien på bru er lavere enn alle målepunktene utenfor brua. Dette gjelder kun for 157 av de 2019 bruene, eller 7,8 % av datasettene. Videre er det studert hvor mye friksjonsverdien endrer seg fra målepunktene utenfor brua til målepunktene på brua for disse tilfellene. I Tabell 4.3 vises differansen mellom laveste friksjonsverdi utenfor brua og laveste friksjonsverdi på brua. Tabellen fremhever også totalfriksjonsverdiene til de to datasettene med lavest totalfriksjon på hvert sett innenfor sitt intervall. Resultatene viser at det er svært sjeldent at friksjonsverdien blir markant lavere på ei bru sammenlignet med vegen opptil 100 meter før og etter brua. Med andre ord burde ikke lave friksjonsverdier på bru komme overraskende på sjåførene med unntak av ytterst få tilfeller.

Differanse i friksjonsverdi	Antall forekomster	Minste friksjonsverdi	Nest minste friksjonsverdi
0,00 - 0,05	122	0,02	0,04
0,05 - 0,10	22	0,19	0,20
0,10 - 0,15	7	0,28	0,31
0,15+	6	0,13	0,21

Tabell 4.3: Oversikt over tilfeller hvor alle målepunkter på ei bru hadde lavere friksjonsverdi enn målepunktene utenfor brua. Tabellen viser de to laveste målte friksjonsverdiene på bruene og antall forekomster avhengig av differansen mellom minste registrerte friksjonsverdi på bru og minste registrerte friksjonsverdi utenfor bru.

Det andre utvalget tar for seg de 6 tilfellene med størst variasjon i friksjonsverdiene på og utenfor bruene. Friksjonsverdiene sammen med vegoverflatetemperaturen til de ulike settene er vist i Figur 4.3. Sett 1 og 2 viser Gaulabrua på E6 som krysser elva Gaula utenfor Støren. Det første settet ble målt fra nord mot sør 18. desember 2018, mens det andre settet ble målt fra sør mot nord 14. januar 2019. Sett 3 viser Tillerbrua over Nidelva i Trondheim som forbinder fylkesveg 6684 med fylkesveg 6680. Sett 4 viser Omundvågbrua som krysser ei kort navnløs elv på fylkesveg 715 i Fosen. Sett 5 viser brua Meosen som sikrer forbindelse over elva med samme navn for fylkesveg 17 utenfor Namsos. Sett 6 viser brua Heggset i nærheten av Tydalen som bringer trafikantene på fylkesveg 705 tørrskodd over elva Nea. En fellesnevner for bruene er at de alle spenner over elver, og som figuren viser har de jevne vegoverflatetemperaturer over hele datasettet som ligger på frysepunktet eller ned til 4 grader under. Øvrig informasjon om datasettene er gitt i Tabell 4.4. Utvalget er for lite til å trekke slutninger på generelt grunnlag, men det tyder på at vinterdriftsklasse C kan være utsatt for høye variasjoner i friksjonsverdi. Det samme kan gjelde for bruer i spennbetong. Begge disse hypotesene undersøkes dypere i Seksjon 4.1.2. Det mest kritiske tabellene viser at bruer med lave horisontalkurve radiuser kan bli utsatt for høye variasjoner i friksjonsverdi. For Meosen falt totalfriksjonen fra så å si perfekte forhold på 0,5 helt ned til 0,13, noe horisontalkurve radiuser på brua ikke er dimensjonert for. Trafikantene som var ute på brua denne tirsdagen i 17-tiden da målingen ble utført skal være glade for at de likevel kom seg trygt over. Heller ikke Gaulabrua som har en streng vinterdriftsklasse med barvegsstrategi slipper unna fall i friksjonsverdi, heldigvis med mindre kritiske fall enn det som ble registrert på Meosen. Det er likefult bekymringsverdig at Gaulabrua kommer innenfor topp 6 ved 2 anledninger.

Utvalg 2: Datasett med størst variasjon i totalfriksjon målt på og utenfor bru



Figur 4.3: Illustrasjon over friksjonsverdier målt før bru, på bru, og etter bru, samt vegoverflatetemperatur for de ulike målepunktene. Målepunktene før og etter brua har en distanse til bruas endepunkter som varierer mellom 0 og 100 meter i luftlinje.

Sett	Navn	Fartsgrense [km/t]	ÅDT
1 og 2	Gaulabrua	90	7610
3	Tillerbrua	60	3300
4	Omundvågbrua	80	1400
5	Meosen	80	3430
6	Heggset	80	700

Sett	Vinterdrifts- klasse	Horisontal- kurveradius	Materiale
1 og 2	DkB, høy	Udefinert	Spennbetong
3	DkC	Rettlinje	Stein
4	DkC	449	Spennbetong
5	DkC	282	Stål
6	DkD	366	Spennbetong

Tabell 4.4: Øvrig informasjon om de 6 datasettene i Utvalg 2.

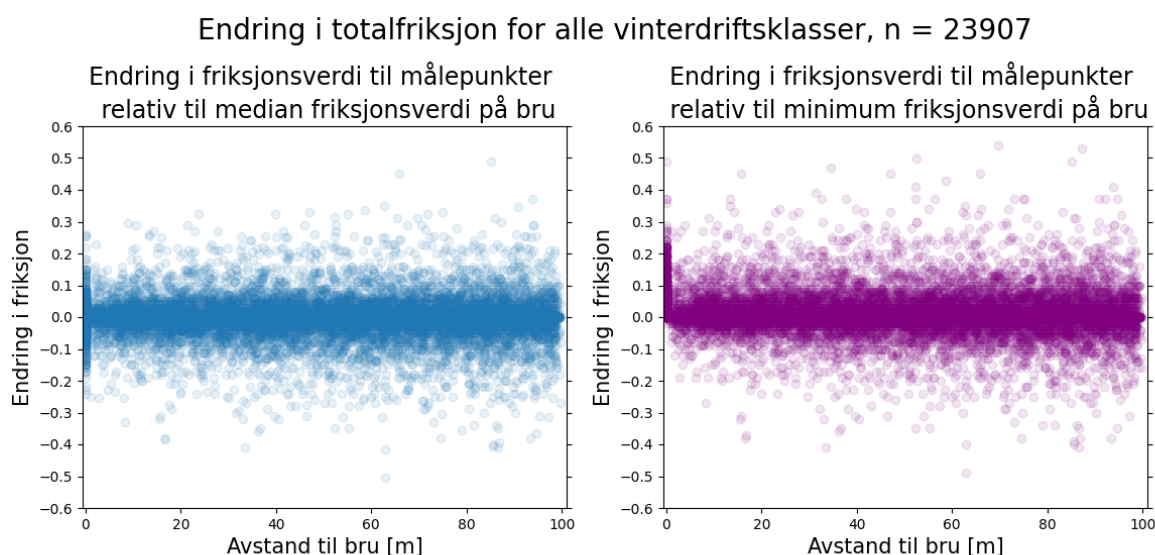
Datasettene utgjør kun en begrenset andel med målepunkter fra større måleserier. Måleseriene som hvert datasett i Utvalg 2 er hentet fra er studert nøyere i ViaTechs friksjonsdatabase. En samling av de utvalgte måleseriene er vist i Figur 4.4 hvor hvert datasett er markert i blått i de tilhørende måleseriene. For hver måleserie vises friksjonsverdi, lufttemperatur og vegoverflate-temperatur. Figuren viser at totalfriksjonsverdien kan variere mye innad i én måleserie. For de to første måleseriene som dekker Gaulabrua blir det tydelig at friksjonsverdien sjeldent synker over den målte distansen utenom på brua. Måleserien til Tillerbrua ser man at er veldig liten, noe som bidrar til å holde friksjonen jevn når man ser bort fra selve brua. I de resterende seriene er det store variasjoner i friksjonsverdiene som gjør det vanskelig å tyde om brua skiller seg ut eller om forholdene generelt sett er dårlige. I den 5. måleserien skiller ikke Meosen seg nevneverdig ut, men det er fine forhold et godt stykke på begge sider av brua som kan gjøre at sjåførene slapper litt mer av enn de burde. For Gaulabrua hvor friksjonen stort sett er god over lengre distanser før man ankommer brua, så kan en plutselig overgang til dårligere friksjonsforhold komme overraskende på sjåfører og bidra til ulykker.



Figur 4.4: Komplette måleserier hentet fra ViaTechs database for hvert av datasettene i Utvalg 2. Andre-
lene av måleseriene som utgjør datasettene er markert i blått. De røde linjene viser totalfriksjonsverdiene,
de svarte linjene viser lufttemperatur, og de grønne linjene viser vegoverflatetemperatur.

4.1.2 Friksjon avhengig av vinterdriftsklasse

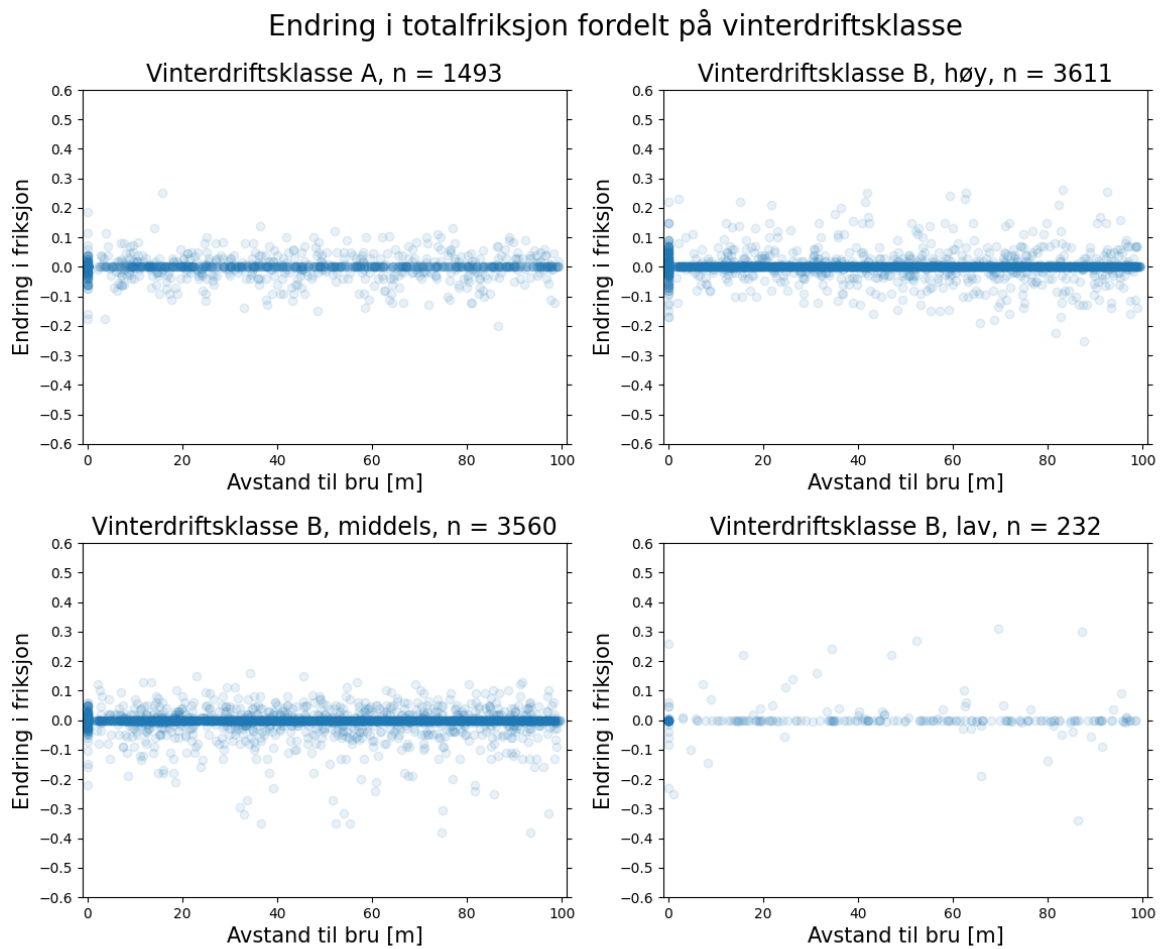
For å kunne sammenligne endringer mellom målepunkter i ett datasett med endringer mellom målepunkter et annet datasett tas det utgangspunkt i totalfriksjonsverdien på bruene. Slik kan man si at friksjonen til et målepunkt er en viss verdi høyere eller lavere enn verdien på brua, slik at friksjonsøkninger og friksjonsfall kan sammenlignes på tvers av datasettene. For lengre bruer er det flere målepunkter, og som Figur 4.2 viser kan det være store variasjoner i friksjon på brua. Figur 4.5 tar for seg utslaget i friksjonsendringen dersom man sammenligner med medianverdiene eller minimumsverdiene for totalfriksjonene på bruene. Her er alle målepunkter med nøyaktig null avstand definert som målepunkter på ei bru, men det er ikke kjent hvor på bruene disse målingene er utført. Ved å bruke minimumsverdiene til friksjonen på bruene ser man at endringene i friksjon er justert litt høyere enn dersom man bruker medianverdiene for friksjonen på bruene. Likevel er det ikke drastiske forskjeller mellom å bruke medianverdi eller minimumsverdi for friksjon på bru. For enkelhets skyld brukes medianverdi for friksjon på bru som sammenligningsgrunnlag videre i undersøkelsene.



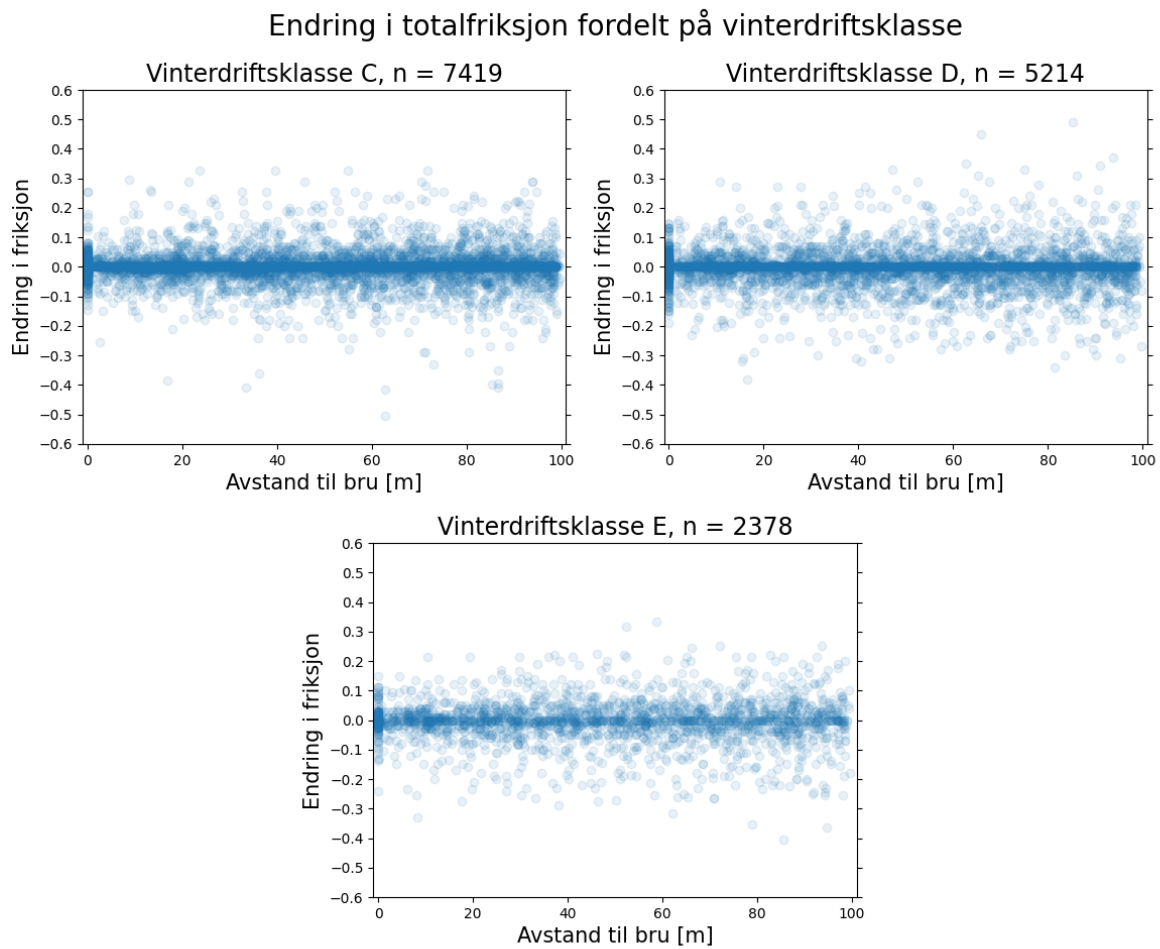
Figur 4.5: Illustrasjon over hvordan totalfriksjon endrer seg med avstand til ei bru. Til venstre er endringen i friksjonsverdi er relativ til medianfriksjon målt på bru. Til høyre er endringen i friksjonsverdi er relativ til minimumfriksjon målt på bru.

Undersøkelsene for vinterdriftsklassene ser på endringer i friksjonsverdier med økende avstand fra brua sammenlignet med median friksjonsverdi på brua. Friksjonsverdier med negativ endring tilsvarer altså målepunkter med lavere friksjon enn på brua, mens friksjonsverdier med positiv endring tilsvarer målepunkter med høyere friksjon enn på brua. Datasettene som gjør opp Figur 4.6 og Figur 4.7 er filtrert slik at det kun tas med målepunkter som har samme vinterdriftsklasse på bru og utenfor bru. Med andre ord ekskluderes målepunkter utenfor brua dersom de ikke deler samme vinterdriftsklasse som målepunktene på brua. Den første figuren består av vinterdriftsklasser med barvegsstrategi, mens den andre figuren består av vinterdriftsklasser som baserer seg på vinterveger, eventuelt i kombinasjon med barvegsstrategi. Ingen av grafene viser klare trender på at friksjonsverdien øker eller synker med avstand, og vinterdriftsklassene ser ikke ut til å påvirke trenden. Derimot ser man med det blotte øye at variasjonen til endringen i friksjonsverdi er lavere for vinterdriftsklassene i den første figuren sammenlignet med vinter-

driftsklassene i den andre figuren. Dette blir også tydelig ved å se på densitesfordelingene til totalfriksjonsverdiene for hver vinterdriftsklasse som vises i Figur 4.8.

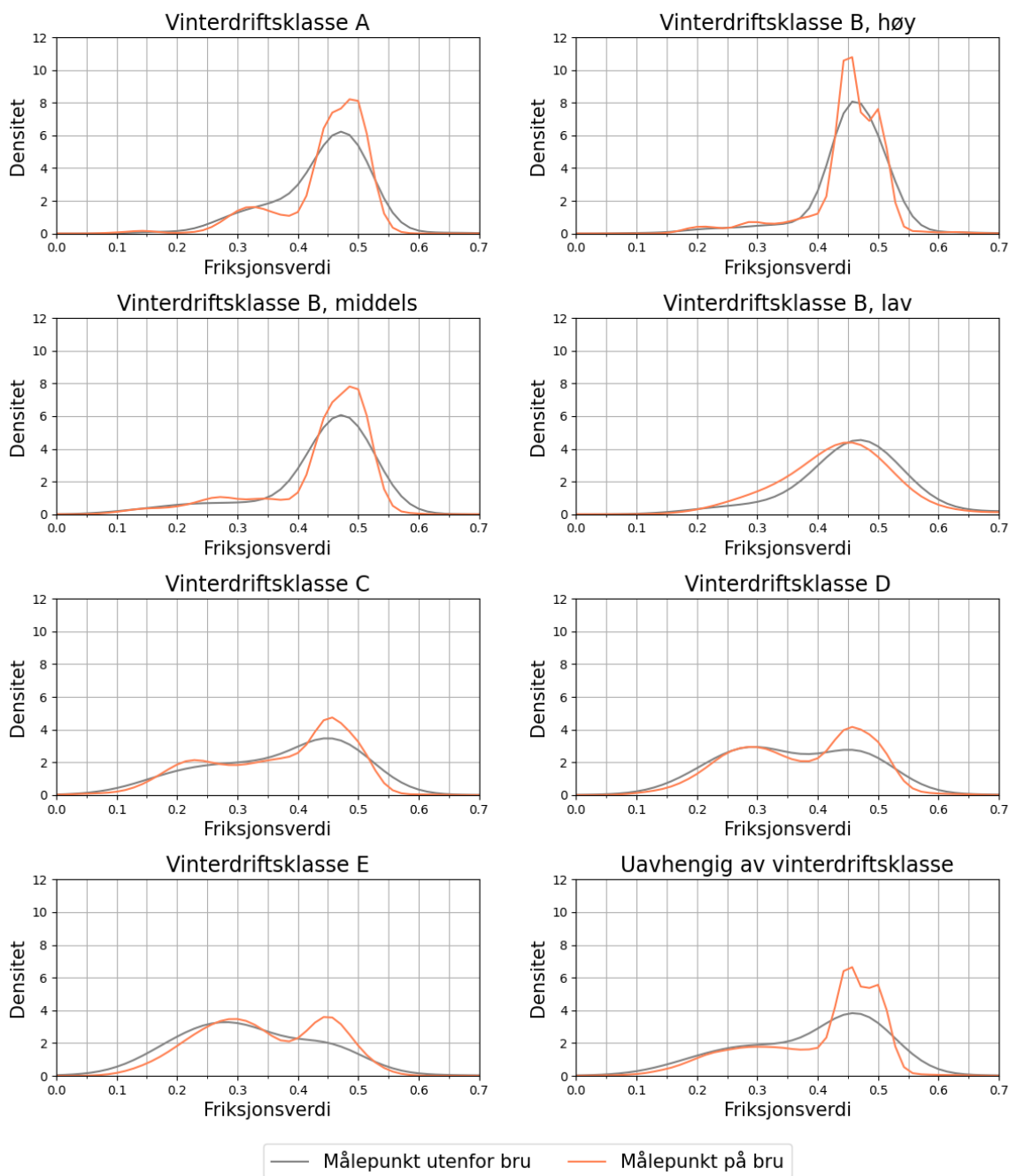


Figur 4.6: Illustrasjon over hvordan friksjon endrer seg med avstand til ei bru basert på vinterdriftsklasse-
ne A og B. Vinterdriftsklasse B er igjen delt opp i B, høy, B, middels, og B, lav. Endringen i friksjonsverdi
er relativ til medianfriksjon målt på bru.



Figur 4.7: Illustrasjon over hvordan friksjon endrer seg med avstand til ei bru basert på vinterdriftsklassene C, D og E. Endringen i friksjonsverdi er relativ til medianfriksjon målt på bru.

Densitetsfordeling til totalfriksjonsverdiene for ulike vinterdriftsklasser



Figur 4.8: Densitetsfordelinger for vinterdriftsklassenes totalfriksjoner for målepunkter målt utenfor bru og målepunkter målt på bru.

Vinterdrifts- klasse	Målepunkter	Totalfriksjonsverdi			n
		5-persentil	15-persentil	Gjennomsnitt	
Alle	På bru	0,20	0,27	0,401	4886
	Utenfor bru	0,19	0,25	0,389	19021
	Alle	0,19	0,26	0,391	23907
A	På bru	0,30	0,35	0,446	427
	Utenfor bru	0,29	0,36	0,438	1066
	Alle	0,29	0,36	0,441	1493
B, høy	På bru	0,29	0,41	0,447	923
	Utenfor bru	0,31	0,45	0,453	2688
	Alle	0,30	0,44	0,451	3611
B, middels	På bru	0,23	0,33	0,435	770
	Utenfor bru	0,21	0,35	0,435	2790
	Alle	0,22	0,34	0,435	3560
B, lav	På bru	0,30	0,36	0,451	37
	Utenfor bru	0,29	0,40	0,481	195
	Alle	0,29	0,40	0,476	232
C	På bru	0,18	0,23	0,369	1406
	Utenfor bru	0,17	0,23	0,370	6013
	Alle	0,17	0,23	0,370	7419
D	På bru	0,20	0,25	0,373	1053
	Utenfor bru	0,20	0,25	0,361	4161
	Alle	0,20	0,25	0,363	5214
E	På bru	0,19	0,23	0,344	270
	Utenfor bru	0,16	0,21	0,319	2108
	Alle	0,16	0,21	0,322	2378

Tabell 4.5: Oversikt over hvordan vinterdriftsklasser påvirker 5-persentil, 15-persentil og gjennomsnittlig totalfriksjonsverdi til målepunkter utført på bru, utenfor bru, eller uavhengig om de ble målt på eller utenfor bru. Målinger på bru som er annerledes fra målinger utenfor bru med p-verdier lavere enn 5 % signifikansnivå er uthevet i fet skrift.

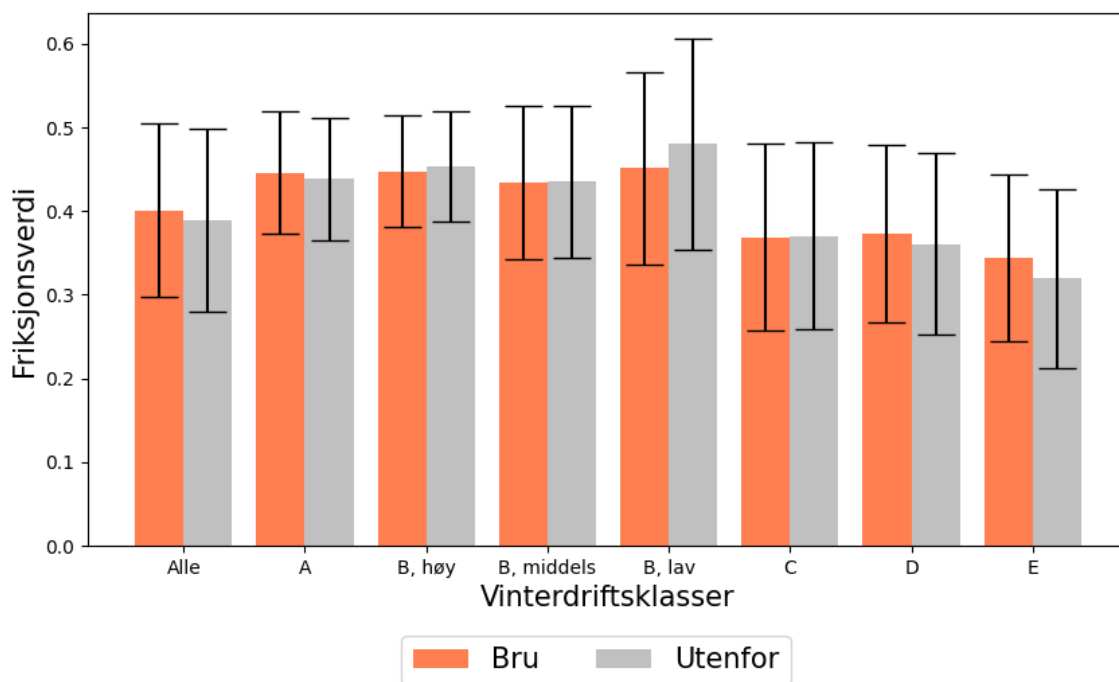
På forrige side fremheves viktige friksjonsverdier for de ulike vinterdrifstklassene og ulikheter mellom dem. Det skilles mellom målepunkter registrert på bru og målepunkter målt utenfor bru, i tillegg til et alternativ som ikke tar hensyn til hvor målepunktet er tatt. Resultatene over hvilke totalfriksjonsverdier som er registrert for de ulike vinterdrifstklassene er samlet i Tabell 4.5. Det gjøres oppmerksom på at vinterdrifstklasse *B, lav* har betydelig færre målepunkter enn de øvrige vinterdrifstklassene. Resultatene viser at dersom man ser bort i fra vinterdrifstklassene vil totalfriksjonen være signifikant høyere på bru enn utenfor. Ser man på vinterdrifstklassene med barvegsstrategi, altså vinterdrifstklasse *A, B, høy, B, middels*, og *B, lav*, så finner man gjennomsnittlige verdier i samme størrelsesorden, mens *B, middels* skiller seg ut med noe lavere 5-persentil. Noe overraskende registreres vinterdrifstklasse *B, høy* med høyere 15-persentil og gjennomsnittsverdi enn vinterdrifstklasse *A*, og det til tross for at begge disse verdiene er signifikant lavere på bru enn utenfor bru. Her bemerkes det at 280 av brumålingene til vinterdrifstklassen er målt på Gaulabrua. Den nest mest målte brua er Korporalsbrua som krysser elva Sokna med 111 målinger, mens den tredje mest registrerte brua med 56 målinger er brua Lundamo. Tilsvarende er det registrert resultater for vinterdrifstklassene med vintervegsstrategi, altså vinterdrifstklasse *C* som også kombineres med barvegsstrategi, og vinterdrifstklassene *D* og *E*. Igjen finner man gjennomsnittlige verdier i samme størrelsesorden, og gunstigere verdier for vinterdrifstklasse *D* enn *C* til tross for at dette skal være en mindre streng vinterdrifstklasse. For både vinterdrifstklasse *D* og *E* er gjennomsnittsverdien signifikant høyere på bru, mens *E* i tillegg kan vise til signifikant høyere verdier også for 5-persentil og 15-persentil. Dette er med på å gjøre 5-persentilen på bru til vinterdrifstklasse *C* mest kritisk for vinterdrifstklassene med vintervegsstrategi selv om vinterdrifstklasse *E* har lavere verdier utenfor bruene.

Grunnlaget for signifikansvurderingene kommer fra p-verdier utregnet med Students t-test. P-verdiene for 5-persentil, 15-persentil og gjennomsnitt for samtlige vinterdrifstklasser er oppsummert i Tabell 4.6. Verdiene kommer fra to-sidige, uavhengige t-tester som sammenligner de ulike utvalgene av totalfriksjonsverdier målt på bru mot totalfriksjonsverdiene målt utenfor bru.

Vinterdrifstklasse	p-verdier			n, bru	n, utenfor
	5-persentil	15-persentil	Gjennomsnitt		
Alle	0,00006	1,55E-09	4,35E-12	4886	19021
A	0,58461	0,97207	0,08318	427	1066
B, høy	0,77951	0,00471	0,02616	923	3611
B, middels	0,31199	0,86832	0,91397	770	2790
B, lav	0,24572	0,68515	0,18988	37	195
C	0,76083	0,49213	0,56706	1406	6013
D	0,90255	0,19496	0,00120	1053	4161
E	0,00336	0,00002	0,00028	270	2108

Tabell 4.6: P-verdier utregnet etter uavhengig Students t-test for å vurdere om totalfriksjonen til målinger på bru skiller seg signifikant fra målinger utenfor bru. P-verdier lavere enn 0,05 er uthevet med fet skrift.

Gjennomsnitt og standardavvik til totalfriksjonsverdier avhengig av vinterdriftsklasse på og utenfor bru

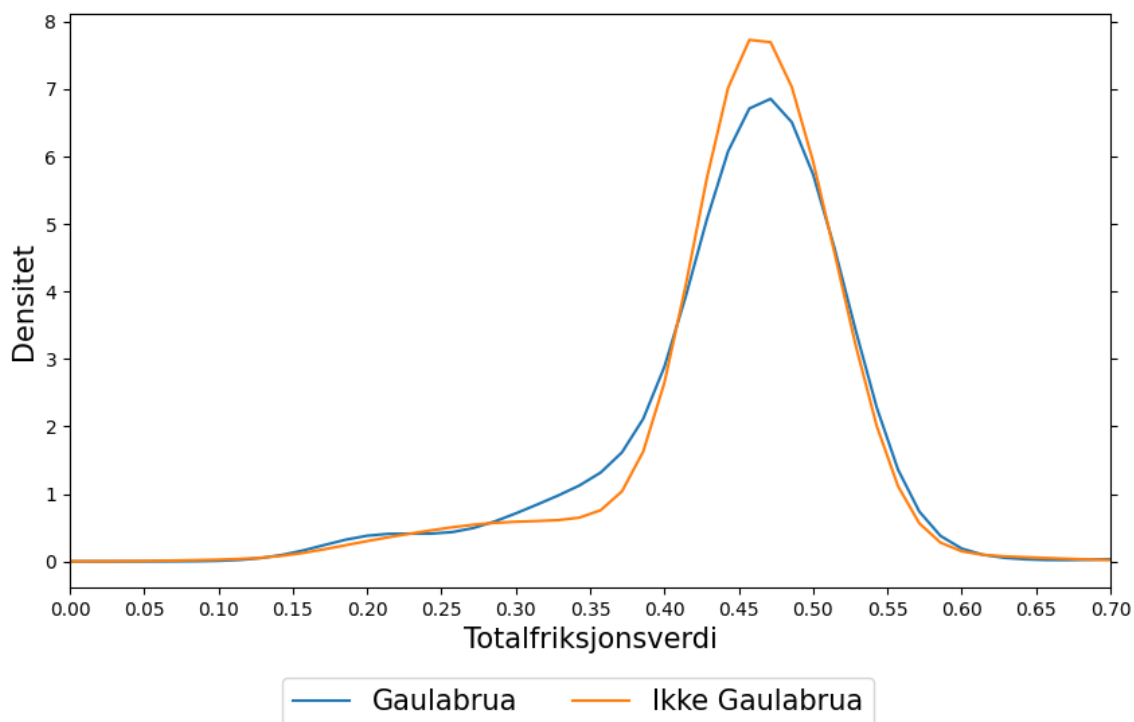


Figur 4.9: Gjennomsnittlige verdier med standardavvik for friksjon avhengig av vinterdriftsklasse og om målingen er utført på eller utenfor bru.

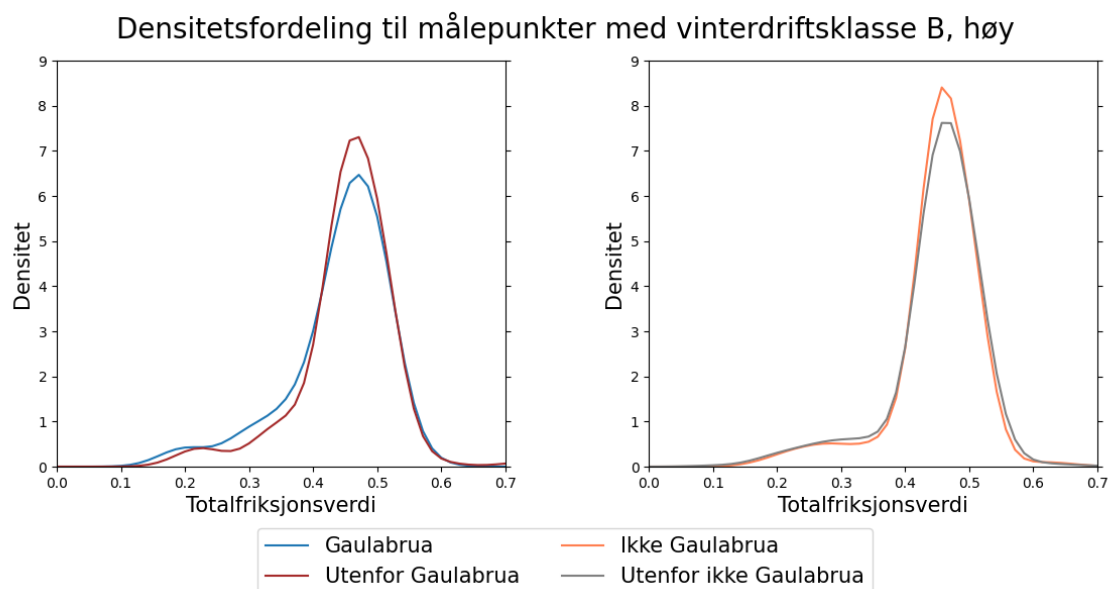
Figur 4.9 viser gjennomsnittsverdiene og standardavvikene til friksjonsverdiene målt på og utenfor bru avhengig av vinterdriftsklasser. Det største standardavvik på bru er på 0,115 og kommer fra vinterdriftsklasse B, lav. Med sine 37 målepunkter har denne vinterdriftsklassen i midlertid 38 ganger færre målepunkter enn vinterdriftsklasse C som står for det nest høyeste standardavviket på 0,112. Den laveste variasjonen kommer derimot fra vinterdriftsklasse B, høy som har er standardavvik på 0,067. Grafene viser tydelig lavere gjennomsnittsverdier med høyere standardavvik for vinterdriftsklassene som ikke baserer seg utelukkende på barvegsstrategi.

De avvikende verdiene for vinterdriftsklasse B, høy sammen med de store variasjonene som ble observert i Figur 4.3 førte til ekstra undersøkelser av Gaulabrua. Her er det forsøkt å si noe om Gaulabrua skiller seg fra de øvrige bruene med vinterdriftsklasse B, høy. Man ser umiddelbart ut fra Figur 4.10 at Gaulabrua er verdt å se nærmere på.

Densitetsfordeling til målepunkter med vinterdriftsklasse B, høy



Figur 4.10: Densitetsfordeling til alle målepunkter på Gaulabrua sammenlignet med andre bruer som har vinterdriftsklasse B, høy, men som ikke er Gaulabrua.



Figur 4.11: Densitetsfordeling til totalfriksjonsverdiene målt på og utenfor Gaulabrua, samt densitetsfordelingene til øvrige bruer med vinterdriftsklasse B, høy og deres totalfriksjonsverdier målt på og utenfor bruene.

Ved å splitte Gaulabrua i punktene som ligger på brua og målepunktene som ligger utenfor brua får man densitetsfordelingene i Figur 4.11. Tilsvarende splittelse er også gjort for de andre bruene for å vise hvordan Gaulabrua skiller seg ut. Gaulabrua opptrer annerledes på brua enn utenfor brua, og verdiene på brua skiller seg signifikant fra verdiene på de andre bruene med en p-verdi lik 0,024. Den relativt hyppige forekomsten av lave friksjonsverdier på brua senker ikke bare gjennomsnittsverdien, men som Tabell 4.7 viser senkes også 5-persentilen og 15-persentilen på brua. Med p-verdiene i Tabell 4.8 får man at både 15-persentilen og gjennomsnittsverdien er signifikant lavere på bru enn utenfor. De øvrige bruene er ikke nærheten av noe lignende. Observante lesere vil se at antallet målepunkter overstiger antallet for vinterdriftsklasse B, høy i de forrige tabellene og grafene. For akkurat denne tabellen er samtlige målepunkter for vinterdriftsklassen inkludert, ikke kun dem som tilhører datasett med minimum ett målepunkt på bru.

Bru	Målepunkter	Totalfriksjonsverdi			n
		5-persentil	15-persentil	Gjennomsnitt	
Gaula	På bru	0,29	0,37	0,440	280
	Utenfor bru	0,32	0,42	0,452	267
	Alle	0,29	0,38	0,446	547
Ikke Gaula	På bru	0,29	0,44	0,450	645
	Utenfor bru	0,28	0,44	0,449	3567
	Alle	0,28	0,44	0,449	4212

Tabell 4.7: 5-persentil, 15-persentil og gjennomsnitt for Totalfriksjonsverdier til Gaulabrua og bruer med vinterdriftsklasse B, høy som ikke er Gaulabrua. Verdier som er signifikant lavere på bru sammenlignet med utenfor er markert med fet skrift.

Målepunkter	Verdi	p-verdi
Gaula	5-persentil	0,057
	15-persentil	0,013
	Gjennomsnitt	0,026
Ikke Gaula	5-persentil	0,960
	15-persentil	0,976
	Gjennomsnitt	0,586

Tabell 4.8: P-verdier beregnet med ensidig, uavhengig t-test for Gaulabrua og bruer med vinterdriftsklasse B, høy som ikke er Gaulabrua. Verdier på bru er sammenlignet med verdier utenfor. P-verdier lavere enn 0,05 er ansett som signifikante og er uthevet med fet skrift i tabellen.

Til slutt er det undersøkt om noen brumaterialtyper fremstår som mer utsatte for lave friksjonsverdier enn andre. Denne undersøkelsen tar for seg hvor hyppig brumaterialtypene opptrer med samme eller lavere friksjonsverdi som 5-persentilen til de ulike vinterdriftsklassene. Det gjøres oppmerksom på at dette øker antallet målepunkter utover 5-persentilen da det er flere målepunkter med samme friksjonsverdi som 5-persentilen for de ulike vinterdriftsklassene. Disse forekomstene sammenlignes med antall forekomster uavhengig av friksjonsverdi for å utrede om enkelte brumaterialtyper opptrer hyppigere for lav friksjons enn ellers. Målepunkter med manglende eller uspesifiserte brumaterialtyper er satt til ukjent. Resultatene er samlet i Ta-

bell 4.9 og viser at bru av betong opptrer signifikant oftere blant de laveste friksjonsverdiene enn ellers, mens spennbetong opptrer signifikant sjeldnere.

		Betong	Stål	Spennbetong	Stein	Tre	Ukjent
Vinterdriftsklasse, antall forekomster i 5-persentil*	A	24	0	0	0	0	0
	B, høy	12	20	21	0	1	0
	B, medium	34	7	2	0	0	0
	B, lav	2	2	0	0	0	0
	C	28	42	19	3	2	0
	D	29	16	9	5	0	0
	E	11	5	0	0	0	0
	Totalt	140	92	51	8	3	0
Alle forekomster		1958	1573	1158	122	41	34
Andel i 5-persentil* [%]		47,6	31,3	17,3	2,7	1,0	0,0
Andel i alle forekomster [%]		40,1	32,2	23,7	2,5	0,8	0,7

Tabell 4.9: Oversikt over hvor hyppig ulike brumaterialtyper opptrer for friksjonsmålingene. Tabellen sammenligner forekomster i 5-persentilen* for friksjonsverdi målt på bru med forekomster hos alle friksjonsverdier målt på bru. Signifikante verdier er markert med fet skrift.

* Det brukes forekomster med samme eller lavere friksjonsverdi som 5-persentilen vist i Tabell 4.5 for hver vinterdriftsklasse. Dermed inkluderes flere elementer enn de elementene som faktisk utgjør 5-persentilen.

P-verdiene som ligger til grunn for signifikansvurderingene kommer fra binomisk fordeling. Beregningene gjentas for hvert materiale, og de positive utfallene tilsvarer antallet ganger materialene forekommer. Dermed blir sannsynligheten lik andelen i 5-persentilene og andelen blant alle forekomstene. Disse verdiene danner grunnlaget for Z-skår som brukes i ensidig test for normalfordeling for å gi p-verdier. P-verdier lavere enn 0,05 anses som signifikante. Verdiene er oppsummert i Tabell 4.10.

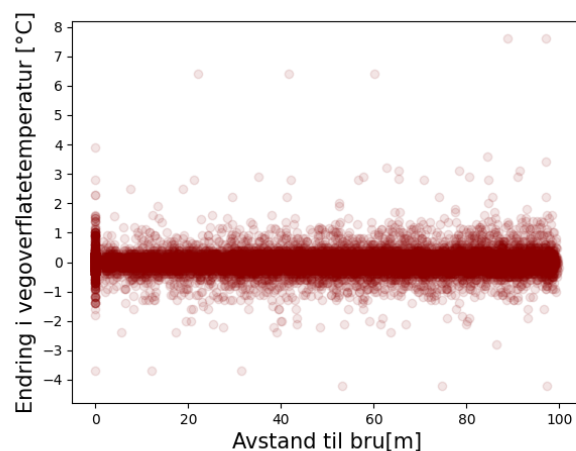
Materiale	Betong	Stål	Spennbetong	Stein	Tre	Ukjent
$\hat{p}_{5-persentil}$	0,476	0,313	0,173	0,027	0,010	0,000
\hat{p}_{Alle}	0,401	0,322	0,237	0,025	0,008	0,007
\hat{p}	0,405	0,321	0,233	0,025	0,008	0,007
S^2	0,00087	0,00079	0,00065	0,00009	0,00003	0,00002
Z	2,560	-0,321	-2,501	0,239	0,329	-1,435
$p - verdi$	0,005	0,374	0,006	0,406	0,371	0,076

Tabell 4.10: Sannsynligheter, Z-skår og p-verdier for brumaterialer etter binomisk fordeling.

4.1.3 Friksjon avhengig av temperatur

Før man kan si så mye om hvordan totalfriksjonen avhenger av temperaturen må man vite litt om hvordan temperaturen oppfører seg på og utenfor bru. Derfor er det utført en temperaturundersøkelse som viser hvordan vegoverflatetemperaturen endrer seg med avstand fra bru. I likhet med forrige seksjon brukes medianverdien målt på bru som sammenligningsgrunnlag. Figur 4.12 illustrer resultatet og viser ingen klare trender på at vegoverflatetemperaturen endrer seg nevneverdig på bruene eller utenfor. Med p-verdi på 0,48 er det ingen signifikant forskjell i vegoverflatetemperatur på og utenfor bru. Tvert i mot er trenden at vegoverflatetemperaturen stort sett er ganske jevn. De få unntakene i figuren med større endringer enn $+4^{\circ}\text{C}$ er alle fra samme datasett.

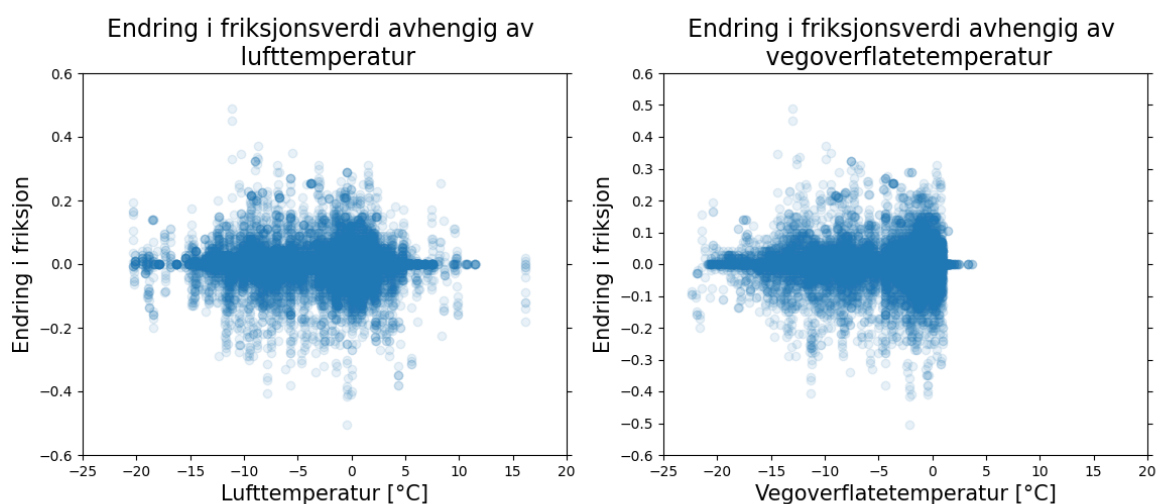
Endring i vegoverflatetemperatur med avstand til bru sammenlignet med median vegoverflatetemperatur på bru, $n = 24219$



Figur 4.12: Illustrasjon over hvordan vegoverflatetemperatur endrer seg med avstand til bru. Vegoverflatetemperaturene er relativ til median vegoverflatetemperatur målt på bru.

Neste undersøkelse studerer endring i friksjonsverdi med temperatur. Tilsvarende som før så er endringen i friksjonsverdi relativ til medianverdien til friksjonen på bru. Figur 4.13 viser hvordan endring i friksjonsverdi er avhengig av henholdsvis lufttemperatur og vegoverflatetemperatur. Fra Figur 4.12 ble det klart at temperaturen ikke varierer nevneverdig verken på eller utenfor bru. Utslag i Figur 4.13 vil dermed stort sett skje vertikalt i grafen. I figuren ser man at det er registrert noen målepunkter med temperaturer som er høyere enn frysepunktet. Som innledningen til resultatkapitlet antyder så vil punkter med høyere lufttemperatur enn 1°C samtidig ha en vegoverflatetemperatur lavere enn 1°C . Tilsvarende vil punkter med høyere vegoverflatetemperatur enn 1°C ha en lufttemperatur lavere enn 1°C . Grafene tyder ikke på at det er en klar trend mellom temperatur og endring i friksjonsverdi, men friksjonsverdiene ser ut til å kunne være mer stabile ved de laveste vegoverflatetemperaturene.

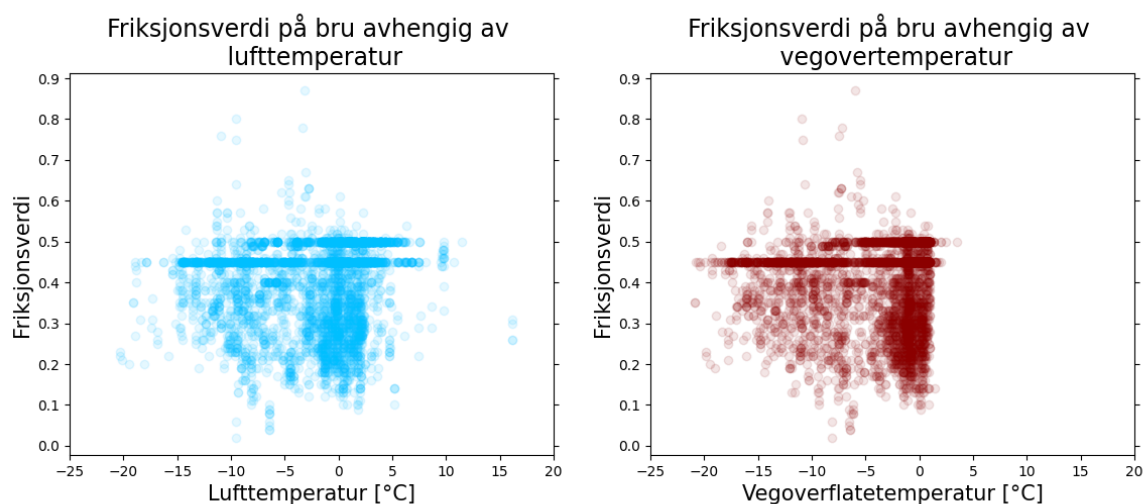
Endring i totalfriksjon avhengig av temperatur for alle målepunkter, n = 24219



Figur 4.13: Illustrasjon over hvordan friksjon endrer seg med lufttemperatur og vegoverflatetemperatur. Endringen i friksjonsverdi er relativ til medianfriksjon målt på bru.

Det er også undersøkt hva totalfriksjonsverdiene faktisk er målt til ved de ulike temperatuene. Figur 4.14 viser målte totalfriksjonsverdier for henholdsvis lufttemperatur og overflatetemperatur. Friksjonsverdiene 0,45 og 0,5 går igjen ved de fleste temperatuene, noe som ikke er overraskende ettersom disse verdiene ble målt hyppigst i datagrunnlaget. Grafene illustrerer tydelig at de fleste målepunktene ble utført rundt 0°C, og at vegoverflaten jevnt over er litt kjøligere enn lufta.

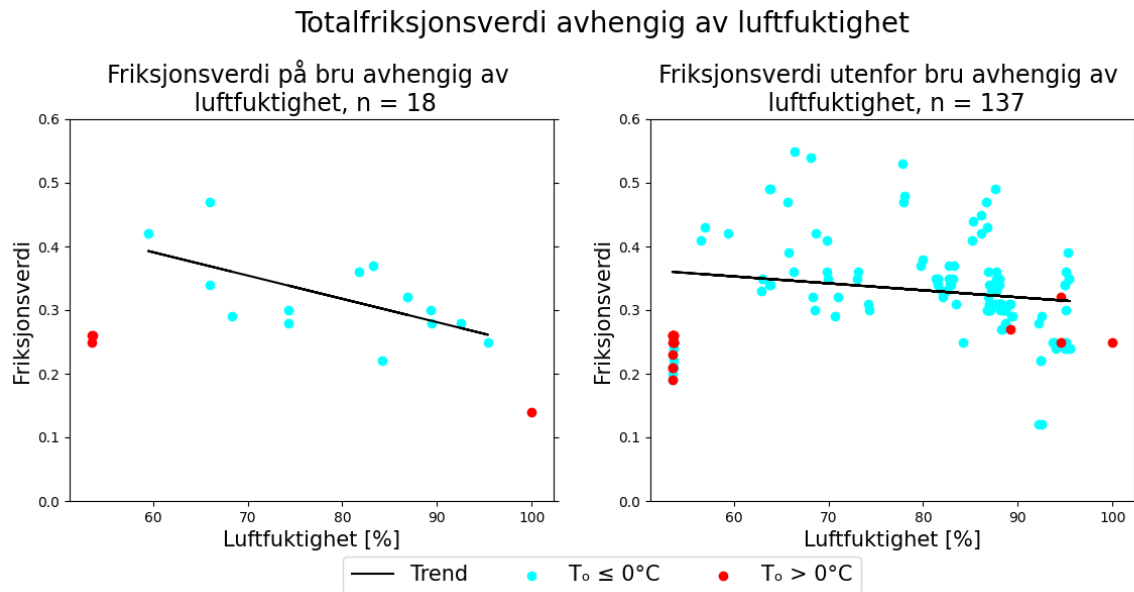
Totalfriksjonsverdi på bru avhengig av temperatur målt på bru, n = 4889



Figur 4.14: Illustrasjon over målte friksjonsverdier avhengig av lufttemperatur og vegoverflatetemperatur.

4.1.4 Friksjon avhengig av luftfuktighet

Siste undersøkelse på bru er ser nærmere på sammenhengen mellom totalfriksjonsverdi og luftfuktighet. Det bemerkes at svært få målepunkter i datagrunnlaget hadde målte verdier for luftfuktighet. Figur 4.15 viser sammenhengen mellom friksjonsverdi og luftfuktighet på bru og utenfor bru. Målepunkter hvor vegoverflatetemperaturen var lavere enn 0°C er markert i lys blå, mens målepunkter med vegoverflatetemperaturer høyere enn 0°C er markert i rødt. Grafene antyder at høy luftfuktighet har lavere friksjonsverdi enn middels til høy luftfuktighet. Trendlinjene er basert på lineærregresjon for punktene med vegtemperatur lavere eller lik 0°C og er i begge tilfeller synkende med økende luftfuktighet.



Figur 4.15: Illustrasjon over målte friksjonsverdier på bru og utenfor bru avhengig av luftfuktighet. Målepunkter med vegoverflatetemperatur lik 0°C eller kjøligere er markert i lys blå. Målepunkter med vegoverflatetemperatur varmere enn 0°C er markert i rødt. Trendlinjene kommer fra lineærregresjoner over de lyseblå punktene.

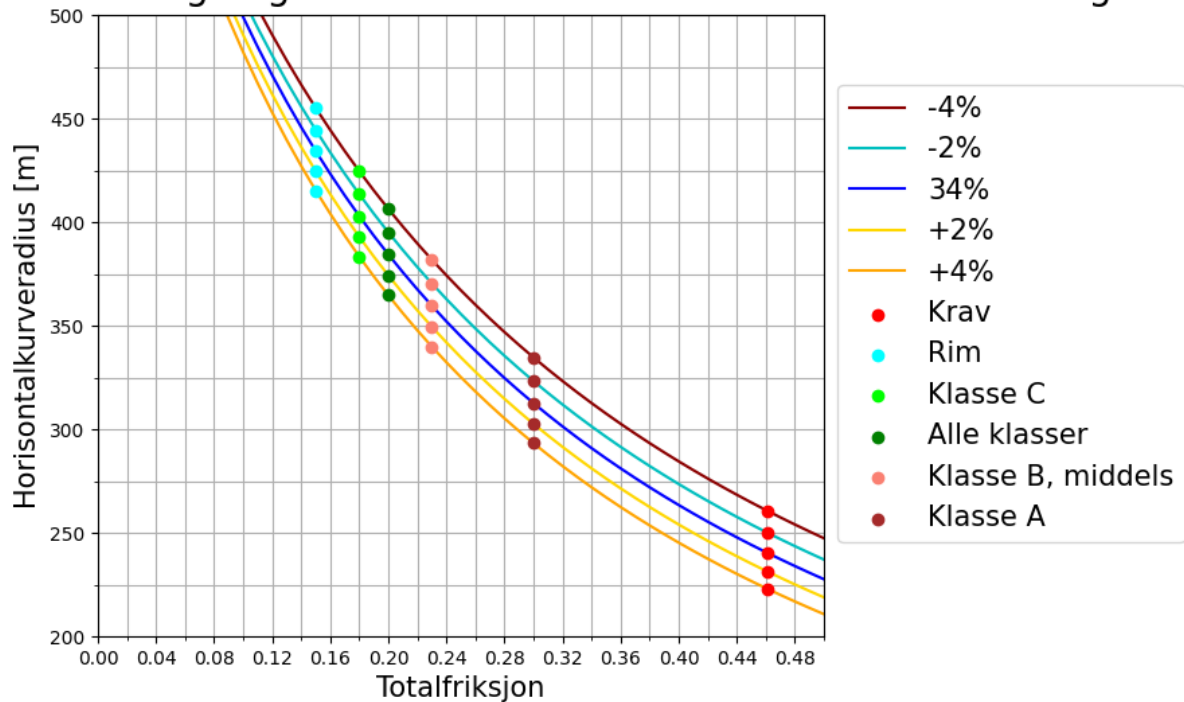
4.2 Sensitivitet til variabler som påvirker dimensjoneringen av minste horisontalkurvatur

Denne seksjonen utforsker sensitiviteten til variabler som inngår i beregningen av minste horisontalkurveradius og sammenligner dem med 50 % økning i minste horisontalkurveradius. Utregningene gjøres for vegklasse H1, og legges til grunn etter Formel 1.1. For de øvrige variablene som ikke justeres benyttes standardiserte verdier for vegklasse H1 etter håndbok V120 [5]. I oppgaven er det valgt å først se nærmere på totalfriksjon kombinert med ulike andeler av totalfriksjon som brukes til sidefriksjon påvirker horisontalkurveradius. Deretter sammenlignes sidefriksjoner fra ulike land og hvordan de påvirker den norske metoden for utregning av minste horisontalkurveradius. Videre gjøres en vurdering av nødvendige sikkerhetsfaktorer for friksjon for å imøtekomme de mest relevante 5-persentilene og 15-persentilene for totalfriksjon som ble funnet i Tabell 4.5. Avslutningsvis undersøkes behovet for sidefriksjon og totalfriksjon når det oppstår avvik i forventet overhøyde.

4.2.1 Totalfriksjon og sidefriksjon

For første vurdering beregnes horisontalkurveradiusen etter ulike kombinasjoner av totalfriksjon og sidefriksjonsandeler. Her er det valgt ut sidefriksjonsandelen for H1 veg på 34 % og sammenlignet mot ± 2 % og ± 4 % som vises i Figur 4.16. Totalfriksjoner som møter dagens krav til 0,461 er markert i rødt mens totalfriksjon for rim på 0,15 er markert i lyseblå. I tillegg er totalfriksjonsverdiene fra 5-persentilene til ulike vinterdriftsklasser markert. Figuren viser at totalfriksjonen er omvendt proporsjonal med horisontalkurveradiusen, og at endringshastigheten til grafen synker med økende totalfriksjon. Dette innebærer at dersom man skal minke de dimensjonerende verdiene for totalfriksjon vil det kreve større økninger i minste horisontalkurveradius etter hvert som totalfriksjonsverdiene blir lavere og lavere. I tillegg vil innflytelsen til sidefriksjonsandelene variere. Det gir større utslag i behovet for horisontalkurveradius dersom man reduserer sidefriksjonsandelen enn dersom man øker den tilsvarende mye. Små endringer i sidefriksjonsandeler påvirker behovet for horisontalkurveradier mer for lave totalfriksjonsverdier enn høye, men etter hvert som endringen i sidefriksjonsandel blir stor, så vil effekten bli størst for store totalfriksjonsverdier. Likevel er begge disse effektene tilnærmert neglisjerbare. Tabell 4.11 viser hvordan behovet for horisontalkurveradius er ulikt avhengig om sidefriksjonsandelen endrer seg mellom høye eller lave verdier, selv om det kun endres med 1 %. Forenklet sett kan man si at for hver prosent sidefriksjonsandelen endres fra dagens krav på 34 % for en H1 veg, så vil behovet for horisontalkurveradius endres med ca 5 meter.

Sensitivitet til totalfriksjon med ulike sidefriksjonsandeler i utregning av minste horisontalkurveradius for H1 veg



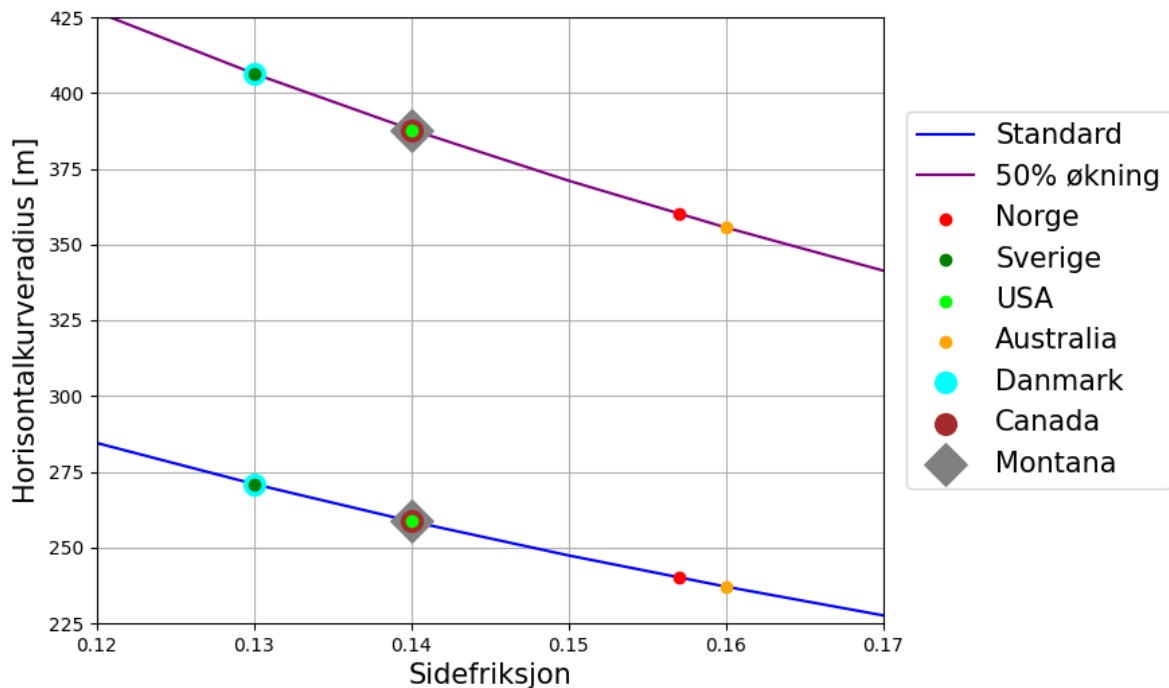
Figur 4.16: Illustrasjon over sensitiviteten til totalfriksjon i utregningen av minste horisontalkurveradius for en H1-veg med ulike sidefriksjonsandeler. Dagens krav til 34 % sidefriksjonsandel for H1-veg følger den mørkeblå linjen, mens de øvrige andelene er relative til dette kravet. Dagens krav til totalfriksjon er markert i rødt mens behovet for horisontalkurveradius for ulike vinterdriftsklasser og rim kan leses av for de andre punktene.

Totalfriksjon	Endring i sidefriksjonsandel [%]	Endring i behov for horisontalkurveradius [m]
0,15	37 → 38	-4,60
	34 → 35	-4,91
	34 → 33	+5,03
	31 → 30	+5,40
0,461	37 → 38	-4,10
	34 → 35	-4,59
	34 → 33	+4,77
	31 → 30	+5,39

Tabell 4.11: Behov for horisontalkurveradius for ulike 1 % endringer for H1-veg.

Neste undersøkelse går rett til sidefriksjonsverdier uten å vurdere sidefriksjonsandeler, og sammenligner behovet for horisontalkurve radius for dimensjonerende verdier hentet fra ulike lands håndbøker. En større oversikt over de ulike landenes dimensjonerende verdier er presentert i Seksjon 2.3.2, mens resultatet av beregningene oppsummeres i Figur 4.17. Igjen representeres dagens norske krav i rødt. Norge skiller seg ut med lavere sidefriksjonsverdier enn samtlige av de andre landene som også opplever vinterforhold regelmessig for veier med hastigheter tilsvarende H1-veg. Mellom sidefriksjonsverdier på 0,13 og 0,16 kan endringer i behov for horisontalkurve radius tilnærmes lineært. En reduksjon i sidefriksjon på 0,01 tilsvarer da en økning i behov for horisontalkurve radius på 11,3 meter for H1-veg. Naturligvis blir 50 % økning av behovet lik 17,0 meter per 0,01 endring i sidefriksjon. Hadde Norge benyttet de svenske og danske verdiene med 50 % økning vil dette tilsvart 46,3 meter høyere horisontalkurve radius på bru enn det gjør nå, eller 62,6 % økning av minste horisontalkurve radius gitt i håndboken for vegklasse H1.

Sensitivitet til sidefriksjon i utregning av minste horisontalkurve radius for H1 veg



Figur 4.17: Ulike lands sidefriksjoner og deres betydning for horisontalkurve radius.

4.2.2 Sikkerhetsfaktor for friksjon

Denne underseksjonen gjør et dypdykk i sikkerhetsfaktorer for friksjon og strekker seg utover sensitivitet. Da håndbok V120 fra 2014 [48] ble erstattet av versjonen fra 2019 [5] gikk man over til et nytt system for totalfriksjonsverdier og sikkerhetsfaktorer. Den gamle versjonen baserte seg på 15-persentiler for totalfriksjon og benyttet større og større sikkerhetsfaktorer for strengere og strengere vegklasser. I den nye versjonen benyttes derimot 5-persentiler for totalfriksjon med minimale sikkerhetsfaktorer. Utslaget sikkerhetsfaktorene til slutt gir på sidefriksjonene som brukes i dimensjoneringen av minste horsiontalkurveradius er oppsummert i Tabell 4.12, før en sammenligning presenteres i Tabell 4.13. Til grunn for sammenligningen av H1 veg brukes H4 veg fra 2014 da denne er likest med hensyn til fartsgrense og ÅDT. Sistnevnte tabell viser at totalfriksjon uten sikkerhetsfaktor naturlig nok er lavere for 5-persentilene. For sidefriksjoner med sikkerhetsfaktorer benyttes lavere verdier i dag for fartsgrensene 40 km/t og 50 km/t, men for høyere fartsgrenser, og særlig 80 km/t og høyere, er sidefriksjonen i dag vesentlig høyere.

Fartsgrense [km/t]	Sidefriksjons- andel [%]	2019: 5-persentil		
		Total- friksjon	Sikkerhets- faktor	Side- friksjon
40	39	0,638	1,00	0,2488
50	39	0,575	1,00	0,2243
60	37	0,528	1,00	0,1954
70	37	0,491	1,00	0,1817
80	34	0,461	1,00	0,1567
90	30	0,437	1,10	0,1192
100	26	0,416	1,10	0,0983
110	20	0,397	1,10	0,0722

Fartsgrense [km/t]	Sidefriksjons- andel [%]	2014, 15-persentil		
		Total- friksjon	Sikkerhets- faktor	Side- friksjon
40	39	0,77	1,10	0,2730
50	39	0,69	1,10	0,2446
60	37	0,63	1,25	0,1865
70	37	0,59	1,25	0,1746
80	34	0,55	1,50	0,1247
90	30	0,52	1,50	0,1040
100	26	0,49	1,75	0,0728
110	20	-	-	-

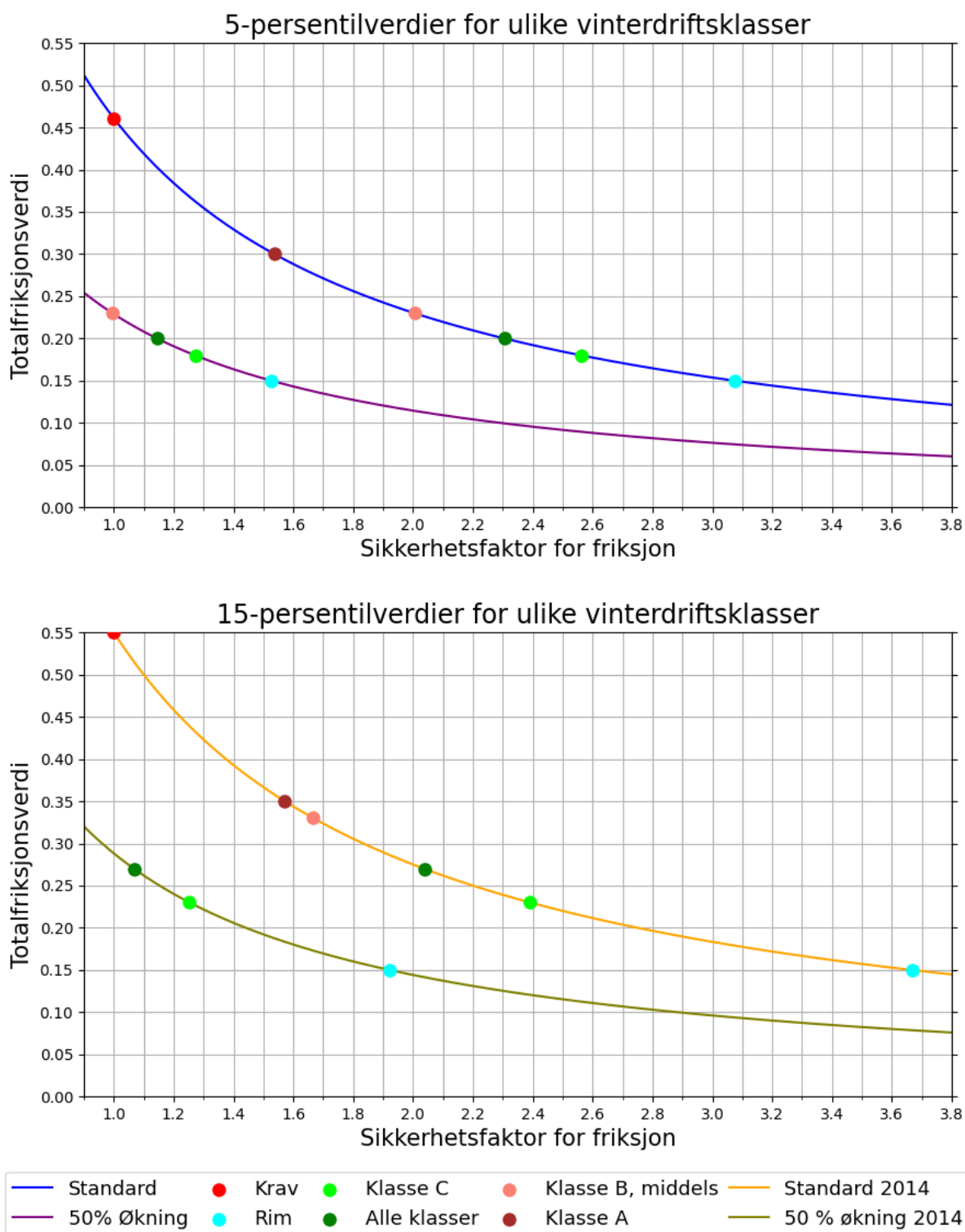
Tabell 4.12: Dimensjonerende verdier for totalfriksjon og sidefriksjon for henholdsvis 2020 og 2014.

Fartsgrense [km/t]	2019	2014	Differanse 2019 - 2014	
	Sidefriksjon	Sidefriksjon	Sidefriksjon	Prosent [%]
40	0,2488	0,2730	-0,024	-8,9
50	0,2243	0,2446	-0,020	-8,3
60	0,1954	0,1865	0,009	4,8
70	0,1817	0,1746	0,007	4,0
80	0,1567	0,1247	0,032	25,7
90	0,1192	0,1040	0,015	14,6
100	0,0983	0,0728	0,026	35,1
110	0,0722	-	-	-

Tabell 4.13: Forskjeller i dimensjonerende verdier for sidefriksjon med sikkerhetsfaktorer. Verdiene fra 2019 sammenlignes mot verdiene fra 2014.

Tilbake til sensitivitet for sikkerhetsfaktorer, både for 5-persentiler og 15-persentiler til totalfriksjon for ulike vinterdriftsklasser. Resultatene for både 5-persentiler og 15 persentiler vises i Figur 4.18. Standard dimensjonerende verdi er markert i rødt, totalfriksjon for rimfrost lik 0,15 er markert i lys blå, mens persentilene for vinterdriftsklassene er markert mellom disse verdiene i ulike farger. Sammenhengen mellom sikkerhetsfaktor og totalfriksjon er omvendt proporsjonal, og endringshastigheten synker etter hvert som sikkerhetsfaktoren økes. For å redusere totalfriksjonen med 25 % kreves dermed kun en sikkerhetsfaktor på 1,3, mens en reduksjon på 50 % krever en sikkerhetsfaktor på 2. For 5-persentilverdiene gjøres beregningene etter dagens standard, og det trengs store sikkerhetsfaktorer dersom man skal imøtekomme de ulike 5-persentilene utelukkende med sikkerhetsfaktorer for totalfriksjon. Selv med 50 % økning i minste horisontalkurveradius må enkelte av 5-persentilene håndteres med en ekstra sikkerhetsfaktor for friksjon. Derimot ser man at vinterdriftsklassene med barvegsstrategi dekkes tilstrekkelig med 50 % økning. Det samme gjelder for 50 % økning av minste horisontalkurveradius beregnet med 15-persentilen for totalfriksjon fra 2014. Også da dekkes 15-persentilverdiene til vinterdriftsklassene med barvegsstrategi. Når man ser på resultatene for 15-persentilene må man tenke over hva sikkerhetsfaktoren i grafen oppnår. Disse sikkerhetsfaktorene omgjør 15-persentilen på 0,55 til ulike 15-persentilverdier for vinterdriftsklassene. Dette er greit når man ser på erfaringsverdiene til rim da dette er den endelige totalfriksjonen man vil ta hensyn til, men for vinterdriftsklassene trengs ytterligere sikkerhetsfaktorer for å ta hensyn til at verdiene kun representerer 15-persentiler. Da kan sikkerhetsfaktorene i Tabell 4.12 benyttes. Ved å bruke samme sikkerhetsfaktor som vegklasse H4 fra 2014 på 1,5, så vil eksempelvis den samlede sikkerhetsfaktoren til vinterdriftsklasse C bli $2,4 * 1,5 = 3,8$. Dette tilsvarer en totalfriksjonsverdi under 0,15, noe som er en vesentlig lavere enn den observerte 5-persentilen på 0,18.

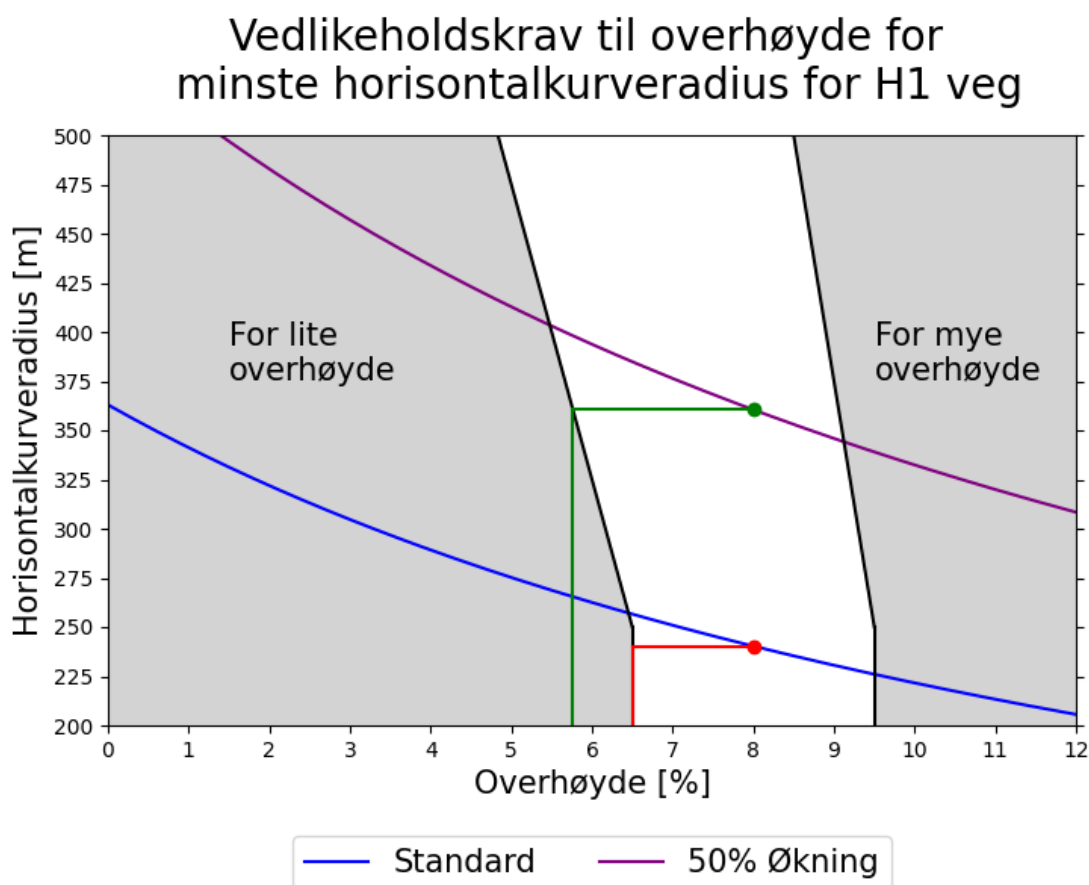
Sensitivitet til sikkerhetsfaktor for friksjon for H1 veg



Figur 4.18: Illustrasjon over hvilke sikkerhetsfaktorer som trengs for å justere kravet til totalfriksjon for H1 veg for å nå 5-persentilverdier eller 15-persentilverdier målt på bruer med ulike vinterdriftsklasser.

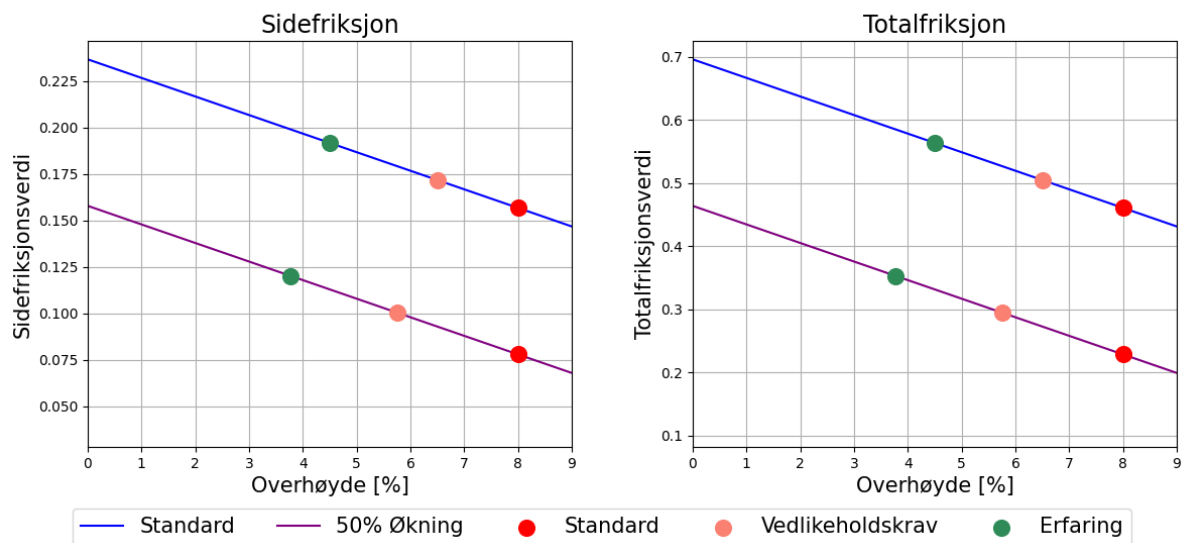
4.2.3 Overhøyde

Siste sensitivitetsanalyse ser på overhøyde. Fra litteraturstudiet er det særlig noen verdier som er interessante. Som presentert i Seksjon 2.2.3 er det bestemte vedlikeholds krav til overhøyde, noe som innebærer at enkelte avvik i overhøyde i utgangspunktet skal tolereres. Disse kravene er oppsummert i Figur 4.19 for H1 veg med standard minste horisontalkurveradius og 50 % økning, slik at det akseptertes overhøyder på henholdsvis 6,5 % og 5,7 %. Litteraturstudiet diskuterte også avvik fra vedlikeholds kravene, og for H1 veg er det sannsynlig at overhøyden reduseres ytterligere med 2 %. Figur 4.20 viser hvordan det kreves høyere sidefriksjon for disse verdiene dersom man legger minste horisontalkurveradius til grunn. Figurene viser at små endringer i overhøyder påvirker friksjonsbehovene lineært. For hver prosent overhøyden reduseres så økes behovet for sidefriksjon med 0,01. Med dagens krav om 34 % sidefriksjonsandel tilsvarende dette økt behov for totalfriksjon på nesten 0,03 per prosent overhøyden reduseres for vegklasse H1.



Figur 4.19: Illustrasjon over vedlikeholdsgrenser for overhøyde og hvordan overhøyden kan avvike fra dimensjonerende verdier avhengig av horisontalkurveradiusen.

Sensitivitet til overhøydens krav til friksjon for H1 veg



Figur 4.20: Behov for sidefriksjon og totalfriksjon etter avvik i overhøyde.

4.3 Hvor godt treffer dagens krav om 50 % økning av minste horisontalkurveradius?

Denne seksjonen viser hvordan ulike verdier for totalfriksjon og overhøyde har ulike krav til minste horisontalkurveradius. Formålet med disse resultatene er å vise hvor godt dagens krav til 50 % økning av minste horisontalkurveradius, som ikke tar hensyn til vegklasser eller vinterdriftsklasser, stemmer overens med det faktiske behovet for økning etter friksjonsforholdene. Det er ikke vurdert andre sidefriksjonsandeler enn de som brukes i dag da disse verdiene stort sett er ganske like enten de svenske eller danske sidefriksjonsandelene slik det ble funnet i Tabell 2.6. Dessuten har sensitivitetsanalysen vist at små endringer i sidefriksjonsandeler ikke gir store utslag på nødvendig horisontalkurveradius. Seksjonen er delt i to deler. Først presenteres det hvordan 5 bestemte totalfriksjonsverdier vil påvirke behovet for horisontalkurveradius avhengig av vegklassen. Til slutt vurderes 3 scenarier med varierende grad av ugunstige forhold som det kan være fornuftig å dimensjonere etter.

4.3.1 Friksjonens behov for horisontalkurveradius

De fleste totalfriksjonsverdiene som brukes i undersøkelsen er valgt ut på bakgrunn av resultatene i Seksjon 4.1.

Utover de laveste friksjonsverdiene til målepunkter uavhengig av vinterdriftsklasse, og de laveste friksjonsverdiene til den vinterdriftsklassen som kom dårligst ut, er det også inkludert friksjonsverdier for vinterdriftsklasser med utelukkende barvegsstrategi. I tillegg til disse verdiene brukes også standardiserte verdier for de ulike vegklassene for å danne et referansegrunnlag som det kan sammenlignes mot, samt erfaringsverdien for friksjon ved rimfrost som representerer et worst case scenario. Utvalgte verdier med beskrivelser er oppsumert i Tabell 4.14 og følger samme struktur som de endelige resultatene i Tabell 4.15.

Friksjonsverdi	Beskrivelse
x	Standardens krav til totalfriksjon avhengig av vegklassen som undersøkes
0,15	Erfaringsverdi for totalfriksjon målt på rimfrost
0,18	Lavesete 5-persentil blant alle vinterdriftsklassene
0,20	5-persentil for alle målepunktene uavhengig av vinterdriftsklasse
0,23	Lavesete 5-persentil blant alle vinterdriftsklassene med utelukkende barvegsstrategi
0,30	Høyeste 5-persentil blant alle vinterdriftsklassene med utelukkende barvegsstrategi

Tabell 4.14: Beskrivelse av ulike totalfriksjonsverdier.

Beregningen av minste horisontalkurvatur gjøres etter Formel 1.1, med standardiserte verdier for dimensjonerende hastigheter, sikkerhetsfaktorer, og overhøyder avhengig av vegklasse. Sidefriksjonen som inngår i formelen beregnes som en andel av totalfriksjonen. Denne beregningen gjøres etter Figur 2.12 og vil dermed avhenge av fartsgrensen til vegklassen. Resultatene er oppsummert i Tabell 4.15, hvor verdiene som vises for sidefriksjoner inkluderer eventuelle sikkerhetsfaktorer som vegklassen krever. Vegklassenes minste horisontalkurveveradier fra håndbok N100 [4] er vist i de grå radene. I parenteser i de samme radene vises de beregningsmessige minste horisontalkurveveradiene som ligger til grunn for håndbokens verdier. Når det er stor avstand mellom verdiene fra håndboken og det beregnede behovet legges det i praksis inn økt sikkerhet. Dermed kreves det mindre økninger i minste horisontalkurveveradius fra håndboka når man skal imøtekomme behovene fra de andre totalfriksjonene. Dette er særlig tydelig for vegklasse H5 hvor radiusen økes med 43 meter før man sammenligner mot andre totalfriksjonsverdier. Dette bidrar til å gjøre de prosentvise økningene mindre for denne vegklassen enn for de andre. Siste kolumnen i tabellen viser nemlig hvor mange prosent håndboksverdiene må økes for å håndtere de ulike totalfriksjonene. Der vegklasse H5 har størst sikkerhet i håndboksverdien har vegklasse H1 minst. Dette kan man også se igjen i siste kolonne hvor den prosentvise endringen er størst for alle totalfriksjonsverdier. Overordnet viser tabellen at dagens krav om 50 % økning av håndboksverdiene treffer rimelig bra for mange totalfriksjonsverdier.

Veg-klasse	Total-friksjon	Side-friksjon	Minste horisontal-kurve radius [m]	Prosentvis- endring [%]
H1	0,461	0,157	250 (240,0)	-
	0,15	0,051	434,3	73,7
	0,18	0,061	402,9	62,2
	0,20	0,071	384,4	53,6
	0,23	0,082	359,6	43,8
	0,30	0,102	312,6	25,0
H5	0,437	0,119	400 (357,1)	-
	0,15	0,041	587,7	46,9
	0,18	0,049	555,5	37,6
	0,20	0,057	528,2	32,0
	0,23	0,065	497,9	24,5
	0,30	0,082	439,2	9,8
H3	0,397	0,072	800 (771,3)	-
	0,15	0,027	1108,7	38,6
	0,18	0,033	1052,5	31,6
	0,20	0,038	1018,2	27,3
	0,23	0,044	970,6	21,3
	0,30	0,055	875,3	9,4
Hø1	0,461	0,157	225 (212,6)	-
	0,15	0,051	384,7	71,0
	0,18	0,061	356,9	58,6
	0,20	0,071	340,5	51,3
	0,23	0,082	318,5	41,6
	0,30	0,102	276,9	23,1
Hø2	0,528	0,195	125 (103,1)	-
	0,15	0,056	209,2	67,4
	0,18	0,067	193,4	54,7
	0,20	0,078	184,1	47,3
	0,23	0,089	171,7	37,4
	0,30	0,111	148,4	18,7

Tabell 4.15: Oversikt over hvordan ulike friksjonsverdier påvirker behovet for minste horisontalkurve radius. Standardiserte verdier er markert i grått med beregnet horisontalkurve radius i parentes. Nødvendige horisontalkurve radii for øvrige friksjonsverdier sammenlignes med standardverdiene.

4.3.2 Scenarioer

Avslutningsvis presenteres 3 ulike scenarioer for totalfriksjon og overhøyde. Scenarioene er delt inn etter hvor ugunstige forhold man tar hensyn til. For meget ugunstige forhold forutsettes rimfrost med totalfriksjon tilsvarende 0,15 og avvik fra krav til overhøyde. Avviket tar utgangspunkt i vedlikeholds kravet til overhøyde [24], med ytterligere 2 % avvik for H1, Hø1 og Hø2 som beskrevet i Seksjon 2.2.3. For ugunstige forhold benyttes rimfrost for totalfriksjon og dimensjonerende verdier for overhøyde. Siste scenario med noe gunstige forhold benytter 5-persentiler for totalfriksjonsverdier avhengig av vinterdriftsklasse fra Tabell 4.5 sammen med dimensjonerende verdier for overhøyde. For H1 benyttes 5-persentil uavhengig av vinterdriftsklasse, H5 og H3 benytter vinterdriftsklasse B, middels, som var den minst gunstige barvegsstrategien, mens Hø1 og Hø2 benytter vinterdriftsklasse C som kom dårligst ut blant vinterdriftsklassene med vintervegstrategi. Horisontalkurve radiusen beregnes etter Formel 1.1 med vegklassenes dimensjonerende verdier for hastighet, sidefriksjonsandel og sikkerhetsfaktor til friksjon. Deretter sammenlignes beregnet horisontalkurve radius med dimensjonerende horisontalkurve radius under normale forhold for å vurdere hvor mange prosent økning som trengs for å møte hvert scenario. Resultatene er oppsummert i Tabell 4.16.

Vegklasse	Gunstighet til forhold	Totalfriksjon	Overhøyde [%]	Horisontalkurve radius [m]	Økning [%]
H1	Meget ugunstig	0,15	4,5	593	137,0
	Ugunstig	0,15	8	434	73,7
	Noe gunstig	0,20	8	384	53,8
H5	Meget ugunstig	0,15	5,5	741	85,2
	Ugunstig	0,15	8	588	46,9
	Noe gunstig	0,23	8	498	24,5
H3	Meget ugunstig	0,15	2,83	2040	155,0
	Ugunstig	0,15	7,5	1109	38,6
	Noe gunstig	0,23	7,5	971	21,3
Hø1	Meget ugunstig	0,15	4,5	525	133,3
	Ugunstig	0,15	8	385	71,0
	Noe gunstig	0,18	8	357	58,6
Hø2	Meget ugunstig	0,15	4,5	282	125,6
	Ugunstig	0,15	8	209	67,4
	Noe gunstig	0,18	8	193	54,7

Tabell 4.16: Horisontalkurve radiusbehov for ulike scenarioer sammenlignet med dimensjonerende minste horisontalkurve radius.

De meget ugunstige forholdene med avvik i overhøyde sammen med rimfrost krever substansielle økninger i minste horisontalkurve radius for å imøtekomme forholdene. Når sidefriksjonen blir så lav som for rimfrost er man særlig avhengig av overhøyde for å ta opp sidekrefter i kurvebevegelsene. Størst utslag skjer for H3 veg hvor man i utgangspunktet aksepterer at overhøyden kan bli så lav som 2,83 % samtidig som kun 20 % av totalfriksjonen antas at kan brukes som sidefriksjon. Dersom man kan garantere at overhøyden opprettholdes på 7,5 % kan man med god samvittighet redusere horisontalkurve radiusen fra 2040 meter til 1109 meter og likevel håndtere rimfrost. Dersom man vil sikre flere av bruene på de andre vegklassene mot rimfrost

kommer man langt med å øke minste horisontalkurveradiene på bru med 70 % for H1, Hø1 og Hø2. For H5 vil dagens krav med 50 % økning være tilstrekkelig, mens for H3 kan man til og med senke kravet til 40 %. Ønsker man derimot å dimensjonere etter 5-persentil-konseptet med realistiske vinterdriftsklasser til hver vegklasse treffer dagens krav til 50 % økning av minste horisontalkurveradius ikke så verst for H1, Hø1 og H2, selv om den burde økes noe. Kravene kan derimot senkes til 25 % økning for H5 og H3. Alt i alt kreves det ikke mye mer fra dagens krav for å sikre god robusthet mot rim, men det forutsetter god kontroll på overhøyde.

Kapittel 5

Diskusjon

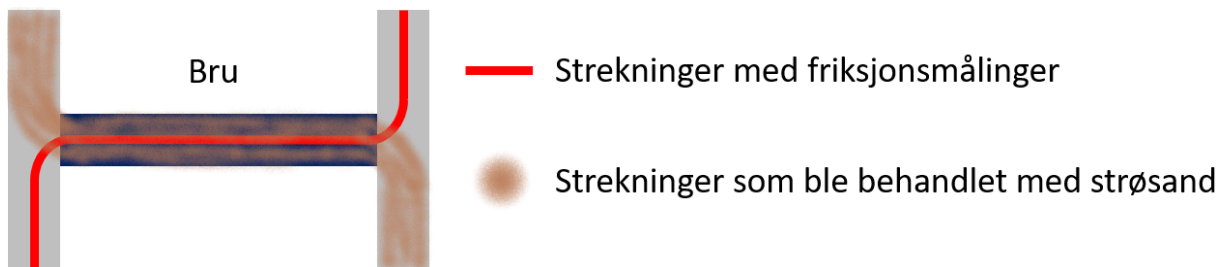
5.1 Hvordan varierer friksjonsforholdene på vegstrekninger på og utenfor bruer under vinterforhold?

Resultatene fra datagrunnlaget som er brukt i oppgaven indikerer at friksjonsforholdene på vegstrekninger på bruer generelt sett er ganske like som vegstrekningene utenfor bruene. Det er lite i datagrunnlaget som tyder på at verken totalfriksjonen eller vegtemperaturen er lavere på bruer, tvert i mot er friksjonen faktisk oftere litt høyere på bruene enn utenfor. Resultatene bryter med tanken om at bruer er mer utsatte for temperaturfall og har dårligere friksjon enn vanlig veg under vinterforhold.

Analysen vurderte derfor ulike vinterdriftsklasser for å se om ulike krav til vintervedlikehold påvirket hvordan friksjonen opptrådte på og utenfor bru. Det viste seg at bekymringen om at friksjonen kunne være lavere på bru enn utenfor kun holdt vann for vinterdriftsklasse *B*, *høy*, men denne vinterdriftsklassen hadde likevel høyere friksjonsverdier enn alle de andre vinterdriftsklassene, inkludert vinterdriftsklasse *A*. Dette kan tyde på at vinterdriftsklassen stort sett driftes svært godt. Nøyaktig hvorfor denne vinterklassen bryter med de andre er vanskelig å fastslå med sikkerhet, men utvalget kan gi en pekepinn i riktig retning. Hele 30 % av målepunktene på bruer for denne vinterklassen kom fra Gaulabrua ved Støren, riktig nok spredt utover ulike dager, måneder og år. Dette tvang frem mer målrettede undersøkelser mot brua. Det er vist at Gaulabrua er signifikant annerledes fra de andre bruene, og med den store andelen brua utgjør av datasettene er det naturlig at dette reflekteres i de generelle resultatene til vinterdriftsklasse *B*, *høy*. Det kan tenkes at bruas geografiske plassering og meteorologiske forhold spiller inn. Uten grundigere undersøkelser utenfor omfanget av denne masteroppgaven er det ikke mulig å trekke endelige konklusjoner om brua.

I motsatt ende ser man gjennomgående høyere friksjonsverdier på bruer med vinterdriftsklasse *E*. Denne vegklassen er mye bedre representert av ulike bruer enn det vinterdriftsklasse *B*, *høy* var, så dette kan ikke forklares like enkelt av enkeltbruer. Mulige forklaringer kan være måten bruene driftes på, hvordan friksjonsverdiene er målt, og hvordan bruene er konstruert. Vinterdriftsklassen forventes å brukes i områder med relativ lav trafikk og større behov for prioriteringer i vinterdriften. I spredtbebygde områder kan det være flere veger med vinterdriftsklasse *E*, men merkbar variasjon i ÅDT. Det er tenkelig at bruer opptrer oftere på de viktigste strekningene, eller simpelthen at det gjøres mer flid med strøinga på bruene. Et kjøretøy som sitter

fast på ei bru vil være en større byrde enn et kjøretøy som står fast på en veg. Det faktumet at bruene ble observert med høyere friksjonsverdier kan også skyldes at det ikke var samsvar mellom strekninger som ble prioritert for vinterdrift og strekninger som endte opp med å bli målt. Ta krysset i Figur 5.1 som eksempel hvor brua forbinder to vegstrekninger. Det er mulig at brua sammen med de høyeste prioriterte retningene på hver side av brua har blitt strødd, men at målebilen kom fra motsatt retning, kjørte over den strødde brua, og fortsatte i den nedprioriterte retningen igjen. Det er nok at målebilen kjører over én nedprioritert strekning for at det skal registreres ulikheter mellom vegen på og utenfor brua.



Figur 5.1: Eksempel på et kryss forbundet med bru hvor den ene retningen på hver side ble behandlet med strøsand, men det ble målt friksjon i de motsatte retningene.

Siste vinterdriftsklasse som bemerket seg var vinterdriftsklasse C. Denne klassen hadde lavest totalfriksjon på bruer av samtlige vinterdriftsklasser. Interessant nok hadde klassen også noen av de likeste totalfriksjonsverdiene på bru som utenfor. At spredningen er lav hjelper nok noe av at det er registrert flere målinger på denne vinterdriftsklassen enn noen annen klasse. Selv om forskjellen mellom friksjonen på bru og utenfor bru er lav, så er variasjonen i de målte verdiene store. Noe av forklaringen kan nok ses i sammenheng med strategien vinterdriftsklassen benytter. Dette er den eneste klassen som benytter både barvegsstrategi og vintervegsstrategi, alt avhengig av om det er milde eller kalde perioder. De andre vinterdriftsklassene forventer flere lengre perioder med stabilt mildvær eller kaldvær, altså utsettes vegene med vinterdriftsklasse C for mer varierende klima. Dette gjør det ekstra krevende å treffe med driftstiltak. Forsøkes det for eksempel å bruke salt for å opprettholde bar veg samtidig som det blir kaldt vil saltet gi lav effekt. I motsatt tilfelle vil det fungere dårlig med strøsand dersom vintervegen utsettes for sykluser av tine- og fryseprosesser slik at sanden til slutt ender opp under snøen eller isen. Veger som er tildelt vinterdriftsklasse C har med andre ord krevende forhold, og det kreves tilsvarende god oppfølging av vegene for å drifte dem optimalt. Slik kan vinterdriftsklasse C ende opp med lavere friksjon enn klasse E som til sammenligning er mindre komplisert å drifte.

Etter å ha sett på karakteristikken til bruene i Utvalg 2, altså bruene med størst differanse mellom laveste friksjonsverdi utenfor bru og på bru, ble det lansert en hypotese om at bruer av spennbetong kan være ekstra utsatt for varierende friksjon. Tabell 4.9 over hvor ofte brumaterialene forekom blant 5-persentilene til de ulike vinterdriftsklassene viste snarere tvert imot at bruer av spennbetong forekom signifikant sjeldnere i 5-persentilen enn de gjorde til vanlig. I tillegg til å komme med data mot hypotesen gir også undersøkelsen underbyggende argumenter for en ny hypotese, nemlig at bruer av betong er mer utsatt enn andre. Dette belyser at man må være forsiktig med å trekke slutninger fra små datasett, men man bør også være kritisk med å trekke slutninger fra større datasett. Hvor gyldig er det egentlig å summere 5-persentilene fra ulike vinterdriftsklasser og sammenligne summen med det generelle bru-utvalget? Eksempelvis forventes det at driftsklasse A brukes på større veger med strengere vegklasse, og da er det

heller ikke særlig sannsynlig at det benyttes bruer i stein eller tre på vegene. Gyldigheten til sammenligningene av antallet forekomster i 5-persentil og det generelle utvalget kan potensielt økes ved å sammenligne 5-persentilene til en driftsklasse med bruer i det generelle utvalget som deler samme driftsklasse. For driftsklasse *B*, høy er det allerede avdekket at Gaulabrua, som er laget av spennbetong, er målt mye oftere enn noen annen bru med samme vinterdriftsklasse. En annen tilnærming kunne vært å sammenligne bruene mot 5-persentiler uten å ta hensyn til vinterdriftsklassene, noe som ville favorisert verdier fra vinterdriftsklassene med vintervegsstrategi. Selv om verdiene som ble funnet med den brukte metoden er signifikante betyr det ikke at de er betydningsfulle nok til å akseptere eller forkaste hypotesen alene. I dette tilfellet burde ytterligere undersøkelser brukes dersom det var et eksplisitt mål å finne svar på hypotesen. Å kun se på 5-persentiler er heller ikke dekkende for å besvare hele spørsmålet om variasjon i friksjonsverdi. Skal man likevel prøve å tolke hvorfor resultatene i Tabell 4.9 ble som de ble, så kan det skyldes at bruer av betong oftere har en vegoverflaten bestående av betong i stedet for asfalt, og at betongoverflater er mer utsatt for å bli glatte enn tilsvarende asfaltoverflater.

Et vesentlig spørsmål som bør reises er hva som egentlig klassifiseres som vinterforhold. Alle resultatene som ligger til grunn for denne oppgaven har basert seg på målinger hvor enten vegoverflaten eller lufta hadde lavere temperatur enn 1°C . Endres terskelverdien til 0°C for datagrunnlaget reduseres antall målepunkter med 16 %, men 5-persentilene blir ikke nødvendigvis lavere av den grunn. Målet med terskelverdien på 1°C var å ikke utelukke interessante verdier da mye av vinterproblematikken ligger rundt frysepunktet. Uten noe slingringsmonn slik at kun kun temperaturer under 0°C hadde blitt undersøkt, så ville man stått igjen med kun halvparten av temperaturene nære frysepunktet. Nye analyser med andre terskelverdier kunne likevel gi interessante resultater. Andre klassifiseringer av hva som gjelder som vinterforhold kan potensielt gi svært ulike 5-persentiler dersom man spisser seg mot, eller beveger seg lenger unna de mest kritiske tilfellene.

5.2 Hvor sensitivt er behovet for minste horisontalkurvatur etter endringer i parametrene friksjon, sikkerhetsfaktor og overhøyde?

Sensitivitetsanalysen som er utført i oppgaven forsøker å gi kontekst til resultatene fra analysene av de historiske friksjonsmålingene. I tillegg hjelper den til med å løfte blikket og evaluere hvor kritisk det er med endringer i øvrige parametre som påvirker horisontalkurveradius. Det ble vurdert endringer i totalfriksjon og sidefriksjon, sikkerhetsfaktorer for friksjon, og overhøyde.

Første analyse studerte sidefriksjonsandeler. Sensitivitetsanalysen som baserte seg på vegklasse H1 med fartsgrense 80 km/t viste at ikke var snakk om store utslag i behov for horisontalkurveradius ved å benytte andre sidefriksjonsandeler enn den dimensjonerende verdien på 34 %. Tabell 4.11 viste hvordan 1 % endring i sidefriksjonsandel ville gi ulike behov for horisontalkurveradius. Det var ulikt hvor stor effekten var av 1 % endring i sidefriksjonsandel avhengig om sidefriksjonsandelen allerede var høy eller lav fra før, og om totalfriksjonen var lav eller høy. Effekten av 1 % endring i sidefriksjonsandel ble mer og mer betydningsfull desto lavere sidefriksjonsandelene ble. Man kan forenklet sett si at for hvert prosent sidefriksjonsandelen reduseres så økes behovet for horisontalkurveradius med ca 5 meter. Det vil si at en reduksjon i sidefriksjonsandel fra 34 % til 29 % på en H1-veg vil øke behovet for horisontalkurveradius

med 25 meter, altså 10 % økning av minste horisontalkurveradius.

Litteraturstudiet hadde gjort det klart at nesten ingen land er enige om hvilke verdier som bør benyttes for totalfriksjon og sidefriksjon, og derav oppstår også ulike sidefriksjonsandeler. Norge har lagt seg på en konservativ linje for de fleste fartsgrensene når det kommer til sidefriksjonsverdier. Som Tabell 2.5 viser så brukes det gjennomgående høyere sidefriksjonsverdier enn samtlige andre undersøkte land med vinterutfordringer frem til en fartsgrense på 90 km/t. Dette viser imidlertid ikke hele bildet ettersom den norske modellen også benytter seg av sikkerhetsfaktorer, fartstillegg og fartsprofiltillegg når det dimensjoneres for horisontalkurveradier. Tar man hensyn til disse sikkerhetstiltakene blir den endelige minste horisontalkurveradiusen for H1 veg faktisk høyere enn for de andre landene som ikke benytter tilsvarende sikkerhetstiltak. Ved å begrense seg til de skandinaviske landene ser man at både Sverige og Danmark er helt enige om dimensjonerende verdier for sidefriksjon på bar veg, men de ulike utgangspunktene for totalfriksjonsverdier gir tidvis svært ulike sidefriksjonsandeler. Til sammenligning fremstår de norske verdiene stort sett en solid middelveg mellom de skandinaviske naboenes verdier frem til fartsgrenser på 90 km/t slik som Tabell 2.6 viser. Dette styrker argumentet om at de norske sidefriksjonsverdiene er godt plassert med dagens verdier, i alle fall for fartsgrenser på 80 km/t og lavere. Der Norge virkelig skiller seg fra nabolandene er ved de høyeste fartsgrensene. Da benyttes det lavere sidefriksjonsandeler som er betydelig lavere enn de svenske sidefriksjonsandelene. Dermed er det svært usannsynlig at sidefriksjonsandelene burde være enda lavere for vegklasse H5 og H3. Alt i alt er det lite som peker på at sidefriksjonsandelene som benyttes i den norske standarden burde være enda lavere. Dermed forventes det heller ikke større behov for horisontalkurveradius som følger av reduksjoner i sidefriksjonsandeler.

Neste steg i sensitivitetsanalysen undersøkte sikkerhetsfaktorer for friksjon. Her ble 5-persentilene og 15-persentilene fra de historiske målingene sammenlignet mot henholdvis dagens krav til totalfriksjon på 0,461, og det tidligere kravet om totalfriksjon på 0,55. Analysen gir en oversikt over hvor store sikkerhetsfaktorer som kreves dersom de observerte historiske totalfriksjonsverdiene skal kontrolleres utelukkende ved bruk av sikkerhetsfaktorer i dimensjoneringen. Vegklasse H1 benytter ikke sikkerhetsfaktorer i dag, og det er like greit da det ville blitt krevd sikkerhetsfaktorer større enn 1,5 for å hele tatt komme i nærheten av totalfriksjonsverdiene under vinterforhold. Derimot ser man at dagens krav til 50 % økning av minste horisontalkurveradius på bru omfatter alle 5-persentilene til vinterdriftsklassene med barvegsstrategi. En kombinasjon av sikkerhetsfaktor på 1,5 sammen 50 % økning ville klart å dekke totalfriksjonen til rimfrost. Over til 15-persentilene hvor man kan se at det også er behov for store sikkerhetsfaktorer dersom man skal nå de observerte totalfriksjonsverdiene. Selv om det kreves noe lavere sikkerhetsfaktorer enn for 5-persentilene, så burde ytterligere sikkerhetsfaktorer benyttes for å kompensere for at disse verdiene er 15-persentiler, altså verdier man forventer å observere 15 % av tiden. Det er altså mulig å bruke sikkerhetsfaktorer for å kompensere for lave friksjonsverdier på bru i stedet for å øke den minste horisontalkurveradiusen med en vilkårlig verdi, men en slik fremgangsmåte fremstår som unødvendig kompleks.

I motsetning til sidefriksjonsandelene så har den norske metoden en svakhet i måten overhøyden behandles. I følge litteraturstudiet må man ikke bare forvente overhøyder som følger ganske snille vedlikeholds krav, men i praksis er overhøyden nokså ofte redusert ytterligere. Dette var særlig fremtredende for de minste horisontalkurveradiusen og mindre vanlig for de største radiene. Et viktig poeng som må fremheves her er at det er snakk om ytterligere avvik fra vedlikeholds kravene, og disse kravene tillater store avvik for de største radiene allerede. Eksempelvis kan H3 nå helt ned til overhøyde på 2,83 % uten å falle utenfor kravene. Med ytterligere av-

vik fra kravet, som ble registrert til å skje med mer enn 2 % i over 10 % av tilfellene, hadde overhøyden blitt tilnærmet null. Det er god grunn til å tvile på hvor godt egnet disse vedlikeholdskravene til overhøyde er dersom man forventer at det kan legge seg rimfrost på bruer i kurver som senker totalfriksjonen til 0,15. Den tidligere masteroppgaven som gjorde disse undersøkelsene hadde fokuset på ordinære veger og ikke bruer. Kanskje er det grunn til å tro at bruer som gjør at vegbanen ligger mer rigid motvirker avvikene i overhøyde? Eller er det kanskje grunn til å tro at ujevne setninger lager helninger på vegbanen oftere enn på vanlige vegstrekninger? Uten målrettet forskning er det vanskelig å si med sikkerhet, men det er liten tvil om at ugunstig overhøyde vil få store konsekvenser for lave friksjonsverdier. Dagens vedlikeholdskrav er ikke utformet til å ivareta sikkerheten til trafikanter som er ute på kurver med glatt føre.

5.3 Hvor godt dekker dagens standarder for friksjon og sikkerhetsfaktorer de observerte friksjonsforholdene på bruer under vinterforhold?

Det viser seg at en så enkel regel som 50 % økning av håndbok N100 [4] sine minste horisontalkurveradier for de ulike vegklassene faktisk kan fungere greit til å håndtere vinterforhold på bruer. Ettersom dagens standard benytter 5-persentiler for friksjon på bar veg [5] blir det naturlig å også bruke 5-persentiler for friksjon på bruer med vinterforhold. Dette reiser spørsmålet om hva 5-persentilen for totalfriksjon faktisk er på bruer med vinterforhold. Det enkleste svaret er at dette tilsvarer en totalfriksjon lik 0,2, og dermed dekkes behovet for horisontalkurveradius med 50 % økning av minsteradiene for alle vegklassene utenom H1. Vegklasse H1 krever derimot 53,6 % økning, noe man kan argumentere for at er nærme nok dagens metode med 50 % økning.

En mer sammensatt tilnærming til spørsmålet er å vurdere hvor godt vegklassene håndterer 5-persentilene til ulike vinterdriftsklasser. Det er derimot ikke helt rett frem å knytte en vegdimensjoneringsklasse til en vinterdriftsklasse da disse ikke er direkte relaterte. Det ligger overlappende intervaller for ÅDT mellom både vegklassene og vinterdriftsklassene, og sistnevnte avhenger også av klimatiske forhold. Valget av vinterdriftsklasse er enklest for vegklasse H5 og H3 som begge behøver vinterdriftsklasse *A* eller *B* grunnet de høye trafikkmengdene som vegklassene dimensjoneres for. For disse vegklassene kan man forvente totalfriksjonsverdier som er høyere enn den generelle 5-persentilen som er uavhengig av vinterdriftsklasser. Selv ved skikkelig dårlig føre og rimfrost på bruene vil vegklasse H5 og H3 med 50 % økning være tilstrekkelig. Etter alt å dømme vil dette kravet føre til overdimensjonering av bruer på veger med vegklasse H5 og H3. Ser man litt større på dimensjoneringen og anerkjenner at det er mer enn kun lav totalfriksjon som er risikofult så kan det tenkes at 50 % økning av minste horisontalkurveradius for disse vegklassene ikke er så langt unna det faktiske behovet likevel.

Dersom man forsøker å koble bruer på veger med vegklasse H1, Hø1 eller Hø2 til vinterdriftsklasser så kommer man ikke like godt ut. Teoretisk sett kan alle disse vegklassene kombineres med alle vinterdriftsklassene, alt avhengig av ÅDT-en på vegene og de klimatiske forholdene. Dermed kan man ikke utelukke at det blir brukt vinterdriftsklasse *C* på bruene, altså den vinterdriftsklassen som hadde lavest registrerte 5-persentil for totalfriksjonsverdi på bruer under vinterforhold. I så fall vil totalfriksjonen på 0,18 kreve horisontalkurveradier fra 55 % til 60 % større enn de minste verdiene for horisontalkurveradius i håndbok N100, alt etter hvilken

vegklasse man legger til grunn. Altså vil kravet om 50 % økning ikke være tilstrekkelig for vegklasse H1, Hø1 og Hø2 dersom totalfriksjonsverdien på 0,18 ligger til grunn. Enda verre blir det for tilfeller med rimfrost på bruene. Ved å legge til grunn en totalfriksjon på 0,15 vil behovet bli mellom ca 75 %, 70 % og 65 % økning av minste horisontalkurveradius for henholdsvis vegklasse H1, Hø1 og Hø2.

Hvordan kan man argumentere for at 50 % økning av minste horisontalkurveradius kan være tilstrekkelig selv om det beregnet behovet for horisontalkurveradius er litt større? For det første så utføres friksjonsmålingene med standardiserte dekk, som i utgangspunktet er beregnet for målinger på bar veg med en tynn vannfilm. Skal man dimensjonere for vinterforhold er det rimelig å anta at trafikantene tilpasser seg forholdene med vinterdekk. I vintersesongen burde dermed kjøretøyenes opplevde friksjon være bedre enn verdiene som kommer ut av friksjonsmålingene. For det andre vil det også være store variasjoner i opplevd friksjon avhengig av slitasje på dekk og marktrykk. Friksjonsmålingene vil aldri kunne representere alle kjøretøyene som benytter bruene perfekt.

Det kan også argumenteres for at 50 % økning av minste horisontalkurveradius ikke er tilstrekkelig selv om det beregnede behovet er dekket. Dersom den største bekymringen er rimfrost er det godt tenkelig at ikke alle trafikantene har rukket å legge om til vinterder. Dessuten skal økningen i horisontalkurveradius tross alt ikke bare håndtere friksjonsverdiene. Ved å ligge helt på grensa på behovet til horisontalkurveradius som blir beregnet med lav friksjon så blir marginene veldig små når det kommer til å håndtere andre faktorer. Eksempelvis kan det oppstå avvik i overhøyde, økte sidekrefter som følge av sterke sidevinder, eller rett og slett ugunstige kurvebevegelser som skaper større sidekrefter i en kort periode. En totalvurdering for minste horisontalkurveradius bør ikke gjøres utelukkende på forventede friksjonsverdier.

5.4 Hvilke totalfriksjonsverdier bør ligge til grunn for dimensjonering av minste horisontalkurvatur på norske vegbruer?

I søken etter dimensjonerende verdier for totalfriksjon på bruer har denne oppgaven gjennomgått to ulike tilnæringsmåter. Det vurderes å enten bruke totalfriksjonsverdier som dekker 5-persentilene fra historiske målinger utført på ulike vinterdriftsklasser, eller å bruke erfaringsverdien for totalfriksjon til rimfrost som anses som den mest kritiske verdiene bruene sannsynligvis vil oppleve. De to måtene har ulikt syn på hvor stor risiko som tolereres, noe som påvirker hvor like resultatene blir dagens metode som baserer seg på å øke håndbøkens verdier for minste horisontalkurveradius med 50 % for bruer.

Første tilnærming baserer seg på 5-persentilene til de ulike vinterdriftsklassene, hvor det forsøkes å knyttes fornuftige vinterdriftsklasser til de ulike vegklassene. Ved å gå frem på denne måten aksepteres det at bruene noen ganger får lavere friksjonsverdier enn det de dimensjoneres for, eksempelvis for tilfeller med rimfrost. Vegklasse H5 og H3 kan greit knyttes opp til vinterdriftsklasse *A* og *B* da disse sammenfaller med dimensjonerende trafikkmengder. Den mest kritiske 5-persentilen som er funnet blant disse vinterdriftsklassene er for klasse *B*, *middels* med totalfriksjon lik 0,23.

Det er mulig å argumentere for enda strengere vinterdriftsklasser, særlig for vegklasse H3, slik at det blir forventet gunstigere friksjonsverdier og lavere behov for horisontalkurveradius. Si-

den behovet for horisontalkurveradius allerede er nokså lavt med totalfriksjon lik 0,23, kun i underkant av 25 % økning for vegklasse H5 og H3, vurderes det ikke som hensiktsmessig å ta enda større risiko og dimensjonere for høyere friksjonsverdier når gevinsten ikke fremstår som stor nok. Klarer man derimot å garantere vinterdriftsklasse B, høy kan totalfriksjonen økes til 0,30 og redusere det beregnede behovet til 10 % økning.

For vegklasse H01 og H02 er det rimelig å forvente mindre strenge vinterdriftsklasser. I deres tilfeller vil vinterdriftsklasse C være mest kritisk med 0,18 i totalfriksjon, noe som krever ca 60 % og 55 % økning av minste horisontalkurveradius. Vegklasse H1 er vanskeligere å plassere sammen med en vinterdriftsklasse. Siden vegklassen er en del av hovedvegene bør det forventes høyere prioritet i vinterdrift enn veger som tilhører de øvrige hovedvegene. Som en middelgrunn kan det fremstå fornuftig å følge den generelle 5-persentilen til de historiske friksjonsmålingene som er uavhengig av vinterdriftsklasse. Dette utgangspunktet betyr i så fall å dimensjonere vegklassen etter en totalfriksjonsverdi like 0,20.

Tilnærmingen med 5-persentiler fra de historiske målingene fremstår ikke som en optimal metode. Løfter man blikket ser man at gjennomsnittsverdiene fra de historiske friksjonsmålingene faktisk ikke er så verst. Når friksjonsverdiene fra de historiske målingene synker på bruene så gjør de som oftest det også utenfor bruene. Formålet med de foreslåtte verdiene er å gjøre ferd-selen tryggere over bruer under vinterforhold, men friksjonsforholdene er tilnærmet like på brua som utenfor. Dermed vil sjåførene som oftest enten merke at det er glatt lenge før de ankommer bruene og tilpasse seg forholdene deretter, eller så merker de at de kan håndtere forholdene slik som de er. I ingen av disse tilfellene vil en justering av minste horisontalkurveradius utgjøre en vesentlig forbedring av trafikksikkerheten til trafikantene.

Den andre tilnærmingen baserer seg på å dimensjonere bruene for rimfrost, og dermed benytte totalfriksjon lik 0,15 uavhengig av hvilken vegklasse det er snakk om. Bruer er teoretisk sett særlig godt egnet for rimdannelse siden de effektivt kan kvitte seg med varme samtidig som de ofte har større tilgang på fuktighet enn resten av vegen [1]. Fuktilførselen kommer av den enkle grunnen at bruer ofte spenner over vann. Lokal dannelse av rim på brua kan gi betraktelige fall i friksjonsforholdene på bru sammenlignet med vegen forøvrig, og rimen er ikke nødvendigvis så enkel å oppdage på lang avstand. Som det ble påstått i Seksjon 2.4.1 skjer det ofte ulykker til tross for at det er fullt vintervedlikehold på vegene [35], for eksempel som møteulykken på E6 ved Snåsa så sent som i 2020 [49].

I følge nullvisjonen har vegtrafikksystemet et ansvar for å beskytte trafikantene mot normale feilhandlinger. Kan man ikke stole på det ene tiltaket trengs det flere tiltak for å gi forsvar i dybden. Blant de undersøkte tiltakene i Seksjon 2.4.3 kommer de fleste løsningene dårlig ut. Termiske metoder fungerer ikke bedre enn vanlig vintervedlikehold, og høyfriksjonsdekker reduserer heller ikke risiko. Dynamisk varsling kunne fungert greit i kombinasjon med vintervedlikehold, men også da er man kun avhengig av aktive barrierer. Selve varslingssystemet må fungere, og sjåførene må også registrere varslingen og tilpasse farten deretter. Ved å legge inn et passivt barriere-element i form av høy horisontalkurveradius er trafikantene beskyttet uten at de trenger å ofre en tanke til muligheten for rim på bru. Sammenlignet med de aktive barriere-elementene så reduseres også kompleksiteten. Det er liten frykt for at det passive barriere-elementet plutselig slutter å fungere, og man sparer samtidig vedlikeholdskostnader sammenlignet med det dynamiske varslingssystemet. Dimensjonerer man bruene for rim får man også gitt økt sikkerhet for de øvrige typene glatthet. Det kreves ei betydelige økninger av minste horisontalkurveradius for å ta høyde til rimen. Med 70 % til 75 % økning er man langt på veg for å dekke vegklassene H1,

Hø1 og Hø2. H5 dekkes godt av dagens krav på 50 %, mens kravene til H3 kan senkes ned til 40 %. Skal man redusere risikoen for ulykker etter ALARP-prinsippet vil en økning i horisontalkurveradius som håndterer rimfrost være å foretrekke over 5-persentilene for friksjonsverdier til de ulike vinterdriftsklassene.

Er det fornuftig å være bekymret for at friksjonen plutselig skal falle på bruer når det kun er registrert friksjonsfall høyere enn 0,15 i 0,3 % av tilfellene? Her er det vesentlig å ikke stole blindt på at datagrunnlaget representerer verden på en fullkommen måte. Grunnen til at tilfeller med rimfrost ikke fanges opp av målingene oftere kan skyldes at målingene som regel utføres på dagtid. Temperaturen synker ofte på kvelden, og med synkende temperatur kan ikke lufta holde på like mye fuktighet, helt til en viss grense hvor fuktigheten deponeres som rim dersom vegoverflaten er under frysepunktet. Med flere målinger på kvelds- og nattetid er det mulig at det ville blitt oppdaget flere tilfeller med rim og tilhørende friksjonsfall på bruer.

Det finnes også argumentasjon for at bruer burde øke horisontalkurveradiene uavhengig av vinterforhold. Som det nevnes helt innledningsvis i oppgaven og i litteraturstudiet så har bruer forhøyet ulykkesrisiko sammenlignet med øvrig veg når man tar i betraktning hvor mange flere kilometer det er med veg. Et av tiltakene som etterspørres for å redusere antall utforkjøring, også før selve brua, er å forbedre overflatefriksjonen [20]. I stedet for å gjøre noe med friksjonen er det mulig å gjøre som i denne oppgaven og heller øke horisontalkurveradiusen for å gjøre seg mindre avhengig av friksjonsforholdene.

5.5 Svakheter og alternativer

5.5.1 Datagrunnlagets gyldighet

Et essensielt spørsmål som ble reist allerede før oppgavestart spurte hvor godt egnet datagrunnlaget som lå til grunn for oppgaven egentlig var til formålet om å fastslå dimensjonerende friksjonsverdier på bruer under vinterforhold. Datagrunnlaget samler friksjonsmålinger fra alle regioner i landet, men antallet målepunkter i nærheten av bruer under vinterforhold er svært skjevfordelt. 76,1 % av målepunktene kommer fra Trøndelag og 14,4 % kommer fra Innlandet. Fra Seksjon 4.1.2 vet man at enkelte bruer, slik som Gaulabrua, kan skille seg ut fra de øvrige bruene. Når noen områder blir veldig overrepresentert står man i fare for både å fange opp flere bruer som Gaulabrua som skiller seg ut, men man risikerer også å ikke fange opp slike bruer i stor nok grad.

Videre er datagrunnlaget tidsbegrenset fra våren 2018 til våren 2021 slik at kun det kun er registrert målinger fra tre vintre. Som Figur 2.16 i litteraturstudiet viser så må man forvente store forskjeller i hvilke forhold som observeres fra sesong til sesong. Det er ikke undersøkt hvor stor variasjonen var mellom vintersesongene i datagrunnlaget, men det faktum at Trøndelag er såpass overrepresentert bidrar ikke til å begrense innflytelsen av den relativt korte tidsperioden. Det er fullt mulig at et datagrunnlag som dekket flere vintre med jevnere geografisk fordeling kunne gitt andre resultater. Dette er mulig å oppnå med samme fremgangsmåte som i denne oppgaven, men med en mer taktisk utvelging av hvilke måleutstyr som tas med i datagrunnlaget.

Samtidig bør det stilles spørsmål ved filtreringsmetoden som er beskrevet nærmere i Seksjon 3.3.2. Formålet med filtreringen var å effektivisere behandlingen av filene og fjerne så mye som mulig av datagrunnlaget som ikke var av interesse, men særlig et filtreringskriterium

kan gi uheldige effekter. Ved å filtrere bort filer hvor alle friksjonsverdiene er helt like er det stor sannsynlighet for å miste noen datasett med bruer, noe som kan kunstig minke andelen bruer med høy friksjonsverdi og dermed flytte 5-persentilen nærmere de laveste friksjonsverdiene. Stort sett har filene irregulære friksjonsverdier helt i starten eller på slutten av målingene, noe som begrenser hvor mange filer som faktisk blir berørt av dette filtreringskriteriet. For de tilfellene hvor det er lik friksjon for alle punktene så er det ofte enten registrert friksjonsverdier lik -1, eller så er det svært få målepunkter i utgangspunktet, noe som gir lavere sannsynlighet for at ei bru var målt og faktisk ville resultere i et datasett som kunne blitt brukt i oppgaven. Derfor anslås det faktiske tapet av data som lite, men ved fremtidige anledninger bør dette vurderes å skrote dette filtreringskriteriet.

Sist men ikke minst er 85 % av målingene utført mellom klokken 8 om morgenen og 8 om kvelden. Dette kan anses som en fordel da mesteparten av trafikken i løpet av et døgn er aktiv på disse tidene [50], og dermed kan man påstå at datagrunnlaget representerer de forholdene flest trafikanter opplever. Ulempen er at veier er mer utsatt for termisk utstråling uten tilsvarende innstråling på kveldstid, noe som gir kjøligere vegbane og bedre forhold for rimdannelse. Det er tenkelig at bruer med rim ikke har blitt fanget opp som følge av det ikke har blitt utført så mange målinger på nattetid.

5.5.2 Utførelse av metoden

Slik koden ble satt opp i QGIS brukes vegbruene fra NVDB ganske ukritisk, noen ganger med uheldige resultater. Av og til går vegen som er målt med friksjonsmålinger under ei bru, men i NVDB kan den kryssende brua være registrert med én komponent som følger brua over den aktuelle vegen, men også én komponent på selve vegen. Et eksempel på dette er vist med Follo bru utenfor Orkdal i Figur 5.2. Koden som er utviklet til bruk i denne oppgaven klarer ikke å skille mellom disse komponentene, noe som legger til flere datasett hvor målepunktene faktisk ikke er målt på bruer. I disse tilfellene registreres det falske positive.



Figur 5.2: Skjermdump fra vegkart: Eksempel på bru som er registrert på både bru og underliggende veg.

I motsatt ende klarte ikke koden alltid å knytte målepunktene til steder det faktisk var bruer. Manuelle kontroller av kjøringene av koden avslørte at bruene ikke alltid ble lastet inn riktig,

slik at tilfeldighetene avgjorde hvor vidt målingene til slutt ble knyttet til brua eller ikke. Denne feilen kan skyldes at enkelte bruer er registrert med lange kommentarfelter i NVDB, og når lengden overstiger antallet bokstaver QGIS kan laste inn skjer det uventede feil. I disse tilfellene er det snakk om falske negativer som dermed ikke blir registrert og tatt med videre til analysen. Det ble ikke forsøkt å korrigere denne typen feil da metoden likevel resulterte i et nokså stort antall datasett som kunne brukes i analysen.

I hele oppgaven har friksjonsmålingene blitt omtalt som målepunkter. I realiteten er målepunktene strekningsmålinger som oppgir gjennomsnittsverdier over den målte distansen. Selve punktet som det stadig refereres til er koordinatene til endepunktene av de målte strekningene. For denne oppgaven medfører dette at det første målepunktet på hver bru inneholder gjennomsnittsverdier for vegbanen før selve brua. Tilsvarende vil det første målepunktet etter hver bru også inkludere gjennomsnittsverdier fra deler av bra. Avhengig av målestrekningene kan man risikere at nesten hele strekningen som utgjør det første punktet på brua nesten kun består av verdier fra vegbanen utenfor brua. Tilsvarende kan det første punktet etter brua bestå av verdier nesten utelukkende målt på selve brua. Dette kan være med på å forklare noen av observasjonene i det andre utvalget med datasett som er illustrert i Figur 4.3. Eksempelvis kan det hende at strekningen som representeres av det første målepunktet etter brua for datasett 5 i dette utvalget egentlig inneholder mye av friksjonsforholdene på brua. Det forventes likevel ikke at dette påvirker resultatene på en vesentlig måte.

Det er heller ikke gjort noe forsøk på å skille mellom bruer som spenner over vann fra bruer som ikke er i nærheten av vann. Både tidligere forskning i litteraturstudiet og resultatene som omhandler luftfuktighet i Seksjon 4.1.4 er enige om at luftfuktigheten spiller en rolle for rimdannelse. Det kunne vært interessant å begrense utvalget til bruer hvor det er flytende vann i nærheten, og dermed større sannsynlighet for forhøyede verdier av luftfukt. Dersom målet er å fremheve bruer som er utsatt for rimdannelse uten at den omkringliggende vegen blir tilsvarende glatt, så kunne bruer som krysset elver og innsjøer blitt fremhevet. Man kunne også vurdert å ekskludere områder hvor vegen før og etter bruene gikk parallelt med vannkildene for å redusere sannsynligheten for at vegen ble utsatt for samme fuktnivå som bruene. Det kunne også vært interessant med egne undersøkelser som studerte om det var ulikheter mellom områder hvor vegene lå langs vann før vegene kysset vannene med bruer, og sammenligne dette med tilfeller hvor vegene ikke lå nærme vann før de krysset vannet med bruer. For slike undersøkelser kunne en tilsvarende metode med GIS-verktøy som er brukt i denne oppgaven blitt benyttet. Men man kunne blant annet forsøkt å bruke NVEs elvenettverksdatabase, ELVIS, til å identifisere hvordan vegene og bruene forholdt seg til elver.

5.5.3 Alternative metoder

En alternativ metode for å fastslå friksjonsverdier på bruer under vinterforhold gikk på å sende ut målebiler fra Statens Vegvesen for å utføre målrettede friksjonsmålinger på et begrenset sett av bruer når værforholdene ville være gunstige for å fange opp de mest kritiske situasjonene. En slik metode kunne gitt et tydeligere bilde over hvor lave friksjonsverdiene på bruer faktisk kunne bli, samtidig som man fikk økt innsikt i hvor hyppig disse tilfellene faktisk tiltraff. Et ønske gikk særlig på å dokumentere friksjonsverdier under rimforhold, og undersøke i hvilken grad det kan oppstå forskjeller i friksjon mellom vegoverflaten på bru og vegoverflaten på de øvrige vegene.

Denne metoden ble valgt bort til fordel for metoden som ble benyttet i oppgaven hovedsaklig

grunnet praktiske utfordringer knyttet til gjennomføring av den alternative metoden, og utfordringer med å samle et stort nok datagrunnlag som kunne gi troverdige resultater. Fra den praktiske siden er det vanskelig å treffe med friksjonsforholdene nøyaktig på de værforholdene som man ønsker å undersøke. For det første må værforholdene som tillater rimdannelse inntreffe. En mild vinter kunne drastisk redusert antallet tilfeller med ønskelige værforhold. For det andre må man få utført målingene mens forholdene faktisk inntreffer, gjerne på nattestid. Ved bruk av målebilsjåfører fra Statens Vegvesen bør førerene varsles god tid i forkant av målingen slik at de både rekker å legge strekningen inn i planene sine og faktisk får kjørt bort til strekningen, gitt at de i hele tatt har anledning til å kjøre ut til alle døgnets tider. For å treffe målrettet på rimdannelse kreves en viss grad av prediksjon, noe som i seg selv er uhyre vanskelig. Alternativt må det utføres målinger mer eller mindre regelmessig, noe som ikke nødvendigvis gir betraktelig bedre resultater enn metoden som ble brukt i oppgaven. Det var også en bekymring for at COVID-19-pandemien ville ført til flere utfordringer med en slik eksperimentell metode.

Kapittel 6

Avslutning

6.1 Konklusjon

Denne oppgaven undersøkte historiske friksjonsmålinger for å tilføre ny kunnskap om opptredende totalfriksjonsverdier på vegstrekninger som krysser bruer under vinterforhold. Formålet var å utvikle et alternativt dimensjoneringskrav for minste horisontalkurveradius på bruer basert på totalfriksjonsverdier. Denne metoden står i kontrast til dagens krav om 50 % økning av håndbok N100s [4] verdier for minste horisontalkurveradius når man dimensjonerer for bruer. Dette ble gjort med analyser av historiske friksjonsmålinger hvor totalfriksjonen ble sett i sammenheng med vinterdriftsklasser, temperaturer og luftfuktighet. Målte vegstrekninger på bruer og 100 meter radius utenfor bruens endepunkter ble sammenlignet mot hverandre. Generelt sett ble det funnet noe høyere totalfriksjonsverdier på bruene enn utenfor bruene. Målestrekningene hadde ganske like vegoverflatetemperaturer på bruer og utenfor bruer. Høyere luftfuktighet hang sammen med lavere friksjonsverdier både på og utenfor bru når vegoverflatetemperaturen var under frysepunktet.

Analysen viste at dagens krav til 50 % økning av minste horisontalkurveradius på bruer fungerte nokså godt til å håndtere 5-persentilverdiene til de historiske friksjonsmålingene sine totalfriksjoner under vinterforhold. Ved å bruke 5-persentilen for totalfriksjon uavhengig av vinterdriftsklasser for vegklasse H1 ble behovet for horisontalkurveradius 55 % større enn minsteverdien i håndboken. Dersom man antar at vegklasse Hø1 og Hø2 benytter vintervegstrategi under vinterforhold vil behovene for horisontalkurveradius bli henholdsvis 60 % og 55 % større enn vegklassenes minste verdier fra håndboken. Legges barvegsstrategi til grunn for veger med vegklasse H3 eller H5 kan kravet om økning av minste horisontalkurveradius senkes til 25 % økning for begge vegklassene. For disse vurderingene ble totalfriksjonsverdiene i Tabell 6.1 lagt til grunn:

Vegklasse	Totalfriksjon
H1	0,20
H5	0,23
H3	0,23
Hø1	0,18
Hø2	0,18

Tabell 6.1: Totalfriksjonsverdier som dekker 5-persentilene til historiske friksjonsmålinger for hver vegklasse under vinterforhold.

Analysen viste også at dagens krav til 50 % økning av minste horisontalkurveradius ikke gir tilstrekkelig sikring mot rimfrost. For bruer med tynne belegg av rim forventes totalfriksjonsverdier nærmere 0,15, altså lavere verdier enn 5-persentilene til de historiske målingene. Dette kan håndteres ved å øke kravet til minste horisontalkurveradius til 75 % for vegklassene H1, Hø1 og Hø2. Vegklasse H5 dekkes godt av dagens krav, mens kravet kan reduseres til 40 % økning for vegklasse H3. Dette medfører å dimensjonere alle vegklassene for en totalfriksjonsverdi lik 0,15.

I alle tilfeller vil man være avhengig av at de dimensjonerende verdiene for overhøyde i kurvene ikke blir lavere i virkeligheten enn det de ble dimensjonert for. Sensitivitetsanalysen viste at små utslag i overhøyde gir store utslag i nødvendig horisontalkurveradius når friksjonsverdiene blir lave.

6.2 Anbefaling

Det anbefales å dimensjonere minste horisontalkurveradius på bru etter totalfriksjonsverdien til rimfrost. Generelt sett er totalfriksjonsverdiene på bruer like eller litt høyere enn vegstreknin-gene utenfor bruene. Som oftest er det like sannsynlig at utfordringer med glatt føre inntreffer før brua. Hovedutfordringen for bruer oppstår når det blir glatt på bruene uten at det blir glatt på vegene som leder til bruene. Siden vegbruer har lavere termisk masse enn vegene rundt, de kan kjøles av vind på undersiden, og de ofte spenner over vann, vil de være mer utsatt for rim-dannelse enn resten av vegene. Dermed er det ikke usannsynlig at det er rim som gjør bruene glatte i de tilfellene friksjonsforholdene på bru blir vesentlig lavere enn for den øvrige vegen. For å realisere nullvisjonens mål om å beskytte trafikanter mot normale feilhandlinger, slik som å ikke forvente plutselig dårligere friksjonsforhold på bru, fremstår dette som det beste tiltaket. Behovet for horisontalkurveradius er heller ikke mye større dersom man legger totalfriksjonen til rim til grunn enn dersom man legger 5-persentilene fra de historiske målingene til grunn. Dette tiltaket kan dermed underbygges av ALARP-prinsippet.

Videre anbefales det å utvikle vedlikeholds krav til overhøyde på bru som sikrer at overhøyden ikke avviker nevneverdig fra de dimensjonerende verdiene.

6.3 Videre arbeid

Før det tas stilling til hvilke økninger av minste horisontalkurveradius på bru som bør brukes i dimensjonering bør flere variabler granskes nærmere. Konkret fremheves friksjonskoeffisienten til vinterdekk, sidefriksjonsandeler og overhøyde.

Målingene som er utført og brukt i oppgaven er alle gjort med standardiserte metoder med standardiserte dekk. Dersom det skal planlegges for vinterforhold bør målingene reflektere utstyret som forventes at det brukes av kjøretøyene. Større innsikt i hvordan totalfriksjonene fra målingene med standardiserte dekk skiller seg fra vinterdekk kan være med på å skape store endringer i behovet for totalfriksjon.

Friksjonsverdiene som er brukt til å regne ut minste horisontalkurveradius i oppgaven kommer alle fra totalfriksjonsverdier som er regnet om til sidefriksjonsverdier med bruk av sidefriksjonsandeler. Disse andelene avhenger kun av fartsgrenser, og de norske verdiene skiller seg fra flere land. Sensitivitetsanalysen har vist at moderate endringer i sidefriksjonsandeler ikke gir store utslag, men når metoden for å fastsette den minste nødvendige horisontalkurveradiusen på bru under vinterforhold vitenskapeliggjøres bør også sidefriksjonsandelene tildeles en tilsvarende stor rolle.

Sensitivitetsanalysen viste derimot at overhøyden spiller en vesentlig rolle i behovet for horisontalkurveradius når friksjonsverdiene blir lave. Tidligere masteroppgaver ved NTNU har vurdert avvik i overhøyde på veger, men tilsvarende undersøkelser rettet mot bruer ville hjelpe på å kartlegge hvor stort behovet for tiltak faktisk er. I tråd med anbefalingen burde det også undersøkes hvordan man kan gå fram for å sikre at det ikke blir store avvik i overhøyde på bruer.

Resultatene gir grunnlag for grundigere undersøkelser av Gaulabrua utenfor Støren da den skiller seg ut fra tilsvarende bruer med samme vinterdriftsklasse. Brua har signifikant lavere totalfriksjonsverdier målt på brua enn utenfor brua uten at det finnes en åpenbar forklaring på hvorfor dette skjer.

Referanser

- [1] Statens havarikommisjon for transport. (2010). *RAPPORT OM MØTEULYKKE MEL- LOM VOGNTOG OG TO PERSONBILER PÅ E16 I FLÅM*. (VEI 2010/04). Statens ha- varikommisjon for transport. Hentet 22. september 2020, fra [https://havarikommisjonen. no/Vei/Avgitte-rapporter/2010-04](https://havarikommisjonen.no/Vei/Avgitte-rapporter/2010-04)
- [2] TV2. (2010). *Farlig bro kostet et menneskeliv*. Hentet 1. oktober 2020, fra [https://www. tv2.no/a/3302356](https://www.tv2.no/a/3302356)
- [3] Aftenposten. (2010). *Får kritikk etter dødsulykke*. Hentet 22. september 2020, fra <https://www.aftenposten.no/norge/i/41xvq/faar-kritikk-etter-doedsulykke>
- [4] Statens Vegvesen. (2019a). *Håndbok N100: Veg- og gateutforming*. Hentet 1. oktober 2020, fra [https://www.vegvesen.no/_attachment/61414/binary/1355470?fast_title=H% 5C%C3%5C%A5ndbok+N100+Veg-+og+gateutforming+%5C%286+MB%5C%29.pdf](https://www.vegvesen.no/_attachment/61414/binary/1355470?fast_title=H%5C%C3%5C%A5ndbok+N100+Veg-+og+gateutforming+%5C%286+MB%5C%29.pdf)
- [5] Statens Vegvesen. (2019b). *Håndbok V120: Premisser for geometrisk utforming av veger*. Hentet 1. oktober 2020, fra [https://www.vegvesen.no/_attachment/61500/binary/ 1327507?fast_title=H%5C%C3%5C%A5ndbok+V120+Premisser+for+geometrisk+ utforming+av+veger.pdf](https://www.vegvesen.no/_attachment/61500/binary/1327507?fast_title=H%5C%C3%5C%A5ndbok+V120+Premisser+for+geometrisk+ utforming+av+veger.pdf)
- [6] Elvik, R. (2017). Can evolutionary theory explain the slow development of knowledge about the level of safety built into roads? *Accident Analysis & Prevention*, 106, 166–172.
- [7] Statistisk sentralbyrå. (2020). *Personer drept eller hardt skadd i veitrafikkulykker, etter statistikkvariabel og år*. Hentet 28. november 2020, fra [https://www.ssb.no/statbank/ table/12043/](https://www.ssb.no/statbank/table/12043/)
- [8] Vegkart. (2020). *Trafikkulykker mellom 01.01.2010 og 26.08.2019*. Hentet 28. november 2020, fra <https://vegkart.atlas.vegvesen.no/>
- [9] Nordahl, R. S. & Hoff, I. (2020). *Vei*. Store norske leksikon på [snl.no](https://snl.no/vei). Hentet 28. novem- ber 2020, fra <https://snl.no/vei>
- [10] Øderud, H. T. & Nordahl, R. S. (2020). *Bro*. Store norske leksikon på [snl.no](https://snl.no/bro). Hentet 28. november 2020, fra <https://snl.no/bro>
- [11] ViaTech. (2021). *ViaTechs friksjonsdatabase*. Hentet 3. november 2020, fra [http://www. viatech.no/ViaFrictionWEB/Account/LogOn?ReturnUrl=%5C%2fViaFrictionWEB% 5C%2f](http://www.viatech.no/ViaFrictionWEB/Account/LogOn?ReturnUrl=%5C%2fViaFrictionWEB%5C%2f)

-
- [12] Meld. St. 20 (2020-2021). (2021). *Nasjonal transportplan 2022–2033*. Hentet 4. juni 2021, fra <https://www.regjeringen.no/no/dokumenter/meld.-st.-20-20202021/id2839503/?ch=1>
- [13] Kjellén, U. & Albrechtsen, E. (2017). *Prevention of accidents and unwanted occurrences: Theory, methods, and tools in safety management*. CRC Press.
- [14] Aven, T. & Renn, O. (2009). On risk defined as an event where the outcome is uncertain. *Journal of risk research*, 12(1), 1–11.
- [15] Moen, B. (2008). *Risk perception, priority of safety and demand for risk mitigation in transport* (Doktoravhandling). NTNU: Fakultet for samfunnsvitenskap og teknologiledelse.
- [16] Haddon, W. (1980). The basic strategies for reducing damage from hazards of all kinds. *Hazard Prevention*, 16:8–12.
- [17] Jones-Lee, M. & Aven, T. (2011). ALARP—What does it really mean? *Reliability Engineering & System Safety*, 96(8), 877–882.
- [18] Rødseth, K. L., Wangsness, P. B., Veisten, K., Høye, A. K., Elvik, R., Klæboe, R., Thune-Larsen, H., Fridstrøm, L., Lindstad, E., Rialland, A., Odolinski, K. & Nilsson, J.-E. (2019). *Eksterne kostnader ved transport i Norge – Estimer av marginale skadekostnader for person- og godstransport* (Nr. 1704/2019). TØI. Hentet 4. juni 2021, fra <https://www.toi.no/publikasjoner/eksterne-kostnader-ved-transport-i-norge-estimer-av-marginale-skadekostnader-for-person-og-godstransport-article35997-8.html>
- [19] Selvik, J. T., Elvik, R. & Abrahamsen, E. B. (2020). Can the use of road safety measures on national roads in Norway be interpreted as an informal application of the ALARP principle? *Accident Analysis & Prevention*, 135, 105363.
- [20] Elvik, R., Sagberg, F. & Langeland, P. A. (2019). An analysis of factors influencing accidents on road bridges in Norway. *Accident Analysis & Prevention*, 129, 1–6.
- [21] Sagberg, F., Elvik, R. & Langeland, P. A. (2020). Crash risk on entrance versus exit zones of road bridges in Norway. *Accident Analysis & Prevention*, 134, 105247.
- [22] Hovd, A. 2014. Horisontalkurvatur. *TBA4203 Vegplanlegging*. Hentet 7. juni 2021, fra <https://ntnu.blackboard.com>
- [23] Spacek, P. (2005). Track behavior in curve areas: attempt at typology. *Journal of transportation engineering*, 131(9), 669–676.
- [24] Klein-Paste, A. & Nonstad, B. (2015). *Lærebok: Drift og vedlikehold av veier* (nr. 365). Statens Vegvesen. Hentet 20. oktober 2020, fra https://www.vegvesen.no/_attachment/1011337/binary/1056061?fast_title=SVV+rapport+365+L%5C%C3%5C%A6rebok+Drift+og+vedlikehold+av+veger.pdf
- [25] Bogdashova, N. (2012). *En studie av sammenheng mellom horisontalkurvatur, tverrfall og trafikkulykker på vegnettet* (Masteroppgave). NTNU.
- [26] Andersen, G. A. & Wasmuth, Ø. (2018). *Grunnparametere innen vegutforming - analyse av fart og friksjon* (Masteroppgave). NTNU.
-

-
- [27] American Association of State Highway and Transportation Officials. (2011). *A Policy on Geometric Design of Highways and Streets* (6th ed.). American Association of State Highway and Transportation Officials.
- [28] American Association of State Highway and Transportation Officials. (2001). *A Policy on Geometric Design of Highways and Streets* (4th ed.). American Association of State Highway and Transportation Officials.
- [29] Trafikverket. (2020). *Vägars och gators utformning - begrepp och grundvärden*. Hentet 20. mai 2021, fra https://trafikverket.ineko.se/Files/sv-SE/71831/Ineko.Product.RelatedFiles/2020_030_vgu_begrepp_och_grundvarden.pdf
- [30] Vejdirektoratet. (2018). *Grundlag for udformning af trafikarealer*. Hentet 27. mai 2021, fra <http://vejregler.lovportaler.dk/showdoc.aspx?q=Grundlag+for+udformning+af+trafikarealer+&docId=vd20180154-full>
- [31] Austroads. (2021). *Guide to Road Design*. Hentet 27. mai 2021, fra <https://austroads.com.au/publications/road-design/agrd-set>
- [32] Transportation Association of Canada. (2011). *Geometric Design Guide for Canadian Roads Part 1*. Hentet 27. mai 2021, fra <https://www.tac-atc.ca/sites/tac-atc.ca/files/site/2011-gdg-revisions.pdf>
- [33] Montana Department of Transportation. (2016). *ROAD DESIGN MANUAL*. Hentet 27. mai 2021, fra <https://www.mdt.mt.gov/publications/manuals.shtml#des>
- [34] Statens Vegvesen. (2018). *Håndbok R211: Feltundersøkelser*. Hentet 17. desember 2020, fra https://www.vegvesen.no/_attachment/111560/binary/1297147?fast_title=H%5C%C3%5C%A5ndbok+R211+Feltunders%5C%C3%5C%B8kelsler%5C%2C+retningslinjer+%5C%2818+MB%5C%29.pdf
- [35] Norrman, J., Eriksson, M. & Lindqvist, S. (2000). Relationships between road slipperiness, traffic accident risk and winter road maintenance activity. *Climate Research*, 15(3), 185–193.
- [36] Statens Vegvesen. (2003). *Temahefte til håndbok 111 Standard for drift og vedlikehold*. Hentet 19. mai 2021, fra https://www.vegvesen.no/_attachment/61431/binary/14138?fast_title=H%5C%C3%5C%A5ndbok+111+Drift+og+vedlikehold%5C%2C+temahefte.pdf
- [37] Dash, J., Rempel, A. & Wettlaufer, J. (2006). The physics of premelted ice and its geophysical consequences. *Reviews of modern physics*, 78(3), 695.
- [38] Karlsson, M. (2001). Prediction of hoar-frost by use of a Road Weather information System. *Meteorological Applications: A journal of forecasting, practical applications, training techniques and modelling*, 8(1), 95–105.
- [39] Ou, T., Hu, Y., Gustavsson, T. & Bogren, J. (2019). On the relationship between the risk of hoar frost on roads and a changing climate in Sweden. *International Journal of Climatology*, 39(5), 2601–2611.
- [40] Fjærestad, J. S. (2020). *Hoar frost formation on a salted road surface* (Doktoravhandling). NTNU.
-

-
- [41] Klein-Paste, A. & Dalen, R. (2018). The Fundamentals of Plowing, Anti-icing, De-icing and Sanding. *Sustainable Winter Road Operations*, 82–100.
- [42] Veneziano, D., Ye, Z. & Turnbull, I. (2013). Speed impacts of an icy curve warning system. *IET Intelligent Transport Systems*, 8(2), 93–101.
- [43] Zhang, J., Das, D. & Peterson, R. (2009). Selection of effective and efficient snow removal and ice control technologies for cold-region bridges. *Journal of Civil, Environmental, and Architectural Engineering*, 3(1), 1–14.
- [44] Liston, L. C., Farnam, Y., Krafcik, M., Weiss, J., Erk, K. & Tao, B. Y. (2016). Binary mixtures of fatty acid methyl esters as phase change materials for low temperature applications. *Applied Thermal Engineering*, 96, 501–507.
- [45] Dave, E. V., Kostick, R. D. & Dailey, J. (2017). Performance of high friction bridge deck overlays in crash reduction. *Journal of Performance of Constructed Facilities*, 31(2), 04016094.
- [46] Jensen, J. K. (2021). *NVDB api V3*. Hentet 7. juni 2021, fra <https://github.com/LtGlahn/nvdbapi-V3>
- [47] Statens Vegvesen. (2021). *Nasjonalt vegdatabank Datakatalog*. Hentet 7. juni 2021, fra <https://datakatalogen.vegvesen.no/>
- [48] Statens Vegvesen. (2013). *Håndbok V120: Premisser for geometrisk utforming av veier*. Hentet 24. mai 2021, fra <https://vegvesen.brage.unit.no/vegvesen-xmli/bitstream/handle/11250/196181/HB-265-2013.pdf?sequence=1&isAllowed=y>
- [49] Statens havarikommisjon for transport. (2021). *RAPPORT OM MØTEULYKKE PÅ E6 VED HAMMER I SNÅSA KOMMUNE 2. FEBRUAR 2020*. (VEI 2021/02). Statens havarikommisjon for transport. Hentet 9. juni 2021, fra <https://havarikommisjonen.no/Vei/Avgitte-rapporter/2021-02>
- [50] Statens Vegvesen. (2011). *Håndbok V714: Veileder i trafikkdata*. Hentet 3. juni 2021, fra https://www.vegvesen.no/_attachment/256135/binary/997080

Vedlegg

Vedlegg A: Datafiler med friksjonsmålinger

Vedlegg A1: Original fil

Date: 07/12/2018 11:05

ViaTrip: 1.0

Filnavn: VTS_2018-1207-110547.dat

Programvareversjon: 1.0

Sted: RV003

Operatør:

Vær:

Målingskommentarer:

Spor (I / M / Y):

Utstyr: 4R61

Utstyrsmærknader:

Bilnummer:

Vegnettserverversjon:

Vegnettsversjon:

Fylke: 50

Vei: RV003

Retning: Mot

Fil: 2

Start Hp: 1

Start: 13414

Type: Fast slipp

Formål:

Stopp Hp: 0

Stopp: 0

Utkjørt distanse: 13375

Tid [ms]	v_TW [km/t]	Slip [%]	μ []	TotalDist [m]	OmT [°C]	OvT [°C]	Fukt [%]	Lat [°]	Long [°]	Veg []	Parsell []	MeterVal [m]	Usikker []	Grenseverdi []
636797775520000000	60.0	18.0	0,45	57	-2,8	-2,6	62,75049	9,98443	RV003	1	13358	false	false	
636797775570000000	60.0	18.0	0,45	138	-2,7	-2,4	62,74981	9,98403	RV003	1	13278	false	false	
636797775620000000	60.0	18.0	0,45	233	-2,7	-2	62,74898	9,98364	RV003	1	13183	false	false	
636797775670000000	60.0	18.0	0,45	337	-2,6	-2,2	62,74804	9,98326	RV003	1	13078	false	false	
636797775720000000	60.0	18.0	0,45	442	-2,6	-2,2	62,74711	9,98298	RV003	1	12973	false	false	
636797775770000000	60.0	18.0	0,45	550	-2,5	-2,6	62,74616	9,98271	RV003	1	12865	false	false	
636797775830000000	60.0	18.0	0,46	660	-2,5	-2,6	62,74519	9,98244	RV003	1	12756	false	false	
636797775880000000	60.0	18.0	0,45	766	-2,4	-2,5	62,74425	9,98219	RV003	1	12650	false	false	
636797775930000000	60.0	18.0	0,45	867	-2,4	-2,4	62,74331	9,98231	RV003	1	12548	false	false	
636797775980000000	60.0	18.0	0,45	965	-2,3	-2,3	62,74251	9,98302	RV003	1	12451	false	false	
636797776030000000	60.0	18.0	0,45	1061	-2,3	-2,3	62,74186	9,98423	RV003	1	12354	false	false	
636797776080000000	60.0	18.0	0,45	1161	-2,2	-2,4	62,74138	9,98585	RV003	1	12255	false	false	
636797776130000000	60.0	18.0	0,45	1263	-2,2	-2	62,7409	9,98754	RV003	1	12153	false	false	
636797776180000000	60.0	18.0	0,45	1368	-2,1	-2	62,74016	9,98892	RV003	1	12048	false	false	
636797776230000000	60.0	18.0	0,45	1475	-2,1	-2,2	62,73926	9,98965	RV003	1	11940	false	false	
636797776280000000	60.0	18.0	0,45	1583	-2,1	-2,3	62,73832	9,99003	RV003	1	11833	false	false	
636797776330000000	60.0	18.0	0,45	1689	-2	-2,3	62,73745	9,99077	RV003	1	11726	false	false	
636797776380000000	60.0	18.0	0,45	1795	-1,9	-2,2	62,7367	9,992	RV003	1	11620	false	false	
636797776430000000	60.0	18.0	0,45	1898	-1,9	-2,1	62,73605	9,99343	RV003	1	11518	false	false	
636797776480000000	60.0	18.0	0,45	2001	-1,8	-2,1	62,73534	9,99485	RV003	1	11415	false	false	
636797776530000000	60.0	18.0	0,45	2103	-1,8	-1,9	62,73458	9,99597	RV003	1	11312	false	false	

636797776580000000 60.0 18.0 0,45 2206 -1,7 -2 62,73371 9,99662 RV003
1 11210 false false
636797776630000000 60.0 18.0 0,45 2309 -1,7 -1,8 62,73289 9,99745
RV003 1 11107 false false
636797776680000000 60.0 18.0 0,45 2415 -1,6 -1,7 62,73209 9,99855
RV003 1 11001 false false
636797776730000000 60.0 18.0 0,45 2523 -1,6 -1,9 62,73124 9,99973
RV003 1 10892 false false
636797776780000000 60.0 18.0 0,45 2629 -1,5 -2 62,73054 10,0011 RV003
1 10786 false false
636797776830000000 60.0 18.0 0,45 2733 -1,5 -1,9 62,72986 10,00248
RV003 1 10682 false false
636797776880000000 60.0 18.0 0,45 2834 -1,4 -1,9 62,7291 10,00353
RV003 1 10582 false false
636797776930000000 60.0 18.0 0,45 2930 -1,4 -2 62,72833 10,0043 RV003
1 10486 false false
636797776980000000 60.0 18.0 0,45 3022 -1,3 -2 62,72775 10,00563
RV003 1 10394 false false
636797777030000000 60.0 18.0 0,45 3111 -1,3 -2,1 62,72741 10,00719
RV003 1 10305 false false
636797777080000000 60.0 18.0 0,45 3203 -1,2 -2,1 62,72701 10,00879
RV003 1 10212 false false
636797777130000000 60.0 18.0 0,45 3300 -1,2 -2,2 62,72638 10,01008
RV003 1 10116 false false
636797777190000000 60.0 18.0 0,45 3398 -1,1 -2 62,7257 10,01131 RV003
1 10017 false false
636797777240000000 60.0 18.0 0,45 3500 -1,1 -1,9 62,72512 10,0128
RV003 1 9915 false false
636797777290000000 60.0 18.0 0,45 3604 -1 -2,1 62,72453 10,01449
RV003 1 9791 false false
636797777340000000 60.0 18.0 0,45 3707 -1 -2,2 62,72394 10,01603
RV003 1 9688 false false
636797777390000000 60.0 18.0 0,45 3807 -0,9 -2 62,7235 10,01773 RV003
1 9588 false false
636797777440000000 60.0 18.0 0,45 3907 -0,9 -2,2 62,72304 10,0194
RV003 1 9489 false false
636797777490000000 60.0 18.0 0,45 4002 -0,8 -2,4 62,72256 10,02097
RV003 1 9393 false false
636797777540000000 60.0 18.0 0,45 4093 -0,8 -2,3 62,72209 10,02243
RV003 1 9303 false false
636797777590000000 60.0 18.0 0,45 4181 -0,8 -2,3 62,72161 10,02387
RV003 1 9215 false false
636797777640000000 60.0 18.0 0,45 4262 -0,7 -2,5 62,72104 10,02489
RV003 1 9134 false false
636797777690000000 60.0 18.0 0,45 4341 -0,7 -2,6 62,72047 10,02578
RV003 1 9054 false false
636797777740000000 60.0 18.0 0,45 4415 -0,7 -2,5 62,72002 10,02681

RV003 1 8981 false false
636797777790000000 60.0 18.0 0,45 4487 -0,6 -2,4 62,71962 10,02793
RV003 1 8908 false false
636797777784000000 60.0 18.0 0,45 4562 -0,6 -2,2 62,7193 10,02918
RV003 1 8833 false false
636797777789000000 60.0 18.0 0,45 4638 -0,6 -2,4 62,71896 10,03054
RV003 1 8757 false false
636797777794000000 60.0 18.0 0,45 4712 -0,5 -2,5 62,71854 10,03168
RV003 1 8683 false false
636797777799000000 60.0 18.0 0,45 4785 -0,5 -2,5 62,71801 10,03253
RV003 1 8611 false false
636797777804000000 60.0 18.0 0,45 4864 -0,4 -2,6 62,7174 10,03321
RV003 1 8532 false false
636797777809000000 60.0 18.0 0,45 4945 -0,4 -2,4 62,71712 10,03466
RV003 1 8450 false false
636797777814000000 60.0 18.0 0,45 5030 -0,4 -2,4 62,7169 10,03621
RV003 1 8366 false false
636797777819000000 60.0 18.0 0,45 5117 -0,3 -2,4 62,71665 10,0379
RV003 1 8279 false false
636797777824000000 60.0 18.0 0,45 5205 -0,3 -2,6 62,71616 10,03925
RV003 1 8191 false false
636797777829000000 60.0 18.0 0,45 5294 -0,3 -2,5 62,71563 10,04052
RV003 1 8101 false false
636797777834000000 60.0 18.0 0,45 5386 -0,3 -2,6 62,71516 10,04181
RV003 1 8010 false false
636797777839000000 60.0 18.0 0,45 5477 -0,2 -2,5 62,71488 10,04342
RV003 1 7918 false false
636797777844000000 60.0 18.0 0,45 5566 -0,2 -2,6 62,71459 10,04509
RV003 1 7829 false false
636797777849000000 60.0 18.0 0,45 5657 -0,2 -2,7 62,71407 10,04646
RV003 1 7739 false false
636797777854000000 60.0 18.0 0,45 5748 -0,2 -2,5 62,71368 10,04801
RV003 1 7648 false false
636797777859000000 60.0 18.0 0,45 5838 -0,2 -2,6 62,71314 10,04934
RV003 1 7558 false false
636797777865000000 60.0 18.0 0,45 5924 -0,1 -2,7 62,71278 10,05079
RV003 1 7471 false false
636797777870000000 60.0 18.0 0,45 6012 -0,1 -2,8 62,71244 10,05234
RV003 1 7383 false false
636797777875000000 60.0 18.0 0,45 6101 -0,1 -2,9 62,71194 10,0537
RV003 1 7295 false false
636797777880000000 60.0 18.0 0,45 6188 -0,1 -2,8 62,7117 10,05535
RV003 1 7207 false false
636797777885000000 60.0 18.0 0,45 6272 -0,1 -2,9 62,71138 10,05685
RV003 1 7123 false false
636797777890000000 60.0 18.0 0,45 6359 -0,1 -2,8 62,71099 10,05828
RV003 1 7037 false false

636797778950000000 60.0 18.0 0,45 6445 0 -2,7 62,71077 10,05989 RV003
1 6950 false false
636797779000000000 60.0 18.0 0,45 6531 0 -2,9 62,71058 10,06151 RV003
1 6864 false false
636797779050000000 60.0 18.0 0,45 6619 0 -2,8 62,71062 10,06318 RV003
1 6777 false false
636797779100000000 60.0 18.0 0,45 6710 0 -2,9 62,71071 10,06502 RV003
1 6685 false false
636797779150000000 60.0 18.0 0,45 6803 0 -3 62,71059 10,06681 RV003
1 6593 false false
636797779200000000 60.0 18.0 0,45 6897 0 -3,1 62,7103 10,06856 RV003
1 6499 false false
636797779250000000 60.0 18.0 0,45 6988 0 -3 62,70998 10,07022 RV003
1 6407 false false
636797779300000000 60.0 18.0 0,45 7082 0,1 -3 62,70965 10,0719 RV003
1 6314 false false
636797779350000000 60.0 18.0 0,45 7178 0,1 -2,8 62,70931 10,07363
RV003 1 6218 false false
636797779400000000 60.0 18.0 0,45 7274 0,1 -2,8 62,70896 10,07541
RV003 1 6122 false false
636797779450000000 60.0 18.0 0,45 7368 0,1 -2,7 62,7087 10,07715
RV003 1 6027 false false
636797779500000000 60.0 18.0 0,45 7461 0,1 -2,7 62,70854 10,07892
RV003 1 5935 false false
636797779550000000 60.0 18.0 0,45 7556 0,1 -2,8 62,70844 10,08065
RV003 1 5840 false false
636797779600000000 60.0 18.0 0,45 7654 0,1 -2,9 62,70833 10,08257
RV003 1 5741 false false
636797779650000000 60.0 18.0 0,45 7754 0,1 -2,7 62,70821 10,0845
RV003 1 5642 false false
636797779700000000 60.0 18.0 0,45 7856 0,2 -2,7 62,70791 10,08645
RV003 1 5540 false false
636797779750000000 60.0 18.0 0,45 7954 0,2 -2,7 62,70769 10,08829
RV003 1 5441 false false
636797779800000000 60.0 18.0 0,45 8050 0,2 -3 62,70757 10,09013 RV003
1 5345 false false
636797779850000000 60.0 18.0 0,45 8144 0,2 -2,8 62,70719 10,09177
RV003 1 5251 false false
636797779900000000 60.0 18.0 0,45 8237 0,2 -2,9 62,70685 10,09347
RV003 1 5158 false false
636797779950000000 60.0 18.0 0,45 8330 0,2 -2,9 62,7065 10,09509
RV003 1 5066 false false
636797780010000000 60.0 18.0 0,45 8421 0,2 -3 62,70597 10,09646 RV003
1 4974 false false
636797780060000000 60.0 18.0 0,45 8515 0,2 -3 62,7054 10,09775 RV003
1 4880 false false
636797780110000000 60.0 18.0 0,45 8607 0,2 -3 62,70498 10,09936 RV003

1 4788 false false
636797780160000000 60.0 18.0 0,45 8700 0,2 -3,2 62,70455 10,10091
RV003 1 4696 false false
636797780210000000 60.0 18.0 0,45 8792 0,2 -3,1 62,70427 10,1026
RV003 1 4603 false false
636797780260000000 60.0 18.0 0,45 8881 0,2 -3 62,70387 10,10409 RV003
1 4494 false false
636797780310000000 60.0 18.0 0,45 8967 0,2 -3 62,70344 10,10549 RV003
1 4408 false false
636797780360000000 60.0 18.0 0,45 9055 0,2 -2,9 62,70308 10,107 RV003
1 4319 false false
636797780410000000 60.0 18.0 0,45 9145 0,2 -3 62,70281 10,10871 RV003
1 4230 false false
636797780460000000 60.0 18.0 0,45 9239 0,2 -3 62,70253 10,11043 RV003
1 4136 false false
636797780510000000 60.0 18.0 0,45 9334 0,2 -3 62,70224 10,11218 RV003
1 4040 false false
636797780560000000 60.0 18.0 0,45 9430 0,2 -3,2 62,70203 10,11398
RV003 1 3945 false false
636797780610000000 60.0 18.0 0,45 9525 0,2 -3,2 62,70192 10,11581
RV003 1 3850 false false
636797780660000000 60.0 18.0 0,45 9622 0,2 -3,1 62,70193 10,11775
RV003 1 3753 false false
636797780710000000 60.0 18.0 0,45 9722 0,2 -3,1 62,7021 10,11966
RV003 1 3652 false false
636797780760000000 60.0 18.0 0,45 9820 0,2 -3,1 62,70241 10,12144
RV003 1 3555 false false
636797780810000000 60.0 18.0 0,45 9915 0,1 -3,3 62,70266 10,12323
RV003 1 3460 false false
636797780860000000 60.0 18.0 0,45 10011 0,1 -3,4 62,70259 10,1251
RV003 1 3364 false false
636797780910000000 60.0 18.0 0,45 10111 0,1 -3,2 62,70223 10,12691
RV003 1 3264 false false
636797780960000000 60.0 18.0 0,45 10204 0,1 -3,3 62,70205 10,12879
RV003 1 3171 false false
636797781010000000 60.0 18.0 0,45 10295 0,1 -3,3 62,7019 10,13054
RV003 1 3080 false false
636797781060000000 60.0 18.0 0,45 10384 0,1 -3,4 62,70149 10,13207
RV003 1 2990 false false
636797781110000000 60.0 18.0 0,45 10471 0,1 -3,5 62,70085 10,13299
RV003 1 2904 false false
636797781160000000 60.0 18.0 0,45 10561 0,1 -3,5 62,70009 10,13348
RV003 1 2814 false false
636797781210000000 60.0 18.0 0,45 10646 0,1 -3,5 62,69944 10,13437
RV003 1 2729 false false
636797781260000000 60.0 18.0 0,45 10739 0,1 -3,6 62,69883 10,13549
RV003 1 2636 false false

636797781310000000 60.0 18.0 0,45 10834 0 -3,8 62,69812 10,13641
RV003 1 2541 false false
636797781370000000 60.0 18.0 0,45 10933 0 -3,6 62,69728 10,13696
RV003 1 2442 false false
636797781420000000 60.0 18.0 0,45 11035 0 -3,5 62,69637 10,13699
RV003 1 2340 false false
636797781470000000 60.0 18.0 0,45 11133 0 -3,4 62,69545 10,13699
RV003 1 2242 false false
636797781520000000 60.0 18.0 0,45 11227 0 -3,6 62,69464 10,13755
RV003 1 2148 false false
636797781570000000 60.0 18.0 0,45 11322 0 -3,3 62,69393 10,13849
RV003 1 2053 false false
636797781620000000 60.0 18.0 0,45 11416 0 -3,2 62,69315 10,1392 RV003
1 1959 false false
636797781670000000 60.0 18.0 0,43 11514 0 -3,5 62,6923 10,13968 RV003
1 1860 false false
636797781720000000 60.0 18.0 0,45 11615 0 -3,6 62,69144 10,14015
RV003 1 1760 false false
636797781770000000 60.0 18.0 0,45 11715 0 -3,5 62,69052 10,14052
RV003 1 1660 false false
636797781820000000 60.0 18.0 0,45 11814 0 -3,4 62,68965 10,14067
RV003 1 1561 false false
636797781870000000 60.0 18.0 0,45 11907 0 -3,6 62,68884 10,14104
RV003 1 1468 false false
636797781920000000 60.0 18.0 0,45 12002 0 -3,5 62,68806 10,14177
RV003 1 1373 false false
636797781970000000 60.0 18.0 0,45 12096 0 -3,5 62,68731 10,14274
RV003 1 1278 false false
636797782020000000 60.0 18.0 0,41 12192 0 -3,6 62,68655 10,1436 RV003
1 1183 false false
636797782070000000 60.0 18.0 0,4 12291 0 -3,6 62,68572 10,1442 RV003
1 1084 false false
636797782120000000 60.0 18.0 0,42 12392 0 -3,5 62,68482 10,14449
RV003 1 983 false false
636797782170000000 60.0 18.0 0,49 12493 0 -3,4 62,68392 10,14448
RV003 1 882 false false
636797782220000000 60.0 18.0 0,45 12595 0 -3,6 62,68335 10,14439
RV003 1 780 false false
636797782270000000 60.0 18.0 0,41 12691 0 -3,5 62,68335 10,14439
RV003 1 684 false false
636797782320000000 60.0 18.0 0,45 12786 0 -3,2 62,68335 10,14439
RV003 1 589 false false
636797782370000000 60.0 18.0 0,45 12884 0 -3,3 62,6806 10,14486 RV003
1 491 false false
636797782420000000 60.0 18.0 0,45 12981 0 -3,4 62,67959 10,14538
RV003 1 394 false false
636797782470000000 60.0 18.0 0,45 13073 0 -3,2 62,67876 10,14582

RV003 1 302 false false
636797782520000000 60.0 18.0 0,45 13161 0 -3,3 62,67798 10,14615
RV003 1 214 false false
636797782570000000 60.0 18.0 0,45 13241 0 -3,4 62,67726 10,14628
RV003 1 134 false false
636797782630000000 60.0 18.0 0,45 13309 0 -3,4 62,67665 10,14628
RV003 1 65 false false
636797782680000000 60.0 18.0 0,45 13361 0 -3,4 62,67618 10,14627
RV003 1 14 false false

Vedlegg A2: Behandlet fil

Date: 07/12/2018 11:05

ViaTrip: 1.0

Filnavn: VTS_2018-1207-110547.dat

Programvareversjon: 1.0

Sted: RV003

Operatør:

Vær:

Målingskommentarer:

Spor (I / M / Y):

Utstyr: 4R61

Utstyrsmærknader:

Bilnummer:

Vegnettserverversjon:

Vegnettsversjon:

Fylke: 50

Vei: RV003

Retning: Mot

Fil: 2

Start Hp: 1

Start: 13414

Type: Fast slipp

Formål:

Stopp Hp: 0

Stopp: 0

Utkjørt distanse: 13375

Tid [ms]	v_TW [km/t]	Slip [%]	μ []	TotalDist [m]	OmT [°C]	OvT [°C]	Fukt [%]	Lat [°]	Long [°]	Veg []	Parsell []	MeterVal [m]	Usikker []	Grenseverdi []	nvdbid	Byggverkstype	Materialtype	Brulengde [m]	Byggear	Avstand til bru [m]	Driftsklasse	Radius
636797775980000000	60.0	18.0	0.45	965	-2.3	-2.3	NULL	62.74251	9.98302													
RV003 1 12451	false	false	356850407	Platebærerbru (380)	Stål	90.9																
NULL 10.787515382841244	DkC	296																				
636797776030000000	60.0	18.0	0.45	1061	-2.3	-2.3	NULL	62.74186	9.98423													
RV003 1 12354	false	false	356850407	Platebærerbru (380)	Stål	90.9																
NULL 0	DkC	296																				
636797776080000000	60.0	18.0	0.45	1161	-2.2	-2.4	NULL	62.74138	9.98585													
RV003 1 12255	false	false	356850407	Platebærerbru (380)	Stål	90.9																
NULL 93.19671255995065	DkC	0																				
636797781010000000	60.0	18.0	0.45	10295	0.1	-3.3	NULL	62.7019	10.13054													
RV003 1 3080	false	false	272336957	Bjelkebru, NOB (331)	Spennbetong																	

11.95 NULL 92.69437963266316 DkC -1439
636797781060000000 60.0 18.0 0.45 10384 0.1 -3.4 NULL 62.70149 10.13207
RV003 1 2990 false false 272336957 Bjelkebru, NOB (331) Spennbetong
11.95 NULL 4.383760925681488 DkC 567
636797781110000000 60.0 18.0 0.45 10471 0.1 -3.5 NULL 62.70085 10.13299
RV003 1 2904 false false 272336957 Bjelkebru, NOB (331) Spennbetong
11.95 NULL 70.14458781653259 DkC -199
636797781570000000 60.0 18.0 0.45 11322 0.0 -3.3 NULL 62.69393 10.13849
RV003 1 2053 false false 377058380 Platebærerbru (372) Stål 100.0
NULL 44.18539724972288 DkC 0
636797781620000000 60.0 18.0 0.45 11416 0.0 -3.2 NULL 62.69315 10.1392
RV003 1 1959 false false 377058380 Platebærerbru (372) Stål 100.0
NULL 0 DkC 0
636797781670000000 60.0 18.0 0.43 11514 0.0 -3.5 NULL 62.6923 10.13968
RV003 1 1860 false false 377058380 Platebærerbru (372) Stål 100.0
NULL 51.26513194242684 DkC 0

Vedlegg B: Koder

Vedlegg B1: Filtrering av datagrunnlag

```
1 import glob
2 import csv
3
4 # Oppgi sti hvor mappen Programmering ligger lagret
5 sti = 'C:\\Programmering\\'
6
7 # Samler stiene til filene som passerer grovsilingen i en vektor
8 relevanteFiler = []
9
10 # Variabler som teller hvor mange filer som passerer hver enkelt test i
    grovsilingen
11 filTeller = 0
12 tomFilTeller = 0
13 liteDataTeller = 0
14 hoyTemperaturTeller = 0
15 ikkeUlikePunkterTeller = 0
16 konstantFriksjonTeller = 0
17 ugyldigVegTeller = 0
18 vellykedeTeller = 0
19
20 # De første for-løkkene henter og starter behandling av filene
21 filSti = sti + 'Friksjonsmålinger\\Rådata\\*'
22 mappeListe = glob.glob(filSti)
23 for mappe in mappeListe:
24     filListe = glob.glob(mappe + '\\*')
25     filTeller = filTeller + len(filListe)
26     for fil in filListe:
27
28         testScore = [-1, -1, -1, -1, -1, -1] # Lager en vektor som holder
styr på hvilke tester filen passerer, 1 = godkjent test, -1 = ikke
godkjent test
29         friksjonVerdierListe = [] # Lager en vektor som holder
friksjonsverdiene
30         lengdegradListe = [] # Lager en vektor som holder lengdegrader
31         breddegradListe = [] # Lager en vektor som holder breddegrader
32         linjeTeller = -1 # Lager en teller for antall linjer i filene
33
34         with open(fil, "r") as originalFil:
35
36             # Teller antall linjer i fila og noterer relevant metadata
underveis
37             for linje in originalFil:
38                 linjeTeller = linjeTeller + 1
39                 if linjeTeller == 17:
40                     veg = linje[5:].strip()
41                 if linjeTeller == 27:
42                     if linje[18:] != '':
43                         distanse = int(linje[18:])
```

```

44         else:
45             distanse = 0
46
47         # Nulte test sjekker om filen har data og dermed om den kan
48         testes for flere testkriterier
49         if linjeTeller > 29:
50             testScore[0] = 1
51
52         # Første test sjekker om filen har mer enn tre linjer som
53         dekker 300 meter
54         if linjeTeller > 31 and distanse >= 300:
55             testScore[1] = 1
56
57         originalFil.seek(0) # Går til starten av filen igjen
58
59         # Leser av andre nyttige verdier på CSV-format
60         aktivFil_reader = csv.reader(originalFil, delimiter='\t')
61         for i in range(30): # Bruker for-løkken til å hoppe over de fø
62         rste 30 linjene med metadata
63             next(aktivFil_reader)
64         for linje in aktivFil_reader:
65             friksjonVerdierListe.append(float(linje[3].replace(',', '.')))
66             lengdegradListe.append(float(linje[8].replace(',', '.')))
67             breddegradListe.append(float(linje[9].replace(',', '.')))
68
69         # Andre test sjekker om det finnes målinger for
70         overflatetemperatur eller lufttemperatur som er lavere enn 1.
71         if linje[5] != '' and linje[6] != '':
72             if float(linje[5].replace(',', '.')) < 1 or float(linje
73             [6].replace(',', '.')) < 1: # Sjekker om det finnes målinger for
74             overflatetemperatur eller lufttemperatur som er lavere enn 1.
75             testScore[2] = 1
76
77         # Tredje test sjekker om lengdegrad og breddegrad endrer seg
78         for i in range(len(lengdegradListe) - 1): # Sjekker om
79         lengdegrad og breddegrad endrer seg
80             if lengdegradListe[i] != lengdegradListe[i + 1] or
81             breddegradListe[i] != breddegradListe[i + 1]:
82                 testScore[3] = 1
83
84         # Fjerde test
85         for i in range(len(friksjonVerdierListe) - 1): # Sjekker om
86         friksjonsverdien endrer seg
87             if friksjonVerdierListe[i] != friksjonVerdierListe[i + 1]:
88                 testScore[4] = 1
89
90         # Femte test sjekker at filen har oppgit vegen som er målt på
91         gyldig format
92         if veg != '': # Sjekker at vegen har et navn
93             if len(veg) - len(veg.replace(' ', '')) == 0: # Sjekker at
94             vegen ikke har mellomrom i navnet
95                 if veg[:2] == 'FV' or veg[:2] == 'EV' or veg[:2] == 'KV
96                 ': # Sjekker om vegen er en fylkesveg, europaveg eller kommunal veg
97                 testScore[5] = 1
98                 elif veg[:2] == 'RV' and veg != 'RV000': # Sjekker om
99                 vegen er en gyldig riksveg

```

```

87         testScore[5] = 1
88
89     # Teller hvilke tester filen har passert
90     if testScore[0] == 1:
91         if testScore[1] == -1:
92             liteDataTeller = liteDataTeller + 1
93         if testScore[2] == -1:
94             hoyTemperaturTeller = hoyTemperaturTeller + 1
95         if testScore[3] == -1:
96             ikkeUlikePunkterTeller = ikkeUlikePunkterTeller + 1
97         if testScore[4] == -1:
98             konstantFriksjonTeller = konstantFriksjonTeller + 1
99         if testScore[5] == -1:
100             ugyldigVegTeller = ugyldigVegTeller + 1
101     else:
102         tomFilTeller = tomFilTeller + 1
103
104     # Samler stier til filer som har bestått alle tester i en vektor
105     if sum(testScore) == 6:
106         vellykedeTeller = vellykedeTeller + 1
107         relevanteFiler.append(fil)
108
109 # Samler stier til alle relevante filer i et tekstdokumentre
110 QGIS_input = sti + 'Friksjonsmålinger\\QGIS_input\\Input.txt'
111 with open(QGIS_input, 'w+') as output:
112     for filer in relevanteFiler:
113         splittetSti = filer.split('\\')
114         output.write(splittetSti[-4] + '\\\\' + splittetSti[-3] + '\\\\' +
115             splittetSti[-2] + '\\\\' + splittetSti[-1] + '\\n')
116
117 # Printer ut informasjon om antall filer som har passert testene
118 print('Antall filer: {} stykker'.format(filTeller))
119 print('Antall tomme filer: {} stykker'.format(tomFilTeller))
120 print('Antall filer med data: {} stykker = 100%'.format(filTeller-
121     tomFilTeller))
122 print('Antall filer med for lite data: {} stykker = {:.2f}%'.format(
123     liteDataTeller, liteDataTeller/(filTeller-tomFilTeller)*100))
124 print('Antall filer med for høy temperatur: {} stykker = {:.2f}%'.format(
125     hoyTemperaturTeller, hoyTemperaturTeller/(filTeller-tomFilTeller)*100))
126 print('Antall filer som kun måler ett punkt: {} stykker = {:.2f}%'.format(
127     ikkeUlikePunkterTeller, ikkeUlikePunkterTeller/(filTeller-tomFilTeller)
128     *100))
129 print('Antall filer med konstant friksjon: {} stykker = {:.2f}%'.format(
130     konstantFriksjonTeller, konstantFriksjonTeller/(filTeller-tomFilTeller)
131     *100))
132 print('Antall filer med ugyldige veger: {} stykker = {:.2f}%'.format(
133     ugyldigVegTeller, ugyldigVegTeller/(filTeller-tomFilTeller)*100))
134 print('Antall vellykkede filer: {} stykker = {:.2f}%'.format(
135     vellykedeTeller, vellykedeTeller/(filTeller-tomFilTeller)*100))

```

Vedlegg B2: QGIS main

```
1 # Skriv inn stien som leder til der mappen 'Programmering' er lagret
2 # Eks: 'C:\\Bruker\\Dokumenter\\Programmerg\\
3 sti = 'C:\\Programmering\\'
4
5 import sys
6
7 # Definerer søkestier som funksjoner skal hentes fra
8 NVDB_bibliotek= sti+'QGIS/nvdbapi-V3-master/'
9 kodeSti = sti+'QGIS/Custom'
10 if not [k for k in sys.path if 'nvdbapi-V3' in k]:
11     sys.path.append(NVDB_bibliotek)
12 if not [k for k in sys.path if 'Custom' in k]:
13     sys.path.append(kodeSti)
14
15 # Importerer funksjonene som skal brukes
16 from Hjelpfunksjoner import FinnFiler, FilInformasjon, LeggTilMaalinger,
17     LeggTilBru, TestBruEksistens, LeggTilDriftsklasse, LeggTilKurvatur,
18     LagBuffer, ArvNVDB_Bru, ArvNVDB_Annet, FerdigFil, Nullstill
19
20 # Lager en vektor med filnavnene som skal behandles
21 samlingFilnavn = FinnFiler(sti)
22
23 # Starter løkken som behandler én fil om gangen
24 for fil in samlingFilnavn:
25
26     # Leser filens verdi for dato, system, fylke, vegreferanse og distanse
27     metaData = FilInformasjon(fil)
28     # Definerer variablene som brukes i funksjonene
29     metaDataLinjer = metaData[0]
30     filnavn = metaData[1]
31     veg = metaData[2]
32
33     # Importerer friksjonsmålingene til QGIS og lager et område som dekker
34     # alle målepunktene
35     kartutsnitt = LeggTilMaalinger(fil)
36
37     # Henter bruer fra NVDB og legger dem til i listen over lag
38     LeggTilBru(veg, kartutsnitt)
39
40     # Behandler kun filer hvor det måles friksjon på bru
41     bruTest = TestBruEksistens()
42     if bruTest == 1:
43
44         # Legger til driftsklassene
45         LeggTilDriftsklasse(veg, kartutsnitt)
46         # Legger til kurvatur
47         LeggTilKurvatur(veg, kartutsnitt)
48         # Lager buffer rundt bru, driftsklasse og kurvatur
49         LagBuffer()
50         # Knytter friksjonsmålingene til brudata fra NVDB og beregner
51         # avstander
52         ArvNVDB_Bru()
53         # Legger til ytterligere informasjon angående driftsklasse og
```

```
radius fra NVDB
50     ArvNVDB_Annet()
51     # Skriver den nye informasjonen til en DAT-fil
52     FerdigFil(sti, filnavn, metaDataLinjer)
53
54     # Nullstiller arbeidsområdet for å klargjøre nye kjøring
55     Nullstill()
56 print('Behandlet {} filer'.format(len(samlingFilnavn)))
```

Vedlegg B3: QGIS hjelpefunksjoner

```
1 def FinnFiler(sti):
2     import codecs
3
4     filliste = []
5     inputFiler = sti + 'Friksjonsmålinger\\QGIS_input\\Input.txt'
6
7     # Henter filstiene til filene som skal behand
8     with codecs.open(inputFiler, 'r', 'latin-1') as QGIS_input: # Åpner
9         for fil in QGIS_input:
10             filliste.append(sti + fil.rstrip())
11     return(filliste)
12
13
14 def FilInformasjon(fil):
15     # Definerer en vektor som holder data underveis
16     originaleLinjer = []
17
18     # Leser metadata fra filen og legger det til vektoren
19     f = open(fil, "r", encoding="latin-1")
20     for position, line in enumerate(f):
21         originaleLinjer.append(line)
22         if position == 28:
23             break
24     f.close()
25
26     # Noterer informasjon filnavn og veg
27     filnavn = originaleLinjer[3][9:].strip()
28     veg = originaleLinjer[17][5:].strip()
29
30     return(originaleLinjer, filnavn, veg)
31
32
33 def LeggTilMaalinger(fil):
34     import processing
35     from qgis.core import QgsVectorLayer, QgsVectorFileWriter,
36     QgsCoordinateReferenceSystem, QgsProject
37
38     # Definerer hvordan filen skal leses og behandles
39     uri = 'file:///{}?encoding=latin1&type=csv&delimiter=%5Ct&skipLines=28&
40     maxFields=10000&detectTypes=yes&decimalPoint=, &xField=Long%20[%C2%B0]&
41     yField=Lat%20[%C2%B0]&crs=EPSG:4326&spatialIndex=no&subsetIndex=no&
42     watchFile=no'.format(fil.replace('å', '%C3%A5'))
43
44     # Ber QGIS om å ta inn filen som et vektorlag
45     friksjonspunkter = QgsVectorLayer(uri, "FriksjonPunkter", "
46     delimitedtext")
47     QgsProject.instance().addMapLayer(friksjonspunkter)
48
49     # Lager en utstrekning som dekker friksjonspunktene
50     processing.runAndLoadResults("native:polygonfromlayerextent", {'INPUT':
51     friksjonspunkter,
52     'ROUND_TO': 0,
```

```

47         'OUTPUT': 'TEMPORARY_OUTPUT'})
48     utstrekning = QgsProject.instance().mapLayersByName('Extent')[0]
49
50     # Omprojiserer utstrekningen fra WGS84 til UTM33
51     processing.runAndLoadResults("native:reprojectlayer", {'INPUT':
utstrekning,
52         'TARGET_CRS': QgsCoordinateReferenceSystem('EPSG:5973'),
53         'OPERATION': '+proj=pipeline +step +proj=unitconvert +
xy_in=deg +xy_out=rad +step +proj=utm +zone=33 +ellps=GRS80',
54         'OUTPUT': 'TEMPORARY_OUTPUT'})
55     omprojisertUtstrekning = QgsProject.instance().mapLayersByName('
Reprojected')[0]
56
57     # Endrer koordinatsystemet i QGIS til UTM33
58     QgsProject.instance().setCrs(QgsCoordinateReferenceSystem(5973))
59
60     # Definerer koordinatene som dekker utstrekningen
61     ext = omprojisertUtstrekning.extent()
62     kartutsnitt = str(ext.xMinimum()) + ',' + \
63         str(ext.yMinimum()) + ',' + str(ext.xMaximum()) + \
64         ',' + str(ext.yMaximum())
65     return(kartutsnitt)
66
67
68 def LeggTilBru(veg, kartutsnitt):
69     from nvdbapiv3 import nvdbFagdata
70     from nvdbapiV3qgis3 import nvdbSok2qgis
71
72     # Henter bruer fra NVDB
73     sokeobj = nvdbFagdata(60)
74     sokeobj.filter({'vegssystemreferanse': veg}) # Aktuell veg
75     sokeobj.filter({'egenskap': '11317=19262 AND 1263=7304'}) # Kun
trafikkerte vegbruer
76     sokeobj.filter({'srid': 'utm33'}) # Setter CRS for kartutsnittet
77     sokeobj.filter({'kartutsnitt': kartutsnitt}) # Begrenser området det
hentes data fra
78     nvdbSok2qgis(sokeobj)
79
80
81 def TestBruEksistens():
82     from qgis.core import QgsProject
83
84     # Returnerer 1 dersom bruer har blitt importert, ellers returneres -1
85     resultat = -1
86     lagListe = QgsProject.instance().mapLayers()
87     for lag in lagListe:
88         if lag[:3] == 'Bru' and lag[:6] != 'Bru_2d':
89             resultat = 1
90     return(resultat)
91
92
93 def LeggTilDriftsklasse(veg, kartutsnitt):
94     from nvdbapiv3 import nvdbFagdata
95     from nvdbapiV3qgis3 import nvdbSok2qgis
96
97     # Henter vinterdriftsklasser fra NVDB
98     sokeobj = nvdbFagdata(810)

```

```

99     sokeobj.filter({'vegssystemreferanse': veg}) # Aktuell veg
100    sokeobj.filter({'egenskap': '9260!=12909 OR 9260!=12910 OR 9260!=15876'
101    }) # Ikke GsA, GsB eller sideanlegg
102    sokeobj.filter({'srid': 'utm33'}) # Setter CRS for kartutsnittet
103    sokeobj.filter({'kartutsnitt': kartutsnitt}) # Begrenser området det
104    hentes data fra
105    nvdb sok2qgis(sokeobj)
106
107 def LeggTilKurvatur(veg, kartutsnitt):
108     from nvdbapiv3 import nvdbFagdata
109     from nvdbapiV3qgis3 import nvdb sok2qgis
110
111     # Henter horisontalelementer fra NVDB
112     sokeobj = nvdbFagdata(639)
113     sokeobj.filter({'vegssystemreferanse': veg}) # Aktuell veg
114     sokeobj.filter({'srid': 'utm33'}) # Setter CRS for kartutsnittet
115     sokeobj.filter({'kartutsnitt': kartutsnitt}) # Begrenser området det
116     hentes data fra
117     nvdb sok2qgis(sokeobj)
118
119 def LagBuffer():
120     import processing
121     from qgis.core import QgsVectorLayer, QgsProject, QgsField, edit,
122     QgsCoordinateReferenceSystem
123     from PyQt5.QtCore import QVariant
124
125     # Sjekker om det finnes lag for kurvatur og vinterdriftsklasse
126     kurvatur = -1
127     driftsklasse = -1
128     lagListe = QgsProject.instance().mapLayers()
129     for lag in lagListe:
130         if lag[:8] == 'Kurvatur':
131             kurvatur = 1
132         if lag[:18] == 'Vinterdriftsklasse':
133             driftsklasse = 1
134
135     if kurvatur == -1:
136         # Opretter et tomt lag for kurvatur som kan brukes videre i
137         behandlingen
138         tomKurvatur = QgsVectorLayer("Point", "TomKurvatur", "memory")
139         with edit(tomKurvatur):
140             tomKurvatur.addAttribute(QgsField('Radius', QVariant.String))
141             tomKurvatur.updateFields()
142             processing.runAndLoadResults("native:reprojectlayer", {'INPUT':
143             tomKurvatur,
144             'TARGET_CRS':QgsCoordinateReferenceSystem('EPSG:5973'),
145             'OPERATION':'+proj=pipeline +step +proj=unitconvert +
146             xy_in=deg +xy_out=rad +step +proj=utm +zone=33 +ellps=GRS80',
147             'OUTPUT': 'TEMPORARY_OUTPUT'})
148             omprojisertTomKurvatur = QgsProject.instance().mapLayersByName('
149             Reprojected')[0]
150             processing.runAndLoadResults("qgis:renamelayern", {'INPUT':
151             omprojisertTomKurvatur,
152             'NAME': 'Kurvatur, horisontalelement'})

```

```

147     if driftsklasse == -1:
148         # Opretter et tomt lag for vinterdriftsklasse som kan brukes videre
           i behandlingen
149         tomVinterdriftsklasse = QgsVectorLayer("Point", "
TomVinterdriftsklasse", "memory")
150         with edit(tomVinterdriftsklasse):
151             tomVinterdriftsklasse.addAttribute(QgsField('Driftsklasse',
QVariant.String))
152             tomVinterdriftsklasse.updateFields()
153             processing.runAndLoadResults("native:reprojectlayer", {'INPUT':
tomVinterdriftsklasse,
154                 'TARGET_CRS':QgsCoordinateReferenceSystem('EPSG:5973'),
155                 'OPERATION':'+proj=pipeline +step +proj=unitconvert +
xy_in=deg +xy_out=rad +step +proj=utm +zone=33 +ellps=GRS80',
156                 'OUTPUT': 'TEMPORARY_OUTPUT'})
157             omprojisertTomVinterdriftsklasse = QgsProject.instance().
mapLayersByName('Reprojected')[0]
158             processing.runAndLoadResults("qgis:renamelaye", {'INPUT':
omprojisertTomVinterdriftsklasse,
159                 'NAME': 'Vinterdriftsklasse'})
160
161         # Lager en buffer som gir bruene en total bredde på 20 meter og som
           stopper der brua stopper i lengderetning
162         bruer = QgsProject.instance().mapLayersByName('Bru')[0]
163         processing.runAndLoadResults("native:buffer", {'INPUT': bruer,
164             'DISTANCE': 10,
165             'SEGMENTS': 5,
166             'END_CAP_STYLE': 1,
167             'JOIN_STYLE': 1,
168             'MITER_LIMIT': 2,
169             'DISSOLVE': False,
170             'OUTPUT': 'TEMPORARY_OUTPUT'})
171         bruBuffer = QgsProject.instance().mapLayersByName('Buffered')[0]
172         processing.runAndLoadResults("qgis:renamelaye", {'INPUT': bruBuffer,
173             'NAME': 'BruBuffer'})
174
175         # Lager buffer rundt driftsklassene
176         vinterdriftsklasse = QgsProject.instance().mapLayersByName('
Vinterdriftsklasse')[0]
177         processing.runAndLoadResults("native:buffer", {'INPUT':
vinterdriftsklasse,
178             'DISTANCE': 10,
179             'SEGMENTS': 5,
180             'END_CAP_STYLE': 1,
181             'JOIN_STYLE': 1,
182             'MITER_LIMIT': 2,
183             'DISSOLVE': False,
184             'OUTPUT': 'TEMPORARY_OUTPUT'})
185         vinterdriftsklasseBuffer = QgsProject.instance().mapLayersByName('
Buffered')[0]
186         processing.runAndLoadResults("qgis:renamelaye", {'INPUT':
vinterdriftsklasseBuffer,
187             'NAME': 'VinterdriftsklasseBuffer'})
188
189         # Lager buffer rundt kurvaturen
190         kurvatur = QgsProject.instance().mapLayersByName('Kurvatur,
horisontalelement')[0]

```

```

191     processing.runAndLoadResults("native:buffer", {'INPUT': kurvatur,
192         'DISTANCE': 10,
193         'SEGMENTS': 5,
194         'END_CAP_STYLE': 1,
195         'JOIN_STYLE': 1,
196         'MITER_LIMIT': 2,
197         'DISSOLVE': False,
198         'OUTPUT': 'TEMPORARY_OUTPUT'})
199     kurvaturBuffer = QgsProject.instance().mapLayersByName('Buffered')[0]
200     processing.runAndLoadResults("qgis:renamelayer", {'INPUT':
201     kurvaturBuffer,
202         'NAME': 'KurvaturBuffer'})
203
204 def ArvNVDB_Bru():
205     import processing
206     from qgis.core import QgsProject, QgsField, QgsFeature, NULL, edit
207     from PyQt5.QtCore import QVariant
208
209     # Finner endepunktene til bruene
210     bruer = QgsProject.instance().mapLayersByName('Bru')[0]
211     processing.runAndLoadResults("native:extractspecificvertices", {'INPUT':
212     :bruer,
213         'VERTICES': '0, -1',
214         'OUTPUT': 'TEMPORARY_OUTPUT'})
215     bruEndepunkter = QgsProject.instance().mapLayersByName('Vertices')[0]
216     processing.runAndLoadResults("qgis:renamelayer", {'INPUT':
217     bruEndepunkter,
218         'NAME': 'BruEndepunkter'})
219
220     # Lager et buffer rundt endepunktene
221     bruEndepunkter = QgsProject.instance().mapLayersByName('BruEndepunkter'
222     ) [0]
223     processing.runAndLoadResults("native:buffer", {'INPUT':bruEndepunkter,
224         'DISTANCE':100,
225         'SEGMENTS':5,
226         'END_CAP_STYLE':0,
227         'JOIN_STYLE':0,
228         'MITER_LIMIT':2,
229         'DISSOLVE':False,
230         'OUTPUT': 'TEMPORARY_OUTPUT'})
231     bruEndepunkterBuffer = QgsProject.instance().mapLayersByName('Buffered'
232     ) [0]
233     processing.runAndLoadResults("qgis:renamelayer", {'INPUT':
234     bruEndepunkterBuffer,
235         'NAME': 'BruEndepunkterBuffer'})
236
237     # Lager to korrigerede buffer
238     # Første buffer kombinerer brubufferet med bufferet rundt endepunktene
239     # Andre buffer ekskluderer brubufferet fra bufferet til endepunktene
240     bruEndepunkterBuffer = QgsProject.instance().mapLayersByName('
241     BruEndepunkterBuffer')[0]
242     bruBuffer = QgsProject.instance().mapLayersByName('BruBuffer')[0]
243     processing.runAndLoadResults("native:union", {'INPUT':
244     bruEndepunkterBuffer,
245         'OVERLAY':bruBuffer,
246         'OVERLAY_FIELDS_PREFIX':'',

```

```

240         'OUTPUT': 'TEMPORARY_OUTPUT'})
241     altBruBuffer = QgsProject.instance().mapLayersByName('Union')[0]
242     processing.runAndLoadResults("qgis:renamelayer", {'INPUT': altBruBuffer
243     ,
244         'NAME': 'AltBruBuffer'})
245     processing.runAndLoadResults("native:difference", {'INPUT':
246     bruEndepunkterBuffer,
247         'OVERLAY': bruBuffer,
248         'OUTPUT': 'TEMPORARY_OUTPUT'})
249     inngangBruBuffer = QgsProject.instance().mapLayersByName('Difference')
250     [0]
251     processing.runAndLoadResults("qgis:renamelayer", {'INPUT':
252     inngangBruBuffer,
253         'NAME': 'InngangBruBuffer'})
254
255     # Finner alle målepunkter på og ved inngangen til bruene
256     friksjonPunkter = QgsProject.instance().mapLayersByName('
257     FriksjonPunkter')[0]
258     altBruBuffer = QgsProject.instance().mapLayersByName('AltBruBuffer')[0]
259     processing.runAndLoadResults("native:intersection", {'INPUT':
260     friksjonPunkter,
261         'OVERLAY': altBruBuffer,
262         'INPUT_FIELDS': '',
263         'OVERLAY_FIELDS': ['nvdbid', 'Byggverkstype', 'Materialtype
264     ', 'Lengde', 'Byggeår'],
265         'OVERLAY_FIELDS_PREFIX': '',
266         'OUTPUT': 'TEMPORARY_OUTPUT'})
267     alleBruPunkter = QgsProject.instance().mapLayersByName('Intersection')
268     [0]
269     processing.runAndLoadResults("qgis:renamelayer", {'INPUT':
270     alleBruPunkter,
271         'NAME': 'AlleBruPunkter'})
272
273     # Finner målepunkter som er ved inngangen til brua = Friksjonspunkter
274     som er mindre enn 100 meter unna en bru, men ikke på en bru
275     alleBruPunkter = QgsProject.instance().mapLayersByName('AlleBruPunkter'
276     ) [0]
277     inngangBruBuffer = QgsProject.instance().mapLayersByName('
278     InngangBruBuffer')[0]
279     processing.runAndLoadResults("native:intersection", {'INPUT':
280     alleBruPunkter,
281         'OVERLAY': inngangBruBuffer,
282         'INPUT_FIELDS': '',
283         'OVERLAY_FIELDS': '',
284         'OVERLAY_FIELDS_PREFIX': '',
285         'OUTPUT': 'TEMPORARY_OUTPUT'})
286     inngangBruPunkter = QgsProject.instance().mapLayersByName('Intersection
287     ')[0]
288     processing.runAndLoadResults("qgis:renamelayer", {'INPUT':
289     inngangBruPunkter,
290         'NAME': 'InngangBruPunkter'})
291
292     # Finner distanser fra punktene til bruenes endepunkter
293     inngangBruPunkter = QgsProject.instance().mapLayersByName('
294     InngangBruPunkter')[0]
295     bruEndepunkter = QgsProject.instance().mapLayersByName('BruEndepunkter'
296     ) [0] # Gjentar linjen fra tidligere for å forbedre lesbarhet

```

```

280     processing.runAndLoadResults("qgis:distancetonearesthublinetohub", {'
INPUT': inngangBruPunkter,
281         'HUBS': bruEndepunkter,
282         'FIELD': 'nvdbid',
283         'UNIT': 0,
284         'OUTPUT': 'TEMPORARY_OUTPUT'})
285     avstandTilBru = QgsProject.instance().mapLayersByName('Hub distance')
[0]
286     processing.runAndLoadResults("qgis:renamelayer", {'INPUT':
avstandTilBru,
287         'NAME': 'AvstandTilBru'})
288
289     # Legger til distansene til laget med alle inngangspunktene
290     alleBruPunkter = QgsProject.instance().mapLayersByName('AlleBruPunkter'
) [0]
291     avstandTilBru = QgsProject.instance().mapLayersByName('AvstandTilBru')
[0]
292     processing.runAndLoadResults("native:joinattributesbylocation", {'INPUT
': alleBruPunkter,
293         'JOIN': avstandTilBru,
294         'PREDICATE': [0],
295         'JOIN_FIELDS': ['HubDist'],
296         'METHOD': 0,
297         'DISCARD_NONMATCHING': False,
298         'PREFIX': '',
299         'OUTPUT': 'TEMPORARY_OUTPUT'})
300     alleBruPunkterMedLengder = QgsProject.instance().mapLayersByName('
Joined layer') [0]
301     processing.runAndLoadResults("qgis:renamelayer", {'INPUT':
alleBruPunkterMedLengder,
302         'NAME': 'AlleBruPunkterMedLengder'})
303
304     # Oppdaterer verdiene på bruer fra NULL til 0
305     alleBruPunkterMedLengder = QgsProject.instance().mapLayersByName('
AlleBruPunkterMedLengder') [0]
306     egenskapNavn11 = 'Lengde'
307     egenskapNavn12 = 'Brulengde [m]'
308     egenskapNavn21 = 'HubDist'
309     egenskapNavn22 = 'Avstand til bru [m]'
310     egenskapVerdil = NULL
311     egenskapVerdi2 = '0'
312     with edit(alleBruPunkterMedLengder):
313         for felt in alleBruPunkterMedLengder.fields():
314             if felt.name() == egenskapNavn11:
315                 idx = alleBruPunkterMedLengder.fields().indexOfName(felt.
name())
316                 alleBruPunkterMedLengder.renameAttribute(idx,
egenskapNavn12)
317             if felt.name() == egenskapNavn21:
318                 idx = alleBruPunkterMedLengder.fields().indexOfName(felt.
name())
319                 alleBruPunkterMedLengder.renameAttribute(idx,
egenskapNavn22)
320                 alleBruPunkterMedLengder.updateFields()
321             for egenskap in alleBruPunkterMedLengder.getFeatures():
322                 if egenskap[egenskapNavn22] == egenskapVerdil:
323                     egenskap[egenskapNavn22] = egenskapVerdi2

```

```

324         alleBruPunkterMedLengder.updateFeature(egenskap)
325
326
327 def ArvNVDB_Annet():
328     import processing
329     from qgis.core import QgsProject
330
331     alleBruPunkterMedLengder = QgsProject.instance().mapLayersByName('
AlleBruPunkterMedLengder')[0]
332     vinterdriftsklasseBuffer = QgsProject.instance().mapLayersByName('
VinterdriftsklasseBuffer')[0]
333     processing.runAndLoadResults("native:joinattributesbylocation", {'INPUT
': alleBruPunkterMedLengder,
334         'JOIN':vinterdriftsklasseBuffer,
335         'PREDICATE':[0],
336         'JOIN_FIELDS':['Driftsklasse'],
337         'METHOD':0,
338         'DISCARD_NONMATCHING':False,
339         'PREFIX':'',
340         'OUTPUT':'TEMPORARY_OUTPUT'})
341     friksjonPunkter_Bru_DK = QgsProject.instance().mapLayersByName('Joined
layer')[0]
342     processing.runAndLoadResults("qgis:renamelayer", {'INPUT':
friksjonPunkter_Bru_DK,
343         'NAME': 'FriksjonPunkter_Bru_DK'})
344
345     friksjonPunkter_Bru_DK = QgsProject.instance().mapLayersByName('
FriksjonPunkter_Bru_DK')[0]
346     kurvaturBuffer = QgsProject.instance().mapLayersByName('KurvaturBuffer'
)[0]
347     processing.runAndLoadResults("native:joinattributesbylocation", {'INPUT
': friksjonPunkter_Bru_DK,
348         'JOIN':kurvaturBuffer,
349         'PREDICATE':[0],
350         'JOIN_FIELDS':['Radius'],
351         'METHOD':0,
352         'DISCARD_NONMATCHING':False,
353         'PREFIX':'',
354         'OUTPUT':'TEMPORARY_OUTPUT'})
355     friksjonPunkter_Bru_DK_R = QgsProject.instance().mapLayersByName('
Joined layer')[0]
356     processing.runAndLoadResults("qgis:renamelayer", {'INPUT':
friksjonPunkter_Bru_DK_R,
357         'NAME': 'FriksjonPunkter_Bru_DK_R'})
358
359
360 def FerdigFil(sti, filnavn, metaDataLinjer):
361     from qgis.core import QgsProject
362     utSti = sti + 'Friksjonsmålinger\\QGIS_output\\' + filnavn
363
364     friksjonPunkter_Bru_DK_R = QgsProject.instance().mapLayersByName('
FriksjonPunkter_Bru_DK_R')[0]
365
366     feltListe = []
367     for felt in friksjonPunkter_Bru_DK_R.fields():
368         feltListe.append(felt.name())
369

```

```
370     with open(utSti, 'w') as utFil:
371         for linje in metaDataLinjer:
372             utFil.write(linje)
373         for i in range(len(feltListe) - 1):
374             utFil.write('{}\t'.format(feltListe[i]))
375         utFil.write('{}\n'.format(feltListe[len(feltListe)-1]))
376         for felt in friksjonPunkter_Bru_DK_R.getFeatures():
377             for i in range(len(feltListe) - 1):
378                 utFil.write('{}\t'.format(felt[feltListe[i]]))
379                 utFil.write('{}\n'.format(felt[feltListe[len(feltListe)-1]]))
380
381
382 def Nullstill():
383     from qgis.core import QgsProject
384
385     # Tilbakestillert midlertidige lag slik at de kan lukkes uten melding i
brukergrensesnittet
386     for lag in QgsProject.instance().mapLayers().values():
387         lag.startEditing()
388         lag.rollback()
389     QgsProject.instance().removeAllMapLayers()
```

Vedlegg B4: Rensing av filer

```
1 import glob
2 import os
3
4 # Oppgi sti hvor mappen Programmering ligger lagret
5 sti = 'C:\\Programmering\\'
6
7 innSti = sti + 'Friksjonsmålinger\\QGIS_output\\*'
8 filliste = glob.glob(innSti)
9 for fil in filliste:
10     utSti = sti + 'Friksjonsmålinger\\Renset\\' + os.path.basename(fil)
11     with open(fil, 'r') as innFil: # Åpner filen som skal leses
12         with open(utSti, 'w') as utFil: # Åpner filen det skal skrives til
13             linjeteller = -1
14             tidListe = ['0']
15             dataListe = []
16             for linje in innFil:
17                 linjeteller = linjeteller + 1
18                 if linjeteller < 30: # Kopierer de første linjene med
19                     utFil.write(linje)
20                     elif linjeteller >= 30: # Noterer unike tidsmålinger med
21                         tid = linje[:18]
22                         if tid not in tidListe:
23                             tidListe.append(tid)
24                             dataListe.append(linje)
25                 dataListe.sort() # Sorterer data i kronologisk rekkefølge
26                 for linje in dataListe:
27                     utFil.write(linje) # Skriver sortert data til den nye
rensede filen
```

Vedlegg B5: Lage matrise

```
1 import glob
2 import csv
3 import numpy as np
4
5 # Oppgi sti hvor mappen Programmering ligger lagret
6 sti = 'C:Programmering\\'
7
8 # Filer som skal analyseres
9 innSti = sti + 'Friksjonsmålinger\\Renset\\*'
10 utSti = sti + 'Python\\matrise.npy'
11 filListe = glob.glob(innSti)
12
13 # Lister som holder verider
14 dagListe = []
15 maanedListe = []
16 aarListe = []
17 klokkeslettListe = []
18 filnavnListe = []
19 utstyrListe = []
20 fylkeListe = []
21 vegListe = []
22 friksjonListe = []
23 luftTemperaturListe = []
24 vegTemperaturListe = []
25 fuktListe = []
26 bruIDListe = []
27 bruTypeListe = []
28 bruMaterialeListe = []
29 bruLengdeListe = []
30 bruAarListe = []
31 avstandListe = []
32 driftsklasseListe = []
33 radiusListe = []
34
35 for fil in filListe:
36     linjeTeller = -1
37     with open(fil, 'r', encoding='utf-8') as originalFil:
38         for linje in originalFil: # Henter interessant informasjon fra
39             linjeTeller = linjeTeller + 1
40             linjeTeller == 0:
41                 dag = linje[6:8]
42                 maaned = linje[9:11]
43                 aar = linje[12:16]
44                 klokkeslett = linje[17:22]
45             linjeTeller == 3:
46                 filnavn = linje[9:].strip()
47             linjeTeller == 10:
48                 utstyr = linje[8:].strip()
49             linjeTeller == 16:
50                 fylke = linje[7:].strip()
51             linjeTeller == 17:
52                 veg = linje[5:].strip()
```

```

53         if linjeTeller == 18:
54             break
55     originalFil.seek(0) # Går til starten av filen igjen
56
57     aktivFil_reader = csv.reader(originalFil, delimiter='\t')
58     for i in range(30): # Bruker for-løkken til å hoppe over de første
30 linjene med metadata
59         next(aktivFil_reader)
60     for linje in aktivFil_reader: # Henter ut interessant data
61         friksjon = float(linje[3])
62         luftTemperatur = float(linje[5])
63         vegTemperatur = float(linje[6])
64         fukt = linje[7]
65         bruID = linje[15]
66         bruType = linje[16]
67         bruMateriale = linje[17]
68         bruLengde = linje[18]
69         bruAar = linje[19]
70         avstand = float(linje[20])
71         driftsklasse = linje[21]
72         radius = linje[22]
73         if bruID != 'None' and friksjon != -1 and avstand <= 100 and (
luftTemperatur <= 1 or vegTemperatur <= 1): # Legger til data dersom
friksjonen og avstanden er gyldig, samt temperatur lavere enn 1 grad
celsiusus
74             dagListe.append(dag)
75             maanedListe.append(maaned)
76             aarListe.append(aar)
77             klokkeslettListe.append(klokkeslett)
78             filnavnListe.append(filnavn)
79             utstyrListe.append(utstyr)
80             fylkeListe.append(fylke)
81             vegListe.append(veg)
82             friksjonListe.append(friksjon)
83             luftTemperaturListe.append(luftTemperatur)
84             vegTemperaturListe.append(vegTemperatur)
85             fuktListe.append(fukt)
86             bruIDListe.append(bruID)
87             bruTypeListe.append(bruType)
88             bruMaterialeListe.append(bruMateriale)
89             bruLengdeListe.append(bruLengde)
90             bruAarListe.append(bruAar)
91             avstandListe.append(avstand)
92             driftsklasseListe.append(driftsklasse)
93             radiusListe.append(radius)
94
95 matrise = np.array([filnavnListe, bruIDListe, avstandListe, friksjonListe,
driftsklasseListe, luftTemperaturListe, # samler listene i en matrise
96                     vegTemperaturListe, fuktListe, bruTypeListe,
bruMaterialeListe, bruLengdeListe, bruAarListe,
97                     radiusListe, utstyrListe, vegListe, fylkeListe,
aarListe, maanedListe, dagListe, klokkeslettListe])
98 np.save(utSti, np.transpose(matrise))

```

Vedlegg B6: Analyse

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import matplotlib.gridspec as gridspec
4 import matplotlib.ticker as ticker
5 from collections import Counter
6 import scipy.stats
7 from scipy.stats import gaussian_kde
8
9 # Oppgi sti til mappen hvor matrisen ligger lagret
10 sti = 'C:\\Programming\\'
11 innSti = sti + 'Python\\BruMatrise.npy'
12
13 # Kolonne 0 = Filnavn
14 # Kolonne 1 = Bruas ID i NVDB
15 # Kolonne 2 = Målepunktets avstand til brua [m]
16 # Kolonne 3 = Målepunktets friksjonsverdi
17 # Kolonne 4 = Vegens vinterdriftsklasse ved målepunktet
18 # Kolonne 5 = Lufttemperatur [Celsius]
19 # Kolonne 6 = Vegoverflatetemperatur [Celsius]
20 # Kolonne 7 = Luftfuktighet [%] OBS, ofte utilgjengelig
21 # Kolonne 8 = Brutype
22 # Kolonne 9 = Brumateriale
23 # Kolonne 10 = Bruas totale lengde [m]
24 # Kolonne 11 = Bruas byggeår. OBS, ofte utilgjengelig
25 # Kolonne 12 = Vegens horisontalradius [m]
26 # Kolonne 13 = Utstyr brukt under målingen
27 # Kolonne 14 = Vegens referanse (navn)
28 # Kolonne 15 = Fylke
29 # Kolonne 16 = Året målingen ble utført
30 # Kolonne 17 = Måneden målingen ble utført
31 # Kolonne 18 = Dagen målingen ble utført
32 # Kolonne 19 = Klokkeslett målingen ble utført
33
34 matrise = np.load(innSti) # Laster inn matrisen
35
36 # Splitter matrisen med all data opp i mindre matriser hvor hver fil har
   sin matrise
37 unikeFilnavnIndekser = np.unique(matrise[:, 0], True)[1]
38 listeFilerMatriser = []
39 for i in range(len(unikeFilnavnIndekser)):
40     if i < len(unikeFilnavnIndekser) - 1:
41         a = unikeFilnavnIndekser[i]
42         b = unikeFilnavnIndekser[i + 1]
43         listeFilerMatriser.append(matrise[a:b, :])
44     else:
45         a = unikeFilnavnIndekser[i]
46         b = len(matrise)
47         listeFilerMatriser.append(matrise[a:b, :])
48
49 # Splitter fil-matrisene opp slik at hver ny matrise inneholder alle må
   lepunktene forbundet til ei bestemt bru
50 listeBruerMatriser = []
51 for filerMatriser in listeFilerMatriser:
```

```

52     bruIDerIndekser = np.sort(np.unique(filerMatriser[:, 1], True)[1])
53     if len(bruIDerIndekser) > 0:
54         for i in range(len(bruIDerIndekser)):
55             if i < len(bruIDerIndekser) - 1:
56                 a = bruIDerIndekser[i]
57                 b = bruIDerIndekser[i + 1]
58                 listeBruerMatriser.append(filerMatriser[a:b, :])
59             else:
60                 a = bruIDerIndekser[i]
61                 b = len(filerMatriser)
62                 listeBruerMatriser.append(filerMatriser[a:b, :])
63
64 # Sorterer bru-matrisene slik at avstanden til brua øker og noterer de fem
65   settene med flest målepunkter rundt én bru
66 listeBruerMatriserSortert = []
67 lengst1 = []
68 lengst2 = []
69 lengst3 = []
70 lengst4 = []
71 lengst5 = []
72 lengst6 = []
73 for bruerMatriser in listeBruerMatriser:
74     listeBruerMatriserSortert.append(bruerMatriser[np.argsort(bruerMatriser
75  [:, 2])])
76
77 # Noterer bruene med flest punkter:
78 midlertidigListe = []
79 for linje in bruerMatriser:
80     if linje[1] != 'None':
81         midlertidigListe.append(linje)
82     if len(midlertidigListe) > len(lengst1):
83         lengst6 = lengst5
84         lengst5 = lengst4
85         lengst4 = lengst3
86         lengst3 = lengst2
87         lengst2 = lengst1
88         lengst1 = midlertidigListe
89     elif len(midlertidigListe) > len(lengst2):
90         lengst6 = lengst5
91         lengst5 = lengst4
92         lengst4 = lengst3
93         lengst3 = lengst2
94         lengst2 = midlertidigListe
95     elif len(midlertidigListe) > len(lengst3):
96         lengst6 = lengst5
97         lengst5 = lengst4
98         lengst4 = lengst3
99         lengst3 = midlertidigListe
100    elif len(midlertidigListe) > len(lengst4):
101        lengst6 = lengst5
102        lengst5 = lengst4
103        lengst4 = midlertidigListe
104    elif len(midlertidigListe) > len(lengst5):
105        lengst6 = lengst5
106        lengst5 = midlertidigListe
107    elif len(midlertidigListe) > len(lengst6):
108        lengst6 = midlertidigListe

```

```
107
108 # Fordeler de seks utvalgte bruene med tanke på kronologisk målt
    friksjonsverdi
109 teller = -1
110 xAkseLengst1 = []
111 yAkseLengst1 = []
112 xAkseBruLengst1 = []
113 yAkseBruLengst1 = []
114 for linje in lengst1:
115     teller = teller + 1
116     xAkseLengst1.append(teller)
117     yAkseLengst1.append(float(linje[3]))
118     if linje[2] == '0.0':
119         xAkseBruLengst1.append(teller)
120         yAkseBruLengst1.append(float(linje[3]))
121 teller = -1
122 xAkseLengst2 = []
123 yAkseLengst2 = []
124 xAkseBruLengst2 = []
125 yAkseBruLengst2 = []
126 for linje in lengst2:
127     teller = teller + 1
128     xAkseLengst2.append(teller)
129     yAkseLengst2.append(float(linje[3]))
130     if linje[2] == '0.0':
131         xAkseBruLengst2.append(teller)
132         yAkseBruLengst2.append(float(linje[3]))
133 teller = -1
134 xAkseLengst3 = []
135 yAkseLengst3 = []
136 xAkseBruLengst3 = []
137 yAkseBruLengst3 = []
138 for linje in lengst3:
139     teller = teller + 1
140     xAkseLengst3.append(teller)
141     yAkseLengst3.append(float(linje[3]))
142     if linje[2] == '0.0':
143         xAkseBruLengst3.append(teller)
144         yAkseBruLengst3.append(float(linje[3]))
145 teller = -1
146 xAkseLengst4 = []
147 yAkseLengst4 = []
148 xAkseBruLengst4 = []
149 yAkseBruLengst4 = []
150 for linje in lengst4:
151     teller = teller + 1
152     xAkseLengst4.append(teller)
153     yAkseLengst4.append(float(linje[3]))
154     if linje[2] == '0.0':
155         xAkseBruLengst4.append(teller)
156         yAkseBruLengst4.append(float(linje[3]))
157 teller = -1
158 xAkseLengst5 = []
159 yAkseLengst5 = []
160 xAkseBruLengst5 = []
161 yAkseBruLengst5 = []
162 for linje in lengst5:
```

```

163     teller = teller + 1
164     xAkseLengst5.append(teller)
165     yAkseLengst5.append(float(linje[3]))
166     if linje[2] == '0.0':
167         xAkseBruLengst5.append(teller)
168         yAkseBruLengst5.append(float(linje[3]))
169 teller = -1
170 xAkseLengst6 = []
171 yAkseLengst6 = []
172 xAkseBruLengst6 = []
173 yAkseBruLengst6 = []
174 for linje in lengst6:
175     teller = teller + 1
176     xAkseLengst6.append(teller)
177     yAkseLengst6.append(float(linje[3]))
178     if linje[2] == '0.0':
179         xAkseBruLengst6.append(teller)
180         yAkseBruLengst6.append(float(linje[3]))
181
182
183 # Definerer en funksjon som kan si om innholdet er NULL eller ikke
184 def isfloat(value):
185     try:
186         float(value)
187         return True
188     except ValueError:
189         return False
190
191
192 klokkeslett8_20 = []
193 klokkeslett20_8 = []
194 friksjonLavestPaaBruTeller0 = 0
195 friksjonLavestPaaBruTeller1 = 0
196 friksjonLavestPaaBruTeller2 = 0
197 friksjonLavestPaaBruTeller3 = 0
198 minBruFriksjonListe0 = []
199 minBruFriksjonListe1 = []
200 minBruFriksjonListe2 = []
201 minBruFriksjonListe3 = []
202 sortertMatriseTeller = 0
203 bruSett = []
204 fylkesnummerListe = []
205 friksjonsverdierPaaBruListe = []
206 friksjonsverdierPaaBruListeSortert = []
207 luftTemperaturPaaBruListe = []
208 vegTemperaturPaaBruListe = []
209 avstanderTilBruListe = []
210 relativFriksjonsverdierMedianListe = []
211 relativFriksjonsverdierMinListe = []
212 luftfuktighetLavTemperaturPaaBruListe = []
213 luftfuktighetLavTemperaturPaaBruFriksjonsverdierListe = []
214 luftfuktighetLavTemperaturUtenforBruListe = []
215 luftfuktighetLavTemperaturUtenforBruFriksjonsverdierListe = []
216 luftfuktighetHoyTemperaturPaaBruListe = []
217 luftfuktighetHoyTemperaturPaaBruFriksjonsverdierListe = []
218 luftfuktighetHoyTemperaturUtenforBruListe = []
219 luftfuktighetHoyTemperaturUtenforBruFriksjonsverdierListe = []

```

```

220 antallBruer = 0
221 antallBrupunkter = 0
222 avstanderTilBruListeDriftsklasse = []
223 relativFriksjonsverdierMedianListeDriftsklasse = []
224 relativFriksjonsverdierMinListeDriftsklasse = []
225 friksjonsverdierListeDriftsklasseAlle = []
226 luftTemperaturListe = []
227 vegTemperaturListe = []
228 relativVegTemperaturListe = []
229 driftsklasseListe = []
230 bruMaterialeListe = []
231 # Finner friksjonsverdien som er nærmest brua. Dersom det er målt flere
    verdier på brua beregnes medianverdien
232 for bruerMatriserSortert in listeBruerMatriserSortert:
233     sortertMatriseTeller = sortertMatriseTeller + 1
234     usortertMatriseTeller = 0
235     for linje in bruerMatriserSortert:
236
237         # Undersøker hvordan friksjon på bruer påvirkes av lufttemperatur
    og vegoverflatetemperatur
238         if linje[2] == '0.0':
239             friksjonsverdierPaaBruListe.append(float(linje[3]))
240             luftTemperaturPaaBruListe.append(float(linje[5]))
241             vegTemperaturPaaBruListe.append(float(linje[6]))
242             friksjonsverdierPaaBruListeSortert.append(linje[3])
243
244         # Undersøker hvordan friksjon på bruer og utenfor bruer påvirkes av
    luftfuktighet
245         if isfloat(linje[7]) is True:
246             if float(linje[6]) <= 0 and float(linje[7]) > 0: # Målinger
    med vegoverflatetemperatur under frysepunktet
247                 if linje[2] == '0.0': # Målinger på bru
248                     luftfuktighetLavTemperaturPaaBruListe.append(float(
    linje[7]))
249                     luftfuktighetLavTemperaturPaaBruFriksjonsverdierListe.
    append(float(linje[3]))
250                 else: # Målinger utenfor bru
251                     luftfuktighetLavTemperaturUtenforBruListe.append(float(
    linje[7]))
252
    luftfuktighetLavTemperaturUtenforBruFriksjonsverdierListe.append(float(
    linje[3]))
253                 elif float(linje[6]) > 0 and float(
254                     linje[7]) > 0: # Målinger med vegoverflatetemperatur
    over frysepunktet, opp til 1Celsius
255                     if linje[2] == '0.0': # Målinger på bru
256                         luftfuktighetHoyTemperaturPaaBruListe.append(float(
    linje[7]))
257                         luftfuktighetHoyTemperaturPaaBruFriksjonsverdierListe.
    append(float(linje[3]))
258                     else: # Målinger utenfor bru
259                         luftfuktighetHoyTemperaturUtenforBruListe.append(float(
    linje[7]))
260
    luftfuktighetHoyTemperaturUtenforBruFriksjonsverdierListe.append(float(
    linje[3]))
261

```

```

262     friksjonsverdierPaaBruListeSortert.sort() # Sorterer listen med
generelle friksjonsverdier på bruer i stigende rekkefølge
263
264     nullAvstandIndekser = np.where(bruerMatriserSortert[:, 2] == '0.0')[
265         0] # Finner plasseringen i matrisen hvor avstanden til brua er
null
266     if nullAvstandIndekser.size > 0:
267         for linje in bruerMatriserSortert: # Noterer klokkeslett for må
lingene
268             klokkeslett = linje[19]
269             klokkeslettTime = int(klokkeslett[:2])
270             if 8 <= klokkeslettTime < 20:
271                 klokkeslett8_20.append(klokkeslettTime)
272             elif klokkeslettTime < 8 or klokkeslettTime >= 20:
273                 klokkeslett20_8.append(klokkeslettTime)
274
275         antallBruer = antallBruer + 1
276         antallBrupunkter = antallBrupunkter + len(bruerMatriserSortert)
277         nullAvstandfrikjsjonsverdier = np.asarray(bruerMatriserSortert[0:
nullAvstandIndekser.size, 3], dtype=np.float64,
278                                                     order='C')
279         medianBruFriksjon = np.percentile(nullAvstandfrikjsjonsverdier,
280                                           50) # Finner median
frikjsjonsverdi til målepunktene på selve brua
281         minBruFriksjon = np.amin(nullAvstandfrikjsjonsverdier)
282         maxBruFriksjon = np.amax(nullAvstandfrikjsjonsverdier)
283         driftsklasseBru = bruerMatriserSortert[0][4]
284
285         nullAvstandVegtemperaturer = np.asarray(bruerMatriserSortert[0:
nullAvstandIndekser.size, 6], dtype=np.float64,
286                                                 order='C')
287         medianBruVegtemperatur = np.percentile(nullAvstandVegtemperaturer,
288                                                 50) # Finner median
vegtemperatur til målepunktene på selve brua
289
290         fylkesnummer = np.asarray(bruerMatriserSortert[:, 15], dtype=np.
float64, order='C')
291         for fylke in fylkesnummer:
292             fylkesnummerListe.append(fylke)
293
294         # Noterer avstand til bru friksjon og temperatur for alle punkter
295         avstanderTilBru = np.asarray(bruerMatriserSortert[:, 2], dtype=np.
float64, order='C')
296         for avstand in avstanderTilBru:
297             avstanderTilBruListe.append(avstand) # Legger til
avstandsverdiene i en liste (brukes som x-koordinater)
298
299         friksjonsverdier = np.asarray(bruerMatriserSortert[:, 3], dtype=np.
float64, order='C')
300         for friksjon in friksjonsverdier: # Legger til friksjonsverdiene i
en liste (brukes som y-koordinater)
301             relativFriksjonsverdierMedianListe.append(friksjon -
medianBruFriksjon)
302             relativFriksjonsverdierMinListe.append(friksjon -
minBruFriksjon)
303
304         friksjonsverdierRundtBru = np.asarray(bruerMatriserSortert[

```

```

nullAvstandIndekser.size:, 3], dtype=np.float64,
305                                     order='C')
306     if friksjonsverdierRundtBru.size > 0:
307         minFriksjonRundtBru = np.amin(friksjonsverdierRundtBru)
308         if minFriksjonRundtBru > minBruFriksjon:
309             if minFriksjonRundtBru < minBruFriksjon + 0.05:
310                 friksjonLavestPaaBruTeller0 =
friksjonLavestPaaBruTeller0 + 1
311                 minBruFriksjonListe0.append(minBruFriksjon)
312                 if minBruFriksjon + 0.05 <= minFriksjonRundtBru <
minBruFriksjon + 0.1:
313                     friksjonLavestPaaBruTeller1 =
friksjonLavestPaaBruTeller1 + 1
314                     minBruFriksjonListe1.append(minBruFriksjon)
315                     if minBruFriksjon + 0.10 <= minFriksjonRundtBru <
minBruFriksjon + 0.15:
316                         friksjonLavestPaaBruTeller2 =
friksjonLavestPaaBruTeller2 + 1
317                         minBruFriksjonListe2.append(minBruFriksjon)
318                         if minFriksjonRundtBru >= minBruFriksjon + 0.15:
319                             friksjonLavestPaaBruTeller3 =
friksjonLavestPaaBruTeller3 + 1
320                             minBruFriksjonListe3.append(minBruFriksjon)
321                             for bruerMatriser in listeBruerMatriser:
322                                 usortertMatriseTeller = usortertMatriseTeller + 1
323                                 if usortertMatriseTeller == sortertMatriseTeller:
324                                     bruSett.append(bruerMatriser)
325
326         luftTemperaturer = np.asarray(bruerMatriserSortert[:, 5], dtype=np.
float64, order='C')
327         for luftTemperatur in luftTemperaturer:
328             luftTemperaturListe.append(luftTemperatur) # Legger til
vegtemperatur i en liste (brukes som x-koordinater)
329
330         vegTemperaturer = np.asarray(bruerMatriserSortert[:, 6], dtype=np.
float64, order='C')
331         for vegTemperatur in vegTemperaturer:
332             relativVegTemperaturListe.append((
333                 vegTemperatur - medianBruVegtemperatur)) # Legger til
vegtemperatur relativ vegtemperaturen målt på brua i ei liste
334             vegTemperaturListe.append(vegTemperatur) # Legger til
vegtemperatur i en liste
335
336         # Samler friksjon, avstand og brumateriale sortert etter
driftsklasse
337         driftsklasser = np.asarray(bruerMatriserSortert[:, 4], order='C')
338         teller = -1
339         for driftsklasse in driftsklasser:
340             teller = teller + 1
341             if driftsklasse == driftsklasseBru and driftsklasse != 'None':
342                 avstanderTilBruListeDriftsklasse.append(float(
bruerMatriserSortert[teller][2]))
343                 relativFriksjonsverdierMedianListeDriftsklasse.append(
344                     float(bruerMatriserSortert[teller][3]) -
medianBruFriksjon)
345                 relativFriksjonsverdierMinListeDriftsklasse.append(
346                     float(bruerMatriserSortert[teller][3]) - minBruFriksjon

```

```

)
347         friksjonsverdierListeDriftsklasseAlle.append(float (
        bruerMatriserSortert[teller][3]))
348         driftsklasseListe.append(bruerMatriserSortert[teller][4])
349         bruMaterialeListe.append(bruerMatriserSortert[teller][9])
350 minBruFriksjonListe0.sort()
351 minBruFriksjonListe1.sort()
352 minBruFriksjonListe2.sort()
353 minBruFriksjonListe3.sort()
354 bruSettFriksjon = []
355 bruSettFriksjonBru = []
356 bruSettVegoverflate = []
357 brusettTall = []
358 brusettTallBru = []
359 for bruMatrise in bruSett:
360     midlertidigListeFriksjon = []
361     midlertidigListeFriksjonBru = []
362     midlertidigListeTall = []
363     midlertidigListeTallBru = []
364     midlertidigListeVegoverflate = []
365     teller = -1
366     for linje in bruMatrise:
367         teller = teller + 1
368         midlertidigListeFriksjon.append(float(linje[3]))
369         midlertidigListeVegoverflate.append(float(linje[6]))
370         midlertidigListeTall.append(teller)
371         if linje[2] == '0.0':
372             midlertidigListeFriksjonBru.append(float(linje[3]))
373             midlertidigListeTallBru.append(teller)
374         bruSettFriksjon.append(midlertidigListeFriksjon)
375         bruSettFriksjonBru.append(midlertidigListeFriksjonBru)
376         bruSettVegoverflate.append(midlertidigListeVegoverflate)
377         brusettTall.append(midlertidigListeTall)
378         brusettTallBru.append(midlertidigListeTallBru)
379
380 # Fordeler avstand og friksjon basert på driftsklasse
381 avstandListeDkA = []
382 avstandListeDkB1 = []
383 avstandListeDkB2 = []
384 avstandListeDkB3 = []
385 avstandListeDkC = []
386 avstandListeDkD = []
387 avstandListeDkE = []
388 relativFriksjonsverdierMedianListeDkA = []
389 friksjonsverdierListeDkA = []
390 relativFriksjonsverdierMedianListeDkB1 = []
391 friksjonsverdierListeDkB1 = []
392 relativFriksjonsverdierMedianListeDkB2 = []
393 friksjonsverdierListeDkB2 = []
394 relativFriksjonsverdierMedianListeDkB3 = []
395 friksjonsverdierListeDkB3 = []
396 relativFriksjonsverdierMedianListeDkC = []
397 friksjonsverdierListeDkC = []
398 relativFriksjonsverdierMedianListeDkD = []
399 friksjonsverdierListeDkD = []
400 relativFriksjonsverdierMedianListeDkE = []
401 friksjonsverdierListeDkE = []

```

```

402 bruMaterialeListeDkA = []
403 bruMaterialeListeDkB1 = []
404 bruMaterialeListeDkB2 = []
405 bruMaterialeListeDkB3 = []
406 bruMaterialeListeDkC = []
407 bruMaterialeListeDkD = []
408 bruMaterialeListeDkE = []
409 driftsklasseTeller = -1
410 for driftsklasse in driftsklasseListe:
411     driftsklasseTeller = driftsklasseTeller + 1
412     if driftsklasse == 'DkA':
413         relativFriksjonsverdierMedianListeDkA.append(
414             relativFriksjonsverdierMedianListeDriftsklasse[driftsklasseTeller])
415         avstandListeDkA.append(avstanderTilBruListeDriftsklasse[
416             driftsklasseTeller])
417         friksjonsverdierListeDkA.append(
418             friksjonsverdierListeDriftsklasseAlle[driftsklasseTeller])
419         bruMaterialeListeDkA.append(bruMaterialeListe[driftsklasseTeller])
420     elif driftsklasse == 'DkB, høy':
421         relativFriksjonsverdierMedianListeDkB1.append(
422             relativFriksjonsverdierMedianListeDriftsklasse[
423                 driftsklasseTeller])
424         avstandListeDkB1.append(avstanderTilBruListeDriftsklasse[
425             driftsklasseTeller])
426         friksjonsverdierListeDkB1.append(
427             friksjonsverdierListeDriftsklasseAlle[driftsklasseTeller])
428         bruMaterialeListeDkB1.append(bruMaterialeListe[driftsklasseTeller])
429     elif driftsklasse == 'DkB, middels':
430         relativFriksjonsverdierMedianListeDkB2.append(
431             relativFriksjonsverdierMedianListeDriftsklasse[
432                 driftsklasseTeller])
433         avstandListeDkB2.append(avstanderTilBruListeDriftsklasse[
434             driftsklasseTeller])
435         friksjonsverdierListeDkB2.append(
436             friksjonsverdierListeDriftsklasseAlle[driftsklasseTeller])
437         bruMaterialeListeDkB2.append(bruMaterialeListe[driftsklasseTeller])
438     elif driftsklasse == 'DkB, lav':
439         relativFriksjonsverdierMedianListeDkB3.append(
440             relativFriksjonsverdierMedianListeDriftsklasse[
441                 driftsklasseTeller])
442         avstandListeDkB3.append(avstanderTilBruListeDriftsklasse[
443             driftsklasseTeller])
444         friksjonsverdierListeDkB3.append(
445             friksjonsverdierListeDriftsklasseAlle[driftsklasseTeller])
446         bruMaterialeListeDkB3.append(bruMaterialeListe[driftsklasseTeller])
447     elif driftsklasse == 'DkC':
448         relativFriksjonsverdierMedianListeDkC.append(
449             relativFriksjonsverdierMedianListeDriftsklasse[driftsklasseTeller])
450         avstandListeDkC.append(avstanderTilBruListeDriftsklasse[
451             driftsklasseTeller])
452         friksjonsverdierListeDkC.append(
453             friksjonsverdierListeDriftsklasseAlle[driftsklasseTeller])
454         bruMaterialeListeDkC.append(bruMaterialeListe[driftsklasseTeller])
455     elif driftsklasse == 'DkD':
456         relativFriksjonsverdierMedianListeDkD.append(
457             relativFriksjonsverdierMedianListeDriftsklasse[driftsklasseTeller])
458         avstandListeDkD.append(avstanderTilBruListeDriftsklasse[

```

```

driftsklasseTeller])
443     friksjonsverdierListeDkD.append(
friksjonsverdierListeDriftsklasseAlle[driftsklasseTeller])
444     bruMaterialeListeDkD.append(bruMaterialeListe[driftsklasseTeller])
445     elif driftsklasse == 'DkE':
446         relativFriksjonsverdierMedianListeDkE.append(
relativFriksjonsverdierMedianListeDriftsklasse[driftsklasseTeller])
447         avstandListeDkE.append(avstanderTilBruListeDriftsklasse[
driftsklasseTeller])
448         friksjonsverdierListeDkE.append(
friksjonsverdierListeDriftsklasseAlle[driftsklasseTeller])
449         bruMaterialeListeDkE.append(bruMaterialeListe[driftsklasseTeller])
450 friksjonsverdierListeDriftsklasseAlleBru = []
451 friksjonsverdierListeDriftsklasseAlleUtenfor = []
452 bruMaterialeListeDriftsklasseAlleBru = []
453 friksjonsverdierListeDkABru = []
454 friksjonsverdierListeDkAUtenfor = []
455 friksjonsverdierListeDkABruUsortert = []
456 bruMaterialeListeDkABru = []
457 for i in range(len(avstandListeDkA)):
458     if avstandListeDkA[i] == 0:
459         friksjonsverdierListeDkABru.append(friksjonsverdierListeDkA[i])
460         friksjonsverdierListeDkABruUsortert.append(friksjonsverdierListeDkA
[i])
461         friksjonsverdierListeDriftsklasseAlleBru.append(
friksjonsverdierListeDkA[i])
462         bruMaterialeListeDkABru.append(bruMaterialeListeDkA[i])
463         bruMaterialeListeDriftsklasseAlleBru.append(bruMaterialeListeDkA[i]
])
464     else:
465         friksjonsverdierListeDkAUtenfor.append(friksjonsverdierListeDkA[i])
466         friksjonsverdierListeDriftsklasseAlleUtenfor.append(
friksjonsverdierListeDkA[i])
467 friksjonsverdierListeDkB1Bru = []
468 friksjonsverdierListeDkB1Utenfor = []
469 friksjonsverdierListeDkB1BruUsortert = []
470 bruMaterialeListeDkB1Bru = []
471 for i in range(len(avstandListeDkB1)):
472     if avstandListeDkB1[i] == 0:
473         friksjonsverdierListeDkB1Bru.append(friksjonsverdierListeDkB1[i])
474         friksjonsverdierListeDkB1BruUsortert.append(
friksjonsverdierListeDkB1[i])
475         friksjonsverdierListeDriftsklasseAlleBru.append(
friksjonsverdierListeDkB1[i])
476         bruMaterialeListeDkB1Bru.append(bruMaterialeListeDkB1[i])
477         bruMaterialeListeDriftsklasseAlleBru.append(bruMaterialeListeDkB1[i]
])
478     else:
479         friksjonsverdierListeDkB1Utenfor.append(friksjonsverdierListeDkB1[i]
])
480         friksjonsverdierListeDriftsklasseAlleUtenfor.append(
friksjonsverdierListeDkB1[i])
481 friksjonsverdierListeDkB2Bru = []
482 friksjonsverdierListeDkB2Utenfor = []
483 friksjonsverdierListeDkB2BruUsortert = []
484 bruMaterialeListeDkB2Bru = []
485 for i in range(len(avstandListeDkB2)):

```

```

486     if avstandListeDkB2[i] == 0:
487         friksjonsverdierListeDkB2Bru.append(friksjonsverdierListeDkB2[i])
488         friksjonsverdierListeDkB2BruUsortert.append(
friksjonsverdierListeDkB2[i])
489         friksjonsverdierListeDriftsklasseAlleBru.append(
friksjonsverdierListeDkB2[i])
490         bruMaterialeListeDkB2Bru.append(bruMaterialeListeDkB2[i])
491         bruMaterialeListeDriftsklasseAlleBru.append(bruMaterialeListeDkB2[i]
])
492     else:
493         friksjonsverdierListeDkB2Utenfor.append(friksjonsverdierListeDkB2[i]
])
494         friksjonsverdierListeDriftsklasseAlleUtenfor.append(
friksjonsverdierListeDkB2[i])
495 friksjonsverdierListeDkB3Bru = []
496 friksjonsverdierListeDkB3Utenfor = []
497 friksjonsverdierListeDkB3BruUsortert = []
498 bruMaterialeListeDkB3Bru = []
499 for i in range(len(avstandListeDkB3)):
500     if avstandListeDkB3[i] == 0:
501         friksjonsverdierListeDkB3Bru.append(friksjonsverdierListeDkB3[i])
502         friksjonsverdierListeDkB3BruUsortert.append(
friksjonsverdierListeDkB3[i])
503         friksjonsverdierListeDriftsklasseAlleBru.append(
friksjonsverdierListeDkB3[i])
504         bruMaterialeListeDkB3Bru.append(bruMaterialeListeDkB3[i])
505         bruMaterialeListeDriftsklasseAlleBru.append(bruMaterialeListeDkB3[i]
])
506     else:
507         friksjonsverdierListeDkB3Utenfor.append(friksjonsverdierListeDkB3[i]
])
508         friksjonsverdierListeDriftsklasseAlleUtenfor.append(
friksjonsverdierListeDkB3[i])
509 friksjonsverdierListeDkCBru = []
510 friksjonsverdierListeDkCUtenfor = []
511 friksjonsverdierListeDkCBruUsortert = []
512 bruMaterialeListeDkCBru = []
513 for i in range(len(avstandListeDkC)):
514     if avstandListeDkC[i] == 0:
515         friksjonsverdierListeDkCBru.append(friksjonsverdierListeDkC[i])
516         friksjonsverdierListeDkCBruUsortert.append(friksjonsverdierListeDkC
[i])
517         friksjonsverdierListeDriftsklasseAlleBru.append(
friksjonsverdierListeDkC[i])
518         bruMaterialeListeDkCBru.append(bruMaterialeListeDkC[i])
519         bruMaterialeListeDriftsklasseAlleBru.append(bruMaterialeListeDkC[i]
])
520     else:
521         friksjonsverdierListeDkCUtenfor.append(friksjonsverdierListeDkC[i])
522         friksjonsverdierListeDriftsklasseAlleUtenfor.append(
friksjonsverdierListeDkC[i])
523 friksjonsverdierListeDkDBru = []
524 friksjonsverdierListeDkDUtenfor = []
525 friksjonsverdierListeDkDBruUsortert = []
526 bruMaterialeListeDkDBru = []
527 for i in range(len(avstandListeDkD)):
528     if avstandListeDkD[i] == 0:

```

```

529     friksjonsverdierListeDkDBru.append(friksjonsverdierListeDkD[i])
530     friksjonsverdierListeDkDBruUsortert.append(friksjonsverdierListeDkD
[i])
531     friksjonsverdierListeDriftsklasseAlleBru.append(
friksjonsverdierListeDkD[i])
532     bruMaterialeListeDkDBru.append(bruMaterialeListeDkD[i])
533     bruMaterialeListeDriftsklasseAlleBru.append(bruMaterialeListeDkD[i]
])
534     else:
535         friksjonsverdierListeDkDUtenfor.append(friksjonsverdierListeDkD[i])
536         friksjonsverdierListeDriftsklasseAlleUtenfor.append(
friksjonsverdierListeDkD[i])
537 friksjonsverdierListeDkEBru = []
538 friksjonsverdierListeDkEUtenfor = []
539 friksjonsverdierListeDkEBruUsortert = []
540 bruMaterialeListeDkEBru = []
541 for i in range(len(avstandListeDkE)):
542     if avstandListeDkE[i] == 0:
543         friksjonsverdierListeDkEBru.append(friksjonsverdierListeDkE[i])
544         friksjonsverdierListeDkEBruUsortert.append(friksjonsverdierListeDkE
[i])
545         friksjonsverdierListeDriftsklasseAlleBru.append(
friksjonsverdierListeDkE[i])
546         bruMaterialeListeDkEBru.append(bruMaterialeListeDkE[i])
547         bruMaterialeListeDriftsklasseAlleBru.append(bruMaterialeListeDkE[i]
])
548     else:
549         friksjonsverdierListeDkEUtenfor.append(friksjonsverdierListeDkE[i])
550         friksjonsverdierListeDriftsklasseAlleUtenfor.append(
friksjonsverdierListeDkE[i])
551 friksjonsverdierListeDkABru.sort()
552 friksjonsverdierListeDkAUtenfor.sort()
553 friksjonsverdierListeDkA.sort()
554 friksjonsverdierListeDkB1Bru.sort()
555 friksjonsverdierListeDkB1Utenfor.sort()
556 friksjonsverdierListeDkB1.sort()
557 friksjonsverdierListeDkB2Bru.sort()
558 friksjonsverdierListeDkB2Utenfor.sort()
559 friksjonsverdierListeDkB2.sort()
560 friksjonsverdierListeDkB3Bru.sort()
561 friksjonsverdierListeDkB3Utenfor.sort()
562 friksjonsverdierListeDkB3.sort()
563 friksjonsverdierListeDkCBru.sort()
564 friksjonsverdierListeDkCUtenfor.sort()
565 friksjonsverdierListeDkC.sort()
566 friksjonsverdierListeDkDBru.sort()
567 friksjonsverdierListeDkDUtenfor.sort()
568 friksjonsverdierListeDkD.sort()
569 friksjonsverdierListeDkEBru.sort()
570 friksjonsverdierListeDkEUtenfor.sort()
571 friksjonsverdierListeDkE.sort()
572 friksjonsverdierListeDriftsklasseAlleBru.sort()
573 friksjonsverdierListeDriftsklasseAlleUtenfor.sort()
574 friksjonsverdierListeDriftsklasseAlle.sort()
575
576 # Legger til brumaterialet for alle friksjonsverdier
577 bruMaterialeListeDkA5persentil = []

```

```

578 for i in range(len(friksjonsverdierListeDkABruUsortert)):
579     if friksjonsverdierListeDkABruUsortert[i] <=
        friksjonsverdierListeDkABru[round(len(friksjonsverdierListeDkABru) *
        0.05)]:
580         bruMaterialeListeDkA5persentil.append(bruMaterialeListeDkABru[i])
581 bruMaterialeListeDkB15persentil = []
582 for i in range(len(friksjonsverdierListeDkB1BruUsortert)):
583     if friksjonsverdierListeDkB1BruUsortert[i] <=
        friksjonsverdierListeDkB1Bru[round(len(friksjonsverdierListeDkB1Bru) *
        0.05)]:
584         bruMaterialeListeDkB15persentil.append(bruMaterialeListeDkB1Bru[i])
585 bruMaterialeListeDkB25persentil = []
586 for i in range(len(friksjonsverdierListeDkB2BruUsortert)):
587     if friksjonsverdierListeDkB2BruUsortert[i] <=
        friksjonsverdierListeDkB2Bru[round(len(friksjonsverdierListeDkB2Bru) *
        0.05)]:
588         bruMaterialeListeDkB25persentil.append(bruMaterialeListeDkB2Bru[i])
589 bruMaterialeListeDkB35persentil = []
590 for i in range(len(friksjonsverdierListeDkB3BruUsortert)):
591     if friksjonsverdierListeDkB3BruUsortert[i] <=
        friksjonsverdierListeDkB3Bru[round(len(friksjonsverdierListeDkB3Bru) *
        0.05)]:
592         bruMaterialeListeDkB35persentil.append(bruMaterialeListeDkB3Bru[i])
593 bruMaterialeListeDkC5persentil = []
594 for i in range(len(friksjonsverdierListeDkCBruUsortert)):
595     if friksjonsverdierListeDkCBruUsortert[i] <=
        friksjonsverdierListeDkCBru[round(len(friksjonsverdierListeDkCBru) *
        0.05)]:
596         bruMaterialeListeDkC5persentil.append(bruMaterialeListeDkCBru[i])
597 bruMaterialeListeDkD5persentil = []
598 for i in range(len(friksjonsverdierListeDkDBruUsortert)):
599     if friksjonsverdierListeDkDBruUsortert[i] <=
        friksjonsverdierListeDkDBru[round(len(friksjonsverdierListeDkDBru) *
        0.05)]:
600         bruMaterialeListeDkD5persentil.append(bruMaterialeListeDkDBru[i])
601 bruMaterialeListeDkE5persentil = []
602 for i in range(len(friksjonsverdierListeDkEBruUsortert)):
603     if friksjonsverdierListeDkEBruUsortert[i] <=
        friksjonsverdierListeDkEBru[round(len(friksjonsverdierListeDkEBru) *
        0.05)]:
604         bruMaterialeListeDkE5persentil.append(bruMaterialeListeDkEBru[i])
605
606 friksjonsverdierMatriseDkAlleBru = np.asarray(
        friksjonsverdierListeDriftsklasseAlleBru)
607 friksjonsverdierMatriseDkAlleUtenfor = np.asarray(
        friksjonsverdierListeDriftsklasseAlleUtenfor)
608 friksjonsverdierMatriseDkAlle = np.asarray(
        friksjonsverdierListeDriftsklasseAlle)
609 friksjonsverdierMatriseDkABru = np.asarray(friksjonsverdierListeDkABru)
610 friksjonsverdierMatriseDkAUtenfor = np.asarray(
        friksjonsverdierListeDkAUtenfor)
611 friksjonsverdierMatriseDkA = np.asarray(friksjonsverdierListeDkA)
612 friksjonsverdierMatriseDkB1Bru = np.asarray(friksjonsverdierListeDkB1Bru)
613 friksjonsverdierMatriseDkB1Utenfor = np.asarray(
        friksjonsverdierListeDkB1Utenfor)
614 friksjonsverdierMatriseDkB1 = np.asarray(friksjonsverdierListeDkB1)
615 friksjonsverdierMatriseDkB2Bru = np.asarray(friksjonsverdierListeDkB2Bru)

```

```

616 friksjonsverdierMatriseDkB2Utenfor = np.asarray(
        friksjonsverdierListeDkB2Utenfor)
617 friksjonsverdierMatriseDkB2 = np.asarray(friksjonsverdierListeDkB2)
618 friksjonsverdierMatriseDkB3Bru = np.asarray(friksjonsverdierListeDkB3Bru)
619 friksjonsverdierMatriseDkB3Utenfor = np.asarray(
        friksjonsverdierListeDkB3Utenfor)
620 friksjonsverdierMatriseDkB3 = np.asarray(friksjonsverdierListeDkB3)
621 friksjonsverdierMatriseDkCBru = np.asarray(friksjonsverdierListeDkCBru)
622 friksjonsverdierMatriseDkCUtenfor = np.asarray(
        friksjonsverdierListeDkCUtenfor)
623 friksjonsverdierMatriseDkC = np.asarray(friksjonsverdierListeDkC)
624 friksjonsverdierMatriseDkDBru = np.asarray(friksjonsverdierListeDkDBru)
625 friksjonsverdierMatriseDkDUtenfor = np.asarray(
        friksjonsverdierListeDkDUtenfor)
626 friksjonsverdierMatriseDkD = np.asarray(friksjonsverdierListeDkD)
627 friksjonsverdierMatriseDkEBru = np.asarray(friksjonsverdierListeDkEBru)
628 friksjonsverdierMatriseDkEUtenfor = np.asarray(
        friksjonsverdierListeDkEUtenfor)
629 friksjonsverdierMatriseDkE = np.asarray(friksjonsverdierListeDkE)
630
631 friksjonsverdierMatriseDkAlleBru_mean = np.mean(
        friksjonsverdierMatriseDkAlleBru)
632 friksjonsverdierMatriseDkAlleUtenfor_mean = np.mean(
        friksjonsverdierMatriseDkAlleUtenfor)
633 friksjonsverdierMatriseDkAlle_mean = np.mean(friksjonsverdierMatriseDkAlle)
634 friksjonsverdierMatriseDkABru_mean = np.mean(friksjonsverdierMatriseDkABru)
635 friksjonsverdierMatriseDkAUtenfor_mean = np.mean(
        friksjonsverdierMatriseDkAUtenfor)
636 friksjonsverdierMatriseDkA_mean = np.mean(friksjonsverdierMatriseDkA)
637 friksjonsverdierMatriseDkB1Bru_mean = np.mean(
        friksjonsverdierMatriseDkB1Bru)
638 friksjonsverdierMatriseDkB1Utenfor_mean = np.mean(
        friksjonsverdierMatriseDkB1Utenfor)
639 friksjonsverdierMatriseDkB1_mean = np.mean(friksjonsverdierMatriseDkB1)
640 friksjonsverdierMatriseDkB2Bru_mean = np.mean(
        friksjonsverdierMatriseDkB2Bru)
641 friksjonsverdierMatriseDkB2Utenfor_mean = np.mean(
        friksjonsverdierMatriseDkB2Utenfor)
642 friksjonsverdierMatriseDkB2_mean = np.mean(friksjonsverdierMatriseDkB2)
643 friksjonsverdierMatriseDkB3Bru_mean = np.mean(
        friksjonsverdierMatriseDkB3Bru)
644 friksjonsverdierMatriseDkB3Utenfor_mean = np.mean(
        friksjonsverdierMatriseDkB3Utenfor)
645 friksjonsverdierMatriseDkB3_mean = np.mean(friksjonsverdierMatriseDkB3)
646 friksjonsverdierMatriseDkCBru_mean = np.mean(friksjonsverdierMatriseDkCBru)
647 friksjonsverdierMatriseDkCUtenfor_mean = np.mean(
        friksjonsverdierMatriseDkCUtenfor)
648 friksjonsverdierMatriseDkC_mean = np.mean(friksjonsverdierMatriseDkC)
649 friksjonsverdierMatriseDkDBru_mean = np.mean(friksjonsverdierMatriseDkDBru)
650 friksjonsverdierMatriseDkDUtenfor_mean = np.mean(
        friksjonsverdierMatriseDkDUtenfor)
651 friksjonsverdierMatriseDkD_mean = np.mean(friksjonsverdierMatriseDkD)
652 friksjonsverdierMatriseDkEBru_mean = np.mean(friksjonsverdierMatriseDkEBru)
653 friksjonsverdierMatriseDkEUtenfor_mean = np.mean(
        friksjonsverdierMatriseDkEUtenfor)
654 friksjonsverdierMatriseDkE_mean = np.mean(friksjonsverdierMatriseDkE)
655

```

```

656 friksjonsverdierMatriseDkAlleBru_std = np.std(
        friksjonsverdierMatriseDkAlleBru)
657 friksjonsverdierMatriseDkAlleUtenfor_std = np.std(
        friksjonsverdierMatriseDkAlleUtenfor)
658 friksjonsverdierMatriseDkABru_std = np.std(friksjonsverdierMatriseDkABru)
659 friksjonsverdierMatriseDkAUtenfor_std = np.std(
        friksjonsverdierMatriseDkAUtenfor)
660 friksjonsverdierMatriseDkB1Bru_std = np.std(friksjonsverdierMatriseDkB1Bru)
661 friksjonsverdierMatriseDkB1Utenfor_std = np.std(
        friksjonsverdierMatriseDkB1Utenfor)
662 friksjonsverdierMatriseDkB2Bru_std = np.std(friksjonsverdierMatriseDkB2Bru)
663 friksjonsverdierMatriseDkB2Utenfor_std = np.std(
        friksjonsverdierMatriseDkB2Utenfor)
664 friksjonsverdierMatriseDkB3Bru_std = np.std(friksjonsverdierMatriseDkB3Bru)
665 friksjonsverdierMatriseDkB3Utenfor_std = np.std(
        friksjonsverdierMatriseDkB3Utenfor)
666 friksjonsverdierMatriseDkCBru_std = np.std(friksjonsverdierMatriseDkCBru)
667 friksjonsverdierMatriseDkCUtenfor_std = np.std(
        friksjonsverdierMatriseDkCUtenfor)
668 friksjonsverdierMatriseDkDBru_std = np.std(friksjonsverdierMatriseDkDBru)
669 friksjonsverdierMatriseDkDUtenfor_std = np.std(
        friksjonsverdierMatriseDkDUtenfor)
670 friksjonsverdierMatriseDkEBru_std = np.std(friksjonsverdierMatriseDkEBru)
671 friksjonsverdierMatriseDkEUtenfor_std = np.std(
        friksjonsverdierMatriseDkEUtenfor)
672
673 gjennomsnittBru = [friksjonsverdierMatriseDkAlleBru_mean,
        friksjonsverdierMatriseDkABru_mean,
674                     friksjonsverdierMatriseDkB1Bru_mean,
        friksjonsverdierMatriseDkB2Bru_mean,
675                     friksjonsverdierMatriseDkB3Bru_mean,
        friksjonsverdierMatriseDkCBru_mean,
676                     friksjonsverdierMatriseDkDBru_mean,
        friksjonsverdierMatriseDkEBru_mean]
677 gjennomsnittUtenfor = [friksjonsverdierMatriseDkAlleUtenfor_mean,
        friksjonsverdierMatriseDkAUtenfor_mean,
678                          friksjonsverdierMatriseDkB1Utenfor_mean,
        friksjonsverdierMatriseDkB2Utenfor_mean,
679                          friksjonsverdierMatriseDkB3Utenfor_mean,
        friksjonsverdierMatriseDkCUtenfor_mean,
680                          friksjonsverdierMatriseDkDUtenfor_mean,
        friksjonsverdierMatriseDkEUtenfor_mean]
681 standardavvikBru = [friksjonsverdierMatriseDkAlleBru_std,
        friksjonsverdierMatriseDkABru_std,
682                      friksjonsverdierMatriseDkB1Bru_std,
        friksjonsverdierMatriseDkB2Bru_std,
683                      friksjonsverdierMatriseDkB3Bru_std,
        friksjonsverdierMatriseDkCBru_std,
684                      friksjonsverdierMatriseDkDBru_std,
        friksjonsverdierMatriseDkEBru_std]
685 standardavvikUtenfor = [friksjonsverdierMatriseDkAlleUtenfor_std,
        friksjonsverdierMatriseDkAUtenfor_std,
686                          friksjonsverdierMatriseDkB1Utenfor_std,
        friksjonsverdierMatriseDkB2Utenfor_std,
687                          friksjonsverdierMatriseDkB3Utenfor_std,
        friksjonsverdierMatriseDkCUtenfor_std,
688                          friksjonsverdierMatriseDkDUtenfor_std,

```

```

    friksjonsverdierMatriseDkEUtenfor_std]
689
690 DkASignifikans = scipy.stats.ttest_1samp(friksjonsverdierMatriseDkABru,
    friksjonsverdierMatriseDkAlleBru_mean, axis=0, nan_policy='propagate',
    alternative='greater')
691 DkB1Signifikans = scipy.stats.ttest_1samp(friksjonsverdierMatriseDkB1Bru,
    friksjonsverdierMatriseDkAlleBru_mean, axis=0, nan_policy='propagate',
    alternative='greater')
692 DkB2Signifikans = scipy.stats.ttest_1samp(friksjonsverdierMatriseDkB2Bru,
    friksjonsverdierMatriseDkAlleBru_mean, axis=0, nan_policy='propagate',
    alternative='greater')
693 DkB3Signifikans = scipy.stats.ttest_1samp(friksjonsverdierMatriseDkB3Bru,
    friksjonsverdierMatriseDkAlleBru_mean, axis=0, nan_policy='propagate',
    alternative='greater')
694 DkCSignifikans = scipy.stats.ttest_1samp(friksjonsverdierMatriseDkCBru,
    friksjonsverdierMatriseDkAlleBru_mean, axis=0, nan_policy='propagate',
    alternative='less')
695 DkDSignifikans = scipy.stats.ttest_1samp(friksjonsverdierMatriseDkDBru,
    friksjonsverdierMatriseDkAlleBru_mean, axis=0, nan_policy='propagate',
    alternative='less')
696 DkESignifikans = scipy.stats.ttest_1samp(friksjonsverdierMatriseDkEBru,
    friksjonsverdierMatriseDkAlleBru_mean, axis=0, nan_policy='propagate',
    alternative='less')
697
698 DkAGjennomsnittSignifikans = scipy.stats.ttest_1samp(
    friksjonsverdierMatriseDkABru, friksjonsverdierMatriseDkA_mean, axis=0,
    nan_policy='propagate', alternative='greater')
699 DkB1GjennomsnittSignifikans = scipy.stats.ttest_1samp(
    friksjonsverdierMatriseDkB1Bru, friksjonsverdierMatriseDkB1_mean, axis
    =0, nan_policy='propagate', alternative='greater')
700 DkB2GjennomsnittSignifikans = scipy.stats.ttest_1samp(
    friksjonsverdierMatriseDkB2Bru, friksjonsverdierMatriseDkB2_mean, axis
    =0, nan_policy='propagate', alternative='greater')
701 DkB3GjennomsnittSignifikans = scipy.stats.ttest_1samp(
    friksjonsverdierMatriseDkB3Bru, friksjonsverdierMatriseDkB3_mean, axis
    =0, nan_policy='propagate', alternative='greater')
702 DkCGjennomsnittSignifikans = scipy.stats.ttest_1samp(
    friksjonsverdierMatriseDkCBru, friksjonsverdierMatriseDkC_mean, axis=0,
    nan_policy='propagate', alternative='less')
703 DkDGjennomsnittSignifikans = scipy.stats.ttest_1samp(
    friksjonsverdierMatriseDkDBru, friksjonsverdierMatriseDkD_mean, axis=0,
    nan_policy='propagate', alternative='less')
704 DkEGjennomsnittSignifikans = scipy.stats.ttest_1samp(
    friksjonsverdierMatriseDkEBru, friksjonsverdierMatriseDkE_mean, axis=0,
    nan_policy='propagate', alternative='less')
705 DkAlleGjennomsnittSignifikans = scipy.stats.ttest_1samp(
    friksjonsverdierMatriseDkAlleBru, friksjonsverdierMatriseDkAlle_mean,
    axis=0, nan_policy='propagate', alternative='less')
706
707 # Fordeler 5-persentilene i egne lister for på bru, utenfor bru, og alle
708 friksjonsverdierListeDriftsklasseBru = [friksjonsverdierListeDkABru,
    friksjonsverdierListeDkB1Bru,
709
    friksjonsverdierListeDkB2Bru,
    friksjonsverdierListeDkB3Bru,
710
    friksjonsverdierListeDkCBru,
    friksjonsverdierListeDkDBru,
711
    friksjonsverdierListeDkEBru,

```

```

    friksjonsverdierListeDriftsklasseAlleBru]
712 friksjonsverdierListeDkABru5persentil = []
713 friksjonsverdierListeDkB1Bru5persentil = []
714 friksjonsverdierListeDkB2Bru5persentil = []
715 friksjonsverdierListeDkB3Bru5persentil = []
716 friksjonsverdierListeDkCBru5persentil = []
717 friksjonsverdierListeDkDBru5persentil = []
718 friksjonsverdierListeDkEBru5persentil = []
719 friksjonsverdierListeDriftsklasseAlleBru5persentil = []
720 friksjonsverdierListeDriftsklasseBru5persentil = [
    friksjonsverdierListeDkABru5persentil,
    friksjonsverdierListeDkB1Bru5persentil,
721
    friksjonsverdierListeDkB2Bru5persentil,
    friksjonsverdierListeDkB3Bru5persentil,
722
    friksjonsverdierListeDkCBru5persentil,
    friksjonsverdierListeDkDBru5persentil,
723
    friksjonsverdierListeDkEBru5persentil,
    friksjonsverdierListeDriftsklasseAlleBru5persentil]
724 teller = -1
725 for friksjonsverdiliste in friksjonsverdierListeDriftsklasseBru:
726     midlertidigListe = []
727     teller = teller + 1
728     for i in range(round(len(friksjonsverdiliste)*0.05)):
729         midlertidigListe.append(friksjonsverdiliste[i])
730     friksjonsverdierListeDriftsklasseBru5persentil[teller] =
    midlertidigListe
731
732 friksjonsverdierListeDriftsklasseUtenfor = [friksjonsverdierListeDkAUtenfor
    , friksjonsverdierListeDkB1Utenfor,
733
    friksjonsverdierListeDkB2Utenfor, friksjonsverdierListeDkB3Utenfor,
    friksjonsverdierListeDkCUtenfor
734
    , friksjonsverdierListeDkDUtenfor,
    friksjonsverdierListeDkEUtenfor
735
    , friksjonsverdierListeDriftsklasseAlleUtenfor]
736 friksjonsverdierListeDkAUtenfor5persentil = []
737 friksjonsverdierListeDkB1Utenfor5persentil = []
738 friksjonsverdierListeDkB2Utenfor5persentil = []
739 friksjonsverdierListeDkB3Utenfor5persentil = []
740 friksjonsverdierListeDkCUtenfor5persentil = []
741 friksjonsverdierListeDkDUtenfor5persentil = []
742 friksjonsverdierListeDkEUtenfor5persentil = []
743 friksjonsverdierListeDriftsklasseAlleUtenfor5persentil = []
744 friksjonsverdierListeDriftsklasseUtenfor5persentil = [
    friksjonsverdierListeDkAUtenfor5persentil,
    friksjonsverdierListeDkB1Utenfor5persentil,
745
    friksjonsverdierListeDkB2Utenfor5persentil,
    friksjonsverdierListeDkB3Utenfor5persentil,
746
    friksjonsverdierListeDkCUtenfor5persentil,
    friksjonsverdierListeDkDUtenfor5persentil,
747
    friksjonsverdierListeDkEUtenfor5persentil,

```

```

    friksjonsverdierListeDriftsklasseAlleUtenfor5persentil]
748 teller = -1
749 for friksjonsverdiliste in friksjonsverdierListeDriftsklasseUtenfor:
750     midlertidigListe = []
751     teller = teller + 1
752     for i in range(round(len(friksjonsverdiliste)*0.05)):
753         midlertidigListe.append(friksjonsverdiliste[i])
754     friksjonsverdierListeDriftsklasseUtenfor5persentil[teller] =
        midlertidigListe
755
756 friksjonsverdierListeDriftsklasse = [friksjonsverdierListeDkA,
        friksjonsverdierListeDkB1,
757                                     friksjonsverdierListeDkB2,
        friksjonsverdierListeDkB3,
758                                     friksjonsverdierListeDkC,
        friksjonsverdierListeDkD,
759                                     friksjonsverdierListeDkE,
        friksjonsverdierListeDriftsklasseAlle]
760 friksjonsverdierListeDkA5persentil = []
761 friksjonsverdierListeDkB15persentil = []
762 friksjonsverdierListeDkB25persentil = []
763 friksjonsverdierListeDkB35persentil = []
764 friksjonsverdierListeDkC5persentil = []
765 friksjonsverdierListeDkD5persentil = []
766 friksjonsverdierListeDkE5persentil = []
767 friksjonsverdierListeDriftsklasseAlle5persentil = []
768 friksjonsverdierListeDriftsklasse5persentil = [
        friksjonsverdierListeDkA5persentil, friksjonsverdierListeDkB15persentil,
769
        friksjonsverdierListeDkB25persentil, friksjonsverdierListeDkB35persentil
770
        ,
771
        friksjonsverdierListeDkC5persentil, friksjonsverdierListeDkD5persentil,
772
        friksjonsverdierListeDkE5persentil,
        friksjonsverdierListeDriftsklasseAlle5persentil]
773 teller = -1
774 for friksjonsverdiliste in friksjonsverdierListeDriftsklasse:
775     midlertidigListe = []
776     teller = teller + 1
777     for i in range(round(len(friksjonsverdiliste)*0.05)):
778         midlertidigListe.append(friksjonsverdiliste[i])
779     friksjonsverdierListeDriftsklasse5persentil[teller] = midlertidigListe
780
781 friksjonsverdierListeDkABru15persentil = []
782 friksjonsverdierListeDkB1Bru15persentil = []
783 friksjonsverdierListeDkB2Bru15persentil = []
784 friksjonsverdierListeDkB3Bru15persentil = []
785 friksjonsverdierListeDkCBru15persentil = []
786 friksjonsverdierListeDkDBru15persentil = []
787 friksjonsverdierListeDkEBru15persentil = []
788 friksjonsverdierListeDriftsklasseAlleBru15persentil = []
789 friksjonsverdierListeDriftsklasseBru15persentil = [
        friksjonsverdierListeDkABru15persentil,
        friksjonsverdierListeDkB1Bru15persentil,
789
        friksjonsverdierListeDkB2Bru15persentil,

```

```

    friksjonsverdierListeDkB3Bru15persentil,
790     friksjonsverdierListeDkCBru15persentil,
    friksjonsverdierListeDkDBru15persentil,
791     friksjonsverdierListeDkEBru15persentil,
    friksjonsverdierListeDriftsklasseAlleBru15persentil]
792 teller = -1
793 for friksjonsverdiliste in friksjonsverdierListeDriftsklasseBru:
794     midlertidigListe = []
795     teller = teller + 1
796     for i in range(round(len(friksjonsverdiliste)*0.15)):
797         midlertidigListe.append(friksjonsverdiliste[i])
798     friksjonsverdierListeDriftsklasseBru15persentil[teller] =
    midlertidigListe
799
800 friksjonsverdierListeDriftsklasseUtenfor = [friksjonsverdierListeDkAUtenfor
    , friksjonsverdierListeDkB1Utenfor,
801     friksjonsverdierListeDkB2Utenfor, friksjonsverdierListeDkB3Utenfor,
802     , friksjonsverdierListeDkCUtenfor,
    friksjonsverdierListeDkDUtenfor,
803     , friksjonsverdierListeDkEUtenfor
    , friksjonsverdierListeDriftsklasseAlleUtenfor]
804 friksjonsverdierListeDkAUtenfor15persentil = []
805 friksjonsverdierListeDkB1Utenfor15persentil = []
806 friksjonsverdierListeDkB2Utenfor15persentil = []
807 friksjonsverdierListeDkB3Utenfor15persentil = []
808 friksjonsverdierListeDkCUtenfor15persentil = []
809 friksjonsverdierListeDkDUtenfor15persentil = []
810 friksjonsverdierListeDkEUtenfor15persentil = []
811 friksjonsverdierListeDriftsklasseAlleUtenfor15persentil = []
812 friksjonsverdierListeDriftsklasseUtenfor15persentil = [
    friksjonsverdierListeDkAUtenfor15persentil,
    friksjonsverdierListeDkB1Utenfor15persentil,
813     friksjonsverdierListeDkB2Utenfor15persentil,
    friksjonsverdierListeDkB3Utenfor15persentil,
814     friksjonsverdierListeDkCUtenfor15persentil,
    friksjonsverdierListeDkDUtenfor15persentil,
815     friksjonsverdierListeDkEUtenfor15persentil,
    friksjonsverdierListeDriftsklasseAlleUtenfor15persentil]
816 teller = -1
817 for friksjonsverdiliste in friksjonsverdierListeDriftsklasseUtenfor:
818     midlertidigListe = []
819     teller = teller + 1
820     for i in range(round(len(friksjonsverdiliste)*0.15)):
821         midlertidigListe.append(friksjonsverdiliste[i])
822     friksjonsverdierListeDriftsklasseUtenfor15persentil[teller] =
    midlertidigListe
823
824 friksjonsverdierListeDriftsklasse = [friksjonsverdierListeDkA,
    friksjonsverdierListeDkB1,
825     , friksjonsverdierListeDkB2,
    friksjonsverdierListeDkB3,

```

```

826         friksjonsverdierListeDkC,
        friksjonsverdierListeDkD,
827         friksjonsverdierListeDkE,
        friksjonsverdierListeDriftsklasseAlle]
828 friksjonsverdierListeDkA15persentil = []
829 friksjonsverdierListeDkB115persentil = []
830 friksjonsverdierListeDkB215persentil = []
831 friksjonsverdierListeDkB315persentil = []
832 friksjonsverdierListeDkC15persentil = []
833 friksjonsverdierListeDkD15persentil = []
834 friksjonsverdierListeDkE15persentil = []
835 friksjonsverdierListeDriftsklasseAlle15persentil = []
836 friksjonsverdierListeDriftsklasse15persentil = [
        friksjonsverdierListeDkA15persentil,
        friksjonsverdierListeDkB115persentil,
837
        friksjonsverdierListeDkB215persentil,
        friksjonsverdierListeDkB315persentil,
838
        friksjonsverdierListeDkC15persentil, friksjonsverdierListeDkD15persentil
        ,
839
        friksjonsverdierListeDkE15persentil,
        friksjonsverdierListeDriftsklasseAlle15persentil]
840 teller = -1
841 for friksjonsverdiliste in friksjonsverdierListeDriftsklasse:
842     midlertidigListe = []
843     teller = teller + 1
844     for i in range(round(len(friksjonsverdiliste)*0.15)):
845         midlertidigListe.append(friksjonsverdiliste[i])
846     friksjonsverdierListeDriftsklasse15persentil[teller] = midlertidigListe
847
848 # Gjør 5-persentil-listene til matriser og finner gjennomsnittsverdien for
    "alle"
849 friksjonsverdierMatriseDkABru5persentil = []
850 friksjonsverdierMatriseDkB1Bru5persentil = []
851 friksjonsverdierMatriseDkB2Bru5persentil = []
852 friksjonsverdierMatriseDkB3Bru5persentil = []
853 friksjonsverdierMatriseDkCBru5persentil = []
854 friksjonsverdierMatriseDkDBru5persentil = []
855 friksjonsverdierMatriseDkEBru5persentil = []
856 friksjonsverdierMatriseDriftsklasseAlleBru5persentil = []
857 friksjonsverdierMatriseDkAUtenfor5persentil = []
858 friksjonsverdierMatriseDkB1Utenfor5persentil = []
859 friksjonsverdierMatriseDkB2Utenfor5persentil = []
860 friksjonsverdierMatriseDkB3Utenfor5persentil = []
861 friksjonsverdierMatriseDkCUtenfor5persentil = []
862 friksjonsverdierMatriseDkDUtenfor5persentil = []
863 friksjonsverdierMatriseDkEUtenfor5persentil = []
864 friksjonsverdierMatriseDriftsklasseAlleUtenfor5persentil = []
865 friksjonsverdierMatriseDkA5persentil = []
866 friksjonsverdierMatriseDkB15persentil = []
867 friksjonsverdierMatriseDkB25persentil = []
868 friksjonsverdierMatriseDkB35persentil = []
869 friksjonsverdierMatriseDkC5persentil = []
870 friksjonsverdierMatriseDkD5persentil = []
871 friksjonsverdierMatriseDkE5persentil = []

```

```
872 friksjonsverdierMatriseDriftsklasseAlle5persentil = []
873 friksjonsverdierMatriseDkA5persentil_mean = 0
874 friksjonsverdierMatriseDkB15persentil_mean = 0
875 friksjonsverdierMatriseDkB25persentil_mean = 0
876 friksjonsverdierMatriseDkB35persentil_mean = 0
877 friksjonsverdierMatriseDkC5persentil_mean = 0
878 friksjonsverdierMatriseDkD5persentil_mean = 0
879 friksjonsverdierMatriseDkE5persentil_mean = 0
880 friksjonsverdierMatriseDriftsklasseAlle5persentil_mean = 0
881
882 friksjonsverdierMatriseDriftsklasseBru5persentil = [
    friksjonsverdierMatriseDkABru5persentil,
    friksjonsverdierMatriseDkB1Bru5persentil,
883
    friksjonsverdierMatriseDkB2Bru5persentil,
    friksjonsverdierMatriseDkB3Bru5persentil,
884
    friksjonsverdierMatriseDkCBru5persentil,
    friksjonsverdierMatriseDkDBru5persentil,
885
    friksjonsverdierMatriseDkEBru5persentil,
    friksjonsverdierMatriseDriftsklasseAlleBru5persentil]
886 friksjonsverdierMatriseDriftsklasseUtenfor5persentil = [
    friksjonsverdierMatriseDkAUtenfor5persentil,
    friksjonsverdierMatriseDkB1Utenfor5persentil,
887
    friksjonsverdierMatriseDkB2Utenfor5persentil,
    friksjonsverdierMatriseDkB3Utenfor5persentil,
888
    friksjonsverdierMatriseDkCUtenfor5persentil,
    friksjonsverdierMatriseDkDUtenfor5persentil,
889
    friksjonsverdierMatriseDkEUtenfor5persentil,
    friksjonsverdierMatriseDriftsklasseAlleUtenfor5persentil]
890 friksjonsverdierMatriseDriftsklasse5persentil = [
    friksjonsverdierMatriseDkA5persentil,
    friksjonsverdierMatriseDkB15persentil,
891
    friksjonsverdierMatriseDkB25persentil,
    friksjonsverdierMatriseDkB35persentil,
892
    friksjonsverdierMatriseDkC5persentil,
    friksjonsverdierMatriseDkD5persentil,
893
    friksjonsverdierMatriseDkE5persentil,
    friksjonsverdierMatriseDriftsklasseAlle5persentil]
894 friksjonsverdierMatriseDriftsklasse5persentil_mean = [
    friksjonsverdierMatriseDkA5persentil_mean,
    friksjonsverdierMatriseDkB15persentil_mean,
895
    friksjonsverdierMatriseDkB25persentil_mean,
    friksjonsverdierMatriseDkB35persentil_mean,
896
    friksjonsverdierMatriseDkC5persentil_mean,
    friksjonsverdierMatriseDkD5persentil_mean,
897
    friksjonsverdierMatriseDkE5persentil_mean,
```

```

    friksjonsverdierMatriseDriftsklasseAlle5persentil_mean]
898
899 for i in range(len(friksjonsverdierMatriseDriftsklasseBru5persentil)):
900     friksjonsverdierMatriseDriftsklasseBru5persentil[i] = np.asarray(
        friksjonsverdierListeDriftsklasseBru5persentil[i])
901     friksjonsverdierMatriseDriftsklasseUtenfor5persentil[i] = np.asarray(
        friksjonsverdierListeDriftsklasseUtenfor5persentil[i])
902     friksjonsverdierMatriseDriftsklasse5persentil[i] = np.asarray(
        friksjonsverdierListeDriftsklasse5persentil[i])
903     friksjonsverdierMatriseDriftsklasse5persentil_mean[i] = np.mean(
        friksjonsverdierMatriseDriftsklasse5persentil[i])
904
905 # Gjør 5-persentil-listene til matriser og finner gjennomsnittsverdien for
    "alle"
906 friksjonsverdierMatriseDkABru15persentil = []
907 friksjonsverdierMatriseDkB1Bru15persentil = []
908 friksjonsverdierMatriseDkB2Bru15persentil = []
909 friksjonsverdierMatriseDkB3Bru15persentil = []
910 friksjonsverdierMatriseDkCBru15persentil = []
911 friksjonsverdierMatriseDkDBru15persentil = []
912 friksjonsverdierMatriseDkEBru15persentil = []
913 friksjonsverdierMatriseDriftsklasseAlleBru15persentil = []
914 friksjonsverdierMatriseDkAUtenfor15persentil = []
915 friksjonsverdierMatriseDkB1Utenfor15persentil = []
916 friksjonsverdierMatriseDkB2Utenfor15persentil = []
917 friksjonsverdierMatriseDkB3Utenfor15persentil = []
918 friksjonsverdierMatriseDkCUtenfor15persentil = []
919 friksjonsverdierMatriseDkDUtenfor15persentil = []
920 friksjonsverdierMatriseDkEUtenfor15persentil = []
921 friksjonsverdierMatriseDriftsklasseAlleUtenfor15persentil = []
922 friksjonsverdierMatriseDkA15persentil = []
923 friksjonsverdierMatriseDkB115persentil = []
924 friksjonsverdierMatriseDkB215persentil = []
925 friksjonsverdierMatriseDkB315persentil = []
926 friksjonsverdierMatriseDkC15persentil = []
927 friksjonsverdierMatriseDkD15persentil = []
928 friksjonsverdierMatriseDkE15persentil = []
929 friksjonsverdierMatriseDriftsklasseAlle15persentil = []
930 friksjonsverdierMatriseDkA15persentil_mean = 0
931 friksjonsverdierMatriseDkB115persentil_mean = 0
932 friksjonsverdierMatriseDkB215persentil_mean = 0
933 friksjonsverdierMatriseDkB315persentil_mean = 0
934 friksjonsverdierMatriseDkC15persentil_mean = 0
935 friksjonsverdierMatriseDkD15persentil_mean = 0
936 friksjonsverdierMatriseDkE15persentil_mean = 0
937 friksjonsverdierMatriseDriftsklasseAlle15persentil_mean = 0
938
939 friksjonsverdierMatriseDriftsklasseBru15persentil = [
        friksjonsverdierMatriseDkABru15persentil,
        friksjonsverdierMatriseDkB1Bru15persentil,
940
        friksjonsverdierMatriseDkB2Bru15persentil,
        friksjonsverdierMatriseDkB3Bru15persentil,
941
        friksjonsverdierMatriseDkCBru15persentil,
        friksjonsverdierMatriseDkDBru15persentil,
942

```

```

    friksjonsverdierMatriseDkEBru15persentil,
    friksjonsverdierMatriseDriftsklasseAlleBru15persentil]
943 friksjonsverdierMatriseDriftsklasseUtenfor15persentil = [
    friksjonsverdierMatriseDkAUtenfor15persentil,
    friksjonsverdierMatriseDkB1Utenfor15persentil,
944
    friksjonsverdierMatriseDkB2Utenfor15persentil,
    friksjonsverdierMatriseDkB3Utenfor15persentil,
945
    friksjonsverdierMatriseDkCUtenfor15persentil,
    friksjonsverdierMatriseDkDUtenfor15persentil,
946
    friksjonsverdierMatriseDkEUtenfor15persentil,
    friksjonsverdierMatriseDriftsklasseAlleUtenfor15persentil]
947 friksjonsverdierMatriseDriftsklasse15persentil = [
    friksjonsverdierMatriseDkA15persentil,
    friksjonsverdierMatriseDkB115persentil,
948
    friksjonsverdierMatriseDkB215persentil,
    friksjonsverdierMatriseDkB315persentil,
949
    friksjonsverdierMatriseDkC15persentil,
    friksjonsverdierMatriseDkD15persentil,
950
    friksjonsverdierMatriseDkE15persentil,
    friksjonsverdierMatriseDriftsklasseAlle15persentil]
951 friksjonsverdierMatriseDriftsklasse15persentil_mean = [
    friksjonsverdierMatriseDkA15persentil_mean,
    friksjonsverdierMatriseDkB115persentil_mean,
952
    friksjonsverdierMatriseDkB215persentil_mean,
    friksjonsverdierMatriseDkB315persentil_mean,
953
    friksjonsverdierMatriseDkC15persentil_mean,
    friksjonsverdierMatriseDkD15persentil_mean,
954
    friksjonsverdierMatriseDkE15persentil_mean,
    friksjonsverdierMatriseDriftsklasseAlle15persentil_mean]
955
956 for i in range(len(friksjonsverdierMatriseDriftsklasseBru15persentil)):
957     friksjonsverdierMatriseDriftsklasseBru15persentil[i] = np.asarray(
    friksjonsverdierListeDriftsklasseBru15persentil[i])
958     friksjonsverdierMatriseDriftsklasseUtenfor15persentil[i] = np.asarray(
    friksjonsverdierListeDriftsklasseUtenfor15persentil[i])
959     friksjonsverdierMatriseDriftsklasse15persentil[i] = np.asarray(
    friksjonsverdierListeDriftsklasse15persentil[i])
960     friksjonsverdierMatriseDriftsklasse15persentil_mean[i] = np.mean(
    friksjonsverdierMatriseDriftsklasse15persentil[i])
961
962 friksjonsverdierMatriseDriftsklasseBru = [friksjonsverdierMatriseDkABru,
    friksjonsverdierMatriseDkB1Bru,
963
    friksjonsverdierMatriseDkB2Bru,
    friksjonsverdierMatriseDkB3Bru,
964
    friksjonsverdierMatriseDkCBru,
    friksjonsverdierMatriseDkDBru,
965
    friksjonsverdierMatriseDkEBru,
    friksjonsverdierMatriseDkAlleBru]

```

```

966 friksjonsverdierMatriseDriftsklasseUtenfor = [
    friksjonsverdierMatriseDkAUtenfor, friksjonsverdierMatriseDkB1Utenfor,
967
    friksjonsverdierMatriseDkB2Utenfor, friksjonsverdierMatriseDkB3Utenfor,
968
    friksjonsverdierMatriseDkCUtenfor, friksjonsverdierMatriseDkDUtenfor,
969
    friksjonsverdierMatriseDkEUtenfor, friksjonsverdierMatriseDkAlleUtenfor]
970 friksjonsverdierMatriseDriftsklasse = [friksjonsverdierMatriseDkA,
    friksjonsverdierMatriseDkB1,
971
    friksjonsverdierMatriseDkB2,
    friksjonsverdierMatriseDkB3,
972
    friksjonsverdierMatriseDkC,
    friksjonsverdierMatriseDkC,
973
    friksjonsverdierMatriseDkE,
    friksjonsverdierMatriseDkAlle]
974
975 DkA5persentilSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru5persentil[0],
    friksjonsverdierMatriseDriftsklasseUtenfor5persentil[0], equal_var=True,
    alternative='two-sided')
976 DkB15persentilSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru5persentil[1],
    friksjonsverdierMatriseDriftsklasseUtenfor5persentil[1], equal_var=True,
    alternative='two-sided')
977 DkB25persentilSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru5persentil[2],
    friksjonsverdierMatriseDriftsklasseUtenfor5persentil[2], equal_var=True,
    alternative='two-sided')
978 DkB35persentilSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru5persentil[3],
    friksjonsverdierMatriseDriftsklasseUtenfor5persentil[3], equal_var=True,
    alternative='two-sided')
979 DkC5persentilSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru5persentil[4],
    friksjonsverdierMatriseDriftsklasseUtenfor5persentil[4], equal_var=True,
    alternative='two-sided')
980 DkD5persentilSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru5persentil[5],
    friksjonsverdierMatriseDriftsklasseUtenfor5persentil[5], equal_var=True,
    alternative='two-sided')
981 DkE5persentilSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru5persentil[6],
    friksjonsverdierMatriseDriftsklasseUtenfor5persentil[6], equal_var=True,
    alternative='two-sided')
982 DkAlle5persentilSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru5persentil[7],
    friksjonsverdierMatriseDriftsklasseUtenfor5persentil[7], equal_var=True,
    alternative='two-sided')
983
984 DkA15persentilSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru15persentil[0],
    friksjonsverdierMatriseDriftsklasseUtenfor15persentil[0], equal_var=True
    , alternative='two-sided')
985 DkB115persentilSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru15persentil[1],
    friksjonsverdierMatriseDriftsklasseUtenfor15persentil[1], equal_var=True

```

```

    , alternative='two-sided')
986 DkB215persentilSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru15persentil[2],
    friksjonsverdierMatriseDriftsklasseUtenfor15persentil[2], equal_var=True
    , alternative='two-sided')
987 DkB315persentilSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru15persentil[3],
    friksjonsverdierMatriseDriftsklasseUtenfor15persentil[3], equal_var=True
    , alternative='two-sided')
988 DkC15persentilSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru15persentil[4],
    friksjonsverdierMatriseDriftsklasseUtenfor15persentil[4], equal_var=True
    , alternative='two-sided')
989 DkD15persentilSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru15persentil[5],
    friksjonsverdierMatriseDriftsklasseUtenfor15persentil[5], equal_var=True
    , alternative='two-sided')
990 DkE15persentilSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru15persentil[6],
    friksjonsverdierMatriseDriftsklasseUtenfor15persentil[6], equal_var=True
    , alternative='two-sided')
991 DkAlle15persentilSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru15persentil[7],
    friksjonsverdierMatriseDriftsklasseUtenfor15persentil[7], equal_var=True
    , alternative='two-sided')
992
993 DkAGjennomsnittSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru[0],
    friksjonsverdierMatriseDriftsklasseUtenfor[0], equal_var=True,
    alternative='two-sided')
994 DkB1GjennomsnittSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru[1],
    friksjonsverdierMatriseDriftsklasseUtenfor[1], equal_var=True,
    alternative='two-sided')
995 DkB2GjennomsnittSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru[2],
    friksjonsverdierMatriseDriftsklasseUtenfor[2], equal_var=True,
    alternative='two-sided')
996 DkB3GjennomsnittSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru[3],
    friksjonsverdierMatriseDriftsklasseUtenfor[3], equal_var=True,
    alternative='two-sided')
997 DkCGjennomsnittSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru[4],
    friksjonsverdierMatriseDriftsklasseUtenfor[4], equal_var=True,
    alternative='two-sided')
998 DkDGjennomsnittSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru[5],
    friksjonsverdierMatriseDriftsklasseUtenfor[5], equal_var=True,
    alternative='two-sided')
999 DkEGjennomsnittSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru[6],
    friksjonsverdierMatriseDriftsklasseUtenfor[6], equal_var=True,
    alternative='two-sided')
1000 DkAlleGjennomsnittSignifikansInd = scipy.stats.ttest_ind(
    friksjonsverdierMatriseDriftsklasseBru[7],
    friksjonsverdierMatriseDriftsklasseUtenfor[7], equal_var=True,

```

```

        alternative='two-sided')
1001
1002
1003 # Gir informasjon om datasettet
1004 print('\nInformasjon om datagrunnlaget:')
1005 print('Antall filer med målepunkter maksimum 100 meter fra ei bru: {}'.
        format(len(listeFilerMatriser)))
1006 print('Antall bruer med målepunkter maksimum 100 meter fra ei bru: {}'.
        format(len(listeBruerMatriser)))
1007 print('Antall bruer med friksjon målt på selve bruene: {}'.format(
        antallBruer))
1008 print('Antall innenfor 100 meter av ei bru: {}'.format(antallBrupunkter))
1009 print('Antall målepunkter mellom klokken 08:00 og 20:00: {}, og mellom
        klokken 20:00 og 08:00: {}'.format(len(klokkeslett8_20), len(
        klokkeslett20_8)))
1010 print('Fordeling av målepunkter på formen *fylkesnummer*: *antall må
        lepunkter*')
1011 print(Counter(fylkesnummerListe))
1012 print('\n')
1013
1014 # Gir informasjon om tilfeller hvor brua har lavere friksjon enn vegen
        rundt
1015 print('Informasjon om målepunkter hvor brua har lavere friksjon enn må
        lepunktene rundt bra:')
1016 print('Antall tilfeller hvor differansen mellom laveste friksjonsverdi på
        bru og rundt bru er mellom 0,00 og 0,05: {}'.format(
        friksjonLavestPaaBruTeller0))
1017 print('Lavest friksjonsverdi: {}, nest lavest friksjonsverdi: {}'.format(
        minBruFriksjonListe0[0], minBruFriksjonListe0[1]))
1018 print('Antall tilfeller hvor differansen mellom laveste friksjonsverdi på
        bru og rundt bru er mellom 0,05 og 0,10: {}'.format(
        friksjonLavestPaaBruTeller1))
1019 print('Lavest friksjonsverdi: {}, nest lavest friksjonsverdi: {}'.format(
        minBruFriksjonListe1[0], minBruFriksjonListe1[1]))
1020 print('Antall tilfeller hvor differansen mellom laveste friksjonsverdi på
        bru og rundt bru er mellom 0,10 og 0,15: {}'.format(
        friksjonLavestPaaBruTeller2))
1021 print('Lavest friksjonsverdi: {}, nest lavest friksjonsverdi: {}'.format(
        minBruFriksjonListe2[0], minBruFriksjonListe2[1]))
1022 print('Antall tilfeller hvor differansen mellom laveste friksjonsverdi på
        bru og rundt bru er høyere enn 0,15: {}'.format(
        friksjonLavestPaaBruTeller3))
1023 print('Lavest friksjonsverdi: {}, nest lavest friksjonsverdi: {}\n'.format(
        minBruFriksjonListe3[0], minBruFriksjonListe3[1]))
1024
1025 # Gir informasjon om 95-persentil for laveste friksjonsverdi og
        gjennomsnittlig friksjonsverdi
1026 print('Informasjon om vinterdriftsklasser:')
1027 print('Alle Dk, Kun bru, 5-persentil: {:.2f}, 15-persentil: {:.2f},
        Gjennomsnitt: {:.3f}, n: {}'.format(
1028     np.percentile(friksjonsverdierMatriseDkAlleBru, 5),
1029     np.percentile(friksjonsverdierMatriseDkAlleBru, 15),
1030     np.average(friksjonsverdierMatriseDkAlleBru),
1031     len(friksjonsverdierListeDriftsklasseAlleBru))
1032 print('Alle Dk, Utenfor bru, 5-persentil: {:.2f}, 15-persentil: {:.2f},
        Gjennomsnitt: {:.3f}, n: {}'.format(
1033     np.percentile(friksjonsverdierMatriseDkAlleUtenfor, 5),

```

```

1034     np.percentile(friksjonsverdierMatriseDkAlleUtenfor, 15),
1035     np.average(friksjonsverdierMatriseDkAlleUtenfor),
1036     len(friksjonsverdierListeDriftsklasseAlleUtenfor)))
1037 print('Alle Dk, Alle målepunkter, 5-persentil: {:.2f}, 15-persentil: {:.2f}
      ), Gjennomsnitt: {:.3f}, n: {}'.format(
1038     np.percentile(friksjonsverdierMatriseDkAlle, 5),
1039     np.percentile(friksjonsverdierMatriseDkAlle, 15),
1040     np.average(friksjonsverdierMatriseDkAlle),
1041     len(friksjonsverdierListeDriftsklasseAlle)))
1042 print('DkA, Kun bru, 5-persentil: {:.2f}, 15-persentil: {:.2f},
      Gjennomsnitt: {:.3f}, p-verdi: {}, n: {}'.format(
1043     np.percentile(friksjonsverdierListeDkABru, 5),
1044     np.percentile(friksjonsverdierListeDkABru, 15),
1045     np.average(friksjonsverdierListeDkABru),
1046     DkASignifikans[1],
1047     len(friksjonsverdierListeDkABru)))
1048 print('DkA, Utenfor bru, 5-persentil: {:.2f}, 15-persentil: {:.2f},
      Gjennomsnitt: {:.3f}, n: {}'.format(
1049     np.percentile(friksjonsverdierMatriseDkAUtenfor, 5),
1050     np.percentile(friksjonsverdierMatriseDkAUtenfor, 15),
1051     np.average(friksjonsverdierMatriseDkAUtenfor),
1052     len(friksjonsverdierListeDkAUtenfor)))
1053 print('DkA, Alle målepunkter, 5-persentil: {:.2f}, 15-persentil: {:.2f},
      Gjennomsnitt: {:.3f}, n: {}'.format(
1054     np.percentile(friksjonsverdierMatriseDkA, 5),
1055     np.percentile(friksjonsverdierMatriseDkA, 15),
1056     np.average(friksjonsverdierMatriseDkA),
1057     len(friksjonsverdierListeDkA)))
1058 print('DkB høy, Kun bru, 5-persentil: {:.2f}, 15-persentil: {:.2f},
      Gjennomsnitt: {:.3f}, p-verdi: {}, n: {}'.format(
1059     np.percentile(friksjonsverdierMatriseDkB1Bru, 5),
1060     np.percentile(friksjonsverdierMatriseDkB1Bru, 15),
1061     np.average(friksjonsverdierMatriseDkB1Bru),
1062     DkB1Signifikans[1],
1063     len(friksjonsverdierListeDkB1Bru)))
1064 print('DkB høy, Utenfor bru, 5-persentil: {:.2f}, 15-persentil: {:.2f},
      Gjennomsnitt: {:.3f}, n: {}'.format(
1065     np.percentile(friksjonsverdierMatriseDkB1Utenfor, 5),
1066     np.percentile(friksjonsverdierMatriseDkB1Utenfor, 15),
1067     np.average(friksjonsverdierMatriseDkB1Utenfor),
1068     len(friksjonsverdierListeDkB1Utenfor)))
1069 print('DkB høy, Alle målepunkter, 5-persentil: {:.2f}, 15-persentil: {:.2f}
      ), Gjennomsnitt: {:.3f}, n: {}'.format(
1070     np.percentile(friksjonsverdierMatriseDkB1, 5),
1071     np.percentile(friksjonsverdierMatriseDkB1, 15),
1072     np.average(friksjonsverdierMatriseDkB1),
1073     len(friksjonsverdierListeDkB1)))
1074 print('DkB middels, Kun bru, 5-persentil: {:.2f}, 15-persentil: {:.2f},
      Gjennomsnitt: {:.3f}, p-verdi: {}, n: {}'.format(
1075     np.percentile(friksjonsverdierMatriseDkB2Bru, 5),
1076     np.percentile(friksjonsverdierMatriseDkB2Bru, 15),
1077     np.average(friksjonsverdierMatriseDkB2Bru),
1078     DkB2Signifikans[1],
1079     len(friksjonsverdierListeDkB2Bru)))
1080 print('DkB middels, Utenfor bru, 5-persentil: {:.2f}, 15-persentil: {:.2f},
      Gjennomsnitt: {:.3f}, n: {}'.format(
1081     np.percentile(friksjonsverdierMatriseDkB2Utenfor, 5),

```

```

1082     np.percentile(friksjonsverdierMatriseDkB2Utenfor, 15),
1083     np.average(friksjonsverdierMatriseDkB2Utenfor),
1084     len(friksjonsverdierListeDkB2Utenfor))
1085 print('DkB middels, Alle målepunkter, 5-persentil: {:.2f}, 15-persentil:
      {:.2f}, Gjennomsnitt: {:.3f}, n: {}'.format(
1086     np.percentile(friksjonsverdierMatriseDkB2, 5),
1087     np.percentile(friksjonsverdierMatriseDkB2, 15),
1088     np.average(friksjonsverdierMatriseDkB2),
1089     len(friksjonsverdierListeDkB2)))
1090 print('DkB lav, Kun bru, 5-persentil: {:.2f}, 15-persentil: {:.2f},
      Gjennomsnitt: {:.3f}, p-verdi: {}, n: {}'.format(
1091     np.percentile(friksjonsverdierMatriseDkB3Bru, 5),
1092     np.percentile(friksjonsverdierMatriseDkB3Bru, 15),
1093     np.average(friksjonsverdierMatriseDkB3Bru),
1094     DkB3Signifikans[1],
1095     len(friksjonsverdierListeDkB3Bru)))
1096 print('DkB lav, Utenfor bru, 5-persentil: {:.2f}, 15-persentil: {:.2f},
      Gjennomsnitt: {:.3f}, n: {}'.format(
1097     np.percentile(friksjonsverdierMatriseDkB3Utenfor, 5),
1098     np.percentile(friksjonsverdierMatriseDkB3Utenfor, 15),
1099     np.average(friksjonsverdierMatriseDkB3Utenfor),
1100     len(friksjonsverdierListeDkB3Utenfor)))
1101 print('DkB lav, Alle målepunkter, 5-persentil: {:.2f}, 15-persentil: {:.2f}
      }, Gjennomsnitt: {:.3f}, n: {}'.format(
1102     np.percentile(friksjonsverdierMatriseDkB3, 5),
1103     np.percentile(friksjonsverdierMatriseDkB3, 15),
1104     np.average(friksjonsverdierMatriseDkB3),
1105     len(friksjonsverdierListeDkB3)))
1106 print('DkC, Kun bru, 5-persentil: {:.2f}, 15-persentil: {:.2f},
      Gjennomsnitt: {:.3f}, p-verdi: {}, n: {}'.format(
1107     np.percentile(friksjonsverdierMatriseDkCBru, 5),
1108     np.percentile(friksjonsverdierMatriseDkCBru, 15),
1109     np.average(friksjonsverdierMatriseDkCBru),
1110     DkCSignifikans[1],
1111     len(friksjonsverdierListeDkCBru)))
1112 print('DkC, Utenfor bru, 5-persentil: {:.2f}, 15-persentil: {:.2f},
      Gjennomsnitt: {:.3f}, n: {}'.format(
1113     np.percentile(friksjonsverdierMatriseDkCUtenfor, 5),
1114     np.percentile(friksjonsverdierMatriseDkCUtenfor, 15),
1115     np.average(friksjonsverdierMatriseDkCUtenfor),
1116     len(friksjonsverdierListeDkCUtenfor)))
1117 print('DkC, Alle målepunkter, 5-persentil: {:.2f}, 15-persentil: {:.2f},
      Gjennomsnitt: {:.3f}, n: {}'.format(
1118     np.percentile(friksjonsverdierMatriseDkC, 5),
1119     np.percentile(friksjonsverdierMatriseDkC, 15),
1120     np.average(friksjonsverdierMatriseDkC),
1121     len(friksjonsverdierListeDkC)))
1122 print('DkD, Kun bru, 5-persentil: {:.2f}, 15-persentil: {:.2f},
      Gjennomsnitt: {:.3f}, p-verdi: {}, n: {}'.format(
1123     np.percentile(friksjonsverdierMatriseDkDBru, 5),
1124     np.percentile(friksjonsverdierMatriseDkDBru, 15),
1125     np.average(friksjonsverdierMatriseDkDBru),
1126     DkDSignifikans[1],
1127     len(friksjonsverdierListeDkDBru)))
1128 print('DkD, Utenfor bru, 5-persentil: {:.2f}, 15-persentil: {:.2f},
      Gjennomsnitt: {:.3f}, n: {}'.format(
1129     np.percentile(friksjonsverdierMatriseDkDUtenfor, 5),

```

```

1130     np.percentile(friksjonsverdierMatriseDkDUtenfor, 15),
1131     np.average(friksjonsverdierMatriseDkDUtenfor),
1132     len(friksjonsverdierListeDkDUtenfor))
1133 print('DkD, Alle målepunkter, 5-persentil: {:.2f}, 15-persentil: {:.2f},
      Gjennomsnitt: {:.3f}, n: {}'.format(
1134     np.percentile(friksjonsverdierMatriseDkD, 5),
1135     np.percentile(friksjonsverdierMatriseDkD, 15),
1136     np.average(friksjonsverdierMatriseDkD),
1137     len(friksjonsverdierListeDkD))
1138 print('DkE, Kun bru, 5-persentil: {:.2f}, 15-persentil: {:.2f},
      Gjennomsnitt: {:.3f}, p-verdi: {}, n: {}'.format(
1139     np.percentile(friksjonsverdierMatriseDkEBru, 5),
1140     np.percentile(friksjonsverdierMatriseDkEBru, 15),
1141     np.average(friksjonsverdierMatriseDkEBru),
1142     DkESignifikans[1],
1143     len(friksjonsverdierListeDkEBru))
1144 print('DkE, Utenfor bru, 5-persentil: {:.2f}, 15-persentil: {:.2f},
      Gjennomsnitt: {:.3f}, n: {}'.format(
1145     np.percentile(friksjonsverdierMatriseDkEUtenfor, 5),
1146     np.percentile(friksjonsverdierMatriseDkEUtenfor, 15),
1147     np.average(friksjonsverdierMatriseDkEUtenfor),
1148     len(friksjonsverdierListeDkEUtenfor))
1149 print('DkE, Alle målepunkter, 5-persentil: {:.2f}, 15-persentil: {:.2f},
      Gjennomsnitt: {:.3f}, n: {}'.format(
1150     np.percentile(friksjonsverdierMatriseDkE, 5),
1151     np.percentile(friksjonsverdierMatriseDkE, 15),
1152     np.average(friksjonsverdierMatriseDkE),
1153     len(friksjonsverdierListeDkE))
1154
1155 # Gir informasjon om noen av 5-persentilene på bru er signifikant høyere
      enn punktene forøvrig for hver vinterdriftsklasse
1156 print('\nVurdering av signifikans, Independent t-test:')
1157 print('5-persentiler:')
1158 print('Vinterdriftsklasse A sin 5-persentil har p-verdi = {} for bru
      sammenlignet med alle punkter'.format(DkA5persentilSignifikansInd[1]))
1159 print('Vinterdriftsklasse B, høy sin 5-persentil har p-verdi = {} for bru
      sammenlignet med alle punkter'.format(DkB15persentilSignifikansInd[1]))
1160 print('Vinterdriftsklasse B, middels sin 5-persentil har p-verdi = {} for
      bru sammenlignet med alle punkter'.format(DkB25persentilSignifikansInd
      [1]))
1161 print('Vinterdriftsklasse B, lav sin 5-persentil har p-verdi = {} for bru
      sammenlignet med alle punkter'.format(DkB35persentilSignifikansInd[1]))
1162 print('Vinterdriftsklasse C sin 5-persentil har p-verdi = {} for bru
      sammenlignet med alle punkter'.format(DkC5persentilSignifikansInd[1]))
1163 print('Vinterdriftsklasse D sin 5-persentil har p-verdi = {} for bru
      sammenlignet med alle punkter'.format(DkD5persentilSignifikansInd[1]))
1164 print('Vinterdriftsklasse E sin 5-persentil har p-verdi = {} for bru
      sammenlignet med alle punkter'.format(DkE5persentilSignifikansInd[1]))
1165 print('Vinterdriftsklasse Alle sin 5-persentil har p-verdi = {} for bru
      sammenlignet med alle punkter'.format(DkAlle5persentilSignifikansInd[1]
      )
1166 print('15-persentiler:')
1167 print('Vinterdriftsklasse A sin 15-persentil har p-verdi = {} for bru
      sammenlignet med alle punkter'.format(DkA15persentilSignifikansInd[1]))
1168 print('Vinterdriftsklasse B, høy sin 15-persentil har p-verdi = {} for bru
      sammenlignet med alle punkter'.format(DkB115persentilSignifikansInd[1]))
1169 print('Vinterdriftsklasse B, middels sin 15-persentil har p-verdi = {} for

```



```

    bru sammenlignet med alle punkter'.format(DkB215persentilSignifikansInd
[1]))
1170 print('Vinterdriftsklasse B, lav sin 15-persentil har p-verdi = {} for bru
    sammenlignet med alle punkter'.format(DkB315persentilSignifikansInd[1]))
1171 print('Vinterdriftsklasse C sin 15-persentil har p-verdi = {} for bru
    sammenlignet med alle punkter'.format(DkC15persentilSignifikansInd[1]))
1172 print('Vinterdriftsklasse D sin 15-persentil har p-verdi = {} for bru
    sammenlignet med alle punkter'.format(DkD15persentilSignifikansInd[1]))
1173 print('Vinterdriftsklasse E sin 15-persentil har p-verdi = {} for bru
    sammenlignet med alle punkter'.format(DkE15persentilSignifikansInd[1]))
1174 print('Vinterdriftsklasse Alle sin 15-persentil har p-verdi = {} for bru
    sammenlignet med alle punkter'.format(DkAlle15persentilSignifikansInd
[1]))
1175 print('Gjennomsnitt:')
1176 print('Vinterdriftsklasse A sitt gjennomsnitt har p-verdi = {} for bru
    sammenlignet med alle punkter'.format(DkAGjennomsnittSignifikansInd[1]))
1177 print('Vinterdriftsklasse B, høy sitt gjennomsnitt har p-verdi = {} for bru
    sammenlignet med alle punkter'.format(DkB1GjennomsnittSignifikansInd
[1]))
1178 print('Vinterdriftsklasse B, middels sitt gjennomsnitt har p-verdi = {} for
    bru sammenlignet med alle punkter'.format(
    DkB2GjennomsnittSignifikansInd[1]))
1179 print('Vinterdriftsklasse B, lav sitt gjennomsnitt har p-verdi = {} for bru
    sammenlignet med alle punkter'.format(DkB3GjennomsnittSignifikansInd
[1]))
1180 print('Vinterdriftsklasse C sitt gjennomsnitt har p-verdi = {} for bru
    sammenlignet med alle punkter'.format(DkCGjennomsnittSignifikansInd[1]))
1181 print('Vinterdriftsklasse D sitt gjennomsnitt har p-verdi = {} for bru
    sammenlignet med alle punkter'.format(DkDGjennomsnittSignifikansInd[1]))
1182 print('Vinterdriftsklasse E sitt gjennomsnitt har p-verdi = {} for bru
    sammenlignet med alle punkter'.format(DkEGjennomsnittSignifikansInd[1]))
1183 print('Vinterdriftsklasse Alle sitt gjennomsnitt har p-verdi = {} for bru
    sammenlignet med alle punkter'.format(DkAlleGjennomsnittSignifikansInd
[1]))
1184
1185
1186 # Gir informasjon om brumaterialene
1187 print('\nInformasjon om brumaterialer og hvor hyppig de forekommer blant 5-
    persentilen for friksjonsverdier til ulike vinterdriftsklasser:')
1188 print('Vinterdriftsklasse A, 5-persentil:', Counter(
    bruMaterialeListeDkA5persentil))
1189 print('Vinterdriftsklasse B1, 5-persentil:', Counter(
    bruMaterialeListeDkB15persentil))
1190 print('Vinterdriftsklasse B2, 5-persentil:', Counter(
    bruMaterialeListeDkB25persentil))
1191 print('Vinterdriftsklasse B3, 5-persentil:', Counter(
    bruMaterialeListeDkB35persentil))
1192 print('Vinterdriftsklasse C, 5-persentil:', Counter(
    bruMaterialeListeDkC5persentil))
1193 print('Vinterdriftsklasse D, 5-persentil:', Counter(
    bruMaterialeListeDkD5persentil))
1194 print('Vinterdriftsklasse E, 5-persentil:', Counter(
    bruMaterialeListeDkE5persentil))
1195 print('Alle forekomster:', Counter(bruMaterialeListeDriftsklasseAlleBru))
1196
1197 # Gir informasjon om brusettene med flest målepunkter
1198 print('\nIdentifikasjon til de seks brusettene med flest målepunkter:')

```

```

1199 print('Bru med flest punkter: Filnavn = {}, Målt med {}, ID på bru i NVDB =
      {}'.format(lengst1[0][0], lengst1[0][13],
1200
                  lengst1[0][1]))
1201 print('Bru med 2. flest punkter: Filnavn = {}, Målt med {}, ID på bru i
      NVDB = {}'.format(lengst2[0][0], lengst2[0][13],
1202
                  lengst2[0][1]))
1203 print('Bru med 3. flest punkter: Filnavn = {}, Målt med {}, ID på bru i
      NVDB = {}'.format(lengst3[0][0], lengst3[0][13],
1204
                  lengst3[0][1]))
1205 print('Bru med 4. flest punkter: Filnavn = {}, Målt med {}, ID på bru i
      NVDB = {}'.format(lengst4[0][0], lengst4[0][13],
1206
                  lengst4[0][1]))
1207 print('Bru med 5. flest punkter: Filnavn = {}, Målt med {}, ID på bru i
      NVDB = {}'.format(lengst5[0][0], lengst5[0][13],
1208
                  lengst5[0][1]))
1209 print('Bru med 6. flest punkter: Filnavn = {}, Målt med {}, ID på bru i
      NVDB = {}'.format(lengst6[0][0], lengst6[0][13],
1210
                  lengst6[0][1]))
1211
1212 # Gir informasjon om brusettene med størst variasjon i friksjon
1213 print('\nIdentifikasjon til de seks brusettene med størst variasjon i
      friksjon:')
1214 print('Brusett 1: Filnavn = {}, Målt med {}, ID på bru i NVDB = {},
      Vinterdriftsklasse = {}'.format(bruSett[0][0][0],
1215
                                      bruSett[0][0][13],
1216
                                      bruSett[0][0][1],
1217
                                      bruSett[0][5][4]))
1218 print('Brusett 2: Filnavn = {}, Målt med {}, ID på bru i NVDB = {},
      Vinterdriftsklasse = {}'.format(bruSett[1][0][0],
1219
                                      bruSett[1][0][13],
1220
                                      bruSett[1][0][1],
1221
                                      bruSett[1][5][4]))
1222 print('Brusett 3: Filnavn = {}, Målt med {}, ID på bru i NVDB = {},
      Vinterdriftsklasse = {}'.format(bruSett[2][0][0],
1223
                                      bruSett[2][0][13],
1224
                                      bruSett[2][0][1],
1225
                                      bruSett[2][5][4]))
1226 print('Brusett 4: Filnavn = {}, Målt med {}, ID på bru i NVDB = {},
      Vinterdriftsklasse = {}'.format(bruSett[3][0][0],
1227
                                      bruSett[3][0][13],
1228

```

```

1229         bruSett[3][0][1],
1230     bruSett[3][5][4]))
1231 print('Brusett 5: Filnavn = {}, Målt med {}, ID på bru i NVDB = {},
1232     Vinterdriftsklasse = {}'.format(bruSett[4][0][0],
1233     bruSett[4][0][13],
1234     bruSett[4][0][1],
1235     bruSett[4][5][4]))
1236 print('Brusett 6: Filnavn = {}, Målt med {}, ID på bru i NVDB = {},
1237     Vinterdriftsklasse = {}'.format(bruSett[5][0][0],
1238     bruSett[5][0][13],
1239     bruSett[5][0][1],
1240     bruSett[5][5][4]))
1241 # Viser en figur over hvordan friksjon relativ til friksjonen på brua
1242     endrer seg med avstand fra brua uavhengig av vinterdriftsklasse
1243 fig = plt.figure(figsize=(12, 6))
1244 fig.suptitle('Endring i totalfriksjon for alle vinterdriftsklasser, n = {}'.
1245     .format(len(friksjonsverdierListeDriftsklasseAlle)), fontsize=20)
1246 fig.subplots_adjust(left=0.06, bottom=0.17, right=0.94, top=0.8, wspace
1247     =0.2, hspace=0.28)
1248 x = avstanderTilBruListeDriftsklasse
1249 y1 = relativFriksjonsverdierMedianListeDriftsklasse
1250 y2 = relativFriksjonsverdierMinListeDriftsklasse
1251 ax1 = fig.add_subplot(121)
1252 ax2 = fig.add_subplot(122)
1253 ax1.set_title('Endring i friksjonsverdi til målepunkter \n relativ til
1254     median friksjonsverdi på bru', fontsize=17)
1255 ax1.scatter(x, y1, alpha=0.1)
1256 ax1.set_ylabel('Endring i friksjon', fontsize=15)
1257 ax1.set_xlabel('Avstand til bru [m]', fontsize=15)
1258 ax1.tick_params(right=True, left=True, bottom=True)
1259 ax2.set_title('Endring i friksjonsverdi til målepunkter \n relativ til
1260     minimum friksjonsverdi på bru', fontsize=17)
1261 ax2.scatter(x, y2, alpha=0.1, c='purple')
1262 ax2.set_ylabel('Endring i friksjon', fontsize=15)
1263 ax2.set_xlabel('Avstand til bru [m]', fontsize=15)
1264 ax2.tick_params(right=True, left=True, bottom=True)
1265 ax1.set_xlim([-0.6, 101])
1266 ax2.set_xlim([-0.6, 101])
1267 ax1.set_ylim([-0.6, 0.6])
1268 ax2.set_ylim([-0.6, 0.6])
1269 tick_spacing = 0.1
1270 ax1.yaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))
1271 ax2.yaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))

```

```

1271
1272 # plt.show()
1273
1274 # Viser en figur over hvordan friksjonsverdiene endrer seg med avstand til
      bru fordelt på de ulike vinterdriftsklassene, Klasse A og B
1275 fig = plt.figure(figsize=(12, 10))
1276 fig.suptitle('Endring i totalfriksjon fordelt på vinterdriftsklasse',
      fontsize=20)
1277 fig.subplots_adjust(left=0.06, bottom=0.1, right=0.94, top=0.9, wspace=0.2,
      hspace=0.28)
1278
1279 x1 = avstandListeDkA
1280 x2 = avstandListeDkB1
1281 x3 = avstandListeDkB2
1282 x4 = avstandListeDkB3
1283
1284 y1 = relativFriksjonsverdierMedianListeDkA
1285 y2 = relativFriksjonsverdierMedianListeDkB1
1286 y3 = relativFriksjonsverdierMedianListeDkB2
1287 y4 = relativFriksjonsverdierMedianListeDkB3
1288
1289 ax = fig.add_subplot(221)
1290 ax2 = fig.add_subplot(222)
1291 ax3 = fig.add_subplot(223)
1292 ax4 = fig.add_subplot(224)
1293
1294 ax.set_title('Vinterdriftsklasse A, n = {}'.format(len(
      friksjonsverdierListeDkA)), fontsize=17)
1295 ax.scatter(x1, y1, alpha=0.1)
1296 ax.set_ylabel('Endring i friksjon', fontsize=15)
1297 ax.set_xlabel('Avstand til bru [m]', fontsize=15)
1298 ax.tick_params(right=True, left=True, bottom=True)
1299
1300 ax2.set_title('Vinterdriftsklasse B, høy, n = {}'.format(len(
      friksjonsverdierListeDkB1)), fontsize=17)
1301 ax2.scatter(x2, y2, alpha=0.1)
1302 ax2.set_ylabel('Endring i friksjon', fontsize=15)
1303 ax2.set_xlabel('Avstand til bru [m]', fontsize=15)
1304 ax2.tick_params(right=True, left=True, bottom=True)
1305
1306 ax3.set_title('Vinterdriftsklasse B, middels, n = {}'.format(len(
      friksjonsverdierListeDkB2)), fontsize=17)
1307 ax3.scatter(x3, y3, alpha=0.1)
1308 ax3.set_ylabel('Endring i friksjon', fontsize=15)
1309 ax3.set_xlabel('Avstand til bru [m]', fontsize=15)
1310 ax3.tick_params(right=True, left=True, bottom=True)
1311
1312 ax4.set_title('Vinterdriftsklasse B, lav, n = {}'.format(len(
      friksjonsverdierListeDkB3)), fontsize=17)
1313 ax4.scatter(x4, y4, alpha=0.1)
1314 ax4.set_ylabel('Endring i friksjon', fontsize=15)
1315 ax4.set_xlabel('Avstand til bru [m]', fontsize=15)
1316 ax4.tick_params(right=True, left=True, bottom=True)
1317
1318 ax.set_xlim([-1, 101])
1319 ax2.set_xlim([-1, 101])
1320 ax3.set_xlim([-1, 101])

```

```

1321 ax4.set_xlim([-1, 101])
1322 ax.set_ylim([-0.6, 0.6])
1323 ax2.set_ylim([-0.6, 0.6])
1324 ax3.set_ylim([-0.6, 0.6])
1325 ax4.set_ylim([-0.6, 0.6])
1326
1327 tick_spacing = 0.1
1328 ax.yaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))
1329 ax2.yaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))
1330 ax3.yaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))
1331 ax4.yaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))
1332
1333 # plt.show()
1334
1335 # Viser en figur over hvordan friksjonsverdiene endrer seg med avstand til
      bru fordelt på de ulike vinterdriftsklassene, Klasse C, D og E
1336 fig = plt.figure(figsize=(12, 10))
1337 fig.suptitle('Endring i totalfriksjon fordelt på vinterdriftsklasse',
      fontsize=20)
1338 fig.subplots_adjust(left=0.06, bottom=0.1, right=0.94, top=0.9, wspace
      =0.51, hspace=0.28)
1339
1340 x5 = avstandListeDkC
1341 x6 = avstandListeDkD
1342 x7 = avstandListeDkE
1343
1344 y5 = relativFriksjonsverdierMedianListeDkC
1345 y6 = relativFriksjonsverdierMedianListeDkD
1346 y7 = relativFriksjonsverdierMedianListeDkE
1347
1348 gs = gridspec.GridSpec(2, 4) # Benytter gridspec fremfor add_subplot for å
      sentrere den tredje grafen i figuren.
1349 ax5 = plt.subplot(gs[0, 0:2])
1350 ax6 = plt.subplot(gs[0, 2:])
1351 ax7 = plt.subplot(gs[1, 1:3])
1352
1353 ax5.set_title('Vinterdriftsklasse C, n = {}'.format(len(
      friksjonsverdierListeDkC)), fontsize=17)
1354 ax5.scatter(x5, y5, alpha=0.1)
1355 ax5.set_ylabel('Endring i friksjon', fontsize=15)
1356 ax5.set_xlabel('Avstand til bru [m]', fontsize=15)
1357 ax5.tick_params(right=True, left=True, bottom=True)
1358
1359 ax6.set_title('Vinterdriftsklasse D, n = {}'.format(len(
      friksjonsverdierListeDkD)), fontsize=17)
1360 ax6.scatter(x6, y6, alpha=0.1)
1361 ax6.set_ylabel('Endring i friksjon', fontsize=15)
1362 ax6.set_xlabel('Avstand til bru [m]', fontsize=15)
1363 ax6.tick_params(right=True, left=True, bottom=True)
1364
1365 ax7.set_title('Vinterdriftsklasse E, n = {}'.format(len(
      friksjonsverdierListeDkE)), fontsize=17)
1366 ax7.scatter(x7, y7, alpha=0.1)
1367 ax7.set_ylabel('Endring i friksjon', fontsize=15)
1368 ax7.set_xlabel('Avstand til bru [m]', fontsize=15)
1369 ax7.tick_params(right=True, left=True, bottom=True)
1370

```

```

1371 ax5.set_xlim([-1, 101])
1372 ax7.set_xlim([-1, 101])
1373 ax6.set_xlim([-1, 101])
1374 ax5.set_ylim([-0.6, 0.6])
1375 ax6.set_ylim([-0.6, 0.6])
1376 ax7.set_ylim([-0.6, 0.6])
1377
1378 tick_spacing = 0.1
1379 ax5.yaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))
1380 ax6.yaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))
1381 ax7.yaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))
1382
1383 # plt.show()
1384
1385 # Lager en figur med temperatur og relativ friksjon
1386 fig = plt.figure(figsize=(12, 6))
1387 fig.suptitle('Endring i totalfriksjon avhengig av temperatur for alle må
    lepunkter, n = {}'.format(
1388     len(relativFriksjonsverdierMedianListe)), fontsize=20)
1389 fig.subplots_adjust(left=0.06, bottom=0.17, right=0.94, top=0.8, wspace
    =0.2, hspace=0.28)
1390
1391 x1 = luftTemperaturListe
1392 x2 = vegTemperaturListe
1393 y = relativFriksjonsverdierMedianListe
1394
1395 ax = fig.add_subplot(121)
1396 ax2 = fig.add_subplot(122)
1397
1398 ax.set_title('Endring i friksjonsverdi avhengig av \n lufttemperatur',
    fontsize=17)
1399 ax.scatter(x1, y, alpha=0.1)
1400 ax.set_ylabel('Endring i friksjon', fontsize=15)
1401 ax.set_xlabel('Lufttemperatur [\'u\'u2103\']', fontsize=15)
1402 ax.tick_params(right=True, left=True, bottom=True)
1403
1404 ax2.set_title('Endring i friksjonsverdi avhengig av \n
    vegoverflatetemperatur', fontsize=17)
1405 ax2.scatter(x2, y, alpha=0.1)
1406 ax2.set_ylabel('Endring i friksjon', fontsize=15)
1407 ax2.set_xlabel('Vegoverflatetemperatur [\'u\'u2103\']', fontsize=15)
1408 ax2.tick_params(right=True, left=True, bottom=True)
1409
1410 ax.set_xlim([-25, 20])
1411 ax2.set_xlim([-25, 20])
1412 ax.set_ylim([-0.6, 0.6])
1413 ax2.set_ylim([-0.6, 0.6])
1414
1415 tick_spacing = 0.1
1416 ax1.yaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))
1417 ax2.yaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))
1418
1419 # plt.show()
1420
1421 # Lager en figur med temperatur og målt friksjon på bru
1422 fig = plt.figure(figsize=(12, 6))
1423 fig.suptitle('Totalfriksjonsverdi på bru avhengig av temperatur målt på bru

```

```

    , n = {}'.format(len(friksjonsverdierPaaBruListe)),
1424         fontsize=20)
1425 fig.subplots_adjust(left=0.06, bottom=0.17, right=0.94, top=0.8, wspace
        =0.2, hspace=0.28)
1426
1427 x1 = luftTemperaturPaaBruListe
1428 x2 = vegTemperaturPaaBruListe
1429 y = friksjonsverdierPaaBruListe
1430
1431 ax = fig.add_subplot(121)
1432 ax2 = fig.add_subplot(122)
1433
1434 ax.set_title('Friksjonsverdi på bru avhengig av \n lufttemperatur',
        fontsize=17)
1435 ax.scatter(x1, y, alpha=0.1, c='deepskyblue')
1436 ax.set_ylabel('Friksjonsverdi', fontsize=15)
1437 ax.set_xlabel('Lufttemperatur [\'u\'u2103\']', fontsize=15)
1438 ax.tick_params(right=True, left=True, bottom=True)
1439
1440 ax2.set_title('Friksjonsverdi på bru avhengig av \n vegovertemperatur',
        fontsize=17)
1441 ax2.scatter(x2, y, alpha=0.1, c='darkred')
1442 ax2.set_ylabel('Friksjonsverdi', fontsize=15)
1443 ax2.set_xlabel('Vegoverflatetemperatur [\'u\'u2103\']', fontsize=15)
1444 ax2.tick_params(right=True, left=True, bottom=True)
1445
1446 ax.set_xlim([-25, 20])
1447 ax2.set_xlim([-25, 20])
1448
1449 tick_spacing = 0.1
1450 ax.yaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))
1451 ax2.yaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))
1452
1453 # plt.show()
1454
1455 # Lager en figur som viser hvordan vegoverflatetemperaturen endres med
        avstand til bru
1456 fig = plt.figure(figsize=(12, 6))
1457 fig.suptitle(
1458     'Endring i vegoverflatetemperatur med avstand til bru sammenlignet \n
        med median vegoverflatetemperatur på bru, n = {}'.format(
1459         len(relativVegTemperaturListe)), fontsize=20)
1460 fig.subplots_adjust(left=0.06, bottom=0.17, right=0.94, top=0.8, wspace
        =0.2, hspace=0.28)
1461
1462 gs = gridspec.GridSpec(1, 4) # Benytter gridspec fremfor add_subplot for å
        sentrere den tredje grafen i figuren.
1463 ax = plt.subplot(gs[0, 1:3])
1464 ax.set_ylabel('Endring i vegoverflatetemperatur [\'u\'u2103\']', fontsize
        =15)
1465 ax.set_xlabel('Avstand til bru[m]', fontsize=15)
1466 ax.scatter(avstanderTilBruListe, relativVegTemperaturListe, alpha=0.1, c='
        darkred')
1467 ax.tick_params(right=True, left=True, bottom=True)
1468
1469 tick_spacing = 1
1470 ax.yaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))

```

```

1471
1472 # plt.show()
1473
1474 # Lager en figur som viser hvordan friksjonsverdi varierer med
      luftfuktighet
1475 fig = plt.figure(figsize=(12, 6))
1476 fig.suptitle('Totalfriksjonsverdi avhengig av luftfuktighet', fontsize=20)
1477 fig.subplots_adjust(left=0.06, bottom=0.17, right=0.94, top=0.8, wspace
      =0.2, hspace=0.28)
1478
1479 x1_1 = luftfuktighetLavTemperaturPaaBruListe
1480 y1_1 = luftfuktighetLavTemperaturPaaBruFriksjonsverdierListe
1481 x1_2 = luftfuktighetHoyTemperaturPaaBruListe
1482 y1_2 = luftfuktighetHoyTemperaturPaaBruFriksjonsverdierListe
1483 x2_1 = luftfuktighetLavTemperaturUtenforBruListe
1484 y2_1 = luftfuktighetLavTemperaturUtenforBruFriksjonsverdierListe
1485 x2_2 = luftfuktighetHoyTemperaturUtenforBruListe
1486 y2_2 = luftfuktighetHoyTemperaturUtenforBruFriksjonsverdierListe
1487
1488 z1 = np.polyfit(x1_1, y1_1, 1)
1489 z2 = np.polyfit(x2_1, y2_1, 1)
1490 p1 = np.poly1d(z1)
1491 p2 = np.poly1d(z2)
1492
1493 ax = fig.add_subplot(121)
1494 ax2 = fig.add_subplot(122)
1495
1496 ax.set_title('Friksjonsverdi på bru avhengig av \n luftfuktighet, n = {}'.
      format(len(x1_1) + len(x1_2)), fontsize=17)
1497 ax.scatter(x1_1, y1_1, c='cyan', label='T'u'\u2103'' 'u'\u2264'' 0'u'\u2103
      ')
1498 ax.scatter(x1_2, y1_2, c='red', label='T'u'\u2103'' > 0'u'\u2103')
1499 ax.plot(x1_1, p1(x1_1), c='black', label='Trend')
1500 ax.set_ylabel('Friksjonsverdi', fontsize=15)
1501 ax.set_xlabel('Luftfuktighet [%]', fontsize=15)
1502 ax.tick_params(right=True, left=True, bottom=True)
1503
1504 ax2.set_title('Friksjonsverdi utenfor bru avhengig av \n luftfuktighet, n =
      {}'.format(len(x2_1) + len(x2_2)),
      fontsize=17)
1505
1506 ax2.scatter(x2_1, y2_1, c='cyan')
1507 ax2.scatter(x2_2, y2_2, c='red')
1508 ax2.plot(x2_1, p2(x2_1), c='black')
1509 ax2.set_ylabel('Friksjonsverdi', fontsize=15)
1510 ax2.set_xlabel('Luftfuktighet [%]', fontsize=15)
1511 ax2.tick_params(right=True, left=True, bottom=True)
1512
1513 ax.set_ylim([0, 0.6])
1514 ax2.set_ylim([0, 0.6])
1515
1516 fig.legend(loc=8, ncol=3, fontsize=15)
1517
1518 # plt.show()
1519
1520 # Lager en figur som viser de seks bruene med flest målepunkter og hvordan
      friksjonen varierer for dem
1521 fig = plt.figure(figsize=(12, 12))

```

```

1522 fig.suptitle('Utvalg 1: Datasett med flest målepunkter og deres
      totalfriksjonsverdier', fontsize=20)
1523 fig.subplots_adjust(left=0.06, bottom=0.11, right=0.94, top=0.9, wspace
      =0.2, hspace=0.5)
1524
1525 ax = fig.add_subplot(321)
1526 ax2 = fig.add_subplot(322)
1527 ax3 = fig.add_subplot(323)
1528 ax4 = fig.add_subplot(324)
1529 ax5 = fig.add_subplot(325)
1530 ax6 = fig.add_subplot(326)
1531
1532 ax.set_title('Sett1, brulengde = {}m'.format(lengst1[0][10]).replace(".", "
      ,"), fontsize=17)
1533 ax.set_ylabel('Friksjonsverdi', fontsize=15)
1534 ax.set_xlabel('Nummer på målepunkt', fontsize=15)
1535 ax.scatter(xAkseLengst1, yAkseLengst1, c='silver', label='Målepunkt utenfor
      bru')
1536 ax.scatter(xAkseBruLengst1, yAkseBruLengst1, c='coral', label='Målepunkt på
      bru')
1537 ax.tick_params(right=True, left=True, bottom=True)
1538
1539 ax2.set_title('Sett2, brulengde = {}m'.format(lengst2[0][10]).replace(".", "
      ,"), fontsize=17)
1540 ax2.set_ylabel('Friksjonsverdi', fontsize=15)
1541 ax2.set_xlabel('Nummer på målepunkt', fontsize=15)
1542 ax2.scatter(xAkseLengst2, yAkseLengst2, c='silver')
1543 ax2.scatter(xAkseBruLengst2, yAkseBruLengst2, c='coral')
1544 ax2.tick_params(right=True, left=True, bottom=True)
1545
1546 ax3.set_title('Sett3, brulengde = {}m'.format(lengst3[0][10]).replace(".", "
      ,"), fontsize=17)
1547 ax3.set_ylabel('Friksjonsverdi', fontsize=15)
1548 ax3.set_xlabel('Nummer på målepunkt', fontsize=15)
1549 ax3.scatter(xAkseLengst3, yAkseLengst3, c='silver')
1550 ax3.scatter(xAkseBruLengst3, yAkseBruLengst3, c='coral')
1551 ax3.tick_params(right=True, left=True, bottom=True)
1552
1553 ax4.set_title('Sett4, brulengde = {}m'.format(lengst4[0][10]).replace(".", "
      ,"), fontsize=17)
1554 ax4.set_ylabel('Friksjonsverdi', fontsize=15)
1555 ax4.set_xlabel('Nummer på målepunkt', fontsize=15)
1556 ax4.scatter(xAkseLengst4, yAkseLengst4, c='silver')
1557 ax4.scatter(xAkseBruLengst4, yAkseBruLengst4, c='coral')
1558 ax4.tick_params(right=True, left=True, bottom=True)
1559
1560 ax5.set_title('Sett5, brulengde = {}m'.format(lengst5[0][10]).replace(".", "
      ,"), fontsize=17)
1561 ax5.set_ylabel('Friksjonsverdi', fontsize=15)
1562 ax5.set_xlabel('Nummer på målepunkt', fontsize=15)
1563 ax5.scatter(xAkseLengst5, yAkseLengst5, c='silver')
1564 ax5.scatter(xAkseBruLengst5, yAkseBruLengst5, c='coral')
1565 ax5.tick_params(right=True, left=True, bottom=True)
1566
1567 ax6.set_title('Sett6, brulengde = {}m'.format(lengst6[0][10]).replace(".", "
      ,"), fontsize=17)
1568 ax6.set_ylabel('Friksjonsverdi', fontsize=15)

```

```

1569 ax6.set_xlabel('Nummer på målepunkt', fontsize=15)
1570 ax6.scatter(xAkseLengst6, yAkseLengst6, c='silver')
1571 ax6.scatter(xAkseBruLengst6, yAkseBruLengst6, c='coral')
1572 ax6.tick_params(right=True, left=True, bottom=True)
1573
1574 ax.set_ylim([0, 0.6])
1575 ax2.set_ylim([0, 0.6])
1576 ax3.set_ylim([0, 0.6])
1577 ax4.set_ylim([0, 0.6])
1578 ax5.set_ylim([0, 0.6])
1579 ax6.set_ylim([0, 0.6])
1580
1581 fig.legend(loc=8, ncol=2, fontsize=15)
1582
1583 # plt.show()
1584
1585 # Lager en figur som viser de seks bruene med størst variasjon i friksjon p
    å og rundt bru
1586 fig = plt.figure(figsize=(12, 15))
1587 fig.suptitle('Utvalg 2: Datasett med størst variasjon i totalfriksjon målt
    på og utenfor bru', fontsize=20)
1588 fig.subplots_adjust(left=0.06, bottom=0.09, right=0.94, top=0.92, wspace
    =0.32, hspace=0.35)
1589
1590 ax1 = fig.add_subplot(321)
1591 ax2 = fig.add_subplot(322)
1592 ax3 = fig.add_subplot(323)
1593 ax4 = fig.add_subplot(324)
1594 ax5 = fig.add_subplot(325)
1595 ax6 = fig.add_subplot(326)
1596
1597 ax1.set_title('Sett1, brulengde = {}m'.format(bruSett[0][0][10]).replace(".",
    ", "), fontsize=17)
1598 ax1.set_ylabel('Friksjonsverdi', fontsize=15)
1599 ax1.set_xlabel('Nummer på målepunkt', fontsize=15)
1600 ax1.scatter(brusetTall[0], bruSettFriksjon[0], c='silver', label='Må
    lepunkt utenfor bru')
1601 ax1.scatter(brusetTallBru[0], bruSettFriksjonBru[0], c='coral', label='Må
    lepunkt på bru')
1602 ax1.tick_params(axis='y', bottom=True)
1603 ax1_1 = ax1.twinx()
1604 ax1_1.plot(brusetTall[0], bruSettVegoverflate[0], c='darkred', label='T'u
    \u2103')
1605 ax1_1.set_ylabel('Tetemperatur [\'u\' \u2103\']', c='darkred', fontsize=15)
1606 ax1_1.tick_params(axis='y', labelcolor='darkred')
1607
1608 ax2.set_title('Sett2, brulengde = {}m'.format(bruSett[1][0][10]).replace(".",
    ", "), fontsize=17)
1609 ax2.set_ylabel('Friksjonsverdi', fontsize=15)
1610 ax2.set_xlabel('Nummer på målepunkt', fontsize=15)
1611 ax2.scatter(brusetTall[1], bruSettFriksjon[1], c='silver')
1612 ax2.scatter(brusetTallBru[1], bruSettFriksjonBru[1], c='coral')
1613 ax2.tick_params(axis='y', bottom=True)
1614 ax2_1 = ax2.twinx()
1615 ax2_1.plot(brusetTall[1], bruSettVegoverflate[1], c='darkred')
1616 ax2_1.set_ylabel('Tetemperatur [\'u\' \u2103\']', c='darkred', fontsize=15)
1617 ax2_1.tick_params(axis='y', labelcolor='darkred')

```

```

1618
1619 ax3.set_title('Sett3, brulengde = {}m'.format(bruSett[2][0][10]).replace(".",
      ", ", ","), fontsize=17)
1620 ax3.set_ylabel('Friksjonsverdi', fontsize=15)
1621 ax3.set_xlabel('Nummer på målepunkt', fontsize=15)
1622 ax3.scatter(brusetTall[2], bruSettFriksjon[2], c='silver')
1623 ax3.scatter(brusetTallBru[2], bruSettFriksjonBru[2], c='coral')
1624 ax3.tick_params(axis='y', bottom=True)
1625 ax3_1 = ax3.twinx()
1626 ax3_1.plot(brusetTall[2], bruSettVegoverflate[2], c='darkred')
1627 ax3_1.set_ylabel('Tetemperatur [°u\u2103]', c='darkred', fontsize=15)
1628 ax3_1.tick_params(axis='y', labelcolor='darkred')
1629
1630 ax4.set_title('Sett4, brulengde = {}m'.format(bruSett[3][0][10]).replace(".",
      ", ", ","), fontsize=17)
1631 ax4.set_ylabel('Friksjonsverdi', fontsize=15)
1632 ax4.set_xlabel('Nummer på målepunkt', fontsize=15)
1633 ax4.scatter(brusetTall[3], bruSettFriksjon[3], c='silver')
1634 ax4.scatter(brusetTallBru[3], bruSettFriksjonBru[3], c='coral')
1635 ax4.tick_params(axis='y', bottom=True)
1636 ax4_1 = ax4.twinx()
1637 ax4_1.plot(brusetTall[3], bruSettVegoverflate[3], c='darkred')
1638 ax4_1.set_ylabel('Tetemperatur [°u\u2103]', c='darkred', fontsize=15)
1639 ax4_1.tick_params(axis='y', labelcolor='darkred')
1640
1641 ax5.set_title('Sett5, brulengde = {}m'.format(bruSett[4][0][10]).replace(".",
      ", ", ","), fontsize=17)
1642 ax5.set_ylabel('Friksjonsverdi', fontsize=15)
1643 ax5.set_xlabel('Nummer på målepunkt', fontsize=15)
1644 ax5.scatter(brusetTall[4], bruSettFriksjon[4], c='silver')
1645 ax5.scatter(brusetTallBru[4], bruSettFriksjonBru[4], c='coral')
1646 ax5.tick_params(axis='y', bottom=True)
1647 ax5_1 = ax5.twinx()
1648 ax5_1.plot(brusetTall[4], bruSettVegoverflate[4], c='darkred')
1649 ax5_1.set_ylabel('Tetemperatur [°u\u2103]', c='darkred', fontsize=15)
1650 ax5_1.tick_params(axis='y', labelcolor='darkred')
1651
1652 ax6.set_title('Sett6, brulengde = {}m'.format(bruSett[5][0][10]).replace(".",
      ", ", ","), fontsize=17)
1653 ax6.set_ylabel('Friksjonsverdi', fontsize=15)
1654 ax6.set_xlabel('Nummer på målepunkt', fontsize=15)
1655 ax6.scatter(brusetTall[5], bruSettFriksjon[5], c='silver')
1656 ax6.scatter(brusetTallBru[5], bruSettFriksjonBru[5], c='coral')
1657 ax6.tick_params(axis='y', bottom=True)
1658 ax6_1 = ax6.twinx()
1659 ax6_1.plot(brusetTall[5], bruSettVegoverflate[5], c='darkred')
1660 ax6_1.set_ylabel('Tetemperatur [°u\u2103]', c='darkred', fontsize=15)
1661 ax6_1.tick_params(axis='y', labelcolor='darkred')
1662
1663 ax1.set_ylim([0, 0.6])
1664 ax2.set_ylim([0, 0.6])
1665 ax3.set_ylim([0, 0.6])
1666 ax4.set_ylim([0, 0.6])
1667 ax5.set_ylim([0, 0.6])
1668 ax6.set_ylim([0, 0.6])
1669 ax1_1.set_ylim([-5, 3])
1670 ax2_1.set_ylim([-5, 3])

```

```

1671 ax3_1.set_ylim([-5, 3])
1672 ax4_1.set_ylim([-5, 3])
1673 ax5_1.set_ylim([-5, 3])
1674 ax6_1.set_ylim([-5, 3])
1675 tick_spacing1 = 1
1676 tick_spacing2 = 2
1677 ax1.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing2))
1678 ax2.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing2))
1679 ax3.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing2))
1680 ax4.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing1))
1681 ax5.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing2))
1682 ax6.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing2))
1683 fig.legend(loc=8, ncol=3, fontsize=15)
1684
1685 # plt.show()
1686
1687
1688 # Densitetsfordeling til friksjonsverdiene
1689 fig = plt.figure(figsize=(9, 6))
1690 fig.suptitle('Densitetsfordeling til målepunktene totalfriksjonsverdier',
1691             fontsize=20)
1692 fig.subplots_adjust(left=0.08, bottom=0.19, right=0.94, top=0.9, wspace
1693                    =0.32, hspace=0.35)
1694 ax = fig.add_subplot(111)
1695
1696 x_verdier = np.linspace(0, 0.7)
1697 densitet1 = gaussian_kde(friksjonsverdierMatriseDkAlle)
1698 densitet1.covariance_factor = lambda: .5
1699 densitet1._compute_covariance()
1700 densitet2 = gaussian_kde(friksjonsverdierMatriseDkAlleBru)
1701 densitet2.covariance_factor = lambda: .5
1702 densitet2._compute_covariance()
1703
1704 ax.plot(x_verdier, densitet1(x_verdier), label='Alle')
1705 ax.plot(x_verdier, densitet2(x_verdier), label='Kun bru')
1706 ax.set_ylabel('Densitet', fontsize=15)
1707 ax.set_xlabel('Friksjonsverdi', fontsize=15)
1708 tick_spacing = 0.05
1709 ax.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))
1710 ax.set_xlim([0, 0.7])
1711 ax.set_ylim([0, 5])
1712 ax.tick_params(right=True, left=True, bottom=True)
1713 fig.legend(loc=8, ncol=2, fontsize=15)
1714
1715 # plt.show()
1716
1717 # Gjennomsnitt og standardavvik til ulike vinterdriftsklasser
1718 fig = plt.figure(figsize=(9, 6))
1719 fig.suptitle('Gjennomsnitt og standardavvik til totalfriksjonsverdier \n
1720             avhengig av vinterdriftsklasse på og utenfor bru', fontsize=20)
1721 fig.subplots_adjust(left=0.08, bottom=0.19, right=0.94, top=0.85, wspace
1722                    =0.32, hspace=0.35)
1723 ax = fig.add_subplot(111)
1724
1725 x = np.arange(8)
1726 bredde = 0.4

```

```

1724
1725 plt.bar(x-0.2, gjennomsnittBru, bredde, yerr=standardavvikBru, color='coral
      ', align='center', ecolor='black', capsiz=10, label='Bru')
1726 plt.bar(x+0.2, gjennomsnittUtenfor, bredde, yerr=standardavvikUtenfor,
      color='silver', align='center', ecolor='black', capsiz=10, label='
      Utenfor')
1727 plt.xticks(x, ['Alle', 'A', 'B, høy', 'B, middels', 'B, lav', 'C', 'D', 'E'
      ])
1728 plt.xlabel("Vinterdriftsklasser", fontsize=15)
1729 plt.ylabel("Friksjonsverdi", fontsize=15)
1730 fig.legend(loc=8, ncol=2, fontsize=15)
1731
1732 # plt.show()
1733
1734 # Densitetsfordeling til friksjonsverdiene
1735 fig = plt.figure(figsize=(9, 6))
1736 fig.suptitle('Densitetsfordeling til målepunktene totalfriksjonsverdier',
      fontsize=20)
1737 fig.subplots_adjust(left=0.08, bottom=0.19, right=0.94, top=0.9, wspace
      =0.32, hspace=0.35)
1738 ax = fig.add_subplot(111)
1739
1740 x_verdier = np.linspace(0, 0.7)
1741 densitet1 = gaussian_kde(friksjonsverdierMatriseDkAlle)
1742 densitet1.covariance_factor = lambda: .5
1743 densitet1._compute_covariance()
1744 densitet2 = gaussian_kde(friksjonsverdierMatriseDkAlleBru)
1745 densitet2.covariance_factor = lambda: .5
1746 densitet2._compute_covariance()
1747
1748 ax.plot(x_verdier, densitet1(x_verdier), label='Alle')
1749 ax.plot(x_verdier, densitet2(x_verdier), label='Kun bru')
1750 ax.set_ylabel('Densitet', fontsize=15)
1751 ax.set_xlabel('Friksjonsverdi', fontsize=15)
1752 tick_spacing = 0.05
1753 ax.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))
1754 ax.set_xlim([0, 0.7])
1755 ax.set_ylim([0, 5])
1756 ax.tick_params(right=True, left=True, bottom=True)
1757 fig.legend(loc=8, ncol=2, fontsize=15)
1758
1759 # plt.show()
1760
1761
1762 # Gjennomsnitt og standardavvik til ulike vinterdriftsklasser
1763 fig = plt.figure(figsize=(9, 6))
1764 fig.suptitle('Gjennomsnitt og standardavvik til totalfriksjonsverdier \n
      avhengig av vinterdriftsklasse på og utenfor bru', fontsize=20)
1765 fig.subplots_adjust(left=0.08, bottom=0.19, right=0.94, top=0.85, wspace
      =0.32, hspace=0.35)
1766 ax = fig.add_subplot(111)
1767
1768 x = np.arange(8)
1769 bredde = 0.4
1770
1771 plt.bar(x-0.2, gjennomsnittBru, bredde, yerr=standardavvikBru, color='coral
      ', align='center', ecolor='black', capsiz=10, label='Bru')

```

```

1772 plt.bar(x+0.2, gjennomsnittUtenfor, bredde, yerr=standardavvikUtenfor,
           color='silver', align='center', ecolor='black', capsiz=10, label='
           Utenfor')
1773 plt.xticks(x, ['Alle', 'A', 'B, høy', 'B, middels', 'B, lav', 'C', 'D', 'E'
                  ])
1774 plt.xlabel("Vinterdriftsklasser", fontsize=15)
1775 plt.ylabel("Friksjonsverdi", fontsize=15)
1776 fig.legend(loc=8, ncol=2, fontsize=15)
1777
1778 # plt.show()
1779
1780 fig = plt.figure(figsize=(12, 14))
1781 fig.suptitle('Densitetsfordeling til totalfriksjonsverdiene for ulike
               vinterdriftsklasser', fontsize=20)
1782 fig.subplots_adjust(left=0.06, bottom=0.09, right=0.94, top=0.92, wspace
                       =0.23, hspace=0.43)
1783
1784 ax1 = fig.add_subplot(421)
1785 ax1.set_title('Vinterdriftsklasse A', fontsize=17)
1786 densitetUtenfor = gaussian_kde(friksjonsverdierMatriseDkAUtenfor)
1787 densitetBru = gaussian_kde(friksjonsverdierMatriseDkABru)
1788 x_verdier = np.linspace(0, 0.7)
1789 densitetUtenfor.covariance_factor = lambda: .5
1790 densitetUtenfor._compute_covariance()
1791 densitetBru._compute_covariance()
1792 ax1.plot(x_verdier, densitetUtenfor(x_verdier), c='grey', label='Målepunkt
           utenfor bru')
1793 ax1.plot(x_verdier, densitetBru(x_verdier), c='coral', label='Målepunkt på
           bru')
1794 ax1.set_ylabel('Densitet', fontsize=15)
1795 ax1.set_xlabel('Friksjonsverdi', fontsize=15)
1796 tick_spacing1 = 0.1
1797 tick_spacing2 = 0.05
1798 ax1.set_xlim([0, 0.7])
1799 ax1.set_ylim([0, 12])
1800 ax1.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing1))
1801 ax1.xaxis.set_minor_locator(ticker.MultipleLocator(tick_spacing2))
1802 ax1.xaxis.grid(True, which='minor')
1803 ax1.grid()
1804
1805 ax2 = fig.add_subplot(422)
1806 ax2.set_title('Vinterdriftsklasse B, høy', fontsize=17)
1807 densitetUtenfor = gaussian_kde(friksjonsverdierMatriseDkB1Utenfor)
1808 densitetBru = gaussian_kde(friksjonsverdierMatriseDkB1Bru)
1809 x_verdier = np.linspace(0, 0.7)
1810 densitetUtenfor.covariance_factor = lambda: .5
1811 densitetUtenfor._compute_covariance()
1812 densitetBru._compute_covariance()
1813 ax2.plot(x_verdier, densitetUtenfor(x_verdier), c='grey')
1814 ax2.plot(x_verdier, densitetBru(x_verdier), c='coral')
1815 ax2.set_ylabel('Densitet', fontsize=15)
1816 ax2.set_xlabel('Friksjonsverdi', fontsize=15)
1817 tick_spacing1 = 0.1
1818 tick_spacing2 = 0.05
1819 ax2.set_xlim([0, 0.7])
1820 ax2.set_ylim([0, 12])
1821 ax2.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing1))

```

```

1822 ax2.xaxis.set_minor_locator(ticker.MultipleLocator(tick_spacing2))
1823 ax2.xaxis.grid(True, which='minor')
1824 ax2.grid()
1825
1826 ax3 = fig.add_subplot(423)
1827 ax3.set_title('Vinterdriftsklasse B, middels', fontsize=17)
1828 densitetUtenfor = gaussian_kde(friksjonsverdierMatriseDkB2Utenfor)
1829 densitetBru = gaussian_kde(friksjonsverdierMatriseDkB2Bru)
1830 x_verdier = np.linspace(0, 0.7)
1831 densitetUtenfor.covariance_factor = lambda: .5
1832 densitetUtenfor._compute_covariance()
1833 densitetBru._compute_covariance()
1834 ax3.plot(x_verdier, densitetUtenfor(x_verdier), c='grey')
1835 ax3.plot(x_verdier, densitetBru(x_verdier), c='coral')
1836 ax3.set_ylabel('Densitet', fontsize=15)
1837 ax3.set_xlabel('Friksjonsverdi', fontsize=15)
1838 tick_spacing1 = 0.1
1839 tick_spacing2 = 0.05
1840 ax3.set_xlim([0, 0.7])
1841 ax3.set_ylim([0, 12])
1842 ax3.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing1))
1843 ax3.xaxis.set_minor_locator(ticker.MultipleLocator(tick_spacing2))
1844 ax3.xaxis.grid(True, which='minor')
1845 ax3.grid()
1846
1847 ax4 = fig.add_subplot(424)
1848 ax4.set_title('Vinterdriftsklasse B, lav', fontsize=17)
1849 densitetUtenfor = gaussian_kde(friksjonsverdierMatriseDkB3Utenfor)
1850 densitetBru = gaussian_kde(friksjonsverdierMatriseDkB3Bru)
1851 x_verdier = np.linspace(0, 0.7)
1852 densitetUtenfor.covariance_factor = lambda: .5
1853 densitetUtenfor._compute_covariance()
1854 densitetBru._compute_covariance()
1855 ax4.plot(x_verdier, densitetUtenfor(x_verdier), c='grey')
1856 ax4.plot(x_verdier, densitetBru(x_verdier), c='coral')
1857 ax4.set_ylabel('Densitet', fontsize=15)
1858 ax4.set_xlabel('Friksjonsverdi', fontsize=15)
1859 tick_spacing1 = 0.1
1860 tick_spacing2 = 0.05
1861 ax4.set_xlim([0, 0.7])
1862 ax4.set_ylim([0, 12])
1863 ax4.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing1))
1864 ax4.xaxis.set_minor_locator(ticker.MultipleLocator(tick_spacing2))
1865 ax4.xaxis.grid(True, which='minor')
1866 ax4.grid()
1867
1868 ax5 = fig.add_subplot(425)
1869 ax5.set_title('Vinterdriftsklasse C', fontsize=17)
1870 densitetUtenfor = gaussian_kde(friksjonsverdierMatriseDkCUtenfor)
1871 densitetBru = gaussian_kde(friksjonsverdierMatriseDkCBru)
1872 x_verdier = np.linspace(0, 0.7)
1873 densitetUtenfor.covariance_factor = lambda: .5
1874 densitetUtenfor._compute_covariance()
1875 densitetBru._compute_covariance()
1876 ax5.plot(x_verdier, densitetUtenfor(x_verdier), c='grey')
1877 ax5.plot(x_verdier, densitetBru(x_verdier), c='coral')
1878 ax5.set_ylabel('Densitet', fontsize=15)

```

```

1879 ax5.set_xlabel('Friksjonsverdi', fontsize=15)
1880 tick_spacing1 = 0.1
1881 tick_spacing2 = 0.05
1882 ax5.set_xlim([0, 0.7])
1883 ax5.set_ylim([0, 12])
1884 ax5.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing1))
1885 ax5.xaxis.set_minor_locator(ticker.MultipleLocator(tick_spacing2))
1886 ax5.xaxis.grid(True, which='minor')
1887 ax5.grid()
1888
1889 ax6 = fig.add_subplot(426)
1890 ax6.set_title('Vinterdriftsklasse D', fontsize=17)
1891 densitetUtenfor = gaussian_kde(friksjonsverdierMatriseDkDUtenfor)
1892 densitetBru = gaussian_kde(friksjonsverdierMatriseDkDBru)
1893 x_verdier = np.linspace(0, 0.7)
1894 densitetUtenfor.covariance_factor = lambda: .5
1895 densitetUtenfor._compute_covariance()
1896 densitetBru._compute_covariance()
1897 ax6.plot(x_verdier, densitetUtenfor(x_verdier), c='grey')
1898 ax6.plot(x_verdier, densitetBru(x_verdier), c='coral')
1899 ax6.set_ylabel('Densitet', fontsize=15)
1900 ax6.set_xlabel('Friksjonsverdi', fontsize=15)
1901 tick_spacing1 = 0.1
1902 tick_spacing2 = 0.05
1903 ax6.set_xlim([0, 0.7])
1904 ax6.set_ylim([0, 12])
1905 ax6.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing1))
1906 ax6.xaxis.set_minor_locator(ticker.MultipleLocator(tick_spacing2))
1907 ax6.xaxis.grid(True, which='minor')
1908 ax6.grid()
1909
1910 ax7 = fig.add_subplot(427)
1911 ax7.set_title('Vinterdriftsklasse E', fontsize=17)
1912 densitetUtenfor = gaussian_kde(friksjonsverdierMatriseDkEUtenfor)
1913 densitetBru = gaussian_kde(friksjonsverdierMatriseDkEBru)
1914 x_verdier = np.linspace(0, 0.7)
1915 densitetUtenfor.covariance_factor = lambda: .5
1916 densitetUtenfor._compute_covariance()
1917 densitetBru._compute_covariance()
1918 ax7.plot(x_verdier, densitetUtenfor(x_verdier), c='grey')
1919 ax7.plot(x_verdier, densitetBru(x_verdier), c='coral')
1920 ax7.set_ylabel('Densitet', fontsize=15)
1921 ax7.set_xlabel('Friksjonsverdi', fontsize=15)
1922 tick_spacing1 = 0.1
1923 tick_spacing2 = 0.05
1924 ax7.set_xlim([0, 0.7])
1925 ax7.set_ylim([0, 12])
1926 ax7.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing1))
1927 ax7.xaxis.set_minor_locator(ticker.MultipleLocator(tick_spacing2))
1928 ax7.xaxis.grid(True, which='minor')
1929 ax7.grid()
1930
1931 ax8 = fig.add_subplot(428)
1932 ax8.set_title('Uavhengig av vinterdriftsklasse', fontsize=17)
1933 densitetUtenfor = gaussian_kde(friksjonsverdierMatriseDkAlleUtenfor)
1934 densitetBru = gaussian_kde(friksjonsverdierMatriseDkAlleBru)
1935 x_verdier = np.linspace(0, 0.7)

```

```
1936 densitetUtenfor.covariance_factor = lambda: .5
1937 densitetUtenfor._compute_covariance()
1938 densitetBru._compute_covariance()
1939 ax8.plot(x_verdier, densitetUtenfor(x_verdier), c='grey')
1940 ax8.plot(x_verdier, densitetBru(x_verdier), c='coral')
1941 ax8.set_ylabel('Densitet', fontsize=15)
1942 ax8.set_xlabel('Friksjonsverdi', fontsize=15)
1943 tick_spacing1 = 0.1
1944 tick_spacing2 = 0.05
1945 ax8.set_xlim([0, 0.7])
1946 ax8.set_ylim([0, 12])
1947 ax8.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing1))
1948 ax8.xaxis.set_minor_locator(ticker.MultipleLocator(tick_spacing2))
1949 ax8.xaxis.grid(True, which='minor')
1950 ax8.grid()
1951
1952 fig.legend(loc=8, ncol=2, fontsize=15)
1953
1954 plt.show()
```

Vedlegg B7: Gaulabrua

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import matplotlib.gridspec as gridspec
4 import matplotlib.ticker as ticker
5 import scipy.stats
6 from scipy.stats import gaussian_kde
7
8 # Oppgi sti til mappen hvor matrisen ligger lagret
9 sti = 'C:\\Programming\\'
10 innSti = sti + 'Python\\BruMatrise.npy'
11
12 # Kolonne 0 = Filnavn
13 # Kolonne 1 = Bruas ID i NVDB
14 # Kolonne 2 = Målepunktets avstand til brua [m]
15 # Kolonne 3 = Målepunktets friksjonsverdi
16 # Kolonne 4 = Vegens vinterdriftsklasse ved målepunktet
17 # Kolonne 5 = Lufttemperatur [Celsius]
18 # Kolonne 6 = Vegoverflatetemperatur [Celsius]
19 # Kolonne 7 = Luftfuktighet [%] OBS, ofte utilgjengelig
20 # Kolonne 8 = Brutype
21 # Kolonne 9 = Brumateriale
22 # Kolonne 10 = Bruas totale lengde [m]
23 # Kolonne 11 = Bruas byggeår. OBS, ofte utilgjengelig
24 # Kolonne 12 = Vegens horisontalradius [m]
25 # Kolonne 13 = Utstyr brukt under målingen
26 # Kolonne 14 = Vegens referanse (navn)
27 # Kolonne 15 = Fylke
28 # Kolonne 16 = Året målingen ble utført
29 # Kolonne 17 = Månedens målingen ble utført
30 # Kolonne 18 = Dagen målingen ble utført
31 # Kolonne 19 = Klokkeslett målingen ble utført
32
33 matrise = np.load(innSti) # Laster inn matrisen
34
35 # Splitter matrisen med all data opp i mindre matriser hvor hver fil har
    sin matrise
36 unikeFilnavnIndekser = np.unique(matrise[:, 0], True)[1]
37 listeFilerMatriser = []
38 for i in range(len(unikeFilnavnIndekser)):
39     if i < len(unikeFilnavnIndekser) - 1:
40         a = unikeFilnavnIndekser[i]
41         b = unikeFilnavnIndekser[i + 1]
42         listeFilerMatriser.append(matrise[a:b, :])
43     else:
44         a = unikeFilnavnIndekser[i]
45         b = len(matrise)
46         listeFilerMatriser.append(matrise[a:b, :])
47
48 # Splitter fil-matrisene opp slik at hver ny matrise inneholder alle må-
    lepunktene forbundet til ei bestemt brua
49 listeBruerMatriser = []
50 for filerMatriser in listeFilerMatriser:
51     bruIDerIndekser = np.sort(np.unique(filerMatriser[:, 1], True)[1])
```

```

52     if len(bruIDerIndekser) > 0:
53         for i in range(len(bruIDerIndekser)):
54             if i < len(bruIDerIndekser) - 1:
55                 a = bruIDerIndekser[i]
56                 b = bruIDerIndekser[i + 1]
57                 listeBruerMatriser.append(filerMatriser[a:b, :])
58             else:
59                 a = bruIDerIndekser[i]
60                 b = len(filerMatriser)
61                 listeBruerMatriser.append(filerMatriser[a:b, :])
62
63 gaulaBru = []
64 gaulaUtenfor = []
65 gaulaAlle = []
66 ikkeGaulaBru = []
67 ikkeGaulaUtenfor = []
68 ikkeGaulaAlle = []
69 for matriseM in listeBruerMatriser:
70     for linje in matriseM:
71         if linje[4] == 'DkB, høy':
72             if linje[1] == '356850353':
73                 gaulaAlle.append(float(linje[3]))
74                 if linje[2] == '0.0':
75                     gaulaBru.append(float(linje[3]))
76                 else:
77                     gaulaUtenfor.append(float(linje[3]))
78             else:
79                 ikkeGaulaAlle.append(float(linje[3]))
80                 if linje[2] == '0.0':
81                     ikkeGaulaBru.append(float(linje[3]))
82                 else:
83                     ikkeGaulaUtenfor.append(float(linje[3]))
84
85 gaulaBru.sort()
86 gaulaUtenfor.sort()
87 gaulaAlle.sort()
88 ikkeGaulaBru.sort()
89 ikkeGaulaUtenfor.sort()
90 ikkeGaulaAlle.sort()
91
92 gaulaBruFriksjon = np.asarray(gaulaBru)
93 gaulaUtenforFriksjon = np.asarray(gaulaUtenfor)
94 gaulaAlleFriksjon = np.asarray(gaulaAlle)
95 ikkeGaulaBruFriksjon = np.asarray(ikkeGaulaBru)
96 ikkeGaulaUtenforFriksjon = np.asarray(ikkeGaulaUtenfor)
97 ikkeGaulaAlleFriksjon = np.asarray(ikkeGaulaAlle)
98
99 gaulaBru5persentil = np.percentile(gaulaBruFriksjon, 5)
100 gaulaUtenfor5persentil = np.percentile(gaulaUtenforFriksjon, 5)
101 gaulaAlle5persentil = np.percentile(gaulaAlleFriksjon, 5)
102 gaulaBru15persentil = np.percentile(gaulaBruFriksjon, 15)
103 gaulaUtenfor15persentil = np.percentile(gaulaUtenforFriksjon, 15)
104 gaulaAlle15persentil = np.percentile(gaulaAlleFriksjon, 15)
105 gaulaBruGjennomsnitt = np.mean(gaulaBruFriksjon)
106 gaulaUtenforGjennomsnitt = np.mean(gaulaUtenforFriksjon)
107 gaulaAlleGjennomsnitt = np.mean(gaulaAlleFriksjon)
108 ikkeGaulaBru5persentil = np.percentile(ikkeGaulaBruFriksjon, 5)

```

```

109 ikkeGaulaUtenfor5persentil = np.percentile(ikkeGaulaUtenforFriksjon, 5)
110 ikkeGaulaAlle5persentil = np.percentile(ikkeGaulaAlleFriksjon, 5)
111 ikkeGaulaBru15persentil = np.percentile(ikkeGaulaBruFriksjon, 15)
112 ikkeGaulaUtenfor15persentil = np.percentile(ikkeGaulaUtenforFriksjon, 15)
113 ikkeGaulaAlle15persentil = np.percentile(ikkeGaulaAlleFriksjon, 15)
114 ikkeGaulaBruGjennomsnitt = np.mean(ikkeGaulaBruFriksjon)
115 ikkeGaulaUtenforGjennomsnitt = np.mean(ikkeGaulaUtenforFriksjon)
116 ikkeGaulaAlleGjennomsnitt = np.mean(ikkeGaulaAlleFriksjon)
117
118 midlertidigListe = []
119 for i in range(round(len(gaulaBruFriksjon)*0.05)):
120     midlertidigListe.append(gaulaBruFriksjon[i])
121 gaulaBruFriksjon5persentilListe = midlertidigListe
122 midlertidigListe = []
123 for i in range(round(len(gaulaUtenforFriksjon)*0.05)):
124     midlertidigListe.append(gaulaUtenforFriksjon[i])
125 gaulaUtenforFriksjon5persentilListe = midlertidigListe
126 midlertidigListe = []
127 for i in range(round(len(gaulaBruFriksjon)*0.15)):
128     midlertidigListe.append(gaulaBruFriksjon[i])
129 gaulaBruFriksjon15persentilListe = midlertidigListe
130 midlertidigListe = []
131 for i in range(round(len(gaulaUtenforFriksjon)*0.15)):
132     midlertidigListe.append(gaulaUtenforFriksjon[i])
133 gaulaUtenforFriksjon15persentilListe = midlertidigListe
134
135 midlertidigListe = []
136 for i in range(round(len(ikkeGaulaBruFriksjon)*0.05)):
137     midlertidigListe.append(ikkeGaulaBruFriksjon[i])
138 ikkeGaulaBruFriksjon5persentilListe = midlertidigListe
139 midlertidigListe = []
140 for i in range(round(len(ikkeGaulaUtenforFriksjon)*0.05)):
141     midlertidigListe.append(ikkeGaulaUtenforFriksjon[i])
142 ikkeGaulaUtenforFriksjon5persentilListe = midlertidigListe
143 midlertidigListe = []
144 for i in range(round(len(ikkeGaulaBruFriksjon)*0.15)):
145     midlertidigListe.append(ikkeGaulaBruFriksjon[i])
146 ikkeGaulaBruFriksjon15persentilListe = midlertidigListe
147 midlertidigListe = []
148 for i in range(round(len(ikkeGaulaUtenforFriksjon)*0.15)):
149     midlertidigListe.append(ikkeGaulaUtenforFriksjon[i])
150 ikkeGaulaUtenforFriksjon15persentilListe = midlertidigListe
151
152 gaulaBruFriksjon5persentilArray = np.asarray(
153     gaulaBruFriksjon5persentilListe)
154 gaulaUtenforFriksjon5persentilArray = np.asarray(
155     gaulaUtenforFriksjon5persentilListe)
156 gaulaBruFriksjon15persentilArray = np.asarray(
157     gaulaBruFriksjon15persentilListe)
158 gaulaUtenforFriksjon15persentilArray = np.asarray(
159     gaulaUtenforFriksjon15persentilListe)
160 ikkeGaulaBruFriksjon5persentilArray = np.asarray(
161     ikkeGaulaBruFriksjon5persentilListe)
162 ikkeGaulaUtenforFriksjon5persentilArray = np.asarray(
163     ikkeGaulaUtenforFriksjon5persentilListe)
164 ikkeGaulaBruFriksjon15persentilArray = np.asarray(
165     ikkeGaulaBruFriksjon15persentilListe)
166 ikkeGaulaUtenforFriksjon15persentilArray = np.asarray(
167     ikkeGaulaUtenforFriksjon15persentilListe)

```

```

159 ikkeGaulaUtenforFriksjon15persentilArray = np.asarray(
    ikkeGaulaUtenforFriksjon15persentilListe)
160
161 gaulaSignifikans = scipy.stats.ttest_ind(gaulaBruFriksjon,
162                                           gaulaUtenforFriksjon,
163                                           equal_var=True, alternative='less'
    )
164 gaula5persentilSignifikans = scipy.stats.ttest_ind(
    gaulaBruFriksjon5persentilArray,
165
    gaulaUtenforFriksjon5persentilArray,
166
    equal_var=True,
    alternative='less')
167 gaula15persentilSignifikans = scipy.stats.ttest_ind(
    gaulaBruFriksjon15persentilArray,
168
    gaulaUtenforFriksjon15persentilArray,
169
    equal_var=True,
    alternative='less')
170
171 ikkeGaulaSignifikans = scipy.stats.ttest_ind(ikkeGaulaBruFriksjon,
    ikkeGaulaUtenforFriksjon,
172
    equal_var=True, alternative='
    less')
173 ikkeGaula5persentilSignifikans = scipy.stats.ttest_ind(
    ikkeGaulaBruFriksjon5persentilArray,
174
    ikkeGaulaUtenforFriksjon5persentilArray,
175
    equal_var=True,
    alternative='less')
176 ikkeGaula15persentilSignifikans = scipy.stats.ttest_ind(
    ikkeGaulaBruFriksjon15persentilArray,
177
    ikkeGaulaUtenforFriksjon15persentilArray,
178
    equal_var=True,
    alternative='less')
179 gaulaBruIkkeGaulaBruSignifikans = scipy.stats.ttest_ind(gaulaBruFriksjon,
    ikkeGaulaBruFriksjon,
180
    equal_var=True,
    alternative='less')
181
182 print(len(gaulaBruFriksjon))
183 print(len(gaulaUtenforFriksjon))
184 print(len(gaulaAlleFriksjon))
185 print(len(ikkeGaulaBruFriksjon))
186 print(len(ikkeGaulaUtenforFriksjon))
187 print(len(ikkeGaulaAlleFriksjon))
188
189 print('Gaula bru: 5-persentil: {}, 15-persentil: {}, Gjennomsnitt: {}'.
    format(gaulaBru5persentil, gaulaBru15persentil, gaulaBruGjennomsnitt))
190 print('Gaula utenfor: 5-persentil: {}, 15-persentil: {}, Gjennomsnitt: {}'.
    format(gaulaUtenfor5persentil, gaulaUtenfor15persentil,
    gaulaUtenforGjennomsnitt))
191 print('Gaula alle: 5-persentil: {}, 15-persentil: {}, Gjennomsnitt: {}'.
    format(gaulaAlle5persentil, gaulaAlle15persentil, gaulaAlleGjennomsnitt)
    )
192 print('Ikke Gaula bru: 5-persentil: {}, 15-persentil: {}, Gjennomsnitt: {}'.

```

```

        .format(ikkeGaulaBru5persentil, ikkeGaulaBru15persentil,
        ikkeGaulaBruGjennomsnitt))
193 print('Ikke Gaula utenfor: 5-persentil: {}, 15-persentil: {}, Gjennomsnitt:
        {}'.format(ikkeGaulaUtenfor5persentil, ikkeGaulaUtenfor15persentil,
        ikkeGaulaUtenforGjennomsnitt))
194 print('Ikke Gaula alle: 5-persentil: {}, 15-persentil: {}, Gjennomsnitt: {}
        '.format(ikkeGaulaAlle5persentil, ikkeGaulaAlle15persentil,
        ikkeGaulaAlleGjennomsnitt))
195 print('\n')
196 print('P-verdi for totalfriksjon, 5-persentil, på bru sammenlignet med
        utenfor bru for Gaulabrua: {}'.format(gaula5persentilSignifikans[1]))
197 print('P-verdi for totalfriksjon, 15-persentil, på bru sammenlignet med
        utenfor bru for Gaulabrua: {}'.format(gaula15persentilSignifikans[1]))
198 print('P-verdi for totalfriksjon, gjennomsnitt, på bru sammenlignet med
        utenfor bru for Gaulabrua: {}'.format(gaulaSignifikans[1]))
199 print('P-verdi for totalfriksjon, 5-persentil, på bru sammenlignet med
        utenfor bru for Ikke Gaulabrua: {}'.format(
        ikkeGaula5persentilSignifikans[1]))
200 print('P-verdi for totalfriksjon, 15-persentil, på bru sammenlignet med
        utenfor bru for Ikke Gaulabrua: {}'.format(
        ikkeGaula15persentilSignifikans[1]))
201 print('P-verdi for totalfriksjon, gjennomsnitt, på bru sammenlignet med
        utenfor bru for Ikke Gaulabrua: {}'.format(ikkeGaulaSignifikans[1]))
202 print('P-verdi for totalfriksjon, gjennomsnitt, på Gaulabrua sammenlignet
        med på bruer som ikke er Gaulabrua: {}'.format(
        gaulaBruIkkeGaulaBruSignifikans[1]))
203
204 fig = plt.figure(figsize=(9, 6))
205 fig.suptitle('Densitetsfordeling til målepunkter med vinterdriftsklasse B,
        høy', fontsize=20)
206 fig.subplots_adjust(left=0.08, bottom=0.19, right=0.94, top=0.9, wspace
        =0.32, hspace=0.35)
207 ax = fig.add_subplot(111)
208
209 x_verdier = np.linspace(0, 0.7)
210 densitet1 = gaussian_kde(gaulaAlleFriksjon)
211 densitet1.covariance_factor = lambda: .5
212 densitet1._compute_covariance()
213 densitet2 = gaussian_kde(ikkeGaulaAlleFriksjon)
214 densitet2.covariance_factor = lambda: .5
215 densitet2._compute_covariance()
216
217 ax.plot(x_verdier, densitet1(x_verdier), label='Gaulabrua')
218 ax.plot(x_verdier, densitet2(x_verdier), label='Ikke Gaulabrua')
219 ax.set_ylabel('Densitet', fontsize=15)
220 ax.set_xlabel('Totalfriksjonsverdi', fontsize=15)
221 tick_spacing = 0.05
222 ax.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))
223 ax.set_xlim([0, 0.7])
224 ax.tick_params(right=True, left=True, bottom=True)
225 fig.legend(loc=8, ncol=2, fontsize=15)
226
227
228 fig = plt.figure(figsize=(12, 6))
229 fig.suptitle('Densitetsfordeling til målepunkter med vinterdriftsklasse B,
        høy', fontsize=20)
230 fig.subplots_adjust(left=0.08, bottom=0.23, right=0.94, top=0.9, wspace

```

```

    =0.32, hspace=0.35)
231 ax = fig.add_subplot(121)
232 ax2 = fig.add_subplot(122)
233
234 x_verdier = np.linspace(0, 0.7)
235 densitet3 = gaussian_kde(gaulaBruFriksjon)
236 densitet3.covariance_factor = lambda: .5
237 densitet3._compute_covariance()
238 densitet4 = gaussian_kde(gaulaUtenforFriksjon)
239 densitet4.covariance_factor = lambda: .5
240 densitet4._compute_covariance()
241 densitet5 = gaussian_kde(ikkeGaulaBruFriksjon)
242 densitet5.covariance_factor = lambda: .5
243 densitet5._compute_covariance()
244 densitet6 = gaussian_kde(ikkeGaulaUtenforFriksjon)
245 densitet6.covariance_factor = lambda: .5
246 densitet6._compute_covariance()
247
248 ax.plot(x_verdier, densitet3(x_verdier), label='Gaulabrua')
249 ax.plot(x_verdier, densitet4(x_verdier), label='Utenfor Gaulabrua', c='
    brown')
250 ax.set_ylabel('Densitet', fontsize=15)
251 ax.set_xlabel('Totalfriksjonsverdi', fontsize=15)
252 tick_spacing = 0.1
253 ax.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))
254 ax.set_xlim([0, 0.7])
255 ax.set_ylim([0, 9])
256 ax.tick_params(right=True, left=True, bottom=True)
257
258 ax2.plot(x_verdier, densitet5(x_verdier), label='Ikke Gaulabrua', c='coral'
    )
259 ax2.plot(x_verdier, densitet6(x_verdier), label='Utenfor ikke Gaulabrua', c
    ='grey')
260 ax2.set_ylabel('Densitet', fontsize=15)
261 ax2.set_xlabel('Totalfriksjonsverdi', fontsize=15)
262 tick_spacing = 0.1
263 ax2.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))
264 ax2.set_xlim([0, 0.7])
265 ax2.set_ylim([0, 9])
266 ax2.tick_params(right=True, left=True, bottom=True)
267 fig.legend(loc=8, ncol=2, fontsize=15)
268
269 plt.show()

```

Vedlegg B8: Sensitivitet

```
1 import math
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import matplotlib.gridspec as gridspec
5 import matplotlib.ticker as ticker
6
7 vegklasse = 'H1'
8 fartsgrense = 80
9 fartstillegg = 5
10 overhoyde = 8 # I prosent
11 totalfriksjon = 0.461
12 sidefriksjonsandel = 34 # I prosent
13 sikkerhetsfaktor = 1
14
15 minsteHorisontalkurveradius = ((fartsgrense + fartstillegg)**2)/(127*((
    overhoyde/100)+(totalfriksjon*(sidefriksjonsandel/100)/sikkerhetsfaktor)
    ))
16 minsteHorisontalkurveradius50 = minsteHorisontalkurveradius * 1.5
17
18 sidefriksjonNorge = 0.157
19 sidefriksjonSverige = 0.13
20 sidefriksjonDanmark = sidefriksjonSverige
21 sidefriksjonUSA = 0.14
22 sidefriksjonCanada = 0.14
23 sidefriksjonAustralia = 0.16
24 sidefriksjonMontana = 0.14
25
26
27 def nedrunding(tall, base=25):
28     return base * math.floor(tall/base)
29
30
31 def opprunding(tall, base=25):
32     return base * math.ceil(tall/base)
33
34
35 # Totalfriksjon
36 totalfriksjonIntervall = np.arange(0, 1.01, 0.01)
37 totalfriksjonXverdier = []
38 totalfriksjonYverdierP4 = []
39 totalfriksjonYverdierP2 = []
40 totalfriksjonYverdierN0 = []
41 totalfriksjonYverdierM2 = []
42 totalfriksjonYverdierM4 = []
43
44 minsteHorisontalkurveradiusP4 = ((fartsgrense + fartstillegg)**2)/(127*((
    overhoyde/100)+(totalfriksjon*((sidefriksjonsandel+4)/100)/
    sikkerhetsfaktor)))
45 minsteHorisontalkurveradiusP2 = ((fartsgrense + fartstillegg)**2)/(127*((
    overhoyde/100)+(totalfriksjon*((sidefriksjonsandel+2)/100)/
    sikkerhetsfaktor)))
46 minsteHorisontalkurveradiusN0 = ((fartsgrense + fartstillegg)**2)/(127*((
    overhoyde/100)+(totalfriksjon*((sidefriksjonsandel+0)/100)/
```

```

sikkerhetsfaktor)))
47 minsteHorisontalkurveradiusM2 = ((fartsgrense + fartstillegg)**2)/(127*((
overhoyde/100)+(totalfriksjon*((sidefriksjonsandel-2)/100)/
sikkerhetsfaktor)))
48 minsteHorisontalkurveradiusM4 = ((fartsgrense + fartstillegg)**2)/(127*((
overhoyde/100)+(totalfriksjon*((sidefriksjonsandel-4)/100)/
sikkerhetsfaktor)))
49
50 minsteHorisontalkurveradiusP4_015 = ((fartsgrense + fartstillegg)**2)
/ (127*((overhoyde/100)+(0.15*((sidefriksjonsandel+4)/100)/
sikkerhetsfaktor)))
51 minsteHorisontalkurveradiusP2_015 = ((fartsgrense + fartstillegg)**2)
/ (127*((overhoyde/100)+(0.15*((sidefriksjonsandel+2)/100)/
sikkerhetsfaktor)))
52 minsteHorisontalkurveradiusN0_015 = ((fartsgrense + fartstillegg)**2)
/ (127*((overhoyde/100)+(0.15*((sidefriksjonsandel+0)/100)/
sikkerhetsfaktor)))
53 minsteHorisontalkurveradiusM2_015 = ((fartsgrense + fartstillegg)**2)
/ (127*((overhoyde/100)+(0.15*((sidefriksjonsandel-2)/100)/
sikkerhetsfaktor)))
54 minsteHorisontalkurveradiusM4_015 = ((fartsgrense + fartstillegg)**2)
/ (127*((overhoyde/100)+(0.15*((sidefriksjonsandel-4)/100)/
sikkerhetsfaktor)))
55
56 minsteHorisontalkurveradiusP4_018 = ((fartsgrense + fartstillegg)**2)
/ (127*((overhoyde/100)+(0.18*((sidefriksjonsandel+4)/100)/
sikkerhetsfaktor)))
57 minsteHorisontalkurveradiusP2_018 = ((fartsgrense + fartstillegg)**2)
/ (127*((overhoyde/100)+(0.18*((sidefriksjonsandel+2)/100)/
sikkerhetsfaktor)))
58 minsteHorisontalkurveradiusN0_018 = ((fartsgrense + fartstillegg)**2)
/ (127*((overhoyde/100)+(0.18*((sidefriksjonsandel+0)/100)/
sikkerhetsfaktor)))
59 minsteHorisontalkurveradiusM2_018 = ((fartsgrense + fartstillegg)**2)
/ (127*((overhoyde/100)+(0.18*((sidefriksjonsandel-2)/100)/
sikkerhetsfaktor)))
60 minsteHorisontalkurveradiusM4_018 = ((fartsgrense + fartstillegg)**2)
/ (127*((overhoyde/100)+(0.18*((sidefriksjonsandel-4)/100)/
sikkerhetsfaktor)))
61
62 minsteHorisontalkurveradiusP4_020 = ((fartsgrense + fartstillegg)**2)
/ (127*((overhoyde/100)+(0.20*((sidefriksjonsandel+4)/100)/
sikkerhetsfaktor)))
63 minsteHorisontalkurveradiusP2_020 = ((fartsgrense + fartstillegg)**2)
/ (127*((overhoyde/100)+(0.20*((sidefriksjonsandel+2)/100)/
sikkerhetsfaktor)))
64 minsteHorisontalkurveradiusN0_020 = ((fartsgrense + fartstillegg)**2)
/ (127*((overhoyde/100)+(0.20*((sidefriksjonsandel+0)/100)/
sikkerhetsfaktor)))
65 minsteHorisontalkurveradiusM2_020 = ((fartsgrense + fartstillegg)**2)
/ (127*((overhoyde/100)+(0.20*((sidefriksjonsandel-2)/100)/
sikkerhetsfaktor)))
66 minsteHorisontalkurveradiusM4_020 = ((fartsgrense + fartstillegg)**2)
/ (127*((overhoyde/100)+(0.20*((sidefriksjonsandel-4)/100)/
sikkerhetsfaktor)))
67
68 minsteHorisontalkurveradiusP4_023 = ((fartsgrense + fartstillegg)**2)

```

```

        / (127 * ((overhoyde/100) + (0.23 * ((sidefriksjonsandel+4)/100) /
        sikkerhetsfaktor)))
69 minsteHorisontalkurveradiusP2_023 = ((fartsgrense + fartstillegg)**2)
        / (127 * ((overhoyde/100) + (0.23 * ((sidefriksjonsandel+2)/100) /
        sikkerhetsfaktor)))
70 minsteHorisontalkurveradiusN0_023 = ((fartsgrense + fartstillegg)**2)
        / (127 * ((overhoyde/100) + (0.23 * ((sidefriksjonsandel+0)/100) /
        sikkerhetsfaktor)))
71 minsteHorisontalkurveradiusM2_023 = ((fartsgrense + fartstillegg)**2)
        / (127 * ((overhoyde/100) + (0.23 * ((sidefriksjonsandel-2)/100) /
        sikkerhetsfaktor)))
72 minsteHorisontalkurveradiusM4_023 = ((fartsgrense + fartstillegg)**2)
        / (127 * ((overhoyde/100) + (0.23 * ((sidefriksjonsandel-4)/100) /
        sikkerhetsfaktor)))
73
74 minsteHorisontalkurveradiusP4_030 = ((fartsgrense + fartstillegg)**2)
        / (127 * ((overhoyde/100) + (0.30 * ((sidefriksjonsandel+4)/100) /
        sikkerhetsfaktor)))
75 minsteHorisontalkurveradiusP2_030 = ((fartsgrense + fartstillegg)**2)
        / (127 * ((overhoyde/100) + (0.30 * ((sidefriksjonsandel+2)/100) /
        sikkerhetsfaktor)))
76 minsteHorisontalkurveradiusN0_030 = ((fartsgrense + fartstillegg)**2)
        / (127 * ((overhoyde/100) + (0.30 * ((sidefriksjonsandel+0)/100) /
        sikkerhetsfaktor)))
77 minsteHorisontalkurveradiusM2_030 = ((fartsgrense + fartstillegg)**2)
        / (127 * ((overhoyde/100) + (0.30 * ((sidefriksjonsandel-2)/100) /
        sikkerhetsfaktor)))
78 minsteHorisontalkurveradiusM4_030 = ((fartsgrense + fartstillegg)**2)
        / (127 * ((overhoyde/100) + (0.30 * ((sidefriksjonsandel-4)/100) /
        sikkerhetsfaktor)))
79
80 totalfriksjonXverdiP4 = (((fartsgrense + fartstillegg)**2) / (
        minsteHorisontalkurveradiusP4 * 127) - (overhoyde/100)) / ((
        sidefriksjonsandel+4)/100)/sikkerhetsfaktor
81 totalfriksjonXverdiP2 = (((fartsgrense + fartstillegg)**2) / (
        minsteHorisontalkurveradiusP2 * 127) - (overhoyde/100)) / ((
        sidefriksjonsandel+2)/100)/sikkerhetsfaktor
82 totalfriksjonXverdiN0 = (((fartsgrense + fartstillegg)**2) / (
        minsteHorisontalkurveradiusN0 * 127) - (overhoyde/100)) / ((
        sidefriksjonsandel+0)/100)/sikkerhetsfaktor
83 totalfriksjonXverdiM2 = (((fartsgrense + fartstillegg)**2) / (
        minsteHorisontalkurveradiusM2 * 127) - (overhoyde/100)) / ((
        sidefriksjonsandel-2)/100)/sikkerhetsfaktor
84 totalfriksjonXverdiM4 = (((fartsgrense + fartstillegg)**2) / (
        minsteHorisontalkurveradiusM4 * 127) - (overhoyde/100)) / ((
        sidefriksjonsandel-4)/100)/sikkerhetsfaktor
85
86 for i in totalfriksjonIntervall:
87     totalfriksjonXverdier.append(i)
88     minsteHorisontalkurveradiusTotalfriksjonP4 = ((fartsgrense +
        fartstillegg) ** 2) / (127 * ((overhoyde/100) + (i * ((
        sidefriksjonsandel+4)/100)/sikkerhetsfaktor)))
89     minsteHorisontalkurveradiusTotalfriksjonP2 = ((fartsgrense +
        fartstillegg) ** 2) / (127 * ((overhoyde/100) + (i * ((
        sidefriksjonsandel+2)/100)/sikkerhetsfaktor)))
90     minsteHorisontalkurveradiusTotalfriksjonN0 = ((fartsgrense +
        fartstillegg) ** 2) / (127 * ((overhoyde/100) + (i * ((

```

```

sidefriksjonsandel+0)/100)/sikkerhetsfaktor)))
91 minsteHorisontalkurveradiusTotalfriksjonM2 = ((fartsgrense +
fartstillegg) ** 2) / (127 * ((overhoyde/100) + (i * ((
sidefriksjonsandel-2)/100)/sikkerhetsfaktor)))
92 minsteHorisontalkurveradiusTotalfriksjonM4 = ((fartsgrense +
fartstillegg) ** 2) / (127 * ((overhoyde/100) + (i * ((
sidefriksjonsandel-4)/100)/sikkerhetsfaktor)))
93
94 totalfriksjonYverdierP4.append(
minsteHorisontalkurveradiusTotalfriksjonP4)
95 totalfriksjonYverdierP2.append(
minsteHorisontalkurveradiusTotalfriksjonP2)
96 totalfriksjonYverdierN0.append(
minsteHorisontalkurveradiusTotalfriksjonN0)
97 totalfriksjonYverdierM2.append(
minsteHorisontalkurveradiusTotalfriksjonM2)
98 totalfriksjonYverdierM4.append(
minsteHorisontalkurveradiusTotalfriksjonM4)
99
100
101 # Sidefriksjon
102 landNavn = ['Norge', 'Sverige', 'Danmark', 'USA', 'Canada', 'Australia']
103 landFarger = ['red', 'green', 'cyan', 'lime', 'brown', 'orange']
104 sidefriksjonerLand = [sidefriksjonNorge, sidefriksjonSverige,
sidefriksjonDanmark, sidefriksjonUSA, sidefriksjonCanada,
sidefriksjonAustralia]
105 unikeSidefriksjonsverdierLandPosisjoner = [sidefriksjonerLand.index(
sidefriksjonLand) for sidefriksjonLand in set(sidefriksjonerLand)]
106
107 sidefriksjonerLandUnike = []
108 sidefriksjonerLandDuplikat = []
109 landNavnUnike = []
110 landNavnDuplikat = []
111 fargerUnike = []
112 fargerDuplikat = []
113 for i in range(len(sidefriksjonerLand)):
114     if i in unikeSidefriksjonsverdierLandPosisjoner:
115         sidefriksjonerLandUnike.append(sidefriksjonerLand[i])
116         landNavnUnike.append(landNavn[i])
117         fargerUnike.append(landFarger[i])
118     else:
119         sidefriksjonerLandDuplikat.append(sidefriksjonerLand[i])
120         landNavnDuplikat.append(landNavn[i])
121         fargerDuplikat.append(landFarger[i])
122 horisontalkurveradiususerLandUnike = (((fartsgrense + fartstillegg)**2)
/ (127*((overhoyde/100)+(sidefriksjonLand/sikkerhetsfaktor))) for
sidefriksjonLand in sidefriksjonerLandUnike]
123 horisontalkurveradiususerLandDuplikat = (((fartsgrense + fartstillegg)**2)
/ (127*((overhoyde/100)+(sidefriksjonLand/sikkerhetsfaktor))) for
sidefriksjonLand in sidefriksjonerLandDuplikat]
124 horisontalkurveradiususerLandUnike50 = [horisontalkurveradiusLand * 1.5 for
horisontalkurveradiusLand in horisontalkurveradiususerLandUnike]
125 horisontalkurveradiususerLandDuplikat50 = [horisontalkurveradiusLand * 1.5
for horisontalkurveradiusLand in horisontalkurveradiususerLandDuplikat]
126 horisontalkurveradiusMontana = ((fartsgrense + fartstillegg)**2)/(127*((
overhoyde/100)+(sidefriksjonMontana/sikkerhetsfaktor)))
127 horisontalkurveradiusMontana50 = horisontalkurveradiusMontana * 1.5

```

```

128
129 sidefriksjonIntervall = np.arange(0, 0.25, 0.01)
130 sidefriksjonXverdier = []
131 sidefriksjonYverdier = []
132 for i in sidefriksjonIntervall:
133     minsteHorisontalkurveradiusSidefriksjon = ((fartsgrense + fartstillegg)
134     ** 2) / (127 * ((overhoyde / 100) + (i / sikkerhetsfaktor)))
135     sidefriksjonXverdier.append(i)
136     sidefriksjonYverdier.append(minsteHorisontalkurveradiusSidefriksjon)
137 sidefriksjonYverdier50 = [horisontalkurveradius * 1.5 for
138     horisontalkurveradius in sidefriksjonYverdier]
139
140 # Overhøyde
141 overhoydeIntervall = np.arange(-3, 12.1, 0.1)
142 overhoydeXverdier = []
143 overhoydeYverdier = []
144 overhoydeYverdiDimensjonerende = ((fartsgrense + fartstillegg)**2)/(127*(
145     overhoyde/100+(totalfriksjon*(sidefriksjonsandel/100)/sikkerhetsfaktor)
146     )
147 overhoydeYverdiDimensjonerende50 = overhoydeYverdiDimensjonerende * 1.5
148 beregnetOverhoyde = 0
149 for i in overhoydeIntervall:
150     overhoydeXverdier.append(i)
151     minsteHorisontalkurveradiusOverhoyde = ((fartsgrense + fartstillegg)
152     **2)/(127*(i/100+(totalfriksjon*(sidefriksjonsandel/100)/
153     sikkerhetsfaktor)))
154     overhoydeYverdier.append(minsteHorisontalkurveradiusOverhoyde)
155     overhoydeYverdier50.append(minsteHorisontalkurveradiusOverhoyde*1.5)
156 if minsteHorisontalkurveradius < 250:
157     overhoydeGrenseX = [6.5, overhoyde]
158     overhoydeGrenseY = [minsteHorisontalkurveradius,
159     minsteHorisontalkurveradius]
160     overhoydeLinjeX = [6.5, 6.5]
161     overhoydeLinjeY = [0, minsteHorisontalkurveradius]
162 else:
163     beregnetOverhoyde = 6.5 - ((minsteHorisontalkurveradius - 250) / (1000
164     - 250) * (6.5 - 1.5))
165     overhoydeGrenseX = [beregnetOverhoyde, overhoyde]
166     overhoydeGrenseY = [minsteHorisontalkurveradius,
167     minsteHorisontalkurveradius]
168     overhoydeLinjeX = [beregnetOverhoyde, beregnetOverhoyde]
169     overhoydeLinjeY = [0, minsteHorisontalkurveradius]
170 if minsteHorisontalkurveradius*1.5 < 250:
171     overhoydeGrenseX50 = [6.5, overhoyde]
172     overhoydeGrenseY50 = [minsteHorisontalkurveradius,
173     minsteHorisontalkurveradius]
174     overhoydeLinjeX50 = [6.5, 6.5]
175     overhoydeLinjeY50 = [0, minsteHorisontalkurveradius * 1.5]
176 else:
177     beregnetOverhoyde = 6.5 - ((minsteHorisontalkurveradius*1.5 - 250) /
178     (1000 - 250) * (6.5 - 1.5))
179     overhoydeGrenseX50 = [beregnetOverhoyde, overhoyde]
180     overhoydeGrenseY50 = [minsteHorisontalkurveradius*1.5,
181     minsteHorisontalkurveradius*1.5]
182     overhoydeLinjeX50 = [beregnetOverhoyde, beregnetOverhoyde]

```

```

173     overhoydeLinjeY50 = [0, minsteHorisontalkurveradius*1.5]
174
175 # Tilgjengelig sidefriksjon/totalfriksjon med redusert overhøyde
176 erfaringsverdiOverhoydeAvvik = 2
177 sidefriksjonBehov = ((fartsgrense + fartstillegg)**2)/(127*
    minsteHorisontalkurveradius) - 8/100
178 sidefriksjonBehov50 = ((fartsgrense + fartstillegg)**2)/(127*
    minsteHorisontalkurveradius50) - 8/100
179 sidefriksjonBehovDrift = ((fartsgrense + fartstillegg)**2)/(127*
    minsteHorisontalkurveradius) - 6.5/100
180 sidefriksjonBehovDrift50 = ((fartsgrense + fartstillegg)**2)/(127*
    minsteHorisontalkurveradius50) - beregnetOverhoyde/100
181 sidefriksjonBehovErfaring = ((fartsgrense + fartstillegg)**2)/(127*
    minsteHorisontalkurveradius) - (6.5-erfaringsverdiOverhoydeAvvik)/100
182 sidefriksjonBehovErfaring50 = ((fartsgrense + fartstillegg)**2)/(127*
    minsteHorisontalkurveradius50) - (beregnetOverhoyde-
    erfaringsverdiOverhoydeAvvik)/100
183
184 sidefriksjonBehovXverdier = []
185 sidefriksjonBehovYverdier = []
186 sidefriksjonBehovYverdier50 = []
187 sidefriksjonBehovIntervall = np.arange(0, 12, 0.01)
188
189 for i in sidefriksjonBehovIntervall:
190     minsteHorisontalkurveradiusSidefriksjon = ((fartsgrense + fartstillegg)
    **2)/(127*minsteHorisontalkurveradius) - i/100
191     sidefriksjonBehovXverdier.append(i)
192     sidefriksjonBehovYverdier.append(
    minsteHorisontalkurveradiusSidefriksjon)
193     minsteHorisontalkurveradiusSidefriksjon50 = ((fartsgrense +
    fartstillegg) ** 2) / (127 * minsteHorisontalkurveradius50) - i / 100
194     sidefriksjonBehovYverdier50.append(
    minsteHorisontalkurveradiusSidefriksjon50)
195
196 totalfriksjonBehov = sidefriksjonBehov / (sidefriksjonsandel/100)
197 totalfriksjonBehov50 = sidefriksjonBehov50 / (sidefriksjonsandel/100)
198 totalfriksjonBehovDrift = sidefriksjonBehovDrift / (sidefriksjonsandel/100)
199 totalfriksjonBehovDrift50 = sidefriksjonBehovDrift50 / (sidefriksjonsandel
    /100)
200 totalfriksjonBehovErfaring = sidefriksjonBehovErfaring / (
    sidefriksjonsandel/100)
201 totalfriksjonBehovErfaring50 = sidefriksjonBehovErfaring50 / (
    sidefriksjonsandel/100)
202 totalfriksjonBehovXverdier = sidefriksjonBehovXverdier
203 totalfriksjonBehovYverdier = [sidefriksjonBehov / (sidefriksjonsandel/100)
    for sidefriksjonBehov in sidefriksjonBehovYverdier]
204 totalfriksjonBehovYverdier50 = [sidefriksjonBehov / (sidefriksjonsandel
    /100) for sidefriksjonBehov in sidefriksjonBehovYverdier50]
205
206
207 # Sikkerhetsfaktor friksjon
208 totalfriksjonsverdier5persentil = [totalfriksjon, 0.15, 0.18, 0.20, 0.23,
    0.30]
209 totalfriksjonsverdier15persentil = [0.55, 0.15, 0.23, 0.27, 0.33, 0.35]
210 totalfriksjonsnavn = ['Krav', 'Rim', 'Klasse C', 'Alle klasser', 'Klasse B,
    middels', 'Klasse A']
211 totalfriksjonsfarger = ['red', 'cyan', 'lime', 'green', 'salmon', 'brown']

```

```

212
213 minsteHorisontalkurveradius15 = (((fartsgrense + fartstillegg)**2)/(127*((
    overhoyde/100)+(0.55*(sidefriksjonsandel/100))))
214
215 sikkerhetsfaktor0461 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius))-overhoyde/100)/((sidefriksjonsandel/100) *
    0.461)
216 sikkerhetsfaktor015 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius))-overhoyde/100)/((sidefriksjonsandel/100) *
    0.15)
217 sikkerhetsfaktor018 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius))-overhoyde/100)/(sidefriksjonsandel/100 *
    0.18)
218 sikkerhetsfaktor020 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius))-overhoyde/100)/(sidefriksjonsandel/100 *
    0.20)
219 sikkerhetsfaktor023 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius))-overhoyde/100)/(sidefriksjonsandel/100 *
    0.23)
220 sikkerhetsfaktor030 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius))-overhoyde/100)/(sidefriksjonsandel/100 *
    0.30)
221 sikkerhetsfaktorerBeregnet5persentil = [sikkerhetsfaktor0461,
    sikkerhetsfaktor015, sikkerhetsfaktor018,
222         sikkerhetsfaktor020,
    sikkerhetsfaktor023, sikkerhetsfaktor030]
223
224 sikkerhetsfaktor055_15 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius15))-overhoyde/100)/((sidefriksjonsandel/100)
    * 0.55)
225 sikkerhetsfaktor015_15 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius15))-overhoyde/100)/((sidefriksjonsandel/100)
    * 0.15)
226 sikkerhetsfaktor023_15 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius15))-overhoyde/100)/((sidefriksjonsandel/100)
    * 0.23)
227 sikkerhetsfaktor027_15 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius15))-overhoyde/100)/((sidefriksjonsandel/100)
    * 0.27)
228 sikkerhetsfaktor033_15 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius15))-overhoyde/100)/((sidefriksjonsandel/100)
    * 0.33)
229 sikkerhetsfaktor035_15 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius15))-overhoyde/100)/((sidefriksjonsandel/100)
    * 0.35)
230 sikkerhetsfaktorerBeregnet15persentil = [sikkerhetsfaktor055_15,
    sikkerhetsfaktor015_15, sikkerhetsfaktor023_15,
231         sikkerhetsfaktor027_15,
    sikkerhetsfaktor033_15, sikkerhetsfaktor035_15]
232
233 sikkerhetsfaktor0461_50 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius50))-overhoyde/100)/(sidefriksjonsandel/100 *
    0.461)
234 sikkerhetsfaktor015_50 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius50))-overhoyde/100)/(sidefriksjonsandel/100 *
    0.15)
235 sikkerhetsfaktor018_50 = (((fartsgrense + fartstillegg)**2)/(127 *

```

```

    minsteHorisontalkurveradius50))-overhoyde/100)/(sidefriksjonsandel/100 *
    0.18)
236 sikkerhetsfaktor020_50 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius50))-overhoyde/100)/(sidefriksjonsandel/100 *
    0.20)
237 sikkerhetsfaktor023_50 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius50))-overhoyde/100)/(sidefriksjonsandel/100 *
    0.23)
238 sikkerhetsfaktor030_50 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius50))-overhoyde/100)/(sidefriksjonsandel/100 *
    0.30)
239 sikkerhetsfaktorerBeregnet5persentil50 = [sikkerhetsfaktor0461_50,
    sikkerhetsfaktor015_50, sikkerhetsfaktor018_50,
240         sikkerhetsfaktor020_50,
    sikkerhetsfaktor023_50, sikkerhetsfaktor030_50]
241
242 sikkerhetsfaktor055_15_50 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius15 * 1.5))-overhoyde/100)/((
    sidefriksjonsandel/100) * 0.55)
243 sikkerhetsfaktor015_15_50 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius15 * 1.5))-overhoyde/100)/((
    sidefriksjonsandel/100) * 0.15)
244 sikkerhetsfaktor023_15_50 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius15 * 1.5))-overhoyde/100)/((
    sidefriksjonsandel/100) * 0.23)
245 sikkerhetsfaktor027_15_50 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius15 * 1.5))-overhoyde/100)/((
    sidefriksjonsandel/100) * 0.27)
246 sikkerhetsfaktor033_15_50 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius15 * 1.5))-overhoyde/100)/((
    sidefriksjonsandel/100) * 0.33)
247 sikkerhetsfaktor035_15_50 = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius15 * 1.5))-overhoyde/100)/((
    sidefriksjonsandel/100) * 0.35)
248 sikkerhetsfaktorerBeregnet15persentil_50 = [sikkerhetsfaktor055_15_50,
    sikkerhetsfaktor015_15_50, sikkerhetsfaktor023_15_50,
249         sikkerhetsfaktor027_15_50,
    sikkerhetsfaktor033_15_50, sikkerhetsfaktor035_15_50]
250
251 sikkerhetsfaktorIntervall = np.arange(0.9, 4.01, 0.01)
252 sikkerhetsfaktorXverdier = []
253 sikkerhetsfaktorYverdier = []
254 sikkerhetsfaktorYverdier50 = []
255 for i in sikkerhetsfaktorIntervall:
256     sikkerhetsfaktorXverdier.append(i)
257     totalfriksjonBeregnet = (((fartsgrense + fartstillegg)**2)/(127 *
    minsteHorisontalkurveradius))-(overhoyde / 100)) / ((sidefriksjonsandel
    / 100) * i)
258     sikkerhetsfaktorYverdier.append(totalfriksjonBeregnet)
259     totalfriksjonBeregnet50 = (((fartsgrense + fartstillegg) ** 2) / (127
    * minsteHorisontalkurveradius50)) - (overhoyde / 100)) / ((
    sidefriksjonsandel / 100) * i)
260     sikkerhetsfaktorYverdier50.append(totalfriksjonBeregnet50)
261
262 sikkerhetsfaktorIntervall = np.arange(0.9, 4.01, 0.01)
263 sikkerhetsfaktorXverdier_15 = []
264 sikkerhetsfaktorYverdier_15 = []

```

```

265 sikkerhetsfaktorYverdier_15_50 = []
266
267 for i in sikkerhetsfaktorIntervall:
268     sikkerhetsfaktorXverdier_15.append(i)
269     totalfriksjonBeregnet_15 = (((fartsgrense + fartstillegg)**2)/(127 *
minsteHorisontalkurveradius15))-(overhoyde / 100)) / ((
sidefriksjonsandel / 100) * i)
270     sikkerhetsfaktorYverdier_15.append(totalfriksjonBeregnet_15)
271     totalfriksjonBeregnet_15_50 = (((fartsgrense + fartstillegg) ** 2) /
(127 * minsteHorisontalkurveradius15 * 1.5)) - (overhoyde / 100)) / ((
sidefriksjonsandel / 100) * i)
272     sikkerhetsfaktorYverdier_15_50.append(totalfriksjonBeregnet_15_50)
273
274
275 # lager figurer for henholdsvis totalfriksjon, sidefriksjonsandel, overhø
yde og sikkerhetsfaktor for friksjon
276 fig = plt.figure(figsize=(9, 6))
277 fig.suptitle('Sensitivitet til totalfriksjon med ulike sidefriksjonsandeler
\n i utregning av minste horisontalkurveradius for {} veg'.format(
vegklasse), fontsize=20)
278 fig.subplots_adjust(left=0.09, bottom=0.1, right=0.69, top=0.87, wspace
=0.2, hspace=0.367)
279
280 ax1 = fig.add_subplot(111)
281 ax1.scatter(totalfriksjonXverdiP4, minsteHorisontalkurveradiusP4, c='red',
zorder=3, label='Krav')
282 ax1.scatter(totalfriksjonXverdiP2, minsteHorisontalkurveradiusP2, c='red',
zorder=3)
283 ax1.scatter(totalfriksjonXverdiN0, minsteHorisontalkurveradiusN0, c='red',
zorder=3)
284 ax1.scatter(totalfriksjonXverdiM2, minsteHorisontalkurveradiusM2, c='red',
zorder=3)
285 ax1.scatter(totalfriksjonXverdiM4, minsteHorisontalkurveradiusM4, c='red',
zorder=3)
286
287 ax1.scatter(0.15, minsteHorisontalkurveradiusP4_015, c='cyan', zorder=3,
label='Rim')
288 ax1.scatter(0.15, minsteHorisontalkurveradiusP2_015, c='cyan', zorder=3)
289 ax1.scatter(0.15, minsteHorisontalkurveradiusN0_015, c='cyan', zorder=3)
290 ax1.scatter(0.15, minsteHorisontalkurveradiusM2_015, c='cyan', zorder=3)
291 ax1.scatter(0.15, minsteHorisontalkurveradiusM4_015, c='cyan', zorder=3)
292
293 ax1.scatter(0.18, minsteHorisontalkurveradiusP4_018, c='lime', zorder=3,
label='Klasse C')
294 ax1.scatter(0.18, minsteHorisontalkurveradiusP2_018, c='lime', zorder=3)
295 ax1.scatter(0.18, minsteHorisontalkurveradiusN0_018, c='lime', zorder=3)
296 ax1.scatter(0.18, minsteHorisontalkurveradiusM2_018, c='lime', zorder=3)
297 ax1.scatter(0.18, minsteHorisontalkurveradiusM4_018, c='lime', zorder=3)
298
299 ax1.scatter(0.20, minsteHorisontalkurveradiusP4_020, c='green', zorder=3,
label='Alle klasser')
300 ax1.scatter(0.20, minsteHorisontalkurveradiusP2_020, c='green', zorder=3)
301 ax1.scatter(0.20, minsteHorisontalkurveradiusN0_020, c='green', zorder=3)
302 ax1.scatter(0.20, minsteHorisontalkurveradiusM2_020, c='green', zorder=3)
303 ax1.scatter(0.20, minsteHorisontalkurveradiusM4_020, c='green', zorder=3)
304
305 ax1.scatter(0.23, minsteHorisontalkurveradiusP4_023, c='salmon', zorder=3,

```

```

        label='Klasse B, middels')
306 ax1.scatter(0.23, minsteHorisontalkurveradiusP2_023, c='salmon', zorder=3)
307 ax1.scatter(0.23, minsteHorisontalkurveradiusN0_023, c='salmon', zorder=3)
308 ax1.scatter(0.23, minsteHorisontalkurveradiusM2_023, c='salmon', zorder=3)
309 ax1.scatter(0.23, minsteHorisontalkurveradiusM4_023, c='salmon', zorder=3)
310
311 ax1.scatter(0.30, minsteHorisontalkurveradiusP4_030, c='brown', zorder=3,
        label='Klasse A')
312 ax1.scatter(0.30, minsteHorisontalkurveradiusP2_030, c='brown', zorder=3)
313 ax1.scatter(0.30, minsteHorisontalkurveradiusN0_030, c='brown', zorder=3)
314 ax1.scatter(0.30, minsteHorisontalkurveradiusM2_030, c='brown', zorder=3)
315 ax1.scatter(0.30, minsteHorisontalkurveradiusM4_030, c='brown', zorder=3)
316
317 ax1.plot(totalfriksjonXverdier, totalfriksjonYverdierM4, zorder=2, c='
        darkred', label='-4%')
318 ax1.plot(totalfriksjonXverdier, totalfriksjonYverdierM2, zorder=2, c='c',
        label='-2%')
319 ax1.plot(totalfriksjonXverdier, totalfriksjonYverdierN0, zorder=2, c='blue'
        , label='34%')
320 ax1.plot(totalfriksjonXverdier, totalfriksjonYverdierP2, zorder=2, c='gold'
        , label='+2%')
321 ax1.plot(totalfriksjonXverdier, totalfriksjonYverdierP4, zorder=2, c='
        orange', label='+4%')
322
323 ax1.set_ylabel('Horisontalkurveradius [m]', fontsize=15)
324 ax1.set_xlabel('Totalfriksjon', fontsize=15)
325 ax1.xaxis.set_major_locator(ticker.MultipleLocator(0.04))
326 ax1.xaxis.set_minor_locator(ticker.MultipleLocator(0.02))
327 ax1.yaxis.set_major_locator(ticker.MultipleLocator(50))
328 ax1.yaxis.set_minor_locator(ticker.MultipleLocator(25))
329 ax1.set_xlim([0, 0.5])
330 ax1.set_ylim([200, 500])
331 ax1.grid()
332 ax1.xaxis.grid(True, which='minor')
333 ax1.yaxis.grid(True, which='minor')
334 fig.legend(loc=5, ncol=1, fontsize=15)
335
336 # Sidefriksjon
337 fig = plt.figure(figsize=(9, 6))
338 fig.suptitle('Sensitivitet til sidefriksjon i utregning av \n minste
        horisontalkurveradius for {} veg'.format(vegklasse), fontsize=20)
339 fig.subplots_adjust(left=0.1, bottom=0.1, right=0.75, top=0.85, wspace=0.2,
        hspace=0.367)
340
341 ax2 = fig.add_subplot(111)
342 ax2.plot(sidefriksjonXverdier, sidefriksjonYverdier, c='blue', label='
        Standard')
343 ax2.plot(sidefriksjonXverdier, sidefriksjonYverdier50, c='purple', label='
        50% økning')
344 for i in range(len(sidefriksjonerLandUnike)):
345     ax2.scatter(sidefriksjonerLandUnike[i],
        horisontalkurveradiuserLandUnike[i], zorder=5, c=fargerUnike[i], label=
        landNavnUnike[i])
346     ax2.scatter(sidefriksjonerLandUnike[i],
        horisontalkurveradiuserLandUnike50[i], zorder=5, c=fargerUnike[i])
347 for i in range(len(sidefriksjonerLandDuplikat)):
348     ax2.scatter(sidefriksjonerLandDuplikat[i],

```

```

    horisontalkurveradiusLandDuplikat[i], zorder=4, c=fargerDuplikat[i],
    label=landNavnDuplikat[i], s=2 ** 7)
349 ax2.scatter(sidefriksjonerLandDuplikat[i],
    horisontalkurveradiusLandDuplikat50[i], zorder=4, c=fargerDuplikat[i],
    s=2 ** 7)
350 ax2.scatter(sidefriksjonMontana, horisontalkurveradiusMontana, zorder=3,
    marker='D', c='grey', s=2 ** 8, label='Montana')
351 ax2.scatter(sidefriksjonMontana, horisontalkurveradiusMontana50, zorder=3,
    marker='D', c='grey', s=2 ** 8)
352 ax2.set_ylabel('Horisontalkurveradius [m]', fontsize=15)
353 ax2.set_xlabel('Sidefriksjon', fontsize=15)
354 ax2.set_xlim([min(sidefriksjonerLand)-0.01, max(sidefriksjonerLand)+0.01])
355 ax2.set_ylim([nedrunding(min(horisontalkurveradiusLandUnike)), opprunding
    (max(horisontalkurveradiusLandUnike50))])
356 ax2.yaxis.set_major_locator(ticker.MultipleLocator(25))
357 ax2.grid()
358 fig.legend(loc=5, fontsize=15)
359
360 # Overhøyde
361 fig = plt.figure(figsize=(8, 6))
362 fig.suptitle('Vedlikeholdskrav til overhøyde for \n minste
    horisontalkurveradius for {} veg'.format(vegklasse), fontsize=20)
363 fig.subplots_adjust(left=0.1, bottom=0.2, right=0.9, top=0.85, wspace=0.2,
    hspace=0.367)
364
365 ax3 = fig.add_subplot(111)
366 ax3.plot(overhoydeXverdier, overhoydeYverdier, c='blue', label='Standard',
    zorder=2)
367 ax3.plot(overhoydeXverdier, overhoydeYverdier50, c='purple', label='50% Ø
    kning', zorder=2)
368 ax3.scatter(overhoyde, overhoydeYverdiDimensjonerende, c='red', zorder=3)
369 ax3.scatter(overhoyde, overhoydeYverdiDimensjonerende50, c='green', zorder
    =3)
370 ax3.plot([6.5, 6.5], [0, 250], color='black') # Definerer linjene til
    vedlikeholdsgrensene
371 ax3.plot([1.5, 6.5], [1000, 250], color='black')
372 ax3.plot([9.5, 9.5], [0, 250], c='black')
373 ax3.plot([4.5, 9.5], [1500, 250], c='black')
374 ax3.fill_between([1.5, 6.5], [1000, 250], color='lightgrey') # Gjør
    verdiene utenfor vedlikeholdsgrensene grå
375 ax3.fill_between([-3, 1.5], [1000, 1000], color='lightgrey')
376 ax3.fill_between([4.5, 9.5], [1500, 1500], [1500, 250], color='lightgrey')
377 ax3.fill_between([9.5, 12], [1500, 1500], color='lightgrey')
378 ax3.plot(overhoydeGrenseX, overhoydeGrenseY, c='red') # Lager linjer fra
    punktene til vedlikeholdsgrensene
379 ax3.plot(overhoydeLinjeX, overhoydeLinjeY, c='red')
380 ax3.plot(overhoydeGrenseX50, overhoydeGrenseY50, c='green')
381 ax3.plot(overhoydeLinjeX50, overhoydeLinjeY50, c='green')
382 ax3.text(1.5, 375, 'For lite\noverhøyde', zorder=10, fontsize=15)
383 ax3.text(9.5, 375, 'For mye\noverhøyde', zorder=10, fontsize=15)
384
385 ax3.set_ylabel('Horisontalkurveradius [m]', fontsize=15)
386 ax3.set_xlabel('Overhøyde [%]', fontsize=15)
387 ax3.tick_params(right=True, left=True, bottom=True)
388 tick_spacing3_1 = 1
389 tick_spacing3_2 = 25
390 ax3.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing3_1))

```

```

391 ax3.yaxis.set_major_locator(ticker.MultipleLocator(tick_spacing3_2))
392 ax3.set_xlim([0, 12])
393 ax3.set_ylim([200, 500])
394 fig.legend(loc=8, ncol=3, fontsize=15)
395
396 fig = plt.figure(figsize=(12, 6))
397 fig.suptitle('Sensitivitet til overhøydens krav til friksjon for {} veg'.
              format(vegklasse), fontsize=20)
398 fig.subplots_adjust(left=0.075, bottom=0.18, right=0.975, top=0.85, wspace
                    =0.25, hspace=0.367)
399
400 ax4_1 = fig.add_subplot(121)
401 ax4_1.plot(sidefriksjonBehovXverdier, sidefriksjonBehovYverdier, c='blue',
            label='Standard', zorder=2)
402 ax4_1.plot(sidefriksjonBehovXverdier, sidefriksjonBehovYverdier50, c='
            purple', label='50% Økning', zorder=2)
403 ax4_1.scatter(8, sidefriksjonBehov, c='red', zorder=3, s=2 ** 7, label='
            Standard')
404 ax4_1.scatter(8, sidefriksjonBehov50, c='red', s=2 ** 7, zorder=3)
405 ax4_1.scatter(6.5, sidefriksjonBehovDrift, c='salmon', s=2 ** 7, zorder=3,
            label='Vedlikeholdskrav')
406 ax4_1.scatter(beregnetOverhoyde, sidefriksjonBehovDrift50, c='salmon', s=2
            ** 7, zorder=3)
407 ax4_1.scatter(6.5-erfaringsverdiOverhoydeAvvik, sidefriksjonBehovErfaring,
            c='seagreen', s=2 ** 7, zorder=3, label='Erfaring')
408 ax4_1.scatter(beregnetOverhoyde-erfaringsverdiOverhoydeAvvik,
            sidefriksjonBehovErfaring50, c='seagreen', s=2 ** 7, zorder=3)
409
410 ax4_1.set_title('Sidefriksjon', fontsize=17)
411 ax4_1.set_ylabel('Sidefriksjonsverdi', fontsize=15)
412 ax4_1.set_xlabel('Overhøyde [%]', fontsize=15)
413 ax4_1.set_xlim([0, 9])
414 ax4_1.grid()
415
416 ax4_2 = fig.add_subplot(122)
417 ax4_2.plot(totalfriksjonBehovXverdier, totalfriksjonBehovYverdier, c='blue'
            , zorder=2)
418 ax4_2.plot(totalfriksjonBehovXverdier, totalfriksjonBehovYverdier50, c='
            purple', zorder=2)
419 ax4_2.scatter(8, totalfriksjonBehov, c='red', s=2 ** 7, zorder=3)
420 ax4_2.scatter(8, totalfriksjonBehov50, c='red', s=2 ** 7, zorder=3)
421 ax4_2.scatter(6.5, totalfriksjonBehovDrift, c='salmon', s=2 ** 7, zorder=3)
422 ax4_2.scatter(beregnetOverhoyde, totalfriksjonBehovDrift50, c='salmon', s=2
            ** 7, zorder=3)
423 ax4_2.scatter(6.5-erfaringsverdiOverhoydeAvvik, totalfriksjonBehovErfaring,
            c='seagreen', s=2 ** 7, zorder=3)
424 ax4_2.scatter(beregnetOverhoyde-erfaringsverdiOverhoydeAvvik,
            totalfriksjonBehovErfaring50, c='seagreen', s=2 ** 7, zorder=3)
425
426 ax4_2.set_title('Totalfriksjon', fontsize=17)
427 ax4_2.set_ylabel('Totalfriksjonsverdi', fontsize=15)
428 ax4_2.set_xlabel('Overhøyde [%]', fontsize=15)
429 ax4_2.set_xlim([0, 9])
430 ax4_2.grid()
431
432 fig.legend(loc=8, ncol=5, fontsize=15)
433

```

```

434 # Sikkerhetsfaktor
435 fig = plt.figure(figsize=(9, 12))
436 fig.suptitle('Sensitivitet til sikkerhetsfaktor for friksjon for {} veg'.
              format(vegklasse), fontsize=20)
437 fig.subplots_adjust(left=0.09, bottom=0.114, right=0.97, top=0.914, wspace
                    =0.3, hspace=0.27)
438
439 ax5 = fig.add_subplot(211)
440 ax5.set_title('5-persentilverdier for ulike vinterdriftsklasser', fontsize
              =17)
441 ax5.plot(sikkerhetsfaktorXverdier, sikkerhetsfaktorYverdier, c='blue',
          label='Standard')
442 ax5.plot(sikkerhetsfaktorXverdier, sikkerhetsfaktorYverdier50, c='purple',
          label='50% Økning')
443 for i in range(6):
444     ax5.scatter(sikkerhetsfaktorBeregnet5persentil[i],
                totalfriksjonsverdier5persentil[i], c=totalfriksjonsfarger[i], zorder=5,
                s=2 ** 6, label=totalfriksjonsnavn[i])
445     ax5.scatter(sikkerhetsfaktorBeregnet5persentil150[i],
                totalfriksjonsverdier5persentil[i], c=totalfriksjonsfarger[i], zorder=5,
                s=2 ** 6)
446 ax5.set_ylabel('Totalfriksjonsverdi', fontsize=15)
447 ax5.set_xlabel('Sikkerhetsfaktor for friksjon', fontsize=15)
448 ax5.tick_params(right=True, left=True, bottom=True)
449 tick_spacing5_1 = 0.2
450 tick_spacing5_2 = 0.1
451 tick_spacing5_3 = 0.05
452 ax5.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing5_1))
453 ax5.xaxis.set_minor_locator(ticker.MultipleLocator(tick_spacing5_2))
454 ax5.yaxis.set_major_locator(ticker.MultipleLocator(tick_spacing5_3))
455 ax5.set_xlim([0.9, 3.8])
456 ax5.set_ylim([0, 0.55])
457 ax5.grid()
458 ax5.xaxis.grid(True, which='minor')
459
460 ax6 = fig.add_subplot(212)
461 ax6.set_title('15-persentilverdier for ulike vinterdriftsklasser', fontsize
              =17)
462 ax6.plot(sikkerhetsfaktorXverdier_15, sikkerhetsfaktorYverdier_15, c='
              orange', label='Standard 2014')
463 ax6.plot(sikkerhetsfaktorXverdier_15, sikkerhetsfaktorYverdier_15_50, c='
              olive', label='50 % økning 2014')
464 for i in range(6):
465     ax6.scatter(sikkerhetsfaktorBeregnet15persentil[i],
                totalfriksjonsverdier15persentil[i], c=totalfriksjonsfarger[i], zorder
                =5, s=2 ** 6)
466     ax6.scatter(sikkerhetsfaktorBeregnet15persentil_50[i],
                totalfriksjonsverdier15persentil[i], c=totalfriksjonsfarger[i], zorder
                =5, s=2 ** 6)
467 ax6.set_ylabel('Totalfriksjonsverdi', fontsize=15)
468 ax6.set_xlabel('Sikkerhetsfaktor for friksjon', fontsize=15)
469 ax6.tick_params(right=True, left=True, bottom=True)
470 tick_spacing5_1 = 0.2
471 tick_spacing5_2 = 0.1
472 tick_spacing5_3 = 0.05
473 ax6.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing5_1))
474 ax6.xaxis.set_minor_locator(ticker.MultipleLocator(tick_spacing5_2))

```

```
475 ax6.yaxis.set_major_locator(ticker.MultipleLocator(tick_spacing5_3))
476 ax6.set_xlim([0.9, 3.8])
477 ax6.set_ylim([0, 0.55])
478 ax6.grid()
479 ax6.xaxis.grid(True, which='minor')
480 fig.legend(loc=8, ncol=5, fontsize=13, mode='expand')
481
482 plt.show()
```

