



Norwegian University of  
Science and Technology

# A Feasibility Study of Assessing Bunkering Operations Through AIS Data

**Marion Aarsnes**

Marine Technology

Submission date: June 2018

Supervisor: Bjørn Egil Asbjørnslett, IMT

Co-supervisor: Thomas Mestl, DNV GL

Norwegian University of Science and Technology  
Department of Marine Technology





## **PROJECT THESIS IN MARINE TECHNOLOGY**

**AUTUMN 2018**

**For stud.techn. Marion Aarsnes**

### **A Feasibility Study of Assessing Bunkering Operations Through AIS Data**

#### **Background**

This thesis is intended to construct a framework for identifying bunkering operations, and utilise statistical analysis to establish an index to quantify such operations. Bunker is a generic term given to any fuel used to power ships, and a bunker operation is a process where a vessel is supplied with fuel for operational purpose. The bunker industry encompasses numerous suppliers and buyers, and is characterised by large volumes of fuel transitioning rapidly between owners. Traditionally, this industry has been subject to concealment and little public insight. This lack of transparency prevents regulation and integrity of the bunker industry.

#### **Objective**

The thesis is divided into two objectives; create a framework for identifying bunkering operations and establish an index to quantify bunkering operations. Embedded in the objectives is the sub-objective of conducting statistical analysis of bunkering operations.

#### **Tasks**

The candidate shall/is recommended to cover the following tasks in the project thesis:

- a. Describe the problem and why it is of interest.
- b. Review state-of-the-art within the topic, focusing on previous studies on Automatic Identification Systems (AIS) data.
- c. Create a framework for identifying bunkering operations.
- d. Analyse identified bunkering operations by applying statistical methods.
- e. Develop a quantifying index.
- f. Apply the index on a selection of operations to verify its purpose.
- g. Evaluate results and discuss methodology and shortcomings.

#### **General**

In the thesis the candidate shall present his personal contribution to the resolution of a problem within the scope of the thesis work.

Theories and conclusions should be based on a relevant methodological foundation that through mathematical derivations and/or logical reasoning identify the various steps in the deduction.

The candidate should utilize the existing possibilities for obtaining relevant literature.



The thesis should be organized in a rational manner to give a clear statement of assumptions, data, results, assessments, and conclusions. The text should be brief and to the point, with a clear language. Telegraphic language should be avoided.

The thesis shall contain the following elements: A text defining the scope, preface, list of contents, summary, main body of thesis, conclusions with recommendations for further work, list of symbols and acronyms, reference and (optional) appendices. All figures, tables and equations shall be numerated.

The supervisor may require that the candidate, in an early stage of the work, present a written plan for the completion of the work.

The original contribution of the candidate and material taken from other sources shall be clearly defined. Work from other sources shall be properly referenced using an acknowledged referencing system.

### **Deliverable**

The thesis shall be submitted electronically on DAIM:

- Signed by the candidate.
- The text defining the scope included.
- Computer code, input files, videos and other electronic appendages can be uploaded in a zip-file in DAIM. Any electronic appendages shall be listed in the thesis.

### **Supervision:**

Main supervisor: Prof. Bjørn Egil Asbjørnslett

Co-supervisor: Dr. Scient, Dipl. Ing. Thomas Mestl (DNV GL)

**Deadline: 11.06.2019**



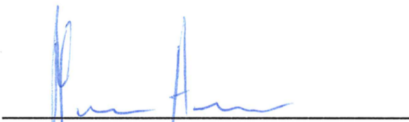
## Preface

This thesis constitutes the final part of my Master of Science degree with specialisation in Marine Systems Design at the Department of Marine Technology. The work was conducted during the spring semester of 2018 at the Norwegian University of Science and Technology (NTNU), under supervision from Professor Bjørn Egil Asbjørnslett and Dr. Scient, Dipl. Ing Thomas Mestl.

The thesis is a continuance of my project thesis "An Introduction to Assessing Bunkering Operations Through AIS Data", written in collaboration with student Daniel Wu. The project thesis is a result from exploring utilisations of AIS data by analysing the behaviour of vessels and bunker barges engaged in bunker operations, and the work was carried out during the autumn semester of 2017.

My motivation for the thesis is initially based on my interest in investigating potential utilisation of AIS data. With the ability to gather large amounts of data, the use of AIS data in studies of maritime transportation has become increasingly prevalent. In addition, available knowledge about the bunker business is scarce, making this an interesting field to explore. The idea for the project was brought up by Thomas Mestl from DNV GL, and the project was completed in cooperation with DNV GL. Thomas Mestl has provided me with necessary data and input throughout the process.

Trondheim, June 11, 2018



---

Marion Aarsnes



## Acknowledgement

I would like to thank the following persons for their great help during this project:

Professor Bjørn Egil Asbjørnslett, my supervisor, for guidance throughout the work with this thesis.

Dr.Scient, Dipl. Ing Thomas Mestl from DNV GL, my co-supervisor, for being an invaluable source of knowledge. I would also like to thank him for always being available to answer my questions, and for devoting a great amount of time to this thesis.

PhD candidate Carl Fredrik Rehn for following up on my progress.

Bjørnar Brende Smestad for providing me with help and guidance on the data handling part of the thesis.

Jens Ellingsen (DNV GL), John Stirling (World Fuel Services) and Bjørn Olav Oddland (VPS) for operational insight to the bunker business.

M.Aa.



## Summary

This thesis is a result of utilising Automatic Identification System data to construct a framework for identifying bunkering operations and conducting a statistical analysis of identified operations. The two objectives of the project are to identify bunkering operations and derive a benchmark that can quantify operation performance. Three relevant fields are studied to gain fundamental knowledge prior to attempting to achieve the objectives. 1) A brief study of the bunker supplier industry, 2) a literature review, and 3) a study of the basics of AIS data.

Knowledge on the bunker industry is to a great extent obtained from industry experts. Effects from lack of transparency are described - focusing on how this prevents monitoring and evaluation of the bunker process. Furthermore, reasons for increasing awareness are recognised. Literature was reviewed in previous work and is divided in two parts. First, literature is categorised according to topics and methods. Second, the most relevant literature is further assessed, where studies related to topics *Risk assessment* and *AIS data handling* are considered the most relevant. Two additional studies regarding mapping of traffic patterns and benchmarking were appraised at later stages. The conclusion of the literature study is that no studies explore the use of AIS data to evaluate bunkering operations, thus forming an unexplored academic field.

Three sub-problems attempt to achieve the objectives of the thesis. The first objective is covered by problem 1 in which a framework for identifying bunkering operations is constructed - resulting in a matching algorithm that matches ships with the barge that most likely performed a bunkering operation. Inputs are AIS data, measured fuel quality data from bunkerings and a list of officially approved bunker barges, restricted to Singapore. Output from the algorithm is bunkering operations that can be verified with high likelihood based on a certainty measure composed by proximity and alignment between ship and barge.

The second objective consists of problems 2 and 3. In problem 2, statistical analysis of identified bunkering operations is conducted. Aggregation and visualisation of data are done in the programming language Python, yielding numerous plots. This has mainly been restricted to scrutinising distributions of time spent at different stages during operations. In problem 3, results from the statistical analysis are utilised to establish an index that quantifies each bunkering operation with time spent before and after the bunkering operation as decisive parameters.

In conclusion, the algorithm is a "proof of concept" which proves that AIS data can be utilised to identify ship-to-ship operations. The index is a guide to evaluate operations,

though insufficient to quantify performance by suppliers. Recommendations for further work are that a Geohashed AIS table should be implemented in the matching algorithm to filter nearby vessels more efficiently, and the algorithm should be expanded to identify perpendicular operations. Parallel data with known fuel quality from bunkering operations should be compared with results from the proposed index. Hence, index parameters can be optimised, and validity be verified. Lastly, methods used in this thesis can be expanded to other types of ship-to-ship operations.

## Sammendrag

Denne oppgaven er et resultat av å bruke *Automatic Identification System* data til å lage et rammeverk for å identifisere bunkringsoperasjoner og utføre statistisk analyse av identifiserte operasjoner. De to målsettingene i prosjektet er å identifisere bunkringsoperasjoner, og utlede et mål for å karakterisere gjennomføringen av bunkringer. Tre relevante felt er studert for å få grunnleggende kunnskap før målene er forsøkt nådd; studie av 1) bunkerleverandørbransjen, 2) litteratur, og 3) grunnleggende kunnskap om AIS data.

Kunnskap om bunkringsindustrien er hovedsakelig hentet fra industriekspertter. Effekter av manglende innsyn i bransjen er beskrevet – med fokus på hvordan dette forhindrer overvåkning og evaluering av bunkringsprosessen. Interessen for å øke innsikten i bunkeringsbransjen er også utdypet. Litteratur ble gjennomgått i tidligere arbeid og er delt i to deler. Første del omfatter å kategorisere litteraturen etter omhandlende emner og metoder. I andre del er den mest relevante litteraturen nøyer gjennomgått, hvor studier relatert til *Risikoanalyse* and *Behandling av AIS data* er vurdert som de viktigste. I tillegg ble to studier vedrørende kartlegging av trafikkmønstre og måling av kvalitet gjennomgått i senere stadier. Konklusjonen er at ingen studier undersøker bruken av AIS data for å evaluere bunkringsoperasjoner eller andre skip-til-skip operasjoner, som medfører at denne oppgaven danner et nytt fagfelt.

Tre delproblemer forsøker å oppnå oppgavens målsettinger. Det første målet dekkes av problem 1 der et rammeverk for å identifisere bunkringsoperasjoner er konstruert – hvor resultatet er en matching-algoritme som matcher skip med det bunkringsskipet som mest sannsynlig har levert drivstoff til skipet. Input til algoritmen er AIS data, målt bunker-kvalitetsdata fra bunkringer og en liste over offisielt godkjente bunkerskip, hvor dataen er begrenset til Singapore. Output fra algoritmen er bunkringsoperasjoner som kan verifiseres med høy sannsynlighet, basert på et mål estimert fra avstand og vinkel mellom skip og bunkringsskip.

Det andre målet dekkes av problem 2 og 3. I problem 2 utføres en statistisk analyse av identifiserte bunkringsoperasjoner. Gruppering og visualisering av data gjøres i programmeringsspråket Python og resulterer i en mengde figurer. Analysen har i hovedsak vært begrenset til å undersøke fordeling av tid brukt på ulike stadier under operasjonen. I problem 3 blir resultater fra statistisk analyse benyttet for å etablere en indeks som kvantifiserer (gir en karakter til) hver bunkringsoperasjon, med tid brukt før og etter operasjonen som avgjørende parametere.

Opgaven konkluderer med at algoritmen et «bevis av konsept» som viser at AIS data kan benyttes for å identifisere skip-til-skip operasjoner. Indeksen er en veiledning for

å evaluere operasjoner, men er ikke omfattende nok til å kvantifisere troverdigheten til leverandører. Anbefalinger for videre arbeid er at en Geohashed AIS-datatabell implementeres i matching-algoritmen for å filtrere nærliggende fartøy mer effektivt, og algoritmen burde utvides til å identifisere perpendikulære operasjoner. Parallelle data med kjent drivstoffkvalitet fra operasjoner burde sammenlignes med resultater fra den foreslåtte indeksen, for å optimalisere indeksparametere og bekrefte at indeksen gir resultater som kan brukes i beslutningstaking. Avslutningsvis kan metoder brukt i denne oppgaven utvides til andre skip-til-skip operasjoner.



# Contents

- Preface . . . . . ii
- Acknowledgement . . . . . iv
- Summary . . . . . vi
- Sammendrag . . . . . viii
  
- 1 Introduction . . . . . 1**
- 1.1 Background . . . . . 1
- 1.2 Previous Research . . . . . 3
- 1.3 Objectives . . . . . 4
- 1.4 Approach . . . . . 4
- 1.5 Contributions . . . . . 5
- 1.6 Limitations . . . . . 5
- 1.7 Outline . . . . . 5
  
- 2 Problem Description . . . . . 7**
- 2.1 The Bunker Operation . . . . . 7
  - 2.1.1 Involved Parties . . . . . 8
- 2.2 Creative Bunker Practices . . . . . 9
- 2.3 Barges in the Centre of Attention . . . . . 12
- 2.4 From an Academic to an Industry Perspective . . . . . 12
- 2.5 Constraints . . . . . 13
  - 2.5.1 Geographical Area and Time Restriction . . . . . 13
  - 2.5.2 Operational Restrictions . . . . . 15
  - 2.5.3 Officially Approved Bunker Barges . . . . . 16
  - 2.5.4 Fuel Quality Samples . . . . . 16
- 2.6 Problem Formulation . . . . . 16
  
- 3 Literature Review . . . . . 17**
- 3.1 Overview of Literature on AIS Data . . . . . 18
  - 3.1.1 Appointing Relevant Fields of Research . . . . . 19
- 3.2 Assessment of Relevant Literature . . . . . 19

3.2.1	Treating Data and Data Gaps . . . . .	19
3.2.2	Analysis of Data . . . . .	20
3.2.3	Utilising AIS Data . . . . .	21
3.2.4	Benchmarking . . . . .	24
3.3	Conclusion of the Literature Survey . . . . .	24
<b>4</b>	<b>Data Sources</b>	<b>27</b>
4.1	AIS Data . . . . .	27
4.1.1	AIS Basics . . . . .	27
4.1.2	Guidelines for Use of AIS . . . . .	28
4.1.3	Use of AIS Data in Industry . . . . .	28
4.1.4	AIS Data Used in Bunkering Matching Algorithm . . . . .	29
4.1.5	Quality Issues . . . . .	32
4.2	VPS data . . . . .	34
4.3	Officially Approved Bunker Barges . . . . .	34
<b>5</b>	<b>Applied Methods</b>	<b>35</b>
5.1	Interpolation . . . . .	35
5.2	Statistical Analysis . . . . .	36
5.2.1	Summarising . . . . .	36
5.2.2	Kernel Density Estimation . . . . .	37
<b>6</b>	<b>Problem 1: Identifying Bunker Operations</b>	<b>39</b>
6.1	Defining the Bunkering Operation . . . . .	39
6.1.1	Bunker Operation Time Line . . . . .	39
6.1.2	Waiting Time, Bunkering Time and Post Bunkering Time . . . . .	40
6.1.3	Bunker Operation as Seen by AIS Data . . . . .	40
6.2	Uncertainties in the Bunkering Process . . . . .	41
6.3	Matching Algorithm . . . . .	42
6.3.1	Previous Work . . . . .	42
6.3.2	Algorithm Structure . . . . .	43
6.3.3	Matching Methodology . . . . .	45
6.3.4	From Distance to Certainty . . . . .	48
6.3.5	Limitations . . . . .	50
6.3.6	Output Data . . . . .	51
6.3.7	Discussion of Algorithm Structure . . . . .	53
<b>7</b>	<b>Problem 2: Analysis</b>	<b>55</b>
7.1	Introduction . . . . .	55
7.2	Requiring Minimum Confidence Measure . . . . .	56

7.3 Ship Types . . . . .	58
7.4 Bunkering Time . . . . .	62
7.5 Bunkering Ships . . . . .	67
7.6 Time Before and After Bunkering . . . . .	71
7.7 Discussion of Analysis . . . . .	74
<b>8 Problem 3: A Quality Index for Bunkering Times</b>	<b>75</b>
8.1 Bunkering Process Time Line . . . . .	75
8.2 Waiting Time and Post Bunkering Time . . . . .	76
8.3 Proposing an Index . . . . .	77
8.4 Quantifying Operations . . . . .	80
8.5 Discussion of Index . . . . .	81
<b>9 Discussion</b>	<b>83</b>
<b>10 Conclusions</b>	<b>85</b>
10.1 Concluding Remarks . . . . .	85
10.2 Recommendations for Further Work . . . . .	86
<b>Bibliography</b>	<b>88</b>
<b>A Acronyms</b>	<b>ii</b>
<b>B Flow Chart of the Study</b>	<b>iv</b>
<b>C CSV File</b>	<b>vi</b>
<b>D Grouping of Literature</b>	<b>viii</b>
<b>E Python code</b>	<b>x</b>
E.1 Distribution of Confidence Mean . . . . .	xi
E.2 Distribution of Ship Types . . . . .	xiii
E.3 Distribution of Ship Lengths . . . . .	xv
E.4 Distribution of Lay Time for Ships . . . . .	xvii
E.5 Distribution of Bunker Barge Lengths and Time . . . . .	xxiv
E.6 Distribution of Lay and Bunkering Time for All Operations . . . . .	xxviii
E.7 Distribution of Time for Ship Types . . . . .	xxxviii
E.8 Distributions Divided by Ship Lengths . . . . .	xlvi
E.9 Timeline Plot of Three Operations . . . . .	lviii
E.10 Normalised Plots of Various Parameters . . . . .	lxi
E.11 Histogram and Scatter Plot of Waiting Time and Post Bunkering Time . . . . .	lxix
E.12 Normalise Waiting Time and Post Bunkering Time . . . . .	lxxiv

E.13 Create Index Plot . . . . . lxxviii  
E.14 Calculate Index . . . . . lxxx  
E.15 List of Electronic Appendages . . . . . lxxxiv

# List of Tables

- 4.1 AIS message components and description . . . . . 30
- 4.2 AIS message components used in matching algorithm . . . . . 30
  
- 6.1 Parameters distinguishing bunkering at anchorage and berth . . . . . 41
- 6.2 Algorithm restrictions and values . . . . . 45
- 6.3 Certainty measure for three barges . . . . . 50
- 6.4 Bunker barge matching result: components and description from algorithm  
output . . . . . 53
  
- 7.1 Expressions and definitions . . . . . 56
- 7.2 Ship types used in analysis . . . . . 59
- 7.3 Median lay time . . . . . 61
- 7.4 Bunkering . . . . . 65
- 7.5 Median bunkering time per ship type . . . . . 69
- 7.6 Median Waiting Time (WT) and Post Bunkering Time (PBT) . . . . . 72
  
- 8.1 Waiting Time (WT) and Post Bunkering Time (PBT) parameters . . . . . 78
- 8.2 Calculated index (QBT) . . . . . 80

# List of Figures

- 2.1 Satellite images showing examples of bunkering at berth (Google (2017a)) and at anchorage (Google (2017b)) . . . . . 8
- 2.2 Illustration from Mestl (2015) (left) and MarineInsight (2015) (right) of methods for under-supplying fuel . . . . . 11
- 2.3 Value for money based on short-lifting of bunker fuel (HFO) per continent (Mestl (2015)). Note that 56% of bunkering operations in Asia and Oceania are considered as loss of money. . . . . 13
- 2.4 Map of Singapore (Google (2018)) . . . . . 14
- 2.5 Restricted geographical area . . . . . 15
  
- 3.1 Approach of literature review . . . . . 17
- 3.2 Distribution of topics . . . . . 18
- 3.3 Distribution of methods . . . . . 18
  
- 4.1 Geographical area under consideration . . . . . 29
- 4.2 Location of antenna given by parameters A, B, C and D in AIS messages . . . . . 32
- 4.3 Ships and bunker barges with close proximity (Mestl (2015)) which can cause interference . . . . . 32
- 4.4 GPS and AIS antenna positions . . . . . 33
  
- 5.1 Illustration of mapping data into two groups . . . . . 36
- 5.2 Illustration of a histogram . . . . . 37
- 5.3 Kernel Density Estimation . . . . . 38
  
- 6.1 Bunkering time line. Note that there may be some Waiting Time and time after bunkering; Post Bunkering Time . . . . . 40
- 6.2 Illustration of area surrounding a ship with radius of 200 meters from which barge candidates are retrieved . . . . . 42
- 6.3 Movement for one bunker barge and six ships in the same geographical area (Wu and Aarsnes (2017)) . . . . . 43
- 6.4 Flow chart of the algorithm . . . . . 44

6.5	Alignment of ship (white) and barge (grey) by a local coordinate system . . . .	46
6.6	Effects from aligning coordinate systems. Global coordinate (left) to local coordinate (right) . . . . .	46
6.7	Ship aligned with barge . . . . .	47
6.8	Ship aligned with several barges . . . . .	47
6.9	Definition of ship and barge alignment area used to calculate the certainty measure. $A_{B1}$ if fully within the area, while $A_{B1}$ is fully outside. . . . .	48
6.10	Ship and barge alignments with respective certainty measures $C_M$ . . . . .	49
6.11	Barge positioning. B1 is fully aligned with the ship and fully within the alignment area, while B3 is positioned perpendicularly to the ship. . . . .	49
6.12	Reducing the problem from geographical coordinates with speed to identifying when $C_M$ is equal to 1 . . . . .	50
6.13	Identified bunker operations in Singapore (2017) plotted in <i>Tableau</i> . . . . .	51
6.14	Designated anchorage areas in Singapore (MPA (2017)). Note that identified bunkering operations are concentrated in the designated areas. . . . .	52
7.1	Bunkering operations identified for one barge plotted in <i>Tableau</i> . . . . .	56
7.2	Distribution of confidence mean. Several operations have values close to 0. . . . .	57
7.3	Distribution of confidence mean above 0.7, 95.3% of all operations have a certainty measure of more than 0.9 . . . . .	58
7.4	Distribution of ship types. The three first ship types constitute 67% of the bunkerings . . . . .	58
7.5	Distribution of ship lengths, median = 230 meters . . . . .	59
7.6	Histogram (left) and normalised plot (right) of lay time all ships. Observe that 75% of all ships had lay time of 23.55 hours or less. . . . .	60
7.7	Histogram (left) and normalised plot (right) of anchorage time. Observe that 75% of ships at anchorage had anchorage time of 23.22 hours or less. . . . .	60
7.8	Histogram (left) and normalised plot (right) of berth time. For comparison, anchorage time was superimposed in the normalised plot. Observe that 75% of ships at berth had berth time of 27.42 hours or less, and that anchorage and berth follow a similar pattern with more short stays at anchorage. . . . .	61
7.9	Histogram (left) and normalised plot (right) of anchorage time for small and large vessels . . . . .	62
7.10	Histogram (left) and normalised plot (right) of berth time for small and large vessels. Note that the normalised plot is transversed from the equivalent plot at anchorage. . . . .	62
7.11	Histogram (left) and normalised plot (right) of barge lengths. Median barge length = 90 meters. . . . .	63

7.12 Histogram (left) and normalised plot (right) of bunkering time for all barges. Median bunkering time = 6.5 hours. Note that 90% of all barges have bunkering times of less than 14 hours . . . . .	63
7.13 Histogram (left) and normalised plot (right) of bunkering time at anchorage . . . . .	64
7.14 Histogram (left) and normalised plot (right) of bunkering time at berth. For comparison, anchorage time was superimposed in the normalised plot. Note that bunkering time at anchorage is usually longer than at berth. . . . .	64
7.15 Histogram (left) and normalised plot (right) of lay time and bunkering time for all operations . . . . .	65
7.16 Histogram (left) and normalised plot (right) of lay time and bunkering of anchorage time. Note that the distribution is almost identical to general lay time for all vessels in the figure above. . . . .	65
7.17 Histogram (left) and normalised plot (right) of lay time and bunkering of berth time . . . . .	66
7.18 Scatter plot of lay time vs bunkering time for all operations. Most operations are located in the lower left corner. . . . .	66
7.19 Histogram (left) and normalised plot (right) of bunkering time for small and large vessels at anchorage. . . . .	67
7.20 Histogram (left) and normalised plot (right) of bunkering time for small and large vessels at berth. Note that distributions for anchorage and berth are quite similar, but bunkering time is longer for <i>large</i> ships. . . . .	67
7.21 Bunkering time for three ship types. Crude Oil Tankers have a longer tail towards longer bunkering time. . . . .	68
7.22 Lay time for container ships, bulk carriers and crude oil tankers. Note that LNG tankers have the longest lay time. . . . .	69
7.23 Bunkering time for container ships, bulk carriers and crude oil tankers. Note that distributions of lay time and bunkering time per ship type is similar. . . . .	70
7.24 Distribution of ship length for three ship types. Note that bulk carriers have three dominating ship lengths; 190, 240 and 300 meters. Crude oil tankers are generally long, with a group of smaller vessels around 250 meters. . . . .	70
7.25 Waiting Time for all ships . . . . .	71
7.26 Waiting Time per bunkering type. Note that waiting time is similar independent of bunkering type. . . . .	71
7.27 Post Bunkering Time for all ships . . . . .	72
7.28 Post Bunkering Time per bunkering type. Note that post bunkering time is usually longer at berth. . . . .	72
7.29 Normalised distribution of WT and PBT. Note that Post Bunkering Time generally is longer than Waiting Time. . . . .	73



7.30 25% and 75% quantiles of Waiting Time and Post Bunkering Time . . . . . 73

8.1 Bunkering and anchorage time lines for three operations: Bunkering at the end, middle and start of anchorage. . . . . 75

8.2 General scatter plot . . . . . 76

8.3 Scatter plot of Waiting Time vs Post Bunkering Time. Note that most operations are located in the sections characterised as good. The linear graph is subject to larger variability at longer Waiting Time and Post Bunkering Time. Pearsonr = 0.2 indicates low correlation between points. . . . . 77

8.4 Index plot illustrating the linearised relationship between good and bad WT and PBT . . . . . 79

8.5 Histogram (left) and normalised plot (right) of QBT. Note that a larger fraction of operations at anchorage have lower scores than at berth. . . . . 81



# Chapter 1

## Introduction

### 1.1 Background

In commercial shipping, bunker expenditure can constitute a large fraction of operational costs (OPEC). Bunker is a generic term given to any fuel used to power ships, and a bunker operation is a process where a vessel is supplied with fuel for operational purpose. The bunker industry encompasses numerous suppliers and buyers, and is characterised by large volumes of fuel transitioning rapidly between owners. Traditionally, this industry has been subject to concealment and little public insight. This lack of transparency prevents regulation and integrity of the bunker industry.

In general, fuel is delivered in volume but paid per mass. The conversion is done based on the fuel density. The majority of delivered fuel is Heavy Fuel Oil (HFO) with a density ranging from approximately 800 to 1010 kg/m<sup>3</sup> is delivered to vessels. Density is stated by the supplier, and the ship owner/charterer can take fuel samples and have them analysed in a lab for true density. Research suggests a bias towards overstatement of the density. Systematic over-reporting is well known within the shipping industry and is assumed to have been practised since the late 1980s (BunkerWorld (2018)). Over-reporting of bunker density leads to reduction in delivered bunker, meaning that the supplier receives payment for a greater amount than supplied. This is offered limited attention from shipping companies and charterers as the fraudulent amount is assumed to be negligible. Nonetheless, the aggregated amount of lost bunker is of significant value (DNV Research & Innovation (2012)), hence the reduction in OPEC is potentially high. This highlights the need for increasing awareness and knowledge about the bunker industry.

Automatic Identification System (AIS) was originally developed as a tool to avoid vessel collision. It has become increasingly prevalent as a multipurpose tool with a wide poten-

tial of use. AIS data contains ship movement and technical data transmitted at irregular intervals, and is an aid for vessel monitoring.

The motivation for this thesis is to investigate the possibility of using AIS data to identify bunkering operations and characterise the integrity of bunker suppliers. With the ability to track vessel movement it may be possible to identify bunker operations, and a statistical analysis may identify normal behaviour during bunker operations and any deviations. This may launch a new application area for AIS data. However, it is worth emphasising that to the knowledge of the author, no attempt has been made to utilise AIS for this purpose, and it is not given that the objectives of this thesis can be reached.

## **Problem Formulation**

This is essentially a two-part study which can be divided into the following problems.

### **1. Identifying potential bunker operations:**

This problem encompasses identification of bunkering operations based on AIS data, fuel quality data from Veritas Petroleum Services (VPS) and a list of bunker barges which are officially approved by the port authority of Singapore. Identification shall be accomplished through a matching algorithm.

### **2. Rating of bunker fuel delivery times:**

The primary goal of this problem is to derive a benchmark that quantifies bunker delivery time with respect to anchoring time, i.e. characterising delivery times within the interval of  $[0,1]$ . Based on this measure, ratings can be given for single bunker barges, barge companies with several barges or entire ports. This would constitute a measure of credibility of bunker suppliers.

## **What Remains to be Done**

The literature review showed that the number of AIS-related studies has expanded rapidly over the last years, although AIS data is mostly utilised on a low-resolution level, e.g. on movement patterns from one position to another. Available research on a higher resolution level is limited, which leaves the author to confirm or reject that this data can be utilised in analysis of ship-to-ship operations.

Impeded transparency in the bunker business and lack of data prevent investigation of this industry. Research specifically focusing on deviating bunker operations is basically non-existent, in this respect this thesis attempts to explore a new academic branch. Due to lack

of research and restricted time, this thesis merely scratches the surface of possibilities of could be further developed.

## 1.2 Previous Research

As of today, no studies are conducted to investigate bunkering operations nor ship-to-ship operations through AIS. However, numerous studies have been published on use of AIS data. The literature review has therefore been devoted to these studies, especially those regarding risk assessments, providing a basis for further utilisation of AIS data. Bole et al. (2014) has been studied to get a general understanding of the regulation around and technical aspects of AIS data.

AIS data is sampled at varying time intervals, and data gaps may be present due to several reasons. Shelmerdine (2015) proposes to close these gaps by linear interpolation. Goldsworthy (2017) has also investigated the accuracy of interpolating, and based on his results Dijkstra's algorithm should be used to improve the accuracy. However, this is proposed to avoid generating points on land, and the AIS data under survey in this thesis will be enclosed to small areas with no risk of interpolating values crossing land, hence linear interpolation will be sufficient for identification of bunkering operations.

Ferrà et al. (2018) propose a method for mapping fishing activity within a defined geographical scope based on filtering data and applying K-means cluster analysis. Combining a similar approach with results obtained by Wu and Aarsnes (2017) will contribute to constructing the framework for a matching algorithm identifying bunkerings. Cluster analysis will not be implemented in the framework for identifying bunkering operations, but filtering the data by operational restrictions and other parameters is considered beneficial.

Identified operations will be subject to statistical analysis. Kernel Density Estimation as suggested by Ristic et al. (2008) will be implemented by Python functions. Using Grid-Based Analysis (Altan and Otay (2017)) is deemed inexpedient for this thesis. Furthermore, results from the analysis will contribute to establishing a quantifying index. Pan et al. (2012) and Qu et al. (2011) have proposed two methods for quantifying risk in maritime shipping areas. This was done by incorporating parameters such as speed and rate of turn into an index that can indicate relative dangerous shipping areas, which can be of inspiration in this thesis.

Benchmarking methodology as presented by Anfindsen et al. (2012) is not transferable to this thesis, as they base their approach on measured parameters obtained from supplied

fuel. It is however an important contributor of knowledge regarding bunkering business practises, as well as providing insight to alleged bunkering malpractice. In addition, important contributors to knowledge regarding bunker operations and the industry are DNV GL and VPS.

### 1.3 Objectives

The main objectives of this thesis are

1. Develop a framework for identifying potential bunker operations.
2. Perform data exploration on bunkering data.
3. Propose criteria for a benchmark quantifying bunker delivery time with respect to lay time.

### 1.4 Approach

A matching algorithm initiated and further developed as a cooperation between the author, student Daniel Wu and Thomas Mestl from DNV GL is used to identify bunkering operations with high certainty, based on proximity between vessels and length of time in proximity. Inputs to the algorithm are AIS data, fuel sample data from VPS and a list of bunker barges officially approved by the port authority of Singapore. Output data from the algorithm are Comma-Separated Values files containing data for identified bunkering operations. The data is aggregated and visualised in various forms, and statistical methods such as Kernel Density Estimators are applied to conduct an analysis. The result from the statistical analysis is the establishment of an index quantifying each bunkering operation with regards to waiting time and post bunkering time recorded for each bunkering operation. A flow chart of the approach can be found in appendix B.

Emphasising ethical issues related to this thesis, it is worth mentioning that a few arise when dealing with vessel identities. First and foremost, analysing motion patterns and possible bunker deviations with known International Maritime Organization (IMO) numbers may incriminate actors without proper evidence. Therefore, no information that can lead to identification of vessels or other involved parties will not be included. Secondly, AIS data used for this project thesis is provided for ships and barges without their knowledge. These parties may not grant permission to use the data, and therefore data is merely included for illustrative purposes without possibility of tracing the data back to respective

vessels or owners. In addition, due to confidentiality, neither the matching algorithm nor VPS fuel sample data are included in the thesis.

## 1.5 Contributions

The thesis will hopefully contribute to increasing knowledge and awareness about the bunker industry, along with emphasising the need of establishing integrity between suppliers and buyers. In addition, this is, to the knowledge of the author, the first time bunkering activities have been scrutinised with regards to deriving lengths of anchorage and bunkering.

On a technical level, this thesis investigates the possibility of utilising AIS data on a high-resolution level to assess ship-to-ship operations. AIS data has previously primarily been exploited on a low-resolution level, whereas this thesis takes a leap into investigation on a micro level.

## 1.6 Limitations

Limitations associated with this study are mainly related to availability of data. Key inputs of this work are fuel quality data from VPS and AIS data provided by DNV GL from a geographic area surrounding Port of Singapore. The data spans over one year (2017), and together yield data for approximately 19 000 bunker samples. Collected VPS samples are restricted to bunker operations involving bunkering of HFO, hence bunkering of other fuel types are not investigated. In addition, all operations are performed by bunker barges officially approved by the port authority of Singapore.

Fuel samples and AIS data utilised in this thesis originate from two individual parties, and consequently the full data is not a complete set of bunkering operations and measured quality. Operations subject to data exploration are therefore restricted to operations available in VPS data.

## 1.7 Outline

- **Chapter 2 - Problem Description:** Establishes the motivation behind the thesis through an elaboration of the bunker business and various methods for under-supplying

bunker. The reader is also presented with an industry perspective which underscores the importance of increased knowledge within this field.

- **Chapter 3 - Literature Review:** Presents literature reviewed for the purpose of establishing the advancement within the field of this thesis and identify relevant fields of study for the work. A major part of the literature was revised during the autumn of 2017. Relevant articles are elaborated with a discussion of potential use of methods they suggest.
- **Chapter 4 - Data Sources:** Addresses fundamentals of AIS and the basic concepts for use of AIS data. The AIS data foundation utilised in this thesis is described, and a brief discussion of quality issues related to AIS data is given. Furthermore, data from VPS and officially approved bunker barges are explained.
- **Chapter 5 - Data Exploration:** Introduces methods applied for creating the matching algorithm, and in the statistical analysis of bunkering operations retrieved by the algorithm.
- **Chapter 6 - Problem 1: Identifying Bunker Operations:** Lays the foundation for the analysis conducted in the thesis. Assumptions and framework laid to ground are presented, and the matching algorithm is outlined. The purpose of the matching algorithm is to identify bunkering operations. The algorithm was created as a cooperation between the author, student Daniel Wu and co-supervisor Thomas Mestl.
- **Chapter 7 - Problem 2: Analysis:** Operations identified by the matching algorithm are aggregated and visualised in various forms.
- **Chapter 8 - Problem 3: A Quality Index for Bunkering Times:** Presents the quality index derived for characterising bunker operations, in addition to calculation of the index for a selection of operations.
- **Chapter 9 - Discussion:** Presents an overall discussion, in addition to the more specific discussions in chapters 4, 5 and 6.
- **Chapter 10 - Conclusions:** Outlines the conclusion of the thesis, and recommends approaches for further work.



# Chapter 2

## Problem Description

This chapter outlines the general bunkering operations and creative bunker practices that deviate from standard procedure. In addition, it presents the academic interest in investigating this field, and the scope of the study is narrowed down. Finally, questions this thesis aims as answering are presented. Theory is mainly based on statements by industry experts and the project thesis *An Introduction to Assessing Bunkering Operations Through AIS Data* by Wu and Aarsnes (2017). In addition, the study by Anfindsen et al. (2012) has provided helpful insight.

### 2.1 The Bunker Operation

Bunker operations involve two parties, namely the supplier and the receiver. The operation is initiated by a demand from the receiver in which the shipowner or the charterer of the ship places an order specifying type and amount of fuel. A commitment between the parties is established, and a time slot and location for the bunkering is usually appointed where the receiver anchors up and waits for the supplier. When the supplier arrives at the predetermined location, it connects to the receiving vessel and initiates bunkering. A quality sample is sometimes taken from the supplied bunker, and both parties must sign a document stating that the operation is completed as agreed before the supplier detaches from the receiver.

The operation can take place while the ship is either at anchor or at berth as displayed in figure 2.1, and the bunker procedure can be summarised as follows:

- Bunker barge moves alongside the ship
- Bunker barge hose connects with fuel tank manifold

- Measuring of bunker quantities at barge and ship before bunkering start
- Commencement of bunkering
- Completion of bunkering
- Measuring of bunker quantities at barge and ship at end of bunkering
- Paperwork and related procedures confirm agreement of transferred bunker quantity
- Disconnection of barge hose
- Bunker barge leaves the ship

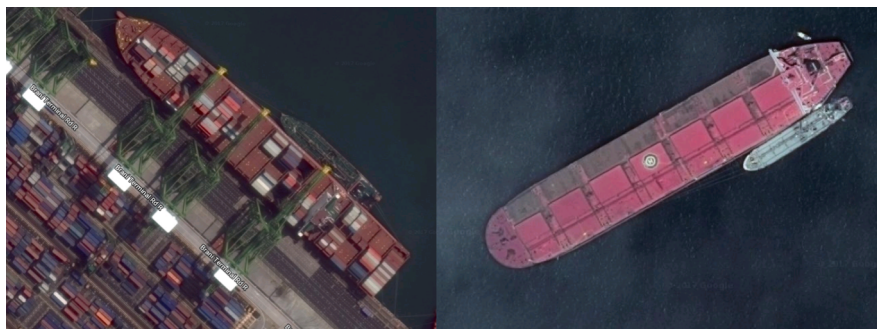


Figure 2.1: Satellite images showing examples of bunkering at berth (Google (2017a)) and at anchorage (Google (2017b))

Paperwork and related procedures are in place to ensure documentation of supplied bunker and compliance with requirements. IMO requirements oblige vessels to document bunker operations through Bunker Delivery Notes (BDN). Wärtsilä (2017) describes bunker delivery notes as follows:

*The standard document required by Annex VI of MARPOL which contains information on fuel oil delivery: name of receiving vessel, port, date, data of a supplier, quantity and characteristics of fuel oil. Every BDN is to be accompanied by a representative sample of the fuel oil delivered. Fuel oil suppliers are to provide the bunker delivery note. The note is to be retained on the vessel, for inspection purposes, for a period of three years after the fuel has been delivered.*

### 2.1.1 Involved Parties

Involved parties of a bunker operation can be grouped into a supplying side and a receiving side. The receiving side is the shipowner or the charterer. If the ship is chartered, the

charterer normally orders and finances the fuel. The supplying side is the bunker supplier who either privately owns or charters bunker barges to deliver fuel from refineries to customers. Chartered barges are often owned by independent bunker barge companies.

## 2.2 Creative Bunker Practices

DNV Research & Innovation (2012) emphasises the lack of performance benchmarks for fuel suppliers. Benchmarks that exist are of fail or pass format, and it is suggested that more nuanced criteria can reflect the quality range among bunker suppliers in a more transparent manner. Among the bunker suppliers there are major refineries involved in a wide range of the value chain, but also smaller actors that operate in the end of the value chain. DNV Research & Innovation (2012) mentions instances where fuel from small bunker providers has been of poor technical quality. The current state of regulation and benchmarking can be said to impede transparency, and some methods exploiting this are presented in the following.

### Delivered in Volume, Paid in Weight

Bunker is delivered in volume, but paid in weight. This makes fuel density one of the most essential properties when determining the monetary value. The conversion in equation 2.1 creates an opportunity for one of the parties to gain on the transaction by stating an inaccurate fuel density.

$$density \cdot volume = mass \quad (2.1)$$

For each delivery the fuel mass is calculated based on the fuel density stated in the BDN. An overstatement of the actual density will favour the supplier as the buyer will pay for more bunker than received - *density short-lifting*. Occurrence of the opposite, an understatement of fuel density, *density long-lifting*, will favour the buyer.

### Split Interests

The main interest of the crew and shipowner is to be on schedule to get paid by the charterer. Bunker is ordered and paid by the charterer, and consequently the crew on board might have limited interest in monitoring the bunker procedure or having a dispute with

the bunker barge regarding the amount that was supplied, as long as the supplied quantity is sufficient for the next leg and the quality is acceptable.

### **Short-Lifting - Stated Bunker Density Deviates From Delivered Density**

Fuel is sampled during bunker operations by filling a container attached to the fuel tank. Short-lifting can be practised by pumping high density fuel at a low rate to fill the sample. When the sample is sufficiently filled, the pump capacity can be raised to pump low density fuel into the customer tank. Thus, the density of the drip sample will not be representative for the density of the received fuel.

DNV Research & Innovation (2012) suggests that if oil density was measured with the intention to give accurate density measurements (unbiased), over-reporting and under-reporting would balance out. However, after systematic analysis of the DNV Petroleum Service (DNVPS) fuel sample database, DNV Research & Innovation has discovered ongoing systematic over-reporting of the density since the late 1980s. Seemingly small inaccuracies in density reporting can result in a significant difference in monetary terms, as illustrated in DNV Research & Innovation (2012):

*A slight over-reporting of the density quickly leads to a financial gain for the seller. For instance, reporting 977 kg/m<sup>3</sup> instead of the actual 960 kg/m<sup>3</sup> with a bunkering volume of 2000 m<sup>3</sup> gives rise to a difference of about 34 Metric Tonnes, worth about US\$ 20,000 at price of US\$ 600/MT in 2010.*

Analysis of over 50,000 samples of fuel oil tested by DNVPS in 2011 concluded with an average over-reporting of 0.6 kg/m<sup>3</sup>. With a global average bunkering of 965 Metric Tonnes (MT) per lift<sup>1</sup>, over-reporting can equal an average loss of about 0.6 MT per 1,000 MT lifted.

*There is no consensus figure for global fuel oil consumption, but assuming the total is 350 million tonnes per annum, and 80% was heavy fuel oil (HFO), that would be 280 million MT of HFO. Using a global average price of 641 USD/MT for 2011, and a loss of 0.6 MT per 1,000 MT, that would add up to 107.7 million USD of HFO paid for, but not delivered.*

### **Questionable Fuel Origins**

Some suppliers may alter the composition in fuel delivered to ships. An example is supplying low value products from refineries posing as HFO. In some cases, the suppliers reach

---

<sup>1</sup>Per bunkering

beyond this and mix the fuel with waste from pharmacies, cooking oil or used lubricants. Even remnants from car tires have been discovered in supplied fuel Mestl (2015). A growing concern in Asia is the increase in piracy with the aim of stealing fuel, mixing it with legally obtained fuel and selling it to buyers. As stated by McCauley (2018):

*Pirated oil is often mixed with legally obtained oil at sea in vessels referred to as mother ships, and it is difficult — if not impossible — to discern whether a given oil supply has been illegitimately obtained. The mixed oil, then, is resold to buyers whose owners, captains or crews may be ignorant of the fuel questionable origin.*

### Deviation in Stated Delivered Volume From Actual Delivered

Other documented methods of over-reporting involve inaccurate reporting of quantity of delivered volume. Methods include tilting the bunker barge to obtain various tank readings as displayed in figure 2.2a, using tampered measure devices or delivering *cappuccino* bunker, one of the most common and widely used ‘malpractice’ in the bunkering industry to-date. The *Cappuccino* effect may be described as frothing or bubbling effect caused by compressed air blown through the delivery hose causing the fuel to foam. The aerated bunkers will give the impression that the fuel is delivered as ordered, but when the entrapped air in suspension settles out - visible by bubbles on the surface (see figure 2.2b), - tank measurements will show significantly lower fuel levels (MarineInsight (2015)). There is a lack of quantitative data documenting such methods, thus there are no benchmarks to regulate delivered volume.



(a) Tilted bunker barge



(b) Visible bubbles from Cappuccino effect

Figure 2.2: Illustration from Mestl (2015) (left) and MarineInsight (2015) (right) of methods for under-supplying fuel

## **Time as Leverage**

Another feature impeding transparency is using time as leverage. Ships or charters are often subject to time pressure due to deadlines for shipment, and maintaining a time schedule is more important than avoiding short-lifting. A supplier can use time pressure to their advantage by deliberately arriving late to a scheduled operation or prolonging it.

In the case of suspicion of over-reporting, the chief engineer of the receiving vessel can refuse to sign the BDN. However, the bunker barge can in turn refuse to detach itself from the ship to prevent it from continuing its journey. In this case, time pressure can make the crew accept and neglect suspicions of over-reporting to maintain schedule. Breaking off an uncompleted operation leaves the receiver to pay for a larger amount than supplied.

## **2.3 Barges in the Centre of Attention**

The methods for impeding transparency presented above, all involve the supplier as the deceiver of the supplied fuel quantity and quality. Some suppliers systematically exploit these methods for monetary gain, but it is worth emphasising that individuals on both the supplier and receiver sides can gain by reporting incorrect supplied amounts. For example, individuals on each party can receive a kickback<sup>2</sup> for confirming the integrity of the operation by signing required documents.

The supplier is not the sole actor motivated to achieve monetary gain, but it is the supplier that will be at the centre of attention in this thesis. The supplier serves as a bottleneck for supplied bunker, and combining this with utilising time as leverage forms the core of the thesis.

## **2.4 From an Academic to an Industry Perspective**

In general, bunker suppliers are hesitant to give information to both industry and academia regarding their operations. Data on bunker prices is a scarce resource, and usually one can only find average prices in ports. Prices stated by individual bunker suppliers are only available on request and often unreliable, and; what may seem as a bargain can be the opposite, if either density or quality of the fuel is poorer than claimed. This lack of data

---

<sup>2</sup>Commission paid to the bribe-taker in exchange for services rendered

prevents general industry and academia from gaining access to information, making this an interesting field to explore.

From an academic point of view, this thesis may provide insight to the feasibility of and methods for using AIS data to identify ship-to-ship operations. In addition, this thesis shall investigate if this can be used to analyse bunkering operations. Along with analysis of these operations, it aims at analysing resulting data and proposing an index to quantify bunker delivery time with respect to anchoring time on a scale ranging from 0 - (*bad*) to 1 - (*good*).

From an industry perspective, the thesis can provide an aid to obtain a more comprehensive understanding of such operations, in addition to addressing consecutive steps in the process of further utilisation. Potential actors in the industry are presumably not interested in methods or aggregated data utilised to obtain the index, but rather the usage areas and how it can be operationalised. An index quantifying the time and thereby the bunker suppliers may be used in decision making, and transparency in the bunkering process can be enhanced. Long term effects of this may be reduced operational costs and increased time efficiency which can reduce delays in shipping schedules.

## 2.5 Constraints

### 2.5.1 Geographical Area and Time Restriction

Bunkering operations in Asia induce higher loss for ship owners than any other continent. Figure 2.3 illustrates general gain and loss for ships in bunkering operations, in which it is evident that bunkerings in Asia have the highest loss of 56%.

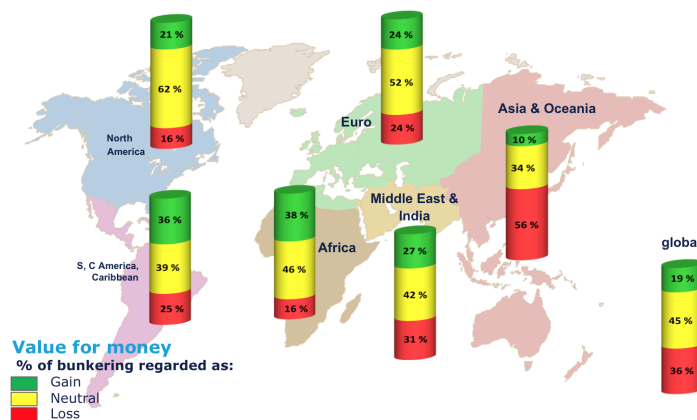


Figure 2.3: Value for money based on short-lifting of bunker fuel (HFO) per continent (Mestl (2015)). Note that 56% of bunkering operations in Asia and Oceania are considered as loss of money.

Port of Singapore is located on the southern end of the Malay Peninsula in Asia, and is the largest publicly owned port in the world. The port offers connectivity to more than 600 ports in 123 countries (Kable (2018)) serving as a gateway for the East-West trading area. It is one of the busiest ports in the world in terms of total shipping tonnage, total cargo tonnage handled and as a transshipment port.

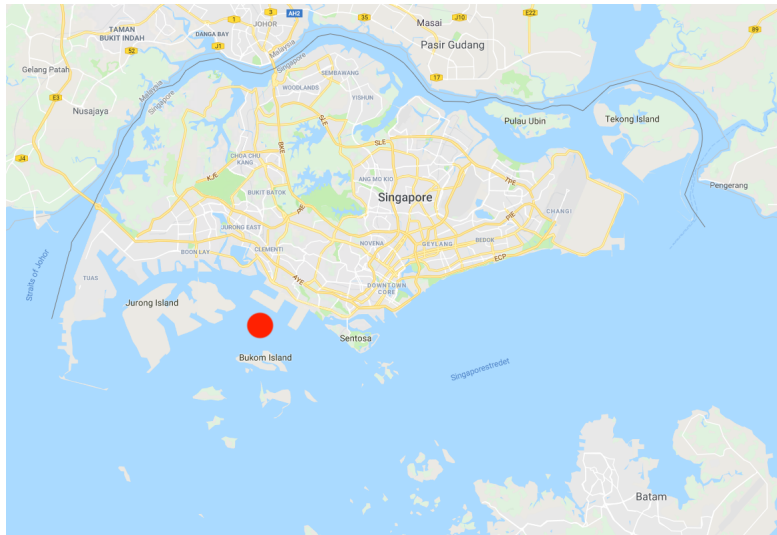


Figure 2.4: Map of Singapore (Google (2018))

As one of the largest marine refuelling hubs in the world, the port has been the scene for a significant amount of documented bunker fraud according to John Stirling from *World Fuel Services* (WFS). As a result, the port authorities have increased regulation of bunkering performed in the port area. It became the first port to mandate use of mass-flow-meters in 2017, making them mandatory for marine fuel bunker barges licensed by the Maritime Port Authority (MPA). Along with a crackdown on short deliveries to bunker fuel customers, the meters were made mandatory to enhance transparency in the bunkering process (Roslan Khasawneh (2018)). The very fact that Singapore is the only bunker port in the world which has mandated the use of mass flow meters on bunker barges may give a clue that volume cheating is indeed a widespread and serious issue.

Traffic flow passing the Port of Singapore is confined into concentrated areas in the Singapore Strait, which narrows down the geographical scope of the study. The geographical area is restricted by a square surrounding Port of Singapore as illustrated in figure 2.5.



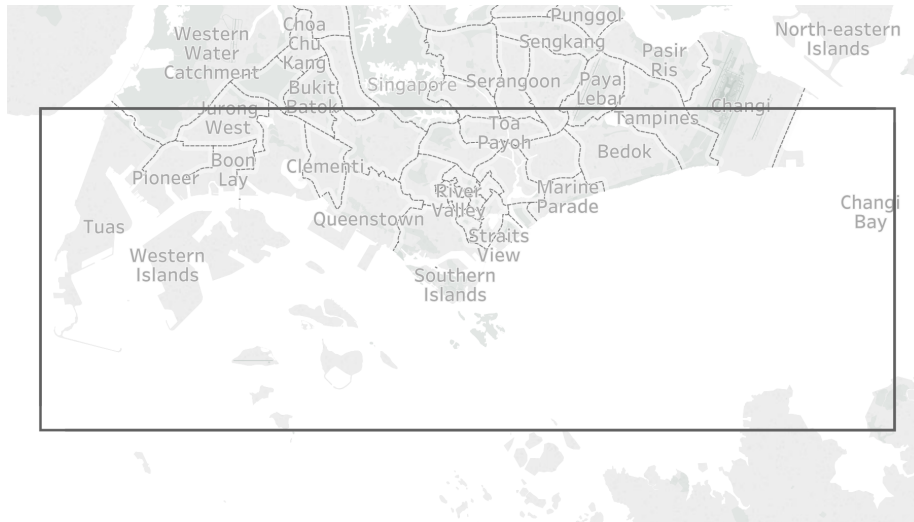


Figure 2.5: Restricted geographical area

Activity, or AIS messages, registered over a time period of one year spanning from 2017.01.01 to 2017.12.31 within the geographical scope form the basis of the thesis. Only IMO numbers from ships that have submitted HFO fuel samples in 2017 and Singapore as bunker port (and are registered by AIS data in this area at the bunkering date) were considered.

## 2.5.2 Operational Restrictions

AIS messages do not indicate if a ship has been engaged in a bunkering operation. Consequently, registered activity will be filtered by several criteria in a matching algorithm, resulting in a data set containing only AIS data from bunkering operations. Firstly, a ship engaged in a bunkering operation is normally anchored up or at berth and subsequently its registered speed should be low. To account for wind and drifting during bunkering at anchorage (open sea), a maximum registered speed of 1 knot is required during an operation. This means that all in-voyage operations with higher Speed Over Ground (SOG) are omitted.

Secondly, the bunkering procedure consists of several steps such as connecting and measuring quantities as introduced in section 2.1. To account for time required to complete necessary activities before commencement of bunkering, a minimum span of two hours of subsequent AIS messages with reported speed of 1 knot and below is further required. Correspondingly, since the average bunkering time is usually not more than 7-8 hours the maximum allowed time span is set to 37 hours to account for some lay time<sup>3</sup> before and after bunkering.

<sup>3</sup>The amount of time allowed to a ship in a voyage charter for loading and unloading of cargo at a port (Kantharia (2015)). In this thesis the term refers to time when the ship is not in-voyage.

### 2.5.3 Officially Approved Bunker Barges

Bunker operations are determined based on registered proximity between a ship and a bunker barge. Only bunker barges officially approved by the Singapore port authority are considered, meaning that bunkering operations with other suppliers are not considered.

### 2.5.4 Fuel Quality Samples

VPS has provided data on fuel quality samples taken during bunkering operations. However, not all bunkering operations are sampled, and the amount of sampled operations is in the range of 40-60%. Hence, a maximum of 40-60% of performed bunker operations can be identified and analysed. In addition, only HFO fuel samples are considered. VPS data includes date, IMO number of ship and fuel parameters, but not which bunker barge delivered fuel.

## 2.6 Problem Formulation

The incentives of the supplier, from now on called bunker barge, influences the behaviour during a bunker operation. It is assumed that this behaviour is reflected in AIS messages broadcasted from the bunker barge combined with AIS messages from the receiving ship.

AIS data from Singapore will be used in a matching algorithm. The algorithm will identify ships at berth or anchor for a predetermined period of time, and attempt to identify bunker barges that have interacted with these ships. Each match will be characterised as an operation, and identified operations will be scrutinised through statistical analysis to evaluate normal operational behaviour. Results from analysis can be utilised to establish a benchmark that characterises the integrity of bunker barges.

The thesis will attempt to answer the following questions:

- Is it possible to use AIS data to identify bunkering operations?
- Is it possible to characterise bunkering operations with respect to time usage?
- Is it possible to prove an index of quality?
- Is this feasible with AIS data or are we at the limit of what can be done with AIS data?

# Chapter 3

## Literature Review

Literature was reviewed during the work of the project thesis *An Introduction to Assessing Bunkering Operations Through AIS Data* (Wu and Aarsnes (2017)) to outline previous studies of AIS data and the bunker business. AIS data have been utilised in a wide spectre of research fields, but none specifically explore the use of AIS data for monitoring of bunkering or similar vessel-to-vessel operations. Some additional literature was reviewed as necessary when conducting this thesis.

45 articles were studied in cooperation with five students working with other AIS related projects. Moreover, the articles were categorised and assessed based on assumed relevance for this thesis. Only a small fraction of these were deemed applicable for the study, and a discussion of these is presented in section 3.2. Figure 3.1 illustrates the approach of the literature review.

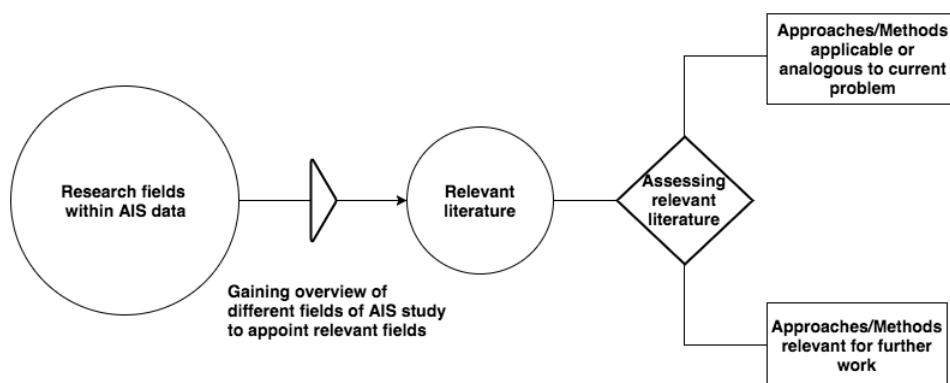


Figure 3.1: Approach of literature review

### 3.1 Overview of Literature on AIS Data

Articles reviewed in cooperation with students Daniel Wu, Erik Grundt, Patrick Næss, Jørgen Jensen Axelsen and Simon Drønen were grouped according to *topic* and *methods*. An example of the groupings is given in Appendix D. One article can comprise multiple topics or methods, and the majority of these considers safety, environmental or economic aspects as presented in figure 3.2. None of them address bunkering operations.

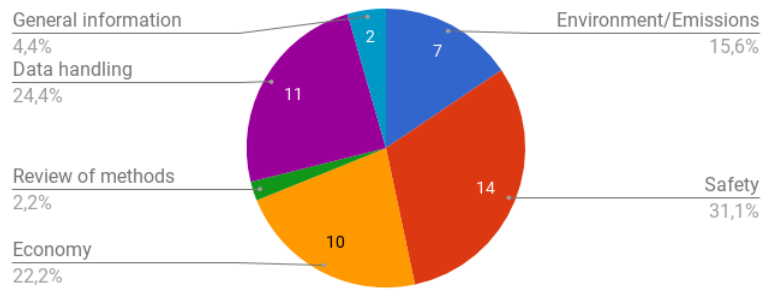


Figure 3.2: Distribution of topics

The topics are further defined in the following:

- *Safety*: Literature focusing on collision avoidance based on various models for pattern recognition, traffic estimation and route planning.
- *Environmental/Emissions*: Literature on calculation of emissions from traffic patterns.
- *Economy*: Literature that utilises traffic estimation and speed selection to increase operational efficiency.
- *Data Handling* and *Review of Methods*: Literature that specifically focuses on handling and utilising AIS data, for instance visualisation of AIS data.

The literature is also categorised by methods assessed for the various objectives of the studies. This categorisation is presented in figure 3.3.

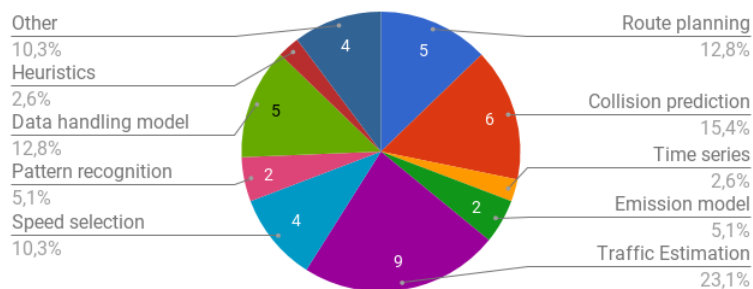


Figure 3.3: Distribution of methods

### 3.1.1 Appointing Relevant Fields of Research

The method categories are self-explanatory, so any further elaboration of these is deemed unnecessary. In the work with the project thesis, the most relevant topic was considered to be *Collision Avoidance* with focus on the methods *Pattern Recognition* and *Collision Prediction* due to the similarities between utilisation of traffic patterns in collision avoidance and behaviour during bunkering operations. This topic is also considered relevant for the current problem formulation.

## 3.2 Assessment of Relevant Literature

A more comprehensive study of literature was performed on work within maritime traffic risk assessment, as the studies focus on an appropriate scope. This scope is regarded as micro resolution, as opposed to other AIS related fields that conduct studies with lower fidelity on a more macro perspective. For instance, motion pattern analysis for individual ships is a prominent element in these studies, in contrary to other fields that calculate traffic flow based on a vast number of vessels. In addition, these articles focus on detection of anomalies in vessel motion patterns and establishing indices or other metrics to predict the level of risk in maritime traffic situations. Therefore, the methodology from this research field have been of inspiration for this thesis.

In the following, some assumed relevant studies are presented. These are grouped according to type of studies, and their potential contribution is discussed in the end of the chapter. Three additional studies by Ferrà et al. (2018), Wu and Aarsnes (2017) and Anfindsen et al. (2012) have been reviewed and elaborated.

### 3.2.1 Treating Data and Data Gaps

#### **Article: Teasing Out the Detail: How Our Understanding of Marine AIS Data Can Better Inform Industries, Developments, and Planning**

Shelmerdine (2015) demonstrates a procedure for processing, analysing and visualising AIS data with outputs and potential uses. Tools used include density mapping, vessel tracks, interpolation and ship type analysis. Density maps were created using ArcGIS<sup>1</sup> for AIS data from a land-based receiver on Shetland, and point data was interpolated at a resolution of 500 meters for vessel dimensions. Higher concentration of data points within

---

<sup>1</sup>Online mapping tool

a 12 NM limit around Shetland yielded the need for a resolution of 50 m. This was further used to analyse vessel types and temporal variation.

The analysis outputs were density maps and vessel tracks which can be used to identify fishing ground and vessel routes, but not anchorage points. The interpolation provided an additional level of querying and could be broken down to be analysed at sector specific level. Within 12 NM limit, interpolations were found to be informative due to increased density of data.

### **Article: Spatial and Temporal Allocation of Ship Exhaust Emissions in Australian Coastal Waters Using AIS Data: Analysis and Treatment of Data Gaps**

Goldsworthy (2017) focuses on allocating emission gaps in coastal regions, that is spatial gaps in the collected AIS data. In several coastal regions, the reports transmitted by the vessels are not collected by the terrestrial network due to absence of ground stations. The boundary regions of the data gaps are identified by specifying the duration (less than but close to the average duration of a gap) and speed at specified levels, and then applying clusters associated with the boundaries so that tracks that span the gaps can be examined.

Further procedure involves generating interpolated vessel tracks that both span the gaps and avoid coming too close to land using a simple shortest path or linear line interpolation. Where these do not produce acceptable results, vessel tracks are steered around land on shortest available paths using a combination of visibility graphs and Dijkstra's algorithm. Goldsworthy (2017) found that this method improved the findings compared to only using a shortest path interpolation.

## **3.2.2 Analysis of Data**

### **Article: Statistical Analysis of Motion Patterns in AIS Data: Anomaly Detection and Motion Prediction**

Ristic et al. (2008) devoted the study to statistical analysis of vessel motion patterns in ports and waterways using AIS ship self-reporting data. Motion patterns are extracted from real historic AIS data and used to construct corresponding motion anomaly<sup>2</sup> detectors through the framework of adaptive kernel density estimation. The anomaly detector

---

<sup>2</sup>Anomalous vessel behaviour refers to vessel movement behaviour that differs from the typical or normal vessel movement behaviour (Sidibé and Shu (2017))

is then sequentially applied to real incoming AIS data to detect anomalies. Motion of vessels is predicted using Gaussian sum tracking filter.

The paper presents some preliminary results using AIS broadcasts. A simple and fast anomaly detector is proposed based on the adaptive kernel density estimation, and the probability of false alarm of this detector can be evaluated numerically. This provides a quantitative measure of performance. The prediction of vessel motion attempts to compute the density of the vessel state in the future. The study concludes that if training data is effectively used, the method of motion prediction is quite straightforward.

#### **Article: Maritime Traffic Analysis of the Strait of Istanbul Based on AIS Data**

Altan and Otay (2017) have investigated navigation patterns using AIS data collected from 309 000 moving vessels over a period of one year in the Istanbul Strait. The data is stored in a Structured Query Language (SQL) database, and grid-based analysis is used to track parameters such as position, heading and course over ground. The study focuses on investigating the effect from local traffic on maritime risk, where speed and course are used as indicators to substantiate whether there is a significant risk.

The grid-based analysis method divides the observed area into sectors. Characteristics of the area are found by analysing the data at the entrance and exit of the area. The size of each sector is decided by introducing a limit for change in the navigational patterns so that the pattern within a sector is assumed to be constant. Vessels registered within a set distance from another sector is assumed to be entering or exiting this sector. Transiting and crossing traffic in the strait has been analysed to obtain the number of vessels sailing in the strait, the main ship types, ship dimensions, SOG and Course over Ground. Altan and Otay (2017) conclude that long-term traffic analysis can be used to understand the complexities of navigation patterns.

### **3.2.3 Utilising AIS Data**

#### **Article: Ship Collision Risk Assessment for the Singapore Strait**

Qu et al. (2011) proposes three ship collision risk indices derived from AIS-data to quantitatively assess the vessel collision risk in the Singapore Strait. The study divides the Strait into 15 legs and calculates the indices for each leg. The first index measures the degree of speed dispersion in each leg, that is the variance of the mean speed of each vessel sailing in the leg. Correlation between speed variance and likelihood of vessel collisions was confirmed by vessel captains.

The second index measures the deceleration and acceleration in each leg. As vessels strive to navigate with constant speed to reduce fuel consumption, a sudden change in speed can relate to a potential collision situation. Such situations emerge where vessels are about to cross, overtake, meet or turn. Thus, degree of deceleration and acceleration can be considered as a metric for collision risk. By calculating the acceleration and deceleration of each vessel in a given leg, and their average squared values, an index is obtained. The third index, vessel domain and collision risk, measures the degree of overlapping vessel domain for each leg. Definition of vessel domain was formulated by Goodwin (1975) as "*the surrounding effective waters which the navigators of a ship want to keep clear of other ships or fixed objects*".

The conclusion of the research points out, based on the three indices, the most high-risk legs in the Strait. They were also able to unveil that a significant portion (around 25%) of the cargo vessels has a sailing speed beyond the speed limit.

#### **Article: An AIS data Visualisation Model for Assessing Maritime Traffic Situation and its Application**

Pan et al. (2012) proposes a novel visualisation model to appraise the maritime traffic situation based on the AIS of a ship. The rate of turn, speed acceleration and ship encounter parameters are incorporated into a new index to indicate relative dangerous shipping areas.

The rate of ship encounters indicates the maritime traffic situation. If there are many potential dangers in the waters, the ship will have an incentive to change its course rapidly and the rate of turn and the speed can therefore also provide indication on the actual traffic situation. The sum of these values, weighted according to research area, constitute the Index of Maritime Traffic Situations (IMTS).

The new model was applied to analyse the maritime traffic situations in Xiamen Bay and Meizhou Wan, and proved to be practical and useful for maritime traffic decision-making and management of marine authorities and mariners. However, it is worth mentioning that the study emphasises the necessity of more theoretical analysis and experiment evaluation.

#### **Article: Expanding the Possibilities of AIS Data with Heuristics**

Smestad et al. (2017) propose heuristics to identify different ship types from AIS data and evaluate the performance of heuristics applied on respective types. The heuristics can be



an alternative to external ship databases. The article emphasises why categorisation of ship type and size is essential when conducting studies on fields such as operational efficiency and greenhouse-gas emissions. For instance, assumptions regarding typical features (block coefficient, design speed) of given ship types can be applied when the ship type is identified.

Satellite-AIS (S-AIS) data consisting of messages from 85,108 ships and a database provided by The Clarksons Group were matched by ship names - forming a candidate group of ships presents in the both S-AIS data and the database. This candidate group was cleaned for erroneous matches, for instance removing vessels classified as Panama ships when the width or draught exceeded the maximum dimension of the canal. The cleaned candidate group was used for heuristic training, which resulted in parameter sets that act as constraints. These parameters constrained features such as maximum speed and draught for a given vessel type.

The heuristics were evaluated through testing for different ship types and their accuracy was listed. In the discussion it is stressed that parameters such as maximum speed and constraints of draught are dependent on operating conditions that again are affected by market forces. Since the S-AIS data spans over 16.5 months, the developed heuristics may have lower accuracy when used on data sets from other time periods. On the other hand, as the time span of the AIS database lengthens, the likelihood of ships exceeding constraints will be greater.

#### **Article: Mapping Change in Bottom Trawling Activity in the Mediterranean Sea Through AIS Data**

Growing concern about depletion of marine resources due to over-exploitation and degradation of ecosystems prompted Ferrà et al. (2018) to map the fishing activity from the Mediterranean Sea. Raw AIS data from fishing vessels operating in the sea from 2012 to 2014 was used. The data was filtered by requiring each vessel to have a minimum of 300 recorded AIS messages per four months. Speed profiles from vessels were used to define speed profiles for different métiers. A K-means cluster analysis was performed to identify vessels using bottom trawl gears, thus impacting the seabed. Spatial filters from the ArcGIS software were applied to exclude false positives.

Subsequently, unique vessel identification numbers were used to create tracking layers from fishing pings, and duration and speed were calculated from the difference of successive pings. The process finally yielded fishing activity maps. Although the available time series data was relatively short, the analysis carried out confirmed the potential of AIS data to map fishing activity over time, resulting in a useful tool to improve marine sciences and

management.

### 3.2.4 Benchmarking

#### **Project Thesis: An Introduction to Assessing Bunkering Operations Through AIS Data**

Wu and Aarsnes (2017) described potential parameters for identifying bunker operations through AIS messages broadcasted from ships and bunker barges. AIS data from Singapore was filtered by requiring speed below 0.5 knots, which led to identifying sector-like shapes in geographical coordinates. These shapes indicate ships at anchor drifting around their fixation point, with potential of being bunkering operations. From this they established idle times - sequences of AIS messages with a reported SOG of e.g. 1 knot and below and a minimum duration of e.g. four hours. Bunker barges with close proximity to idle times were retrieved to identify the barge potentially engaged in a bunkering operation with the associated ship. Speed restriction, a limit for idle times and maximum distance (between ship and candidate) were thereby suggested as parameters for identifying bunker operations. The project thesis is more thoroughly elaborated in section 6.3.

#### **Article: Benchmarking of Marine Bunker Fuel Suppliers: the Good, the Bad, the Ugly**

Anfindsen et al. (2012) propose a methodology to construct a realistic best practice benchmark and develop a methodology for comparison of individual suppliers of marine bunker fuel based on the difference between stated and actual density of supplied fuel. Unfair business behaviour in the bunker market is not uncommon, resulting in financial loss for the buyers. The benchmark is established using the concept of membership functions from fuzzy set theory, a score can be derived from a best practice benchmark histogram. The main advantages of this method are its relative independence both of sample size and of the underlying distribution, as well as being computationally efficient. Based on best practise, Anfindsen et al. (2012) were able to estimate induced loss, gain and loss neutrality for buyers. The presented benchmark methodology is extendable to other (quality and economical) bunkering parameters like viscosity, sulphur or water content, as well as a series of physical and chemical properties.

## 3.3 Conclusion of the Literature Survey

Shelmerdine (2015) used interpolation to increase the informative level of density maps from AIS data for vessels detected by a land-based stations on Shetland. Although density

maps will not be produced for vessel tracking, distance between vessels will be calculated. This requires synchronised time stamps in the AIS data sets, and thereby the proposed interpolation approach can be useful when generating data for gaps. Goldsworthy (2017) achieved improved results when using Dijkstra's algorithm when interpolating over spatial gaps to avoid generating vessel tracks crossing land. Bunkering operations in this master thesis scope will be confined into a smaller geographic area, and linearly interpolated tracks will unlikely cross land. Thus, linear interpolation is regarded as sufficient for closing spatial gaps in data sets.

A matching algorithm will identify bunkering operations, and data from these will be visualised in various forms which requires a method for analysis. Altan and Otay (2017) used a grid-based analysis method to analyse the navigation patterns in the Strait of Istanbul. The results from this study show that long-term data analysis can provide an insight into the complexity of the patterns. However, navigational patterns will not be analysed in this thesis, thus grid-based analysis will not be implemented. Ristic et al. (2008) conducts statistical analysis of motion patterns to detect anomalies based on a kernel density estimation. Investigating anomalies is part of the approach in this thesis, and statistical methods will be applied for analysis. Furthermore, kernel density estimation may be applied in the analysis as an aid, but will in that case be implemented through functions in Python.

Pan et al. (2012) illustrates a method for constructing an index to indicate relatively dangerous shipping areas. Qu et al. (2011) demonstrates how different parameters based on AIS data can be computed for the same purpose. The degree of acceleration and ship domain can be applied in a bunker barge context. An analogy of the procedures proposed in these papers may be implemented to obtain an index expressing the degree of reliability of each bunker provider. Such indices will give bunker purchasers more nuanced insight when selecting bunker providers, than the binary benchmarks described in subsection 2.2. To achieve these metrics on quality/reliability of bunker providers, heuristics as described by Smestad et al. (2017) can be implemented to distinguish types of receiving vessel. Bunker operations for different vessel types can then be scrutinised separately to assign weight to each operation.

Ferrà et al. (2018) propose a method for mapping fishing activity within a defined geographical scope. The method is based on filtering data and applying K-means cluster analysis. Cluster analysis will not be implemented in the framework for identifying bunkering operations, but filtering the data by operational restrictions has proven beneficial by Ferrà et al. (2018). Combining a similar approach with results obtained by Wu and Aarsnes (2017) will contribute to constructing the framework for a matching algorithm identifying bunkerings.

Methodology as presented by Anfindsen et al. (2012) is not transferable to this thesis, as they base their approach on measured parameters obtained from supplied fuel. It is however an important contributor of knowledge regarding bunkering business practises, as well as providing insight to alleged bunkering malpractice.

# Chapter 4

## Data Sources

In this chapter, a brief introduction to AIS data and other key elements essential for this project thesis are given. First, a basic introduction to AIS is outlined, followed by requirements regarding the use of AIS and a general overview of some utilisations and quality issues regarding AIS data. A brief overview of various elements included in AIS messages is presented, and AIS data components used in this thesis are outlined. Furthermore, VPS data and officially approved bunker barges are explained. Some parts of this chapter are based on the project thesis *An Introduction to Assessing Bunkering Operations Through AIS Data* (Wu and Aarsnes (2017)).

### 4.1 AIS Data

#### 4.1.1 AIS Basics

Automatic Identification System (AIS) is a communication system that utilises Very High Frequency (VHF) system to transmit ship movement and technical data at irregular intervals (Smestad et al. (2017)). AIS data can be recorded by ground stations at ports and around the coasts, *terrestrial AIS data*, or by specially equipped satellites, (S-AIS) data, (Goldsworthy (2017)).

A protocol for the communication system specifies what technological equipment shall be utilised and the type of information that shall be transmitted. The data is divided into static data which includes navigational details such as the ship name, ship identification number and dimensions and dynamic data from ship sensors which includes voyage related information such as speed, position, heading and time (Smestad (2015), Smestad et al. (2017)). Semi-static data may be destination and estimated time of arrival.

### 4.1.2 Guidelines for Use of AIS

Regulation 19 of SOLAS Chapter V – *Carriage requirements for ship borne navigational systems and equipment* – defines what navigational equipment must be carried on board ships according to their type.

Carriage of AIS is mandatory for ships of 300 Gross Tonnage (GT) and upwards engaged on international voyages, cargo ships of 500 GT and upwards not engaged on international voyages and all passenger ships built after 2002 or operated after 2008 irrespective of size. Smaller vessels may also use AIS on a voluntary basis (Goldsworthy (2017)). From May 31st of 2013 fishing vessels of more than 15 meters are also required to install AIS (European Commission (2011)). The AIS shall be maintained in operation at all times except where international agreements, rules or standards provide the protection of navigational information (SOLAS (2017)).

### 4.1.3 Use of AIS Data in Industry

The development of AIS was initiated in 1994 as a joint project between several parties such as the International Maritime Organization (IMO) and the International Association of Marine Aids to Navigation and Lighthouse Authorities (IALA). In 1998, IMO 11 amended regulations about use of AIS to the International Convention for the Safety of Life at Sea (SOLAS) (IMO, 1974).

AIS was originally designed as an aid for collision avoidance, and utilises frequencies from the VHF band to broadcast messages. Land based AIS receivers can detect AIS messages up to 40-50 nautical miles offshore. Traffic further offshore will not be detected by these receivers. However, a follow-up study by (Hoye et al. (2008)) found that AIS signals can be detected by satellite-based AIS receivers positioned in altitudes of up to 1000 km (Smestad et al. (2017)). With the capability of gathering data through satellites, the use of AIS data in studies on maritime transportation has become increasingly prevalent. In addition to using it for fleet and cargo tracking and by port control to manage traffic, it can be utilised in estimation of shipping emissions from fuel consumption and gaining insight on technical and operational energy efficiency in shipping (Smith et al. (2014)).

The literature review presented in chapter 3 illustrated a wide range of AIS data utilisation. This underscores that multiple actors are realising the potential of the immense amount of data gathered from maritime traffic. Applying suitable methods can drastically advance shipping as it is known today.

#### 4.1.4 AIS Data Used in Bunkering Matching Algorithm

Satellites move in orbits around the globe and receive AIS data from the area which they travel above. Consequently, the sampling rate of S-AIS data is characterised by larger time intervals. This thesis requires AIS data with a high-resolution level, i.e. small time intervals. Thus, S-AIS data sampling rate is insufficient, and terrestrial AIS data serve as the basis for this work. This also implies that bunkering operations occurring outside the range of land-based stations cannot be detected.

Samples from raw AIS data with a time interval of ten minutes, meaning that each AIS sample is chosen as the one with time difference closest to ten minutes from the last AIS data sample. Raw AIS data is transmitted with varying time intervals depending on the navigation status. For example, when the vessel is anchored or at berth the raw data transmission rate can be four minutes, while it can be ten seconds when the vessel is underway. Since investigation of bunker operations requires data on a high-resolution level the rate is adjusted so it is sufficient for this purpose.

AIS messages broadcasted in a rectangular area surrounding Port of Singapore over period from 01.01.2017 to 31.12.2017 are used in a matching algorithm to identify potential bunkering operations, see chapter 6. The port is chosen due to its position as one of the busiest bunker ports in the world as it is a gateway for the East-West trade area. It has been the scene for a significant amount of documented bunker cheating according to John Stirling from WFS, which makes this an interesting port to investigate.

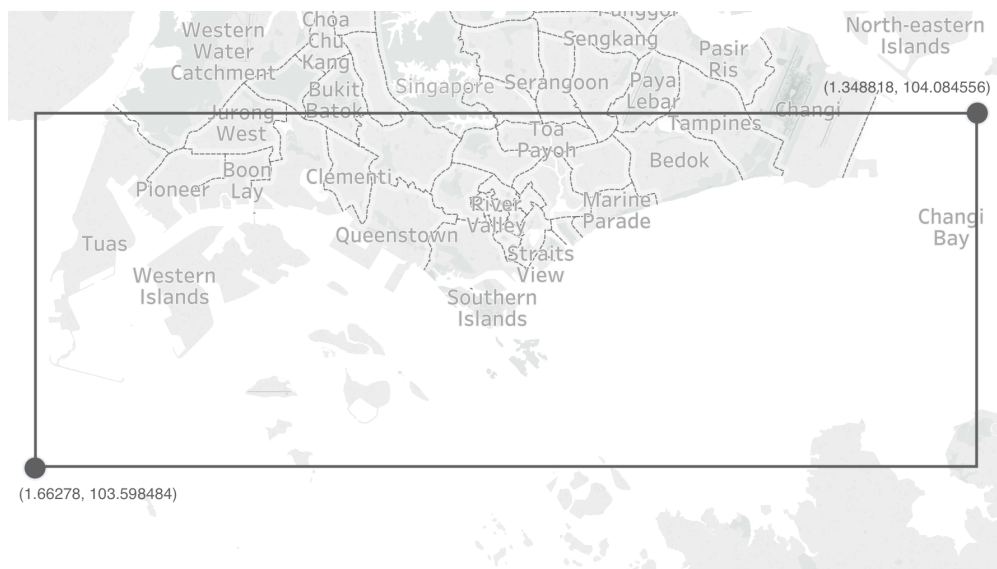


Figure 4.1: Geographical area under consideration

AIS messages contain technical and operational data, and the data is both manually and automatically logged. Some of the most common components are listed in table 4.1.

Table 4.1: AIS message components and description

AIS message component	Description
Message ID number	Excerpt from raw AIS message
IMO number	International Maritime Organization identification system
SOG	Speed Over Ground in knots
COG	Course Over Ground in degrees
UTC Port Time	Time given in time zone at zero degrees longitude
Unixtime	Number of seconds elapsed since 01.01.1970
Latitude	Horizontal geographic coordinate
Longitude	Vertical geographic coordinate
Drought	Vessel specification, drought in meters
Rot	Circular movement
Heading	Course (cardinal direction)
Navigational status	At anchor, under way or not under command

Message components used as input to the algorithm are listed in table 4.2. Some of these are further elaborated below.

Table 4.2: AIS message components used in matching algorithm

Data type	Component Used
Dynamic	Coordinates (latitude, longitude)
	SOG
	Time
	Heading
Static	IMO
	MMSI
	Ship name
	Ship type
	Length
	Width
	Location of GPS antenna

### IMO number

The IMO vessel identification number scheme was introduced through adoption of the IMO Resolution A.600(15) IMO (1987) as a measure aimed at enhancing *maritime safety, and pollution prevention to facilitate the prevention of maritime fraud* (IMO (2017)). It aimed at assigning a permanent number to each vessel for identification purposes, which



would remain unchanged upon transfer of the ship to another flag.

The Resolution was made mandatory through SOLAS regulation XI/3 IMO (1974). All vessels over 100 GT, with exception to vessels engaged solely in fishing, vessels without mechanical means of propulsion, pleasure yachts, vessels engaged on special service, hipper barges, hydrofoils and hovercrafts, floating decks, vessels of war and wooden vessels, shall be identified with an IMO number. The number is required to be permanently marked on the superstructure or the hull itself, and can be found in Static AIS Messages as a seven-digit number (IMO (2017)).

### **SOG**

Measured speeds in the AIS messages are recorded by the Global Positioning System (GPS) as Speed Over Ground (SOG) in knots. Speed through water does not consider the resistance on the vessel or the speed of the vessel relative to the currents of the surrounding water. Speed through water will be different from SOG when there is a tidal stream. For example, a vessel sailing at a speed of 14 knots with a tidal stream acting in the opposite direction at a speed of two knots will experience a speed through water of 16 knots.

### **Latitude**

Latitude is a geographic coordinate that specifies the north-south position of a point on the surface of the Earth. It is an angle ranging from 0 degrees at the Equator to 90 degrees at the poles. Lines of equal latitude run east-west parallel to each other and the Equator, and are often referred to as parallels (UNL (2017)).

### **Longitude**

Longitude specifies the east-west position on the surface of the Earth. Meridians are lines of equal longitudes running north-south, of which one of these passes through the Royal Observatory in Greenwich, England. This meridian has been allocated as the position of zero degrees in longitude (UNL (2017)).

### **Location of GPS antenna**

Vessels are equipped with GPS receivers, and their antenna location is given by dimensions A, B, C and D which are static components in AIS messages. In figure 4.2 the red circle illustrates the location of the GPS antenna, with the respective recorded distances. In some cases vessels have more than one GPS antenna, thus the antenna location is no longer unique.



Figure 4.2: Location of antenna given by parameters A, B, C and D in AIS messages

### 4.1.5 Quality Issues

AIS systems were originally designed as a ship-to-ship communication, and not for space-based receivers. Since a satellite has a much larger coverage area than AIS receivers were designed for, interference problems between the ship AIS signals can occur (Smestad et al. (2017)). This problem is especially prominent in areas with high traffic density, where interference can prevent the identification of individual signals of a vessel, which in turn can give an erroneous tracking of the vessel. An example is displayed in figure 4.3 where three bunkering operations occur in close proximity to each other. Also, in areas where vessels are covered by multiple base stations, the stations can sample AIS data from the same vessel but return different time stamps for the same position.

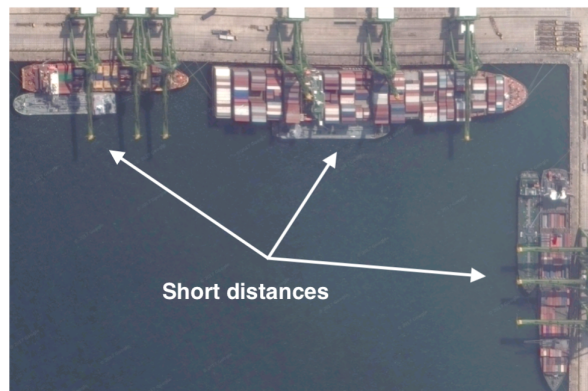


Figure 4.3: Ships and bunker barges with close proximity (Mestl (2015)) which can cause interference

Vessel identifiers used in AIS messages include IMO number, MMSI number, Call Sign and/or Vessel Name. Approximately 50% of AIS Static Data transmissions have errors, and 1/3 of these have at least one incorrect vessel identifier (Winkler (2012)). Winkler states

that no single legal identifier meets the criteria that it is available and issued to every vessel for the purpose of unique tracking, which also complicates the tracking of the history of the vessel. Broadcasting under the identity of another vessel creates safety issues when multiple vessels use the same IMO number in the same area.

Erroneous data come in multiple forms, and a frequent issue is that a data set contains large time or space distances between successive AIS records. This can be caused by turning off the AIS transmitter or sailing in areas with poor terrestrial or satellite coverage. When sailing in open sea the sampling of raw data is done by satellites, and the sampling rate depends on the ship density and number of satellites an AIS provider uses. This can cause large gaps in the data. Erroneous data can also be caused by high proximity between a VHF antenna and the AIS, as the antenna will affect the AIS if they broadcast on the same spectre.

Furthermore, recording of vessel heading may be incorrect. In some instances, this has been observed in the data set used for this thesis. Reasons for this may be incorrect settings on the heading recorder, or that the recorder may be connected to SOG and is restricted from logging heading until the speed reaches a specified limit.

A component in AIS data is specification of GPS antenna location on the vessel. Vessels usually have an AIS antenna for transmitting the AIS signal, in addition to a GPS antenna which receives time signals from GPS satellites. However, if the stated position in the AIS message is not the GPS, but the AIS antenna position, the recorded geographic position of the vessel will be shifted relative to the actual position.

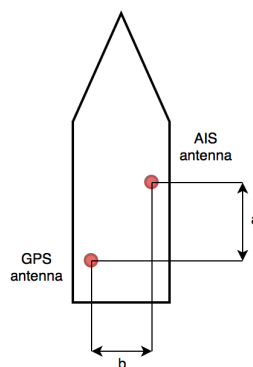


Figure 4.4: GPS and AIS antenna positions

In addition, manually logged data such as draught can be incorrectly recorded by vessel personnel, as they must manually feed the draught into the AIS data. The reliability of manually logged data also depends on the equipment setup on-board.

When using AIS data, several obstacles must be considered in the thesis. For instance, AIS

data may be erroneous or incomplete due to data gaps. Interpolation has been suggested by Goldsworthy (2017) and Wu and Aarsnes (2017) to reduce these, and will be implemented to create data sets with synchronised time stamps. However, in cases of bunkering operations the reported speed of involved vessels are close to zero. Consequently, interpolating over a data set from a ship at berth will yield no new information. For a ship at anchorage, currents and wind should also be accounted for when interpolating due to drifting, but such data was not available while conducting this thesis.

## **4.2 VPS data**

VPS has provided fuel sample data taken by ship crew during bunkering. Samples marked with Singapore as location have been extracted, but no information is available to indicate if a bunkering occurred within or outside the geographical area defined in section 2.5.1.

Fuel quality samples are marked with date of bunkering, location and IMO number when submitted. All specifications in the data are manually recorded by crew, and recorded date is subject to uncertainties. Date may indicate appointed time of bunkering, start of bunkering, end of bunkering or date when the sample was submitted. Only bunkerings involving Heavy Fuel Oil (HFO) are considered, other fuel types are not analysed.

## **4.3 Officially Approved Bunker Barges**

Bunker barges licensed for 2018 by Maritime and Port Authority of Singapore (Singapore (2018)) are addressed. Common for these bunker barges is that they installed mass flow meters on board. Not approved bunker barges are not included in the data.

# Chapter 5

## Applied Methods

This chapter aims to describe the method applied to treat AIS data from bunker operations. Furthermore, it introduces methods used for statistical analysis to interpret the data. The term *Statistical Analysis* encompasses collection, examination, summarising, manipulation and interpretation of quantitative data to discover underlying causes, patterns, relationships and trends (BusinessDictionary (2018)). These methods are outlined in a generalised manner, and utilised in the case study presented in chapter 7. Theory is primarily based on knowledge obtained previously by the author.

### 5.1 Interpolation

Data sets containing significant gaps between successive AIS records can produce less reliable results. This is especially evident when plotting the traces of a ship and a barge engaged in a bunkering operation, where the coordinates in the plots do not match. The simplest form of interpolation assumes linear relation between succeeding time stamps. Gaps in the data can be filled by equation 5.1 (Wu and Aarsnes (2017)).

$$f(x) = f_1 + (f_2 - f_1) \frac{x - x_1}{x_2 - x_1} \quad (5.1)$$

Here,  $f(x)$  is the unknown value which is calculated from known values  $f_1$ ,  $f_2$ ,  $x$ ,  $x_1$  and  $x_2$ . When comparing the traces of a ship and a bunker barge, coordinates will be generated by interpolation to create time synchronised sets of AIS messages. The  $f$  values can be e.g. latitude or longitude, and the unknown of these will be obtained by interpolating between already sampled values, thus interpolated values will be entirely based on sampled values. Replacing  $f$  with latitude  $lat$  (or longitude  $lon$ ) and  $x$  with time stamp  $t$  yields the

following.

$$lat = lat_1 + (lat_2 - lat_1) \frac{t - t_1}{t_2 - t_1}$$

Synchronised lists can be used to calculate distance between ship and barge, and interpolation will be applied in the matching algorithm to generate missing data for the barge from known ship data.

## 5.2 Statistical Analysis

### 5.2.1 Summarising

#### Grouping

The data foundation for this thesis is information that represents qualitative and quantitative attributes of a set of variables for a given entity, here *bunkering operations*. It is initially an ungrouped list of numbers and labels - or raw data, which is not categorised, classified or otherwise grouped.

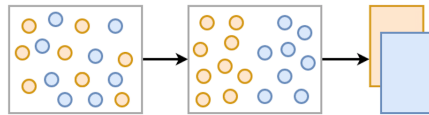


Figure 5.1: Illustration of mapping data into two groups

Grouped data is data that has been organised into smaller groups, where each group has certain properties. Each group has a certain width referred to as *length* which is an important parameter in the process of visualising the data. In this thesis, data will be grouped based on properties such as *vessel length*, *vessel type*, *anchorage time length*, *bunkering time length* and *bunkering type*.

#### Visualisation

Visualisation can be viewed upon as a branch of *Descriptive Statistics* and involves the creation and study of visual representations of data. It will serve as an aid to make the complex data set more accessible and understandable to effectively analyse the data. Plotting will serve as the visualisation aid in this thesis, and restricted to histograms, scatter plots and line plots.

## 5.2.2 Kernel Density Estimation

Kernel Density Estimators (KDE) are generalisations and improvements over histograms. In statistics, it is a non-parametric data smoothing method utilised to estimate the probability density function of a random variable, which can serve as an aid in the data analysis. The mathematical formulation for a histogram as illustrated in figure 5.2 is established in equation 5.2, where  $b$  is the bin size and  $N$  is the number of observations.

$$\begin{aligned}\hat{f}^{hist}(x) &= \frac{1}{N} \frac{\text{Number of } x_i \text{ in same bin as } x}{\text{width of bin containing } x} \\ &= \frac{1}{N} \sum_{i=1}^N \frac{1}{b} \mathbf{1} \left\{ x_i \in \left[ x - \frac{1}{2}b, x + \frac{1}{2}b \right] \right\}\end{aligned}\quad (5.2)$$

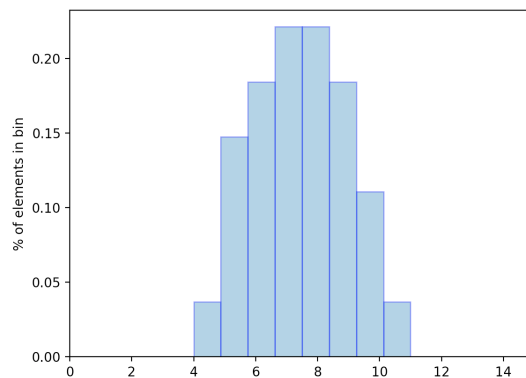


Figure 5.2: Illustration of a histogram

The KDE is based on some of the same intuition as histograms in the sense that it is an average and divided by some bandwidth  $b$  which is regarded as equivalent to the bin width. The average is multiplied with a key function  $K(\cdot)$  that weights the observations  $x_i$  depending on their respective distances from  $x$  and sums them up. As opposed to equation 5.2, the key function applies continuous weight which decreases the further  $x_i$  is positioned from  $x$ .  $K$  denotes the chosen *kernel*, and the choice of kernel throughout the thesis will be the Gaussian - the Probability Density Function (PDF) of the Normal distribution.

$$\hat{f}^{Kernel}(x) = \frac{1}{Nb} \sum_{i=1}^N K\left(\frac{x_i - x}{b}\right)\quad (5.3)$$

Creating the KDE yields a trend line showing the distribution of the initial histogram where the total area beneath the line is equal to one. This will be applied by a built-in function in Python in further analysis, resulting in distribution plots as illustrated in figure 5.3.

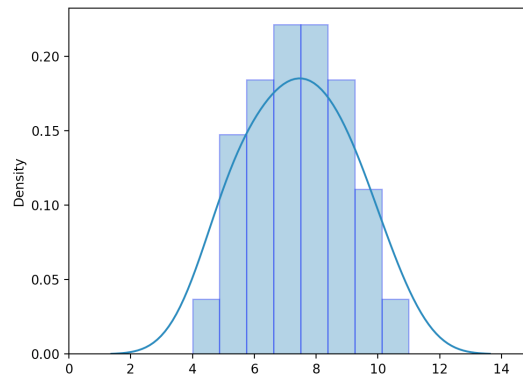


Figure 5.3: Kernel Density Estimation

### Gaussian Kernel

The Gaussian distribution kernel is the most used kernel, and is presented in equation 5.4 (Romeny (2018)).

$$K = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}x^2} \quad (5.4)$$

$\sigma$  determines the width of the Gaussian kernel, i.e. variance of the data.



# Chapter 6

## Problem 1: Identifying Bunker Operations

This chapter presents the framework and a matching algorithm created for identifying bunker operations used for further analysis. The work with this algorithm was initiated as a cooperation between Wu and Aarsnes (2017) and Thomas Mestl (DNV GL), and resulted in the project thesis *An Introduction to Assessing Bunkering Operations Through AIS Data*. Thomas Mestl has further developed the algorithm.

### 6.1 Defining the Bunkering Operation

From now on, the term *Bunkering Operation* encompasses all stages from the ship anchors up and until it withdraws its anchor and leaves the anchorage position.

#### 6.1.1 Bunker Operation Time Line

A bunkering operation time line is illustrated in figure 6.1. The start point is defined by the ship anchoring up at a certain position, preparing for bunkering. When the ship is ready, the bunker barge approaches and connects to the ship. This initiates the bunkering time which includes all steps of a bunkering operation listed in section 2.1. When the operation is completed the barge disconnects from the ship and leaves. After hoisting the anchor, the ship leaves its position which marks the end of anchorage.

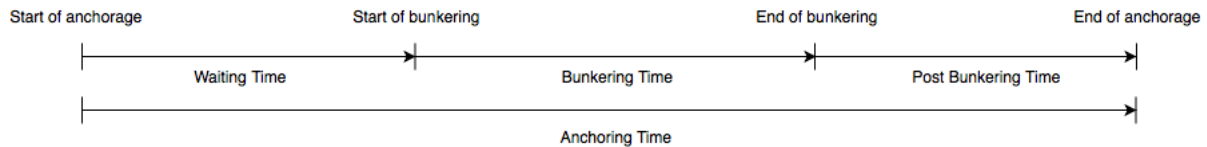


Figure 6.1: Bunkering time line. Note that there may be some Waiting Time and time after bunkering; Post Bunkering Time

Waiting Time is the time from start of anchorage to start of bunkering, Bunkering Time lasts from start of bunkering to end of bunkering and Post Bunkering Time is the final phase from end of bunkering to end of anchorage.

### 6.1.2 Waiting Time, Bunkering Time and Post Bunkering Time

Time spent during a bunkering operation depends on several factors. First and foremost, the amount of supplied fuel dominates bunkering time. Bunkering a full tank or several tanks will require more time than filling half a tank. In addition, bunkering time is prolonged if signing documents, or other, is time demanding. Bunkering time will also depend on the pump rate of the bunker barge.

Waiting Time is also subject to variations. Crew must prepare the ship prior to bunkering, and the barge cannot connect to the ship until approved by ship crew. In case the ship requires several hours to prepare for bunkering, the bunker barge may accept another mission and return later. In this case, Waiting Time will be increased. Also, the ship may have arrived early but ordered bunkering for a later time.

Post Bunkering Time depends on whether the ship is at anchorage or at berth. In case of bunkering at berth, the ship can be delayed by traffic in port. With high traffic density, ships risk queuing which prevents them from leaving port. In other cases, the barge may deny the ship from leaving by not detaching.

### 6.1.3 Bunker Operation as Seen by AIS Data

Bunkering operations are not directly reflected through any AIS message components, but comparing series of data from ships and bunker barges may provide insight to when the ship may have been engaged in such operations. Not-in-voyage bunkering operations will usually have low recorded SOG, typically below 1 knot. In addition, variations in coordinate parameters latitude and longitude will be low due to low speed. AIS data from a ship and a bunker barge engaged in bunkering will have similar geographical coordinates and speed.

Operations may be distinguished between bunkering at anchorage and at berth. At berth, SOG will be zero, and due to the ship being moored at harbour the variation in latitude and longitude will also be zero. In contrast, a ship at anchorage usually has a non-zero speed (below 1 knot) due to wave, wind and current effects. In addition, it will be subject to varying SOG, latitude and longitude. These variations will be detected by calculating the variation in SOG, latitude and longitude of succeeding AIS messages.

Table 6.1: Parameters distinguishing bunkering at anchorage and berth

	SOG	Latitude	Longitude
At berth	$\text{Var}(\text{SOG}) \approx 0$	$\text{Var}(\text{lat}) \approx 0$	$\text{Var}(\text{lon}) \approx 0$
At anchorage	$ \text{SOG}  \leq 1 \text{ knot}$	$\text{Var}(\text{lat}) \neq 0$	$\text{Var}(\text{lon}) \neq 0$

## 6.2 Uncertainties in the Bunkering Process

As mentioned in chapter 2, the shipowner or the charterer orders bunker for a predetermined time and place. The schedule of ship and bunker barge may be affected by delays, causing a shift in both time schedules. This can cause the ship or the bunker barge to arrive at the predetermined position later than agreed. A delay in either schedule will delay the scheduled bunker operation, meaning that either the ship or the bunker barge must wait for the other party to arrive.

Secondly, when a ship is anchored up at the bunker position, the barge will attempt to approach the ship. If the ship is not ready for bunkering, the barge may distance itself from the ship and wait until the ship is ready within a reasonable time otherwise take another assignment. This process may repeat itself until both parties are ready for bunkering.

Thirdly, the bunker barge may alter its position several times during one bunker operation. This can happen if the ship has several fuel tanks requiring filling at different locations on the ship. The bunker barge may for example begin the bunker process on one side of the ship, and sequentially move to the other side to finish the process. Bunkering is not a one-size-fits-all approach, it is ship-specific.

## 6.3 Matching Algorithm

### 6.3.1 Previous Work

Wu and Aarsnes (2017) described potential parameters for identifying bunker operations through AIS messages.

AIS data from ships sailing near Port of Singapore was visualised in the map plotting device *Tableau*. As ship speed during bunkering is typically low, the coordinates were filtered by requiring a reported SOG of 1 knot and below. Through further inspection, sector-like shapes were identified in the plots. These shapes emerge when the ship is anchored up, and currents and wind force the ship to move in circles or sectors around a fixation point.

Ships are either anchored up or at berth during bunkering, and sector-like shapes that stem from the coordinates of the ship can indicate that there has been a bunkering operation. This led to the establishment of idle times - sequences of AIS messages with a reported SOG of e.g. 1 knot and a minimum duration of e.g. two hours. Filtering the AIS data with this requirement yielded several idle times for ships under consideration, and bunker barges registered within 200 meters of the idle time positions were retrieved.

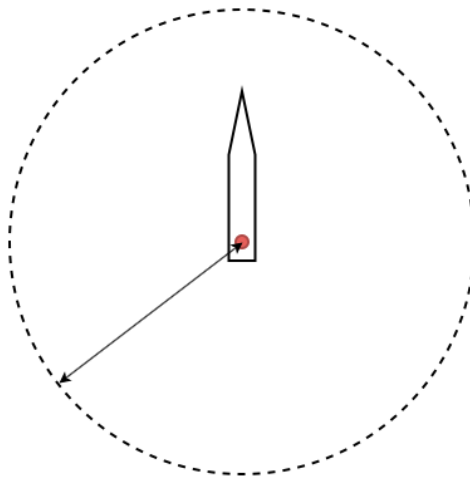


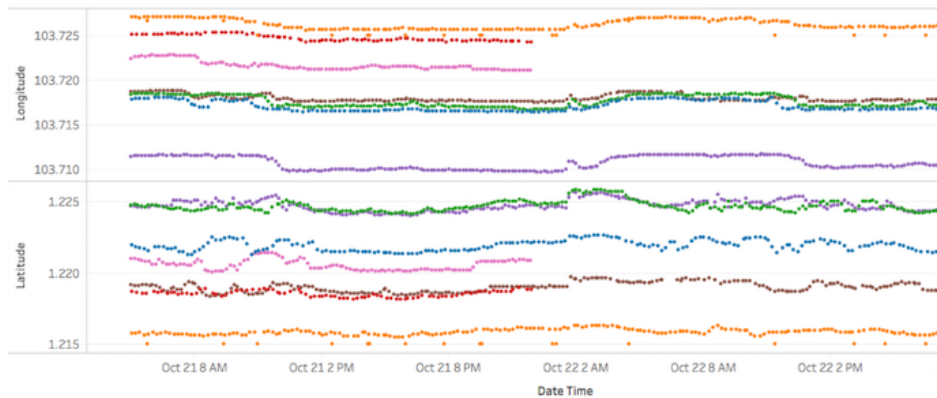
Figure 6.2: Illustration of area surrounding a ship with radius of 200 meters from which barge candidates are retrieved

Bunker barges recorded within 200 meters of the idle times windows may have engaged in a bunker operation with the chosen ship. The variation in latitude and longitude should be similar if both ship and candidate have been anchored up at the same position. However, Wu and Aarsnes (2017) found that these coordinates could be coinciding due to current and weather conditions in the same area, see figure 6.3. Hence, it was concluded

that distance between the vessels should be included as a parameter. Speed restriction, limited time window and maximum distance between ship and a candidate were thereby suggested as parameters for identifying bunker operations.



(a) Sector-like patterns from vessels at anchorage



(b) Similar variations in latitude and longitude although different geographical location

Figure 6.3: Movement for one bunker barge and six ships in the same geographical area (Wu and Aarsnes (2017))

Only bunker operations at berth or at anchorage are considered. The simplest case is while the ship is at berth as its speed will be zero, and the position of the ship will be unchanged. At anchorage in open sea however, the ship will be exposed to currents and winds, which will move the ship in circle-like patterns around the point of fixation depending on sea conditions, see figure 6.3a. In this case, the position of the ship will not be constant.

### 6.3.2 Algorithm Structure

The algorithm reduces the original 4-dimensional problem (lat, lon, SOG, t) to a 2-dimensional problem ( $C_M$ , t) where effects of weather and current variations are removed.

The matching algorithm was developed into an alpha prototype which matches bunker barges with ships within a specified geographical area and time frame. During the ini-

tial identification of bunker operations, the algorithm was restricted to a smaller square surrounding Singapore. It was later expanded to include a larger area in case some ships have bunkered outside this area, but reported Singapore as location. A flow chart of the algorithm structure is displayed in figure 6.4.

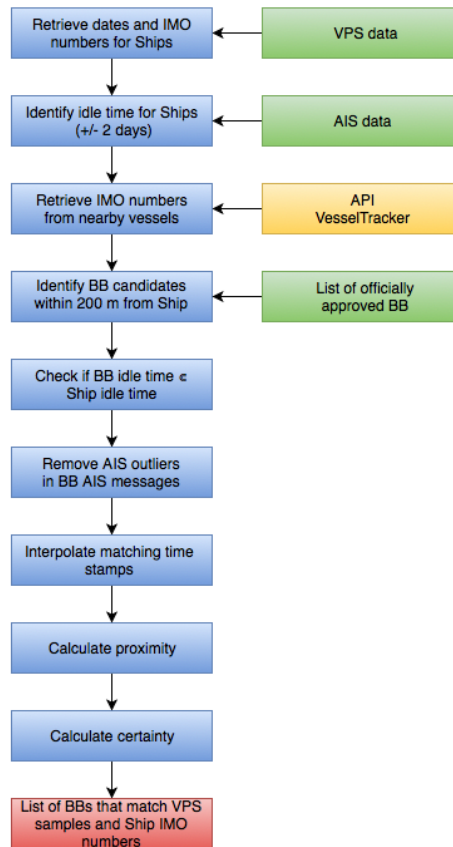


Figure 6.4: Flow chart of the algorithm

Inputs to the algorithm are AIS data, IMO number and dates from VPS and a list of officially approved bunker barges (Singapore (2018)). The algorithm loops through all ship IMO numbers for the aforementioned time limit and geographical location. First, it identifies idle times for ships of minimum 2 and maximum 37 hours where recorded speed is equal to or less than 1 knot, with a slack of +/- 2 days. Minimum 2 hours is set due to investigating solely operations involving bunkering of HFO which usually involves transferring of larger fuel amounts. Second, bunker barges registered with maximum 1 nautical mile (NM) in distance from the ship with the same speed and lay time requirements are retrieved from the Application Programme Interface (API) *VesselTracker*, where these IMO numbers were filtered with respect to the official bunker barges list. Each identified bunker barge is then treated separately. It then calculates a theoretical speed between succeeding AIS messages from ship and barge to detect and remove outliers<sup>1</sup>, i.e points

<sup>1</sup>Unusually large or small observations (Anfindsen et al. (2012))

with sudden jumps between geographical points which would require a significantly high speed when theoretical speed of preceding and succeeding messages are much lower. Distance calculation must be done at matching time stamps, so lacking barge AIS messages are obtained through interpolation using time stamps from ship AIS messages. A confidence measure is calculated for each barge candidate which requires aligned coordinated systems and an area computation. From the area computation, the most likely barge candidate can be extracted.

All identified operations are filtered by requiring that a fuel quality sample has been submitted to VPS and that the bunker barge involved is officially approved by the port authority of Singapore. Positive matches from the algorithm will be regarded as (bunkering) operations.

Table 6.2: Algorithm restrictions and values

Restriction	Value
Geographical point 1 (lat, lon)	(1.166278, 103.598484)
Geographical point 2 (lat, lon)	(1.348818, 104.084556)
Start date	2017-01-01
End date	2017-12-31
Minimum lay time	2 [hours]
Maximum lay time	37 [hours]
Maximum speed	1 [knots]
Maximum distance	1 [NM]

### 6.3.3 Matching Methodology

The sector-like patterns observed in AIS data when anchorage complicates the process of determining if a bunker barge is alongside a ship. However, a sensible transformation of coordinate systems, i.e. from a global (lat,lon) to a local (x,y) coordinate system, will remove these unwanted variations. With the local coordinate system, the heading of the ship is aligned with the y-axis with the origin at the ship position. The barge is transformed into the same coordinate system, as shown in figure 6.5. Distance between ship and barge is thereby no longer just a distance between AIS coordinates, which allows for computation of hull to hull distance. Aligning the coordinate systems is illustrated in figure 6.6 (Mestl (2015)).

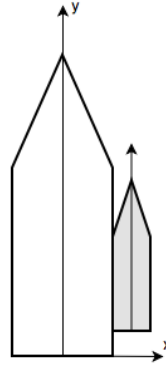


Figure 6.5: Alignment of ship (white) and barge (grey) by a local coordinate system

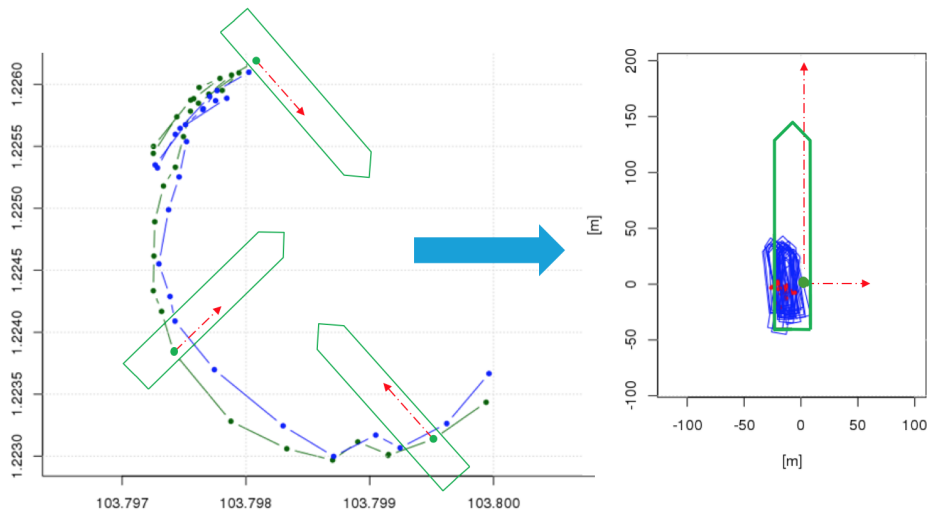


Figure 6.6: Effects from aligning coordinate systems. Global coordinate (left) to local coordinate (right)

Sampling times of AIS messages are usually different for ship and bunker barge, i.e. not synchronised. This may not be an issue for operations at berth, but yields complications when calculating the distance between ship and bunker barge at anchorage due to the drifting which constantly alters the position of each actor. Calculation of the distance between ship and barge at anchorage must therefore be done for coordinates with matching time stamps. This can be solved by applying linear interpolation to fill the data gaps.

Distance between ship and barge is not the distance between their AIS positions, but distance between their hulls when aligned parallel to each other. With the new local coordinate system, this hull-to-hull distance can easily be read from the data. During bunkering, this distance should be small or zero. At berth the ship and barge will be positioned close to each other, but in open sea both vessels will drift during the operation. The movements of ship and barge will not be identical. Consequently, the distance between them will be

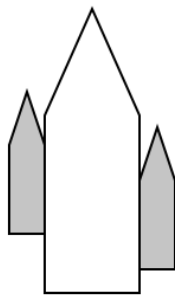


different at various points of the process. When setting a maximum distance between the actors during the operation, one must consider the change in distance between vessels in open sea. If a conservative maximum is set, the risk is that bunker operations with large variations in inter-vessel distance will be rejected as bunker operations.

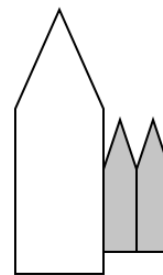


Figure 6.7: Ship aligned with barge

In some cases, the ship can be bunkered by several barges at the same time. These can be positioned on opposite sides of the ship as in figure 6.8a. The bunker barge can also be bunkered by another barge, while simultaneously supplying bunker to a ship. In this case, the second bunker barge will usually be positioned on the outer side of the first barge as illustrated in figure 6.8b.



(a) Ship aligned with two barges



(b) Barge aligned with one ship and one barge

Figure 6.8: Ship aligned with several barges

To calculate the distance between all actors, the sampled headings are manipulated to point in the same direction. This simplifies the process of distance calculation, but the certainty depends on a correctly reported positioning of the AIS antenna on each vessel. An incorrectly reported position can result in e.g. a distance of zero meters, i.e. that the vessels are positioned on top of each other.

### 6.3.4 From Distance to Certainty

Instead of using hull-to-hull distance to determine if a bunker barge and a ship engaged in bunkering, a different approach is used, resulting in a reference measure  $[0,1]$ . *Alignment area*,  $A_{SB}$  is defined as an area around the ship scaled up by the width of the barge and some slack as defined in figure 6.9.

Observe that if there is a bunkering alongside the ship, the bunker barge is fully contained in the  $A_{SB}$  area. Activity is measured from 0 (no activity) to 1 (fully within required distance). Figure 6.9 illustrates two barges registered within close proximity to the ship (200 meters). Barge 1 (B1) is fully within the enclosing limit, whereas barge 2 is completely outside the limit.

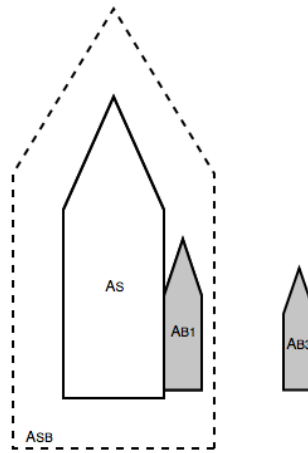


Figure 6.9: Definition of ship and barge alignment area used to calculate the certainty measure.  $A_{B1}$  is fully within the area, while  $A_{B3}$  is fully outside.

To obtain a measure of certainty of a bunkering, the intersection of the barge area  $A_B$  and alignment area  $A_{SB}$  relative to barge area is computed.

$$C_M = \frac{A_{SB_i} \cap A_{B_i}}{A_{B_i}} \quad (6.1)$$

The certainty measure indicates proximity and alignment between ship and barge, and is continuous between 0 and 1. A fully aligned (parallel) positioning will receive a high score, whereas a perpendicular positioning as illustrated rightmost in figure 6.10 will receive a low score.

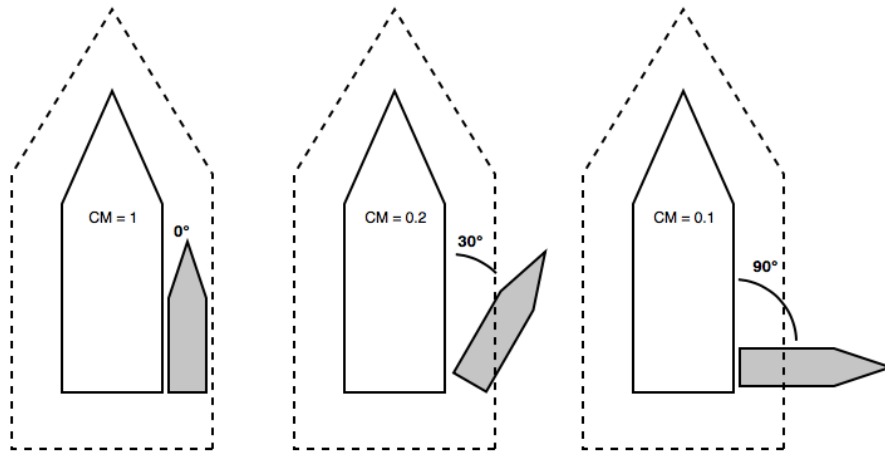


Figure 6.10: Ship and barge alignments with respective certainty measures  $C_M$

Finally, the certainty measure yields a score between 0 and 1. A score of 0 indicates that the distance between the bunker barge was outside the alignment area, i.e. bunker barge and ship most likely not engaged in bunkering, and/or that the vessels were not aligned. In contrast, a score of 1 indicates that the bunker barge was fully within the alignment area and aligned with the ship, i.e. bunker barge and ship likely engaged in bunkering. Figure 6.11 shows three examples of ship-and-barge positionings, and table 6.3 lists their respective scores and reasons for the score.

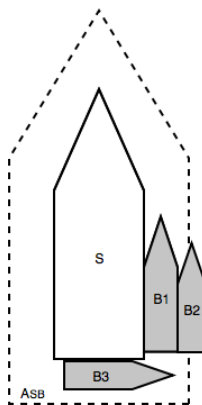


Figure 6.11: Barge positioning. B1 is fully aligned with the ship and fully within the alignment area, while B3 is positioned perpendicularly to the ship.

Table 6.3: Certainty measure for three barges

Barge	Score	Reason	Conclusion
1	1	Aligned and fully within the limit	Bunkering
2	<1	Aligned, but partly within the limit	Undetermined
3	0	Not aligned	No bunkering

Note, a certainty measure can be given to any ship-barge AIS pair with equal time stamps. This means that the original problem consisting of six fluctuating variables - latitude, longitude and SOG for both ship and barge - is now reduced to a one dimensional problem, i.e. when  $C_M$  is 1.

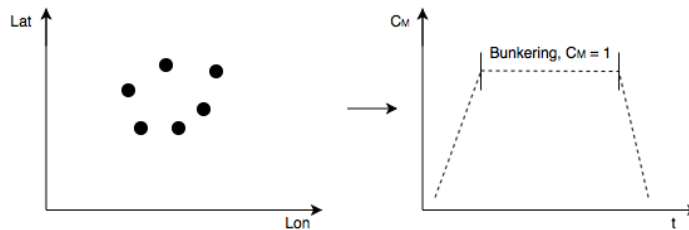


Figure 6.12: Reducing the problem from geographical coordinates with speed to identifying when  $C_M$  is equal to 1

### 6.3.5 Limitations

Due to the structure of the algorithm, the output data will be subject to a few limitations as listed below:

- It is not given that there was an actual bunkering operation; matches are merely based on what the algorithm has proved. However, an identified operation may be a ship-to-ship operation without bunkering.
- Errors in recorded heading can prevent detection.
- The data solely exists of bunkerings that have occurred alongside, meaning that perpendicular bunkerings such as performed by barge 3 in figure 6.11 will be excluded due to low certainty measures.
- Operations involving two bunker barges on each side of the ship are detected as two individual operations.
- All bunkerings are performed at anchorage or berth, excluding all in-voyage bunkerings i.e. recorded SOG above 1 knot.

- The data is limited to operations performed by barges officially approved by the port authorities of Singapore, hence operations performed by other barges or suppliers are not included in the data.

### 6.3.6 Output Data

The output from the algorithm is a Comma-Separated Values (CSV) file containing data for bunker barges within 200 meters proximity. The data spanning a time period of one year yielded approximately 19900 bunker operations performed by officially approved suppliers on a wide variety of ships, and for the first time in history one can actually see *where* the bunkering has occurred. The port authorities have officially designated bunker areas, but where the bunkering has happened in reality is not known - at least not to the public. Figure 6.13 shows the bunker operations identified by the algorithm, distributed over 866 ships and 32 barges.

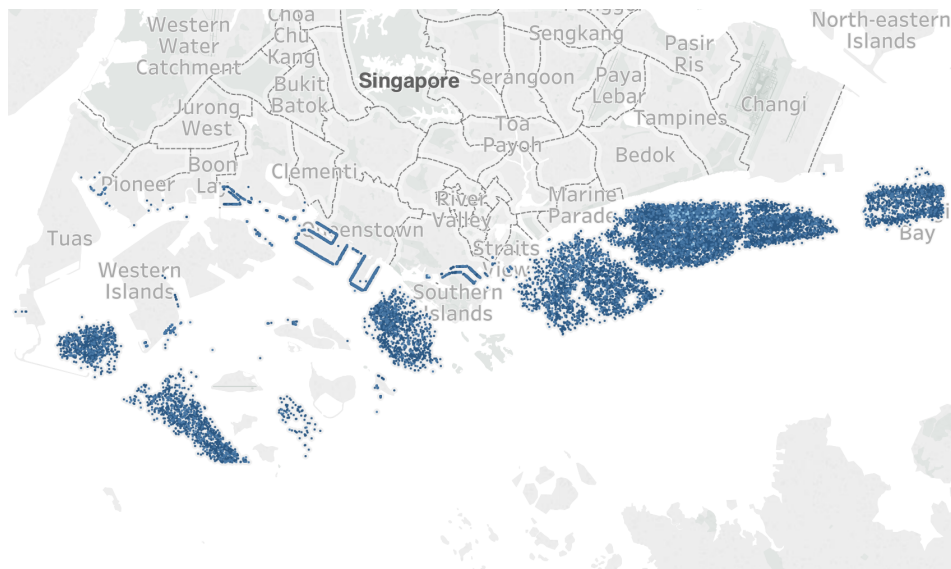


Figure 6.13: Identified bunker operations in Singapore (2017) plotted in *Tableau*

The following figure displays designated anchorage areas in Singapore.

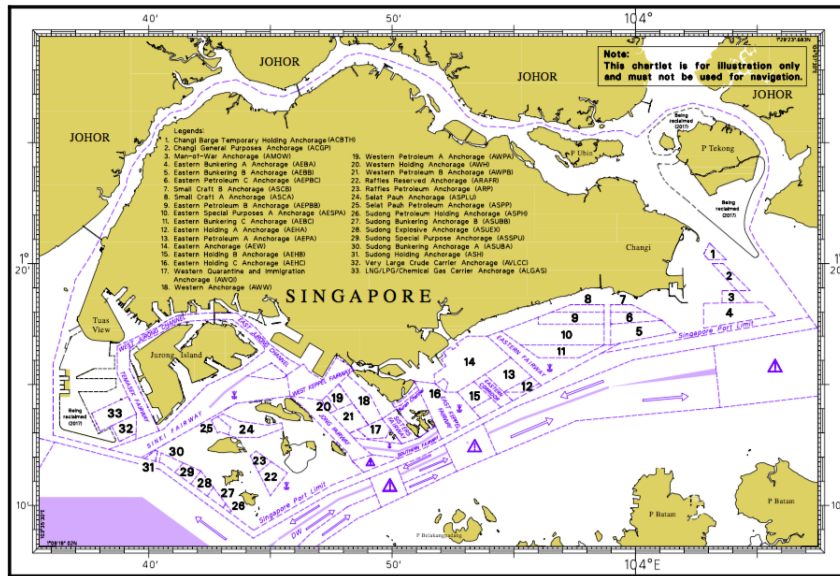


Figure 6.14: Designated anchorage areas in Singapore (MPA (2017)). Note that identified bunkering operations are concentrated in the designated areas.

Comparing the figures above shows that most identified operations are located in the designated areas.

Each row in the CSV file represents one bunker operation, and contains components as shown in table 6.4. An excerpt of one file is given in appendix C, but the IMO numbers and dates removed due to confidentiality reasons of VPS data. The data is not a complete set of all bunker operations performed in Port of Singapore, but will serve its purpose as to map generalities and deviations in bunker operations.

Table 6.4: Bunker barge matching result: components and description from algorithm output

Component	Description
Bunkering number	Numeration of bunker operation
IMO number	International Maritime Organization identification system
Type	Vessel type
Length	Length of vessel [m]
Width	Width of vessel [m]
Ship name	Registered name of ship
VPS sample number	Number of fuel sample taken by VPS
Unixtime	Number of seconds elapsed since 01.01.1970
Bunkering length	Length of bunkering operation [hours]
Median AIS distance	Median distance between AIS antennas [m]
Median hull to hull distance	Median distance between ship and barge hulls [m]
Std. dev. of hull to hull distance	Standard deviation [m]
Anchoring length	Length of ship anchoring [hours]
Time before bunkering	Time from start of anchoring to start of bunkering [hours]
Time after bunkering	Time from end of bunkering to end of anchoring [hours]
Latitude	Horizontal geographic coordinate
Longitude	Vertical geographic coordinate
Bunkering type	At berth or at anchorage
Confidence mean	Mean value of certainty measure
Confidence median	Median value of certainty measure

### 6.3.7 Discussion of Algorithm Structure

Output from the algorithm is referred to as identified operations. However, each identified operation is not necessarily a bunker operation. The algorithm can only to a certain extent confirm a bunker operation through the established certainty measure which is based on alignment and proximity between ship and barge. No data is available to the author to confirm that bunker was actually supplied. We can only assume that identified operations with high certainty measures were bunkering operations. Although an identified operation with a certainty measure of 1 could simply have been a "coffee meeting", such operations will be treated as bunkering operations during analysis. In addition, the minimum required lay time of 2 hours assumes that minimum 2 hours is required to complete bunkering HFO. Shorter lay times are considered as cases where the bunker barge comes alongside the ship but must leave again as the ship was not ready for bunkering.

The matching algorithm utilises data from three different sources; AIS data, VPS data and officially approved bunker barges. Output from the algorithm is restricted by the data sources in different forms. AIS data may be subject to errors due to reasons such as interference or incorrectly calibrated logging equipment, see section 4.1.5. In addition, the algorithm bases its output on operations registered in VPS data. Due to this, the algorithm is not able to identify all operations that have occurred within set geographical and time limits. In addition, VPS fuel quality samples do not contain specifications on which barge performed the operation, thus the algorithm may assign a VPS sample to an incorrect operation or barge. For instance, if two barges bunkered one ship and two samples were submitted, the barges may be matched with the sample submitted on behalf of the other barge. Operations performed by non-approved barges or with missing fuel quality samples will not be identified. Also, an API is used to extract IMO numbers within a certain distance from the ship, and performance of the API is unknown.

Outlier detection in the algorithm is also subject to shortcomings. Not all outliers in retrieved AIS data will be detected. For example a too liberal speed difference restraint between succeeding AIS messages. Undetected outliers in idle times can cause incorrect interpolation between these points, causing interpolated points to receive higher theoretical speed and incorrect geographical coordinates. If interpolated coordinates exceed the distance limit from the ship in the certainty measure, a bunkering operation may be discarded by the algorithm. For example, if a barge is alongside a ship for 3 hours, but AIS messages from this time have an outlier at 1.5 hours, the algorithm will detect this as 2 idle times lasting 1.5 hours each, and not accept them as bunkering operations due to the requirement of minimum 2 hours lay time.



# Chapter 7

## Problem 2: Analysis of Matching Data

This chapter examines the output data from the matching algorithm presented in chapter 6. The output data consists of 19900 assessed bunkering operations. Treatment of data and methods for statistical analysis were implemented in Python. Codes and pseudo codes are included in appendix E. The aim of this chapter is to aggregate data and present results with respect to bunkering time and other parameters.

### 7.1 Introduction

The aim of this case study is to analyse bunkering operations and obtain results which can be used to derive a benchmark that quantifies bunkering operations. Output data from the matching algorithm, presented in section 6.3, is a data frame given as a CSV file where each row represents one bunkering operation, see section 6.3.6 and appendix C.

#### Expressions

To increase the understanding of the reader, expressions used in the thesis must be established. Some of these are self-explanatory, but definitions may be useful for further reading. All necessary expressions are summarised in table 7.1.

Table 7.1: Expressions and definitions

Expression	Description	Acronym
Lay time	Time from start of anchorage to end of anchorage of ship	
Bunkering time	Time from start of bunkering to end of bunkering	BT
Waiting time	Time from start of anchorage to start of bunkering	WT
Post bunkering time	Time from end of bunkering to end of anchorage	PBT
Anchorage time	Lay time at anchorage	
Berth time	Lay time at berth	

### Bunkering Locations

The output provides information on idle times for barges which may be used to investigate which barges are most preoccupied. In addition, it provides information regarding which barge bunkered which ship, when it occurred and for how long. One barge can bunker several ships, and figure 7.1 shows all identified operations performed by one bunker barge.

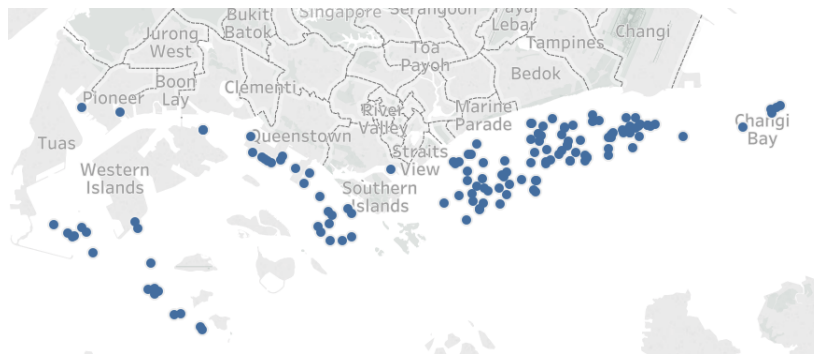


Figure 7.1: Bunkering operations identified for one barge plotted in *Tableau*

## 7.2 Requiring Minimum Confidence Measure

Distribution of certainty measure, from now on called confidence measure, listed in the output is shown by the histogram in figure 7.2. When a ship and a barge are alongside, the confidence measure will be close to 1. However, when these are not alongside the confidence will be close to 0. Confidence measures will be either close to 0 or close to 1 because one can with high certainty confirm or reject a bunkering operation. This explains the two peaks visible in the plot.

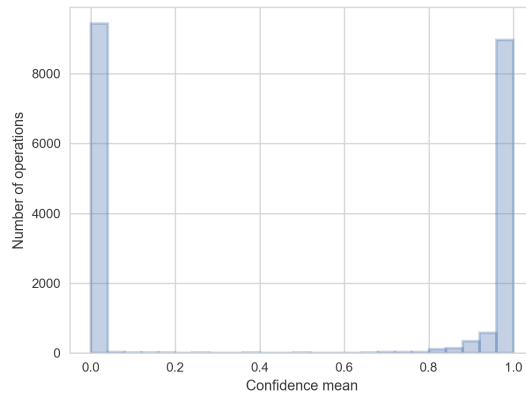


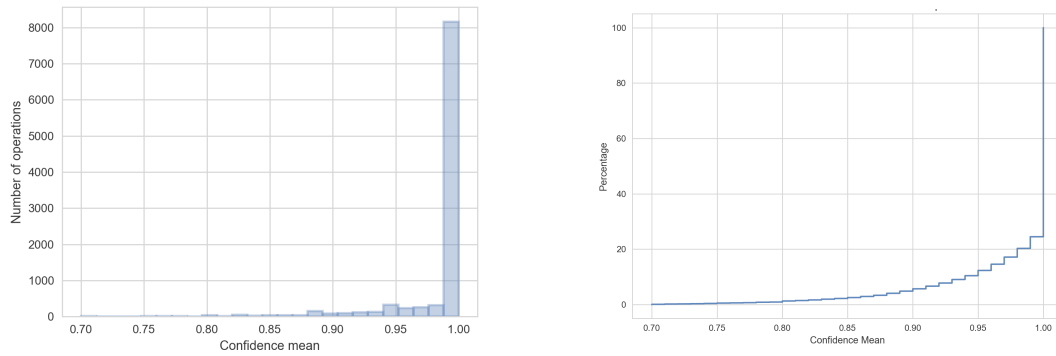
Figure 7.2: Distribution of confidence mean. Several operations have values close to 0.

9468 of the identified operations have a certainty measure of less than 0.1, which is equivalent to 49.8% of all identified operations. The reason why these could not be identified with a higher certainty can be various reasons:

- The ship was bunkered by a not officially approved bunker barge.
- The ship was bunkered outside the geographical scope, but has a few registered AIS messages within the scope with SOG and distance from barge within the required limits.
- The ship and the barge were positioned perpendicular to each other.

Prior to initiating the analysis, the data is filtered by requiring a minimum recorded confidence mean of 0.7. The reason is that operations with lower confidence are with high likelihood not bunkering operations, but rather anchored up with close proximity to each other. Filtering reduces the CSV file 19900 to 10231 operations, which is approximately 51% of the initial data amount.

Figure 7.3a shows the distribution of confidence mean after filtering, while figure 7.3b illustrates confidence mean normalised over number of operations, i.e. 10231 equals 100%, see appendix E.1. Normalised plots are created by sorting the data frame by relevant parameter in ascending or descending order, and enumerating each operation accordingly (from 1 to length of data frame).



(a) Histogram of confidence mean (b) Normalised distribution of confidence mean

Figure 7.3: Distribution of confidence mean above 0.7, 95.3% of all operations have a certainty measure of more than 0.9

Note that a majority of operations in the filtered data have a certainty measure of approximately 1. Lower confidence mean may occur by misalignment or due to quality issues in antenna location.

### 7.3 Ship Types

Each bunkering operation listed in the CSV file includes one ship (receiver) and one bunker barge (supplier). Along with IMO number and vessel length, the vessel types of both actors are specified. 32 vessel types are registered among the receivers and distributed over 10231 bunkering operations with certainty above 0.7. Figure 7.4 illustrates the distribution of the receiver vessel types in descending order, Python code is found in appendix E.2. It is worth noting that the distribution does not consider that 3516 unique IMO numbers are registered, meaning that several IMO numbers are registered with multiple bunkerings.

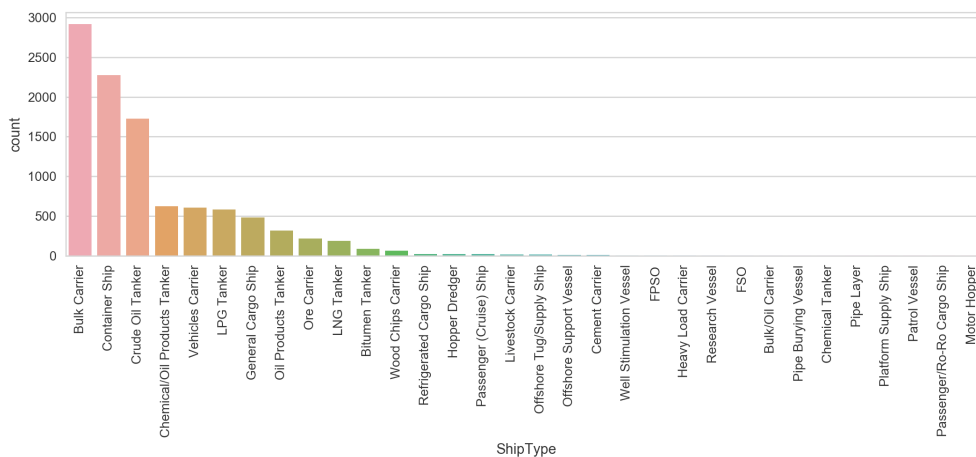


Figure 7.4: Distribution of ship types. The three first ship types constitute 67% of the bunkerings

From the figure it is evident that Bulk Carriers, Container Ships and Crude Oil Tankers are the most prominent types. Several vessel types have fewer than 100 registered bunkering operations and will be excluded from further analysis, as a lower number of operations is insufficient for proper analysis. The data frame will from this point be restricted to include ship types as listed in table 7.2, i.e. vessels to the left of Bitumen Tankers in the figure above, reducing the data to 9929 operations. Median values are listed rather than mean values, as medians are less sensitive to outliers. Note that oil products tankers can be both ship and barge. If it acts as a barge in any of the identified operations, there was a barge-to-barge operation where fuel was transferred from one barge to another. This is known due to the fuel sample submitted to VPS.

Table 7.2: Ship types used in analysis

Ship type	Num of regist. operations	% of total	Median ship length
Bulk Carrier	2919	29.4 %	229
Container Ship	2273	22.9 %	294
Crude Oil Tanker	1724	17.4 %	276
Chemic./Oil Prod. Tanker	626	6.3 %	183
Vehicles Carrier	605	6.1 %	199
LPG Tanker	582	5.9 %	225
General Cargo Ship	481	4.8 %	180
Oil Products Tanker	318	3.2 %	185
Ore Carrier	214	2.2 %	327
LNG Tanker	187	1.8 %	288

Figure 7.5 shows the distribution of ship lengths for ship types with minimum 100 registered bunkerings, see appendix E.3 for Python code. The lengths are not evenly distributed, although most lengths are concentrated around 150-300 meters. Median ship length for all ships is 230 meters.

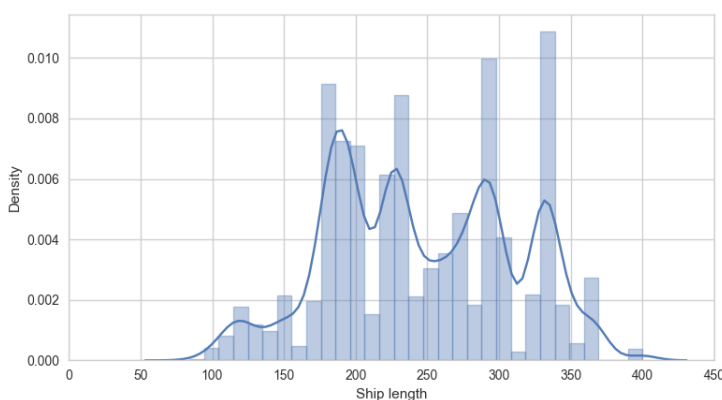


Figure 7.5: Distribution of ship lengths, median = 230 meters

As previously established, bunkering operations can be divided into two categories; *at anchorage* and *at berth*. It is assumed that time at berth is longer than at anchorage due to additional operations such as loading and unloading of goods. To investigate whether this assumption holds, the bunkering operations are divided into type of bunkering. From initial visualising, it was found that several bunkering operations had registered anchorage time of up to 1400 hours. It is reasonable to assume that ships associated with operations with excessive lay times have been engaged in other missions parallel with bunkering, such as repair or lay off. Operations with anchorage time of more than 150 hours are therefore segregated from the data. The distribution of the truncated data is plotted in figures 7.6, 7.7 and 7.8, in which the segregated operations are represented as one individual bin in the rightmost position of the plots (see appendix E.4 for code).

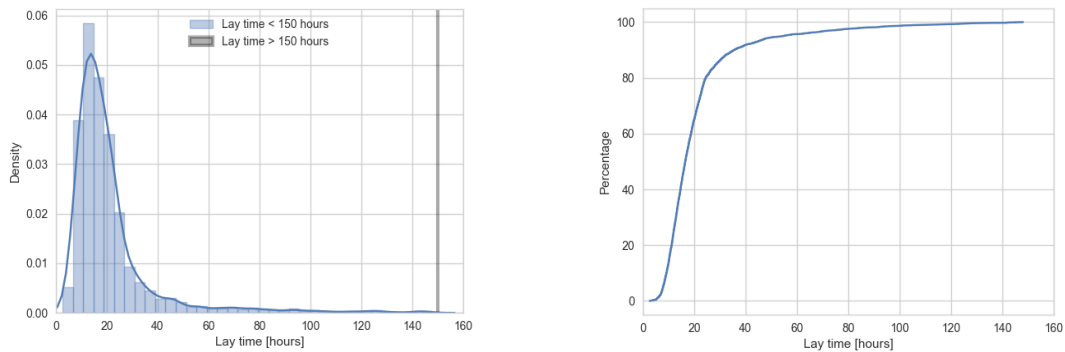


Figure 7.6: Histogram (left) and normalised plot (right) of lay time all ships. Observe that 75% of all ships had lay time of 23.55 hours or less.

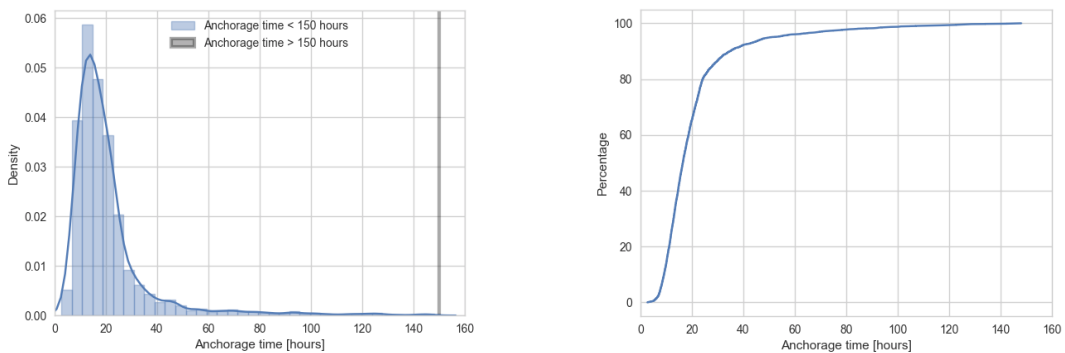


Figure 7.7: Histogram (left) and normalised plot (right) of anchorage time. Observe that 75% of ships at anchorage had anchorage time of 23.22 hours or less.

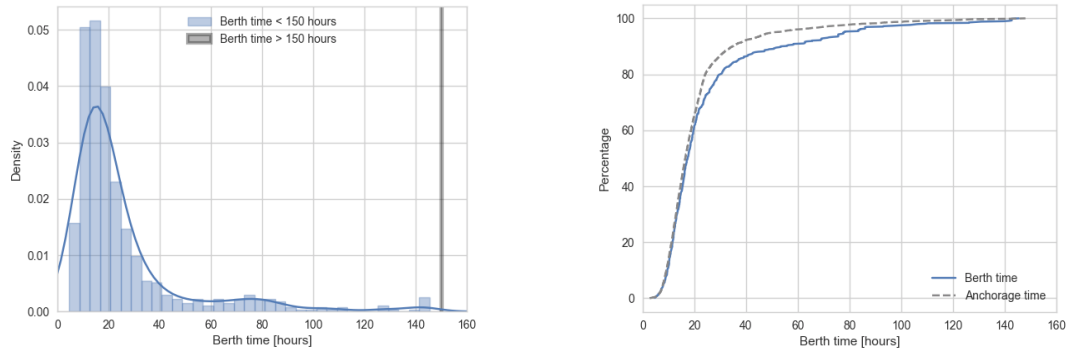


Figure 7.8: Histogram (left) and normalised plot (right) of berth time. For comparison, anchorage time was superimposed in the normalised plot. Observe that 75% of ships at berth had berth time of 27.42 hours or less, and that anchorage and berth follow a similar pattern with more short stays at anchorage.

As seen from the figures above, no operations with registered anchorage time of less than two hours are plotted. This is a result from requiring minimum two hours lay time in the matching algorithm. As anticipated, the concentration of short anchorage periods (less than 30 hours) is slightly higher for ships at anchorage. No obvious outliers are observed in the histogram of figure 7.7. The blue line represents the KDE, which follows the histogram closely. This indicates that the lay time has a top at approximately 16-17 hours, with a large tail to the right. For ships at berth there is a top at 16-17 hours, and a second small peak close to 80 hours. Table 7.3 display median lay time, anchorage time and berth time. Note that median time at berth is longer than at anchorage.

Table 7.3: Median lay time

Parameter	Value
Median lay time	16.6 [hours]
Median anchorage time	16.6 [hours]
Median berth time	17.7 [hours]

The plots to the right in figures 7.7 and 7.8 display normalised distributions of time at anchorage and berth respectively. Further inspection shows that 75% of ships at anchorage have a lay time of 35 hours or less, and 75% of ships at berth lie less than 25 hours, underscoring that ships at berth lie longer than ships at anchorage.

From figure 7.5 it was found that most ships engaged in bunkering operations have lengths exceeding 150 meters. The ships are therefore divided by this limit, creating two groups of ships characterised as *small* and *large* ships. The following figures display lay time for ships at anchorage and at berth. Code is found in appendix E.8.

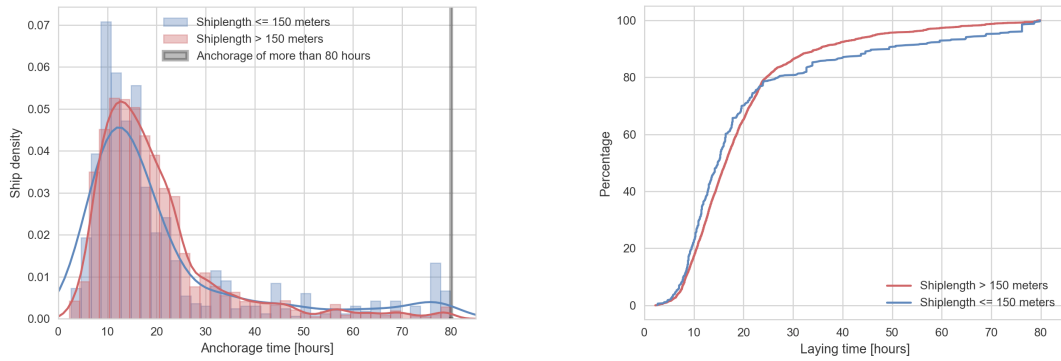


Figure 7.9: Histogram (left) and normalised plot (right) of anchorage time for small and large vessels

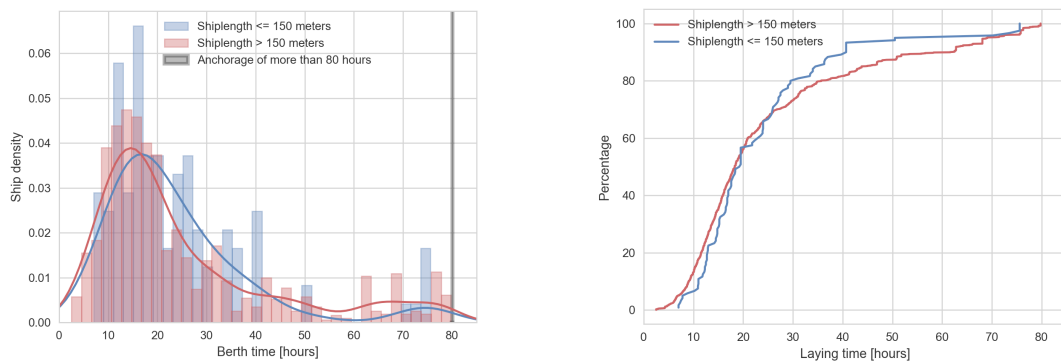


Figure 7.10: Histogram (left) and normalised plot (right) of berth time for small and large vessels. Note that the normalised plot is transversed from the equivalent plot at anchorage.

Distributions of lay time for large ships at anchorage and berth are similar. On the contrary, the distributions for small ships are influenced by spikes at e.g. 40 and 76 hours. Further inspection shows that dominating ship types with more than 70 hours lay time are distributed between general cargo ships and different tankers at berth, and between chemical/oil products tankers, general cargo ships and LPG tankers at anchorage. Small ships with lay time around 40 hours at anchorage are distributed between all ship types under scrutiny. At berth, the dominating ship types are vehicles carriers, general cargo ships and container ships.

## 7.4 Bunkering Time

Bunker barges are the suppliers in a bunkering operation. Among the 9929 registered bunkering operations, 184 individual IMO numbers are distributed over 6 vessel types (reduced from 198 IMO numbers and 7 vessel types in the unfiltered data output). Further



inspection showed that one of these types are *towing vessels*, meaning that (some) towing vessels have been approved by the port authority of Singapore to also deliver fuel, but probably only relatively small quantities. Therefore, towing vessels are included in further analysis. The distribution of the vessel lengths for all bunker barges (left), and barge lengths normalised over number of operations (right) are visualised in figure 7.11 to form an impression of the general bunker barge length. Code is found in appendix E.5.

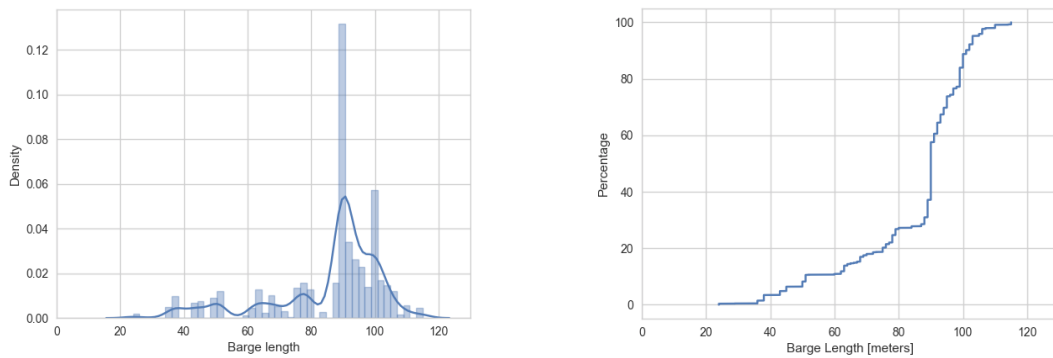


Figure 7.11: Histogram (left) and normalised plot (right) of barge lengths. Median barge length = 90 meters.

No obvious distribution pattern is seen from the barge lengths, although there is a slightly higher concentration around 90-104 meters. Approximately 70% of the 184 bunker barges have lengths of more than 90 meters. Furthermore, the general bunkering time is visualised for barges in figure 7.12.

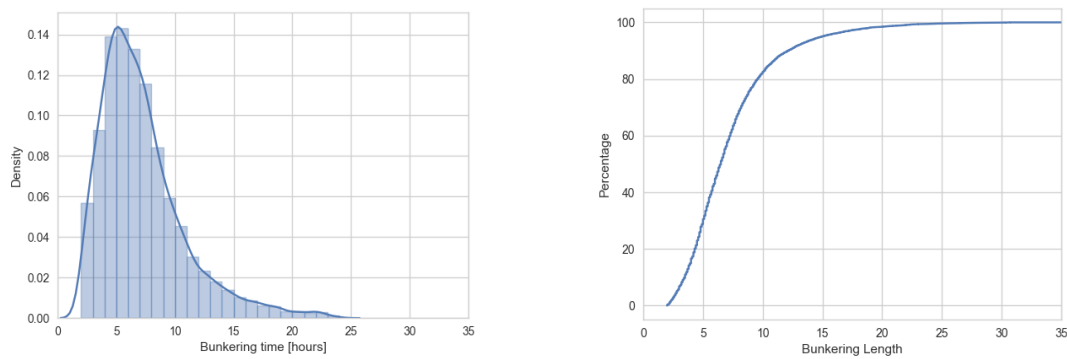


Figure 7.12: Histogram (left) and normalised plot (right) of bunkering time for all barges. Median bunkering time = 6.5 hours. Note that 90% of all barges have bunkering times of less than 14 hours

Due to the requirement of minimum bunkering time in the matching algorithm described in section 6.3, no bunkering operations in the data set have registered bunkering time of less than this limit. Lay time for bunker barges is equivalent to bunkering time. It is evident from the figure that most bunkering times are centred around 2-10 hours, and

the KDE graph is smooth suggesting normal distribution. Of 9929 operations, 9234 were performed at anchorage and 625 at berth. Figures 7.13 and 7.14 display the equivalent plots for bunkering at anchorage and berth, respectively.

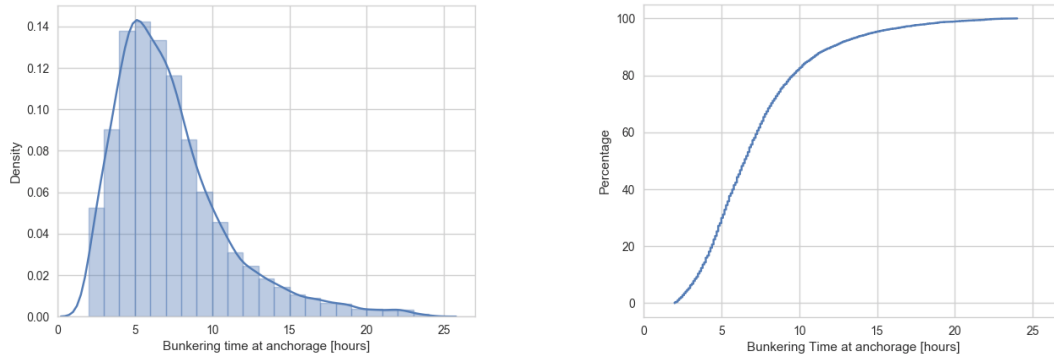


Figure 7.13: Histogram (left) and normalised plot (right) of bunkering time at anchorage

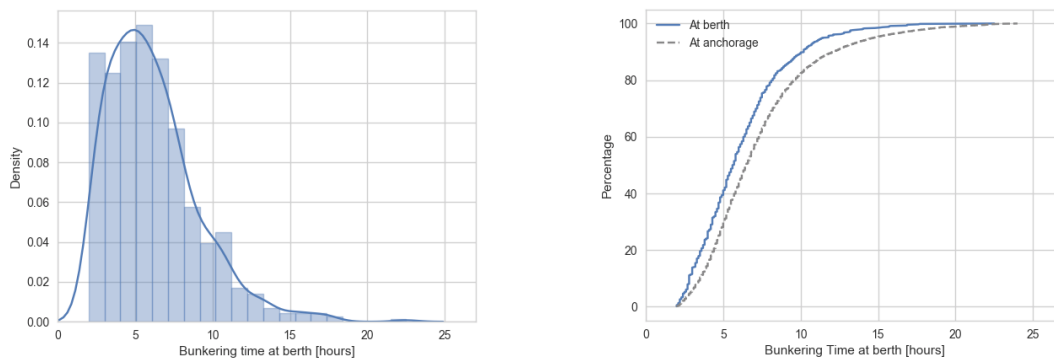


Figure 7.14: Histogram (left) and normalised plot (right) of bunkering time at berth. For comparison, anchorage time was superimposed in the normalised plot. Note that bunkering time at anchorage is usually longer than at berth.

Bunkering time lengths are centred around five hours both at anchorage and at berth. However, fewer operations at berth have bunkering lengths exceeding 20 hours, and the distribution at anchorage is wider. One reason for this may be that operations at anchorage are incorrectly identified by the algorithm due to for example the interpolation.

Another reason may be that ships at berth in general have shorter ship lengths than those at anchorage, due to for example length restrictions at berth where larger ships must bunker at anchorage so larger bunker barges can perform the operation. Time spent connecting ship to barge is generally equal independent of bunkering type, but can be longer at anchorage due to wind and currents. Table 7.4 displays median bunkering time for all operations, operations at anchorage and operations at berth. Note that median bunkering time at anchorage is longer than at berth.

Table 7.4: Bunkering

Parameter	Value
Median bunkering time	6.5 [hours]
Median bunkering time at anchorage	6.5 [hours]
Median bunkering length at berth	5.7 [hours]

Lay time and bunkering time for all operations, operations at anchorage and operations at berth restricted to maximum lay time of 60 hours are plotted in figure 7.15, 7.16 and 7.17 for comparison, see appendix E.6.

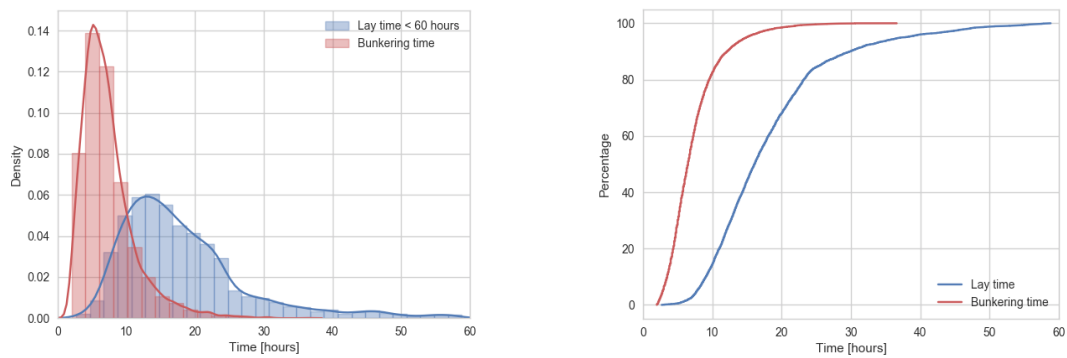


Figure 7.15: Histogram (left) and normalised plot (right) of lay time and bunkering time for all operations

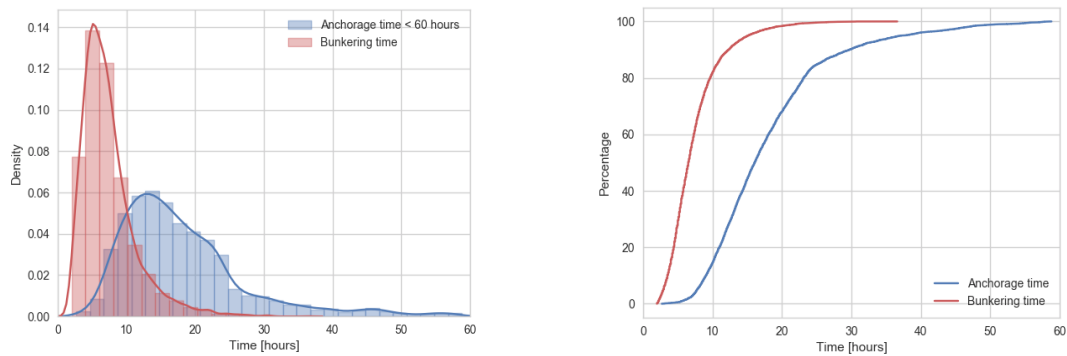


Figure 7.16: Histogram (left) and normalised plot (right) of lay time and bunkering of anchorage time. Note that the distribution is almost identical to general lay time for all vessels in the figure above.

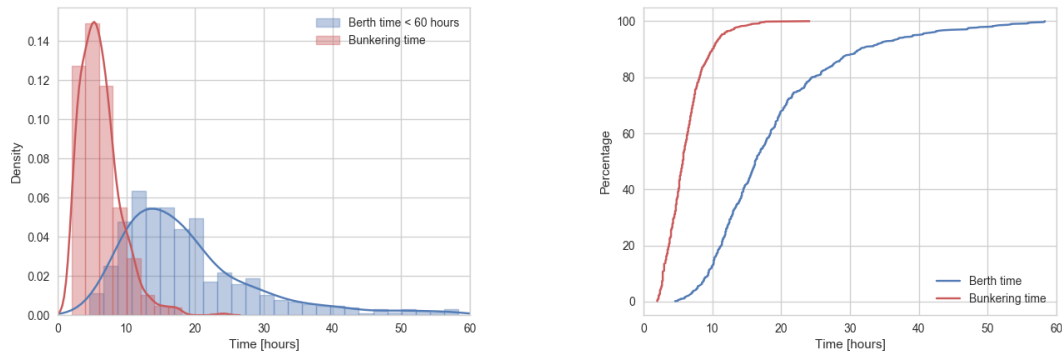


Figure 7.17: Histogram (left) and normalised plot (right) of lay time and bunkering of berth time

In total, 93% of all operations were performed at anchorage, which results in the distributions of time for all operations and at anchorage being quite similar. A larger fraction of the distribution at berth has lay time of more than 20 hours. Bunkering time at berth is however shorter. Median lay and bunkering time at anchor and berth differ with maximum 1.2 hours. The difference is sufficiently small to disregard the separation of bunkering operations at anchorage and berth. Therefore, operations can be analysed without separating them according to bunkering type.

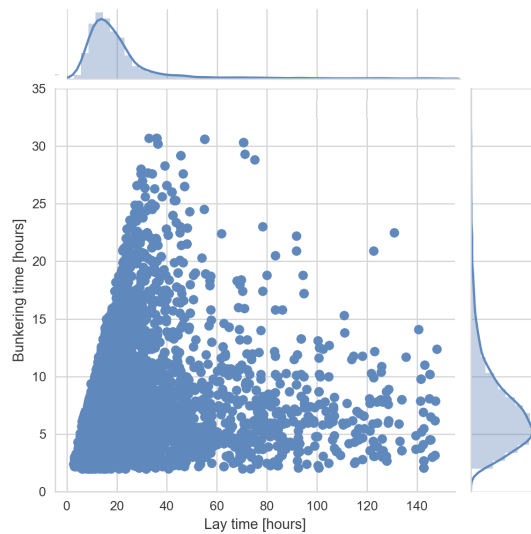


Figure 7.18: Scatter plot of lay time vs bunkering time for all operations. Most operations are located in the lower left corner.

## 7.5 Bunkering Ships

Bunkering times for all ships are identical to bunkering times for all bunker barges presented in figures 7.13 and 7.14 for barges. As for lay time, bunkering times for *small* and *large* ships at anchorage and berth are plotted in figures 7.19 and 7.20, respectively. As expected, bunkering time for large ships is in general longer than for small ships, which is evident in the figures.

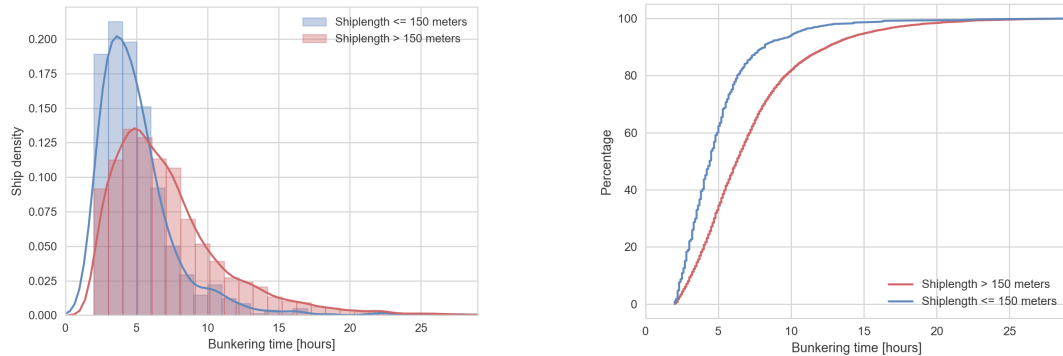


Figure 7.19: Histogram (left) and normalised plot (right) of bunkering time for small and large vessels at anchorage.

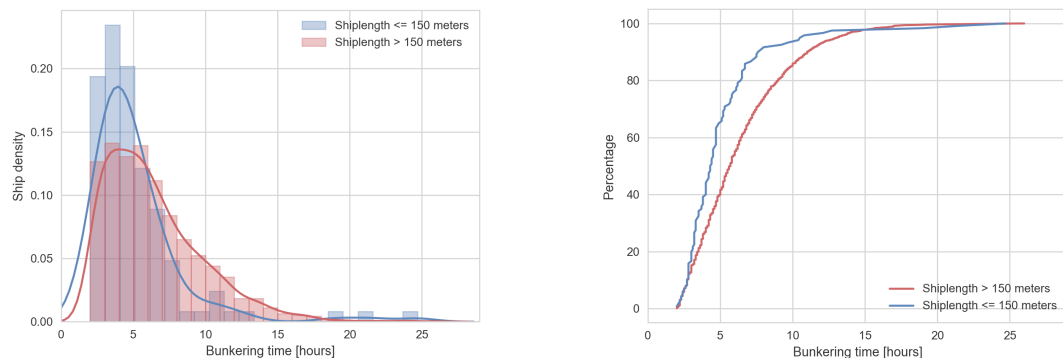


Figure 7.20: Histogram (left) and normalised plot (right) of bunkering time for small and large vessels at berth. Note that distributions for anchorage and berth are quite similar, but bunkering time is longer for *large* ships.

Bunkering lengths for the three dominating ship types are further scrutinised to evaluate whether difference between these may have any impact on the bunkering operation. Figures 7.21a, 7.21b and 7.21c illustrate the distribution of bunkering times for ship types *bulk carriers*, *container ships* and *crude oil tankers* respectively, see appendix E.7.

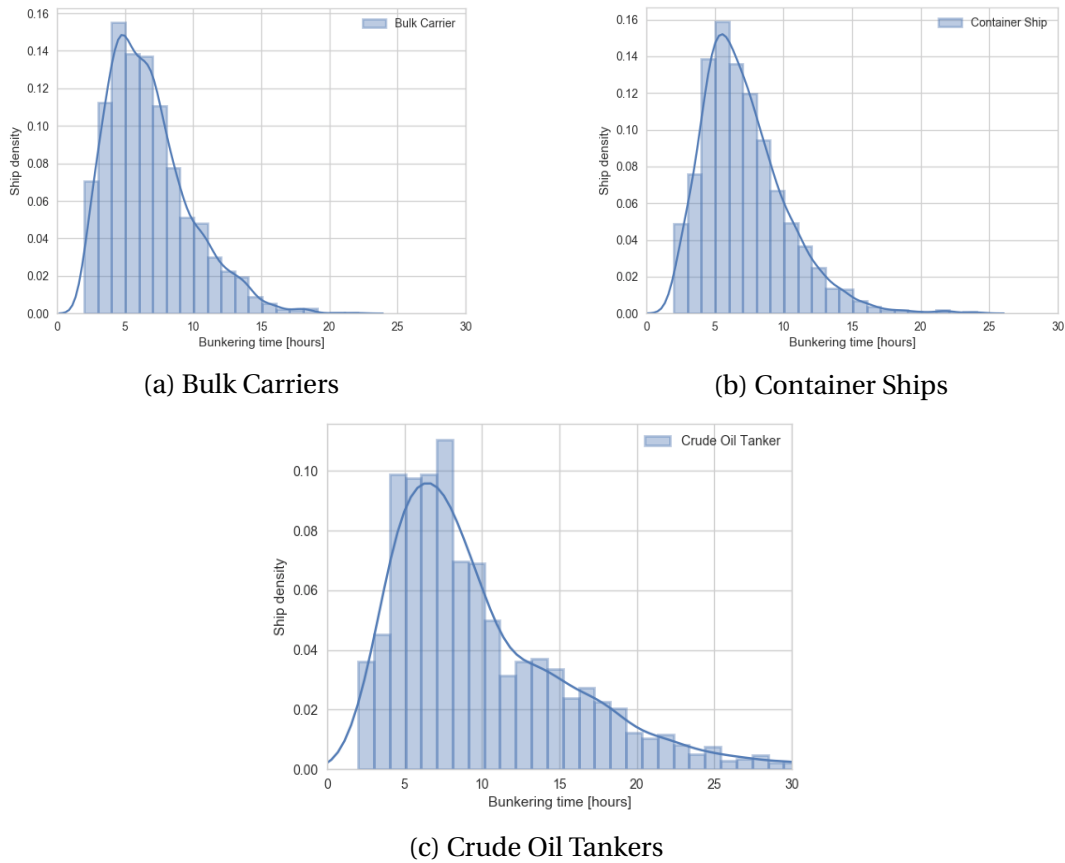


Figure 7.21: Bunkering time for three ship types. Crude Oil Tankers have a longer tail towards longer bunkering time.

As can be seen from the plots above, the distributions of bunkering lengths vary according to ship type. Bulk carriers and containers ship have similar bunkering time distributions. A slightly larger fraction of bulk carriers also has bunkering times of more than 12 hours. Crude Oil Tankers have a longer tail towards long bunkerings. However, it is reasonable to conclude that the variability in bunkering lengths between these three ship types are small relative to each other. Median bunkering time for each ship type is listed in table 7.5.

Table 7.5: Median bunkering time per ship type

Ship type	Num of samples	Med. lay time [hours]	Med. bunk. time [hours]
Bulk Carrier	2919	14.5	6.2
Container Ship	2273	17.0	6.6
Crude Oil Tanker	1724	22.6	8.2
Chem/Oil Prod. Tank.	626	16.2	5.3
Vehicles Carrier	605	12.8	5.8
LPG Tanker	582	16.1	6.3
General Cargo Ship	481	15.3	5.1
Oil Products Tanker	318	16.8	5.7
Ore Carrier	214	24.0	10.1
LNG Tanker	187	24.9	8.7

Median bunkering time per ship type is distributed from 5.1 - 10.1 hours. Ore Tankers possess the highest median, followed by LNG Tankers with a median of 8.7 hours. Higher median bunkering time can be caused by larger ship lengths demanding larger supplies of fuel, meaning that bunkering will require more time. Median bunkering times for ships with most registered operations are spread within a narrow interval, hence, ship lengths for these are plotted in the following figures to investigate whether the assumption of longer ship lengths require longer bunkering operations.

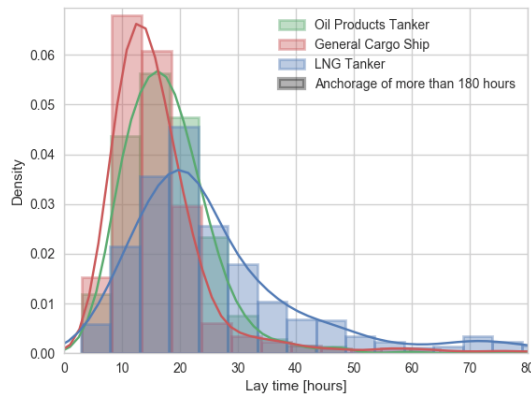


Figure 7.22: Lay time for container ships, bulk carriers and crude oil tankers. Note that LNG tankers have the longest lay time.

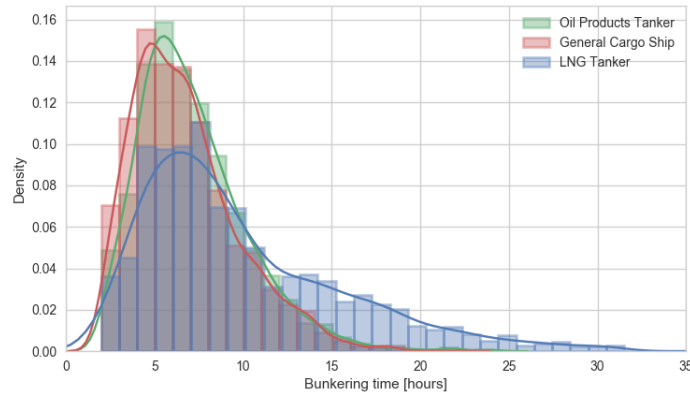
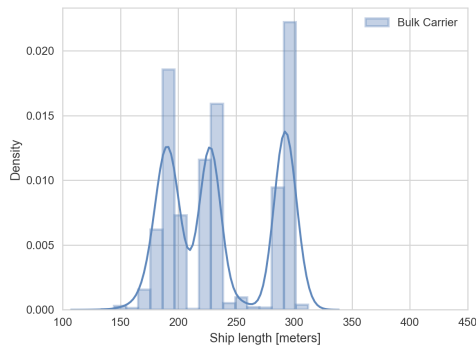
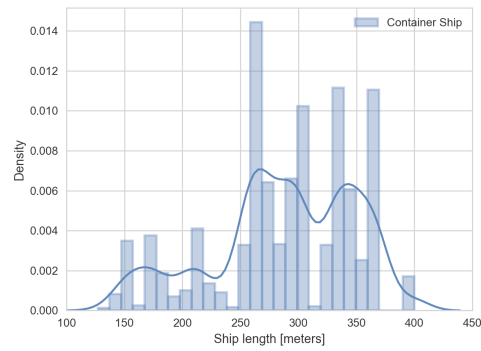


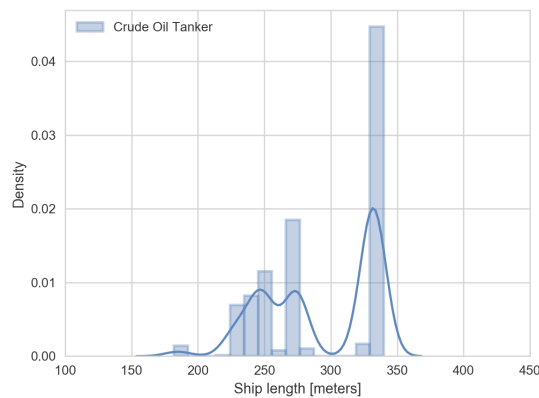
Figure 7.23: Bunkering time for container ships, bulk carriers and crude oil tankers. Note that distributions of lay time and bunkering time per ship type is similar.



(a) Bulk Carriers



(b) Container Ships



(c) Crude Oil Tankers

Figure 7.24: Distribution of ship length for three ship types. Note that bulk carriers have three dominating ship lengths; 190, 240 and 300 meters. Crude oil tankers are generally long, with a group of smaller vessels around 250 meters.

Ship length distributions for bulk carriers, container ships and crude oil follow no specific patterns. From the figures it seems as the increasing median bunkering time follows median ship length. Conferring with table 7.2 rejects this.



## 7.6 Time Before and After Bunkering

Time before bunkering, or *Waiting Time (WT)*, is defined as *time from start of lay time to start of bunkering*. Time after bunkering, or *Post Bunkering Time (PBT)*, is defined as *time from end of bunkering to end of lay time*. Recorded anchorage start time and bunkering start time were used to calculate WT and PBT for each operation as per equation 7.1.

$$\begin{aligned}
 WT &= \text{Unixtime Bunkering Start} - \text{Unixtime Lay Start} \\
 PBT &= \text{Unixtime Lay End} - \text{Unixtime Bunkering End}
 \end{aligned}
 \tag{7.1}$$

Figures 7.25, 7.26a and 7.26b display the distribution of WT for all operations, operations at anchorage and operations at berth respectively, see appendix E.11. No obvious differences can be seen from the plots as the distribution of WT is similar, although WT at berth seems to have fewer registrations exceeding 10 hours. Other than that, it is clear from the distribution that WT for most operations is less than 5 hours.

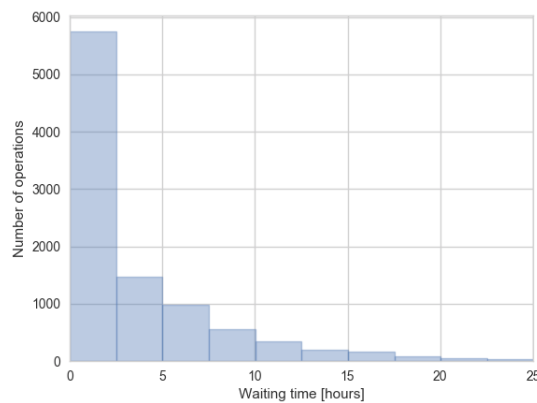


Figure 7.25: Waiting Time for all ships

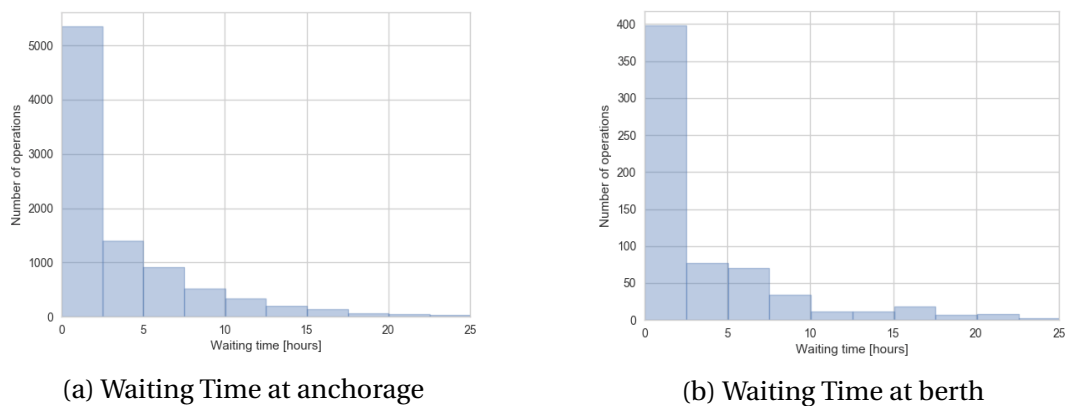


Figure 7.26: Waiting Time per bunkering type. Note that waiting time is similar independent of bunkering type.

Figures 7.27 and 7.28 show PBT for all operations, operations at anchorage and operations at berth respectively to investigate whether these are similar to the WT distributions.

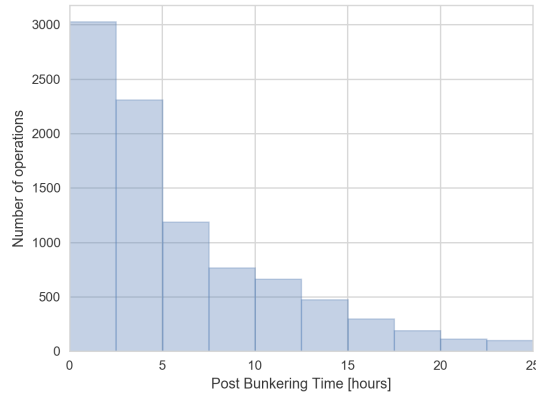


Figure 7.27: Post Bunkering Time for all ships

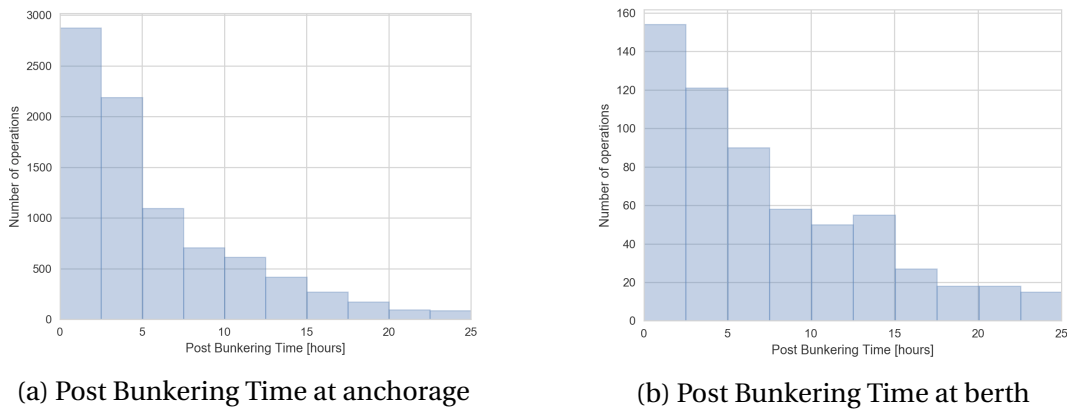


Figure 7.28: Post Bunkering Time per bunkering type. Note that post bunkering time is usually longer at berth.

Figures 7.28a and 7.28b display clear differences. Both decrease with time, but PBT at berth has a long tail towards longer bunkering time than at anchorage. A larger fraction of operations at anchorage have shorter PBT than at berth. Primary reasons for being at berth is loading and unloading. If a bunkered ship is not finished unloading, it will stay at berth for a longer time. In contrast, the primary reason for being at anchorage is bunkering, i.e. after bunkering the ship will leave as soon as possible. Median WT and PBT are displayed in table 7.6 underscoring the plots.

Table 7.6: Median Waiting Time (WT) and Post Bunkering Time (PBT)

Parameter	WT [hours]	PBT [hours]
Median time for all operations	1.8	4.5
Median time at anchorage	1.8	4.4
Median time at berth	1.9	7.0

Merging WT and PBT into one plot yields figure 7.29. Code for the normalised plot is found in appendix E.12.

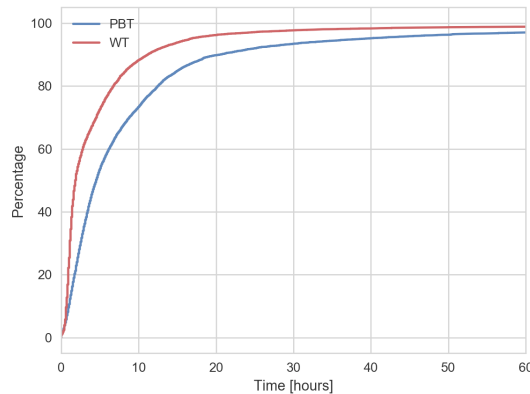


Figure 7.29: Normalised distribution of WT and PBT. Note that Post Bunkering Time generally is longer than Waiting Time.

Focusing on the section of WT and PBT less than 200 hours shows that WT in general is longer than BPT. This indicates that ships usually wait longer to be bunkered than they remain in position post bunkering. Figure 7.30 illustrates the 25% and 75% quantiles of WT and PBT.

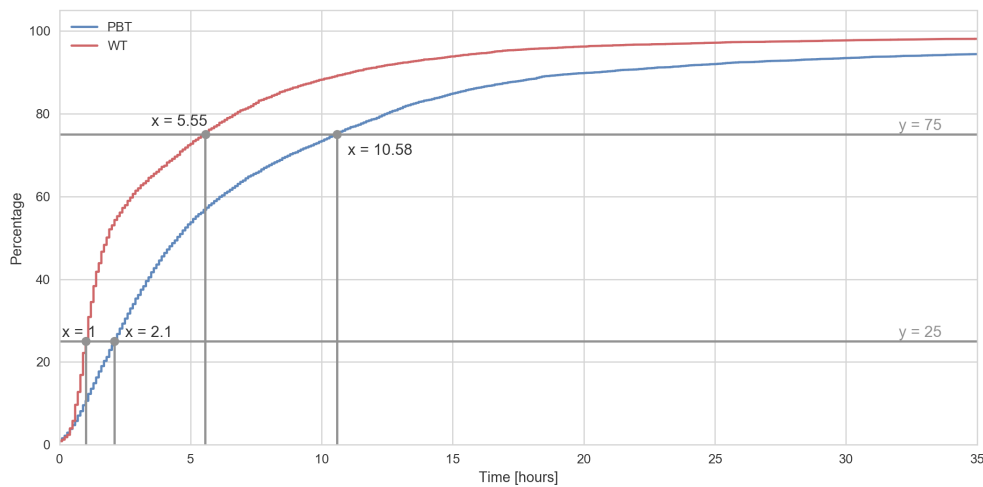


Figure 7.30: 25% and 75% quantiles of Waiting Time and Post Bunkering Time

It is evident from the figure that 75% of all operations have less than hours 5.55 WT and 10.58 hours PBT. Equally, 25% of all operations have less than 1 hour WT and 1.8 hours PBT. Bunkering time and WT were initially the main focus in this thesis. Bunkering time as a parameter is less useful in terms of quantifying bunkering operations as the amount of transferred fuel is unknown. Median WT and PBT for all operations are 1.8 and 4.5 hours respectively, indicating that ships wait for the barge and leave relatively soon post bunkering. It is therefore concluded that these parameters should be implemented in a quality

index. According to John Stirling from WFS, 2 hours WT is usually accepted in the bunkering industry, thus more than 50% of the identified operations are within this limit.

## 7.7 Discussion of Analysis

The analysis is based on data from operations that have been confirmed with a certainty measure of minimum 0.7 by the matching algorithm.

As previously mentioned, reasons for any late arrivals are not known to the author. Both ship and bunker barge can arrive late to a predetermined bunkering operation, which delays the operation. If a ship arrives late, the barge can choose to prioritise another bunkering operation before returning to the initial appointment.

Statistical analysis conducted in this chapter is also based on plots found reasonable to investigate by the author. Numerous plots were constructed to conduct a comprehensive analysis, and the most appropriate and describing plots have been presented.

Dividing between bunkering at anchorage and at berth was regarded as unnecessary due to little differences between plots. However, it is worth noting that operations at anchorage constituted 93% of all identified operations, which means that operations at berth have little impact on the distribution of the data. Identifying more operations at berth would increase the basis for statistical analysis which could yield different results.

# Chapter 8

## Problem 3: A Quality Index for Bunkering Times

This chapter presents a suggested quality index for quantifying bunkering operations and justifications for the structure of the index. Calculation of the index is performed on a selection of operations.

### 8.1 Bunkering Process Time Line

Each bunkering operation is different with regards to time spent before and after bunkering during anchorage. Figure 8.1 illustrates three bunkering operations of different characteristics. The blue lines represent anchorage time for each ship, and the red lines represent bunkering time. Code is found in appendix E.9.

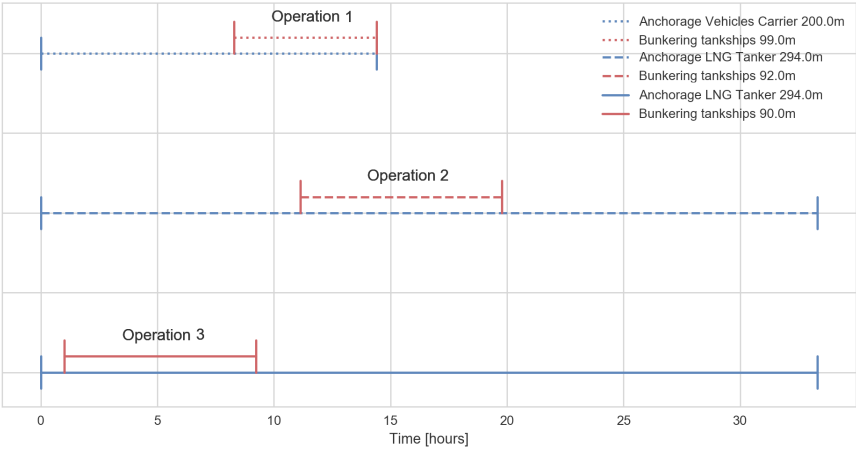


Figure 8.1: Bunkering and anchorage time lines for three operations: Bunkering at the end, middle and start of anchorage.

### Operation 1

As opposed to operations 1 and 2, the waiting time prior to operation 3 is very large relative to time after bunkering. The plot shows that the ship has left the location immediately post-bunkering. With regards to the waiting time, this can indicate that the bunker barge or ship arrival was delayed, and the ship was under pressure to maintain a time schedule.

### Operation 2

Operation 2 is characterised by approximately equal waiting time and time after bunkering. Although the ship waits more than 10 hours for the bunker barge to arrive, it seems to be in no rush to continue its journey as it has a long post-bunkering period. Due to this, operation 2 is also a *good* bunkering.

### Operation 3

Operation 1 may be characterised as a *good* bunkering. The bunker barge arrives for bunkering shortly after the start of anchorage, hence the waiting time is short. The ship also remains anchored for a long period of time after bunkering is completed, and the relationship between time after bunkering and waiting time is high.

## 8.2 Waiting Time and Post Bunkering Time

To get an overview of the general WT-PBT relationship, these parameters are plotted in a scatter plot. Figure 8.2 illustrates a general scatter plot of WT with respect to PBT. In the figure, four sections are marked and characterised from good (sections 1 and 2) to bad (sections 3 and 4). Short WT with short PBT is considered the optimal outcome, while a long WT followed by a short PBT is the worst outcome. Figure 8.3 shows the equivalent scatter plot for the bunkering operations, see appendix E.11.



Figure 8.2: General scatter plot

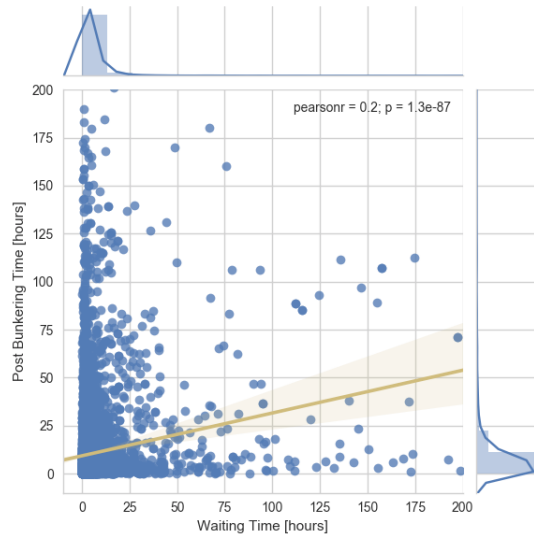


Figure 8.3: Scatter plot of Waiting Time vs Post Bunkering Time. Note that most operations are located in the sections characterised as good. The linear graph is subject to larger variability at longer Waiting Time and Post Bunkering Time. Pearsonr = 0.2 indicates low correlation between points.

Scatter plots of WT and PBT offer a much more grained view of the underlying data. From the scatter plot it is evident that a majority of the operations have WT and PBT in the good sections of the plot, which indicates that most operations are performed within acceptable time frames.

### 8.3 Proposing an Index

Several aspects of bunkering operations were analysed in chapter 7. WT and Bunkering Time were initially regarded as parameters that could be used to derive an index quantifying bunkering operations. If an operation is evaluated merely based on WT, the index can provide a low score due to long WT without considering that the ship can also have a long PBT.

A long WT followed by a long PBT indicates that the ship was in no rush to leave its location. The operation can neither be regarded as good nor bad due to this, since planned time for bunkering cannot be confirmed. However, long PBT indicates that the bunker barge arrived within reasonable time, and it can be assumed that the time frame of the operation was within reasonable limits, qualifying for a high score. On the other hand, a long WT followed by a short PBT can indicate that the operation was delayed either by the barge or the ship arriving later than agreed, which qualifies to a lower score.

The questions are how long it is appropriate to wait prior to bunkering, and at which value WT goes from good to bad. Rather than specifying some arbitrary value, the data decides itself. Quantiles will be used to decide when WT is accepted and not. Quantile values define the maximum or minimum registered value of a parameter within a fraction of the operations when sorting the data according to ascending or descending order of the parameter. Based on this, the 75% quantile values of WT and PBT are retrieved from the analysis and listed in table 8.1. For example, the 60% quantile of WT specifies the maximum WT registered when arranging all operations by ascending WT, and segregating the first 60% of the data. Table 8.1 displays the 25% and 75% quantiles of WT and PBT.

Table 8.1: Waiting Time (WT) and Post Bunkering Time (PBT) parameters

Parameter	Parameter	Value [hours]
WT	25% quantile	1
	75% quantile	5.55
PBT	25% quantile	2.1
	75% quantile	10.58

75% quantiles represent the upper bounds for WT and PBT respectively. This implies that 5.55 hours WT or more is regarded as the worst outcome in a bunkering operation, and 10.58 hours PBT or more is regarded the best outcome. Not all loss is due to late arrivals, so assuming that 25% of bunkering operations are affected by late arrivals yields the lower bound of WT qualifying for a score of zero. The opposite reasoning is laid to ground for limit of PBT. Short WT is desirable while long PBT indicates no rush, and since a 75% quantile yields the lower limit for WT, the equivalent quantile yields the upper limit for PBT.

Contrarily, 25% quantiles should indicate lower bounds for WT and PBT. The 25% quantile for WT is 1 hour. However, according to John Stirling from WFS, a WT of 2 hours is usually accepted, and the lower bound of WT is set accordingly. That is, a  $WT \leq 2$  hours is equivalent to a character of 1, and a WT larger than the 75% quantile qualifies for character 0. Since 25% of all operations have WT within reasonable limits, the lower bound of



PBT should be set accordingly, i.e. lowest acceptable PBT is 2.1 hours. . Similarly, all PBT below the 25% quantile qualify for a character of 0. Code for the following plots are found in appendix E.13. A simple mapping function from WT/PBT to a score could be linearised between the quantiles yielding the following plot, where the x-axis denotes time and y-axis denotes a score from 0 to 1.

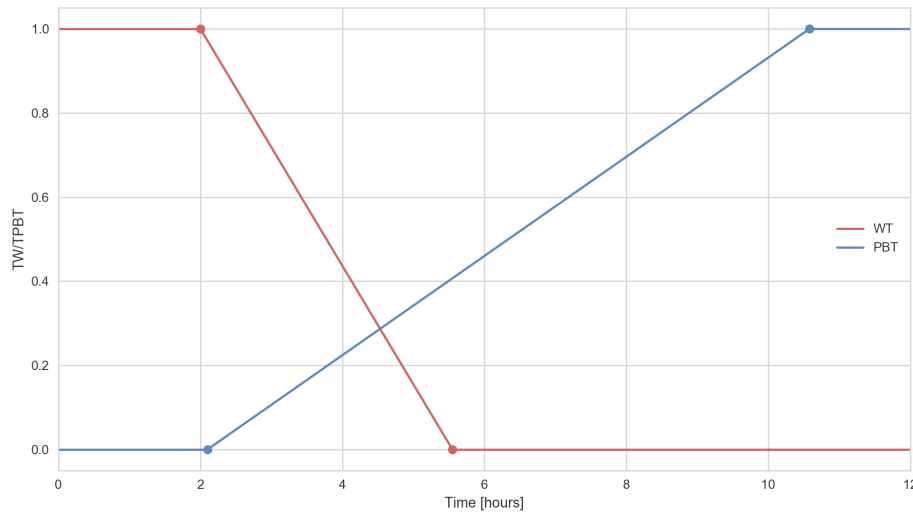


Figure 8.4: Index plot illustrating the linearised relationship between good and bad WT and PBT

Based on the figure above, one can for each operation obtain a score for WT and PBT. A final score should take both parameters into account, to avoid a barge from receiving a penalty based on only one of them. The index should yield high scores if WT or PBT receives character 1. This is justified by the fact that if the ship has a high PBT, i.e. stayed long in port after bunkering, one cannot confirm that the barge arrived later than agreed. Equally, a low WT suggests that the barge arrived on time, and a short PBT should not affect the final score. Therefore, the index should only give penalty when neither of the parameters are scored as 1, yielding the following expression.

$$QBT = T_{WT} + (1 - T_{WT}) \cdot T_{PBT} \quad (8.1)$$

Here, the term *QBT* (*Quantifying Bunkering Time*) is the index per bunkering operation,  $T_W$  and  $T_{PBT}$  are the WT and PBT scores retrieved from figure 8.4.

## 8.4 Quantifying Operations

A selection of bunkering operations is quantified by the index to investigate whether it scores the operations reasonably or not. First, linear equations for increasing and decreasing parts of figure 8.4 are obtained on the form displayed in equation 8.2.

$$y = a + bx \quad (8.2)$$

Following equations are obtained for WT and PBT respectively:

$$\begin{aligned} \text{WT: } I_{WT} &= \frac{111}{71} - \frac{20}{71}WT \\ \text{PBT: } I_{PBT} &= -\frac{105}{424} + \frac{50}{424}PBT \end{aligned} \quad (8.3)$$

Scores are calculated in Python, see appendix E.14, and a selection is displayed in table 8.2. Code for calculation is found in appendix E.14.

Table 8.2: Calculated index (QBT)

Operation number	WT	PBT	$I_{WT}$	$I_{PBT}$	QBT
1	1.1	3.6	1	0.1865	1
2	1.6	1.2	1	0	1
3	10.7	0	0	0	0
4	1	24.1	1	1	1
5	11.1	13.5	0	1	1
6	0.9	68	1	1	1
7	2.7	5.8	0.8028	0.4363	0.8888
8	0.4	4.7	1	0.3066	1
9	1.2	21.3	1	1	1
10	2.4	1.3	0.8873	0	0.8873
11	8.3	0	0	0	0
12	2.6	51.3	0.8309	1	1
13	14.7	3.8	0	0.2005	0.2005
14	2.5	42.4	0.8592	1	1

It is evident from the table that score of PBT becomes effective when the score of WT is between 0 and 1. In all other cases, the score of WT determines the calculated index. These results show that the proposed index is satisfying for quantifying the bunkering time line,

but cannot be used to quantify suppliers without additional verification data. Figure 8.5 displays a histogram of scores for operations at anchorage and at berth.

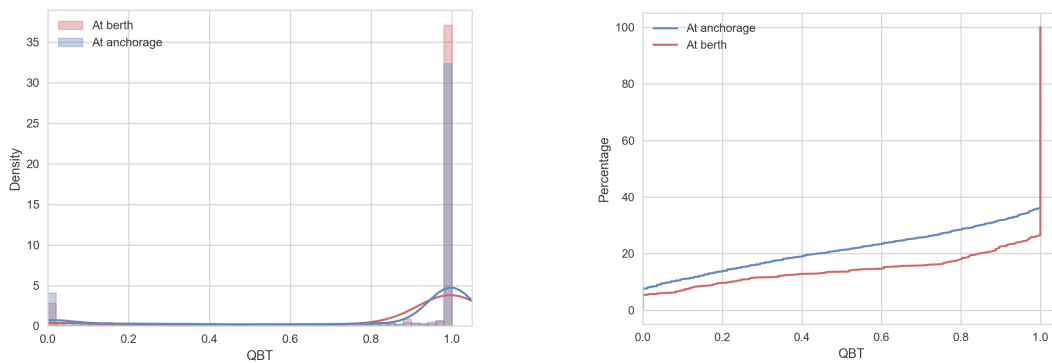


Figure 8.5: Histogram (left) and normalised plot (right) of QBT. Note that a larger fraction of operations at anchorage have lower scores than at berth.

It is evident from the figures that operations at berth in general receive higher scores than operations at anchorage. This suggests that operations at berth are more often performed within reasonable time limits.

## 8.5 Discussion of Index

WT and PBT was a result from the analysis conducted in chapter 7. Based on the analysis it is evident that a quality index should consider both WT and PBT. The reason for this is that if an operation is evaluated merely based on WT, the index can provide a low score due to long WT without considering that the ship can also have a long PBT. WT and PBT are seemingly sufficient for characterising bunkering operations, as the parameter Bunkering Time itself may be inadequate for quantifying operations.

Long WT can be caused by the ship arriving late for the bunkering, leading the bunker barge to prioritise another bunkering appointment before returning to the initial ship. In this case, the barge will receive a low score due to a long WT. A long WT can occur if the ship arrives too early for bunkering, also leading to a low score. It is important to underscore that it is not necessarily the barge that receives a bad score, it is the bunkering event that is quantified - that is both ship and barge.

An index should in theory comprise all aspects necessary for quantification. No fuel quality data is implemented in the index, but expanding the index to comprise these elements should be assessed to achieve a comprehensive decision making parameter. Whether time consumption or fuel quality has the most impact is disputable. Poor fuel quality reduces

the life cycle of machinery, while delays can prevent a ship from completing its missions on time - both inducing higher costs.

Finally, the bunkering time line is subject to variations in each operation. An operation may be performed as planned although the data registers long WT and short PBT. Therefore, the index merely quantifies operations based on set parameters, without verifying any delays. Nonetheless, the innovative benchmarking has proven useful to score operations in general, in forms a framework for further development.

# Chapter 9

## Discussion

This chapter outlines a summarising discussion of each problem presented in previous chapters.

A framework was constructed to identify bunkering operations. Inputs to the matching algorithm are AIS data, VPS data and a list of officially approved bunker barges. AIS data is utilised for identifying idle times for ship and bunker barge and calculate distance between the actors. However, AIS data may be prone to errors due to various reasons, thus yielding incorrect interpolation and hence incorrect identification of idle times and distances. Unsatisfactory results from this identification may lead the algorithm to discard bunkering operations.

Furthermore, initially identified operations match with approximately 60% of operations listed in VPS data, excluding 40% of possible bunkering operations. Since VPS data and officially approved bunker barges are implemented to filter identified operations, bunkerings lacking VPS samples or performed by unapproved suppliers are excluded from the output data. These operations would be valuable to investigate, as the aforementioned factors contribute to reducing the integrity of the suppliers. However, the algorithm has proven useful to identify ship-to-ship operations, and can potentially be developed to identify bunkering operations with higher certainty.

Analysis is based on output data from the algorithm, thus the analysis is only as good as the data itself. Lack of information regarding requested fuel amount, in addition to time and location for the bunkering, complicates the analysis. Hence, the author has relied on guidance from industry experts and assumptions based on visualisation of data. Reasons for late arrivals may be several, and intentional late arrivals can neither be confirmed nor rejected.

WT and PBT were deemed suitable for quantifying bunkering operations. The established

index is based on scoring these parameters according to acceptable and unacceptable values. However, the index quantifies each operation without taking fuel quality into account. Quantifying cannot be done for individual suppliers with this method, but benchmarking can be implemented in industries if access to data of higher quality is provided.

Bunker barges (or suppliers) have been in the centre of attention through analysis and discussions, condemning barges as responsible for malpractices. Nevertheless, ships and crew may also be responsible for such incidents, and the obtained index quantifies the bunkering operation itself and not involved parties.

# Chapter 10

## Conclusions and Recommendations for Further Work

This chapter presents final conclusions from this study, and gives recommendations for further work.

### 10.1 Concluding Remarks

The main objectives for this project were formulated in the introduction as follows:

1. Develop a framework for identifying potential bunker operations.
2. Perform data exploration of bunkering data.
3. Propose criteria for a benchmark quantifying bunker delivery time with respect to lay time.

This thesis has been able to demonstrate that AIS data can be used to identify bunkering operations. However, in some cases the matching algorithm performs poorly. For example, the alignment area  $A_{SB}$  was based on the ship area and scaled by the barge width. In situations where the bunker barge is longer than the ship,  $C_M$  can never be 1. In such cases, ship and barge areas should be swapped. Nevertheless, we were surprised by how well the algorithm worked.

A large fraction of the VPS samples could not be identified. A reason for this may be that the AIS transponder was turned off during the operations, in which case an unlicensed bunker barge can sneak in to perform the operation undetected. Such operations cannot be analysed as no AIS data will be available.

An index was proposed based on WT and PBT. It is concluded that these parameters are sufficient to describe bunkering operations considering the lack of available data such as supplied fuel and agreed time and place for bunkering. However, the index can at this stage not quantify suppliers as it is merely based on time distribution, which is insufficient to assign blame on any actor. Nevertheless, the index can rate the bunkering operations from 0 to 1, and is an innovative method of quantifying operations.

While conducting the thesis, a potentially new data set of operations has been obtained. It may thereby contribute to enhance transparency in the bunkering business. The data set can be utilised to identify when and where an operation occurred, and for how long.

One of the main accomplishments of the thesis is to demonstrate that AIS data, even at a 10 minute time resolution, is sufficient to identify bunkering operations. This thesis has opened up a completely new field of study and application area. The algorithm is a proof of concept, in which bunkering operations have been identified with high likelihood. As the algorithm is still under development, it has wide potential in further development but with AIS data of better quality, one can for the first time in history identify operations with high certainty.

## 10.2 Recommendations for Further Work

The algorithm can be improved to speed up the identification of bunker barges. Use of Geohashed<sup>1</sup> AIS tables can be utilised to check all nearby vessels and not only officially approved barges. With geohashed AIS tables there is no need for the API *VesselTracker*, and it is possible to filter vessels more efficiently.

The algorithm focuses on the ship when identifying nearby barges. In further work, the algorithm can be expanded to identify whether the ship or the barge is largest and set the distance limit accordingly. With the current structure, the distance limit is set depending on the barge width (two times the barge width), but if the barge is larger than the ship the certainty measure does not work well. Setting the limit according to the smallest vessel makes the limit more conservative. In addition, along with identifying alongside bunkering operations, the algorithm may be expanded to identify perpendicular operations.

As previously mentioned, interpolation in the algorithm is done to create synchronised time samples for ship and barge AIS data. The interpolation is conducted based on existing data points without taking weather and current conditions into account. However, if this

---

<sup>1</sup>Domain geocoding based on hierarchical spatial data structure which subdivides space into buckets of grid shape (Finder (2018))



data is available it can be included to obtain more accurate interpolation results. Where to retrieve this data is not known to the author, but it could serve as a good basis for a separate study.

The constructed index can also be improved to be more comprehensive, such as taking bunkering time and VPS data into account. This could be used to gather data regarding bunker barges and other suppliers to establish a framework for choosing supplier.

Results from the index calculation should be compared to VPS fuel quality samples to verify whether barges that receive low scores, also deliver low quality fuel. VPS data is available, hence a statistical analysis is feasible based on results obtained in this thesis. Due to time limitations this could not be conducted by the author.

The method used in this thesis can be expanded to other types of ship-to-ship operations, such as loading/unloading or transfer of oil/fish/cargo to other vessels, as long as AIS data on a high-resolution level is available. An example is in the fishing industry where illegal amounts are fished and vessels distribute the fish to other actors. This is potentially feasible close to shore, but on high seas the satellite coverage is lower and may not be of satisfying quality.



# Bibliography

- Altan, Y. C. and Otay, E. N. (2017). Maritime Traffic Analysis of the Strait of Istanbul based on AIS data. *Journal of Navigation*, pages 1367–1382.
- Anfindsen, O. J., Løvoll, G., and Mestl, T. (2012). Benchmarking of Marine Bunker Fuel Suppliers : the Good, the Bad, the Ugly. *Emerald Insight*, 19(1):109–125.
- Bole, A., Wall, A., and Norris, A. (2014). Chapter 5 - automatic identification system (ais). In Bole, A., Wall, A., and Norris, A., editors, *Radar and {ARPA} Manual (Third Edition)*, pages 255 – 275. Butterworth-Heinemann, Oxford, third edition edition.
- BunkerWorld (2018). *Bunker World*. <http://www.bunkerworld.com>. Accessed: 08.06.2018.
- BusinessDictionary (2018). Definition of *Statistical Analysis*. <http://www.businessdictionary.com/definition/statistical-analysis.html>. Accessed: 10.05.2018.
- DNV Research & Innovation (2012). *Millions lost to density 'shortlifting' in 2011*.
- EuropeanCommission (2011). *COMMISSION DIRECTIVE 2011/15/EU*. <http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:32011L0015>. Accessed: 03.10.2017.
- Ferrà, C., Tassetti, A. N., Grati, F., Pellini, G., Polidori, P., Scarcella, G., and Fabi, G. (2018). Mapping change in bottom trawling activity in the Mediterranean Sea through AIS data. *Marine Policy*, (March):1–7.
- Finder, W. (2018). Definition of *geohash*. <https://findwords.info/term/geohash>. Accessed: 02.06.2018.
- Goldsworthy, B. (2017). Spatial and temporal allocation of ship exhaust emissions in australian coastal waters using ais data: Analysis and treatment of data gaps. *Atmospheric environment*, 163:77–86.
- Goodwin, E. M. (1975). A statistical study of ship domains. *Journal of Navigation*, 28(3):328–344.

Google (2017a). *Satellite picture of bunkering*. [www.google.no/maps/place/Singapore/@1.297564,104.0801778,794m/data=!3m1!1e3!4m5!3m4!1s0x31da11238a8b9375:0x887869cf52abf5c4!8m2!3d1.352083!4d103.819836](http://www.google.no/maps/place/Singapore/@1.297564,104.0801778,794m/data=!3m1!1e3!4m5!3m4!1s0x31da11238a8b9375:0x887869cf52abf5c4!8m2!3d1.352083!4d103.819836). Accessed: 11.11.2017.

Google (2017b). *Satellite picture of bunkering*. <https://www.google.no/maps/place/Singapore/@1.2612999,103.8382553,499m/data=!3m1!1e3!4m5!3m4!1s0x31da11238a8b9375:0x887869cf52abf5c4!8m2!3d1.352083!4d103.819836>. Accessed: 11.11.2017.

Google (2018). *Google Maps picture of Singapore*. <https://www.google.no/maps/@1.309286,103.814453,11.45z>. Accessed: 07.05.2018.

Hoye, G. K., Eriksen, T., Meland, B. J., and Narheim, B. T. (2008). Space-based AIS for global maritime traffic monitoring. *Acta Astronautica*, 62(2-3):240–245.

IMO (2017). *IMO Identification Number Scheme*. <http://www.imo.org/en/ourwork/msas/pages/imo-identification-number-scheme.aspx>. Accessed: 20.10.2017.

Kable (2018). *Port of Singapore*. <https://www.ship-technology.com/projects/portofsingapore/>. Accessed: 06.05.2018.

Kantharia, R. (2015). *What is the Difference between Lay days and Lay time?* <https://www.marineinsight.com/maritime-law/what-is-the-difference-between-lay-days-and-lay-time/>. Accessed: 03.04.2018.

MarineInsight (2015). *What is Cappuccino Bunker Effect On Ships?* <https://www.marineinsight.com/case-studies/cappuccino-bunker-effect-ships/>. Accessed: 05.05.2018.

McCauley, A. (2018). *Pirates in Southeast Asia: The World's Most Dangerous Waters*. <http://time.com/piracy-southeast-asia-malacca-strait/>. Accessed: 05.06.2018.

Mestl, T. (2015). Powerpoint: *Bunkering*.

MPA (2017). *Designated anchorage areas in Singapore*. <https://www.mpa.gov.sg/web/portal/home/port-of-singapore/operations/port-infrastructure/anchorages>. Accessed: 11.06.2018.

Pan, J., Jiang, Q., Hu, J., and Shao, Z. (2012). An AIS data visualization model for assessing maritime traffic situation and its applications. *Procedia Engineering*, 29:365–369.

Qu, X., Meng, Q., and Suyi, L. (2011). Ship collision risk assessment for the Singapore Strait. *Accident Analysis and Prevention*, 43(6):2030–2036.

- Ristic, B., Scala, B. L., Morelande, M., and Gordon, N. (2008). Statistical analysis of motion patterns in AIS Data: Anomaly detection and motion prediction. *2008 11th International Conference on Information Fusion*, pages 40–46.
- Romeny, B. M. T. H. (2018). 03 . The Gaussian Kernel. *Front-End Vision and Multi-Scale Image Analysis*, pages 37–51.
- Roslan Khasawneh, R. (2018). *Singapore port extends compulsory use of flow meters to distillate barges*. <https://www.reuters.com/article/us-singapore-port-bunker/singapore-port-extends-compulsory-use-of-flow-meters-to-distillate-barges-idUSKBN1HY18U>. Accessed: 06.05.2018.
- Shelmerdine, R. L. (2015). Teasing out the detail: How our understanding of marine ais data can better inform industries, developments, and planning. *Marine policy : the international journal of ocean affairs*, 54:17–25.
- Sidibé, A. and Shu, G. (2017). Study of Automatic Anomalous Behaviour Detection Techniques for Maritime Vessels. *Journal of Navigation*, 70(04):847–858.
- Singapore, M. (2018). *Maritime and Port Authority of Singapore - Bunker Tankers*. <https://www.mpa.gov.sg/web/portal/home/port-of-singapore/services/bunkering/bunkering-services-providers>. Accessed: 02.02.2018.
- Smestad, B. B. (2015). *A Study of Satellite AIS Data and the Global Ship Traffic Through the Singapore Strait*. Master Thesis, NTNU.
- Smestad, B. B., Asbjørnslett, B. E., and Rødseth, Ø. J. (2017). Expanding the Possibilities of AIS Data with Heuristics. *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation*, 11(2):93–100.
- Smith, T. W. P., Jalkanen, J. P., Anderson, B. A., Corbett, J. J., Faber, J., Hanayama, S., O’Keeffe, E., Parker, S., Johansson, L., Aldous, L., Raucci, C., Traut, M., Ettinger, S., Nelissen, D., Lee, D. S., Ng, S., Agrawal, A., Winebrake, J. J., and Hoen, M., A. (2014). Third IMO Greenhouse Gas Study 2014. *International Maritime Organization (IMO)*, page 327.
- SOLAS (2017). *Regulation 19 - Carriage requirements for shipborne navigational systems and equipment*. <http://solasv.mcga.gov.uk/regulations/regulation19.htm>. Accessed: 21.10.2017.
- UNL (2017). *Units of Longitude and Latitude*. [http://astro.unl.edu/naap/motion1/tc\\_units.html](http://astro.unl.edu/naap/motion1/tc_units.html). Accessed: 22.10.2017.
- Winkler, D. (2012). AIS Vessel Identification and Techniques.

Wu, D. and Aarsnes, M. (2017). *An Introduction to Assessing Bunkering Operations Through AIS Data*. Master Thesis, NTNU.

Wärtsilä (2017). *Wärtsilä Encyclopedia of Marine Technology*. [https://www.wartsila.com/encyclopedia/term/bunker-delivery-note-\(bdn\)](https://www.wartsila.com/encyclopedia/term/bunker-delivery-note-(bdn)). Accessed: 05.11.2017.

# Appendix A

## Acronyms

**AIS** Automatic Identification System

**API** Application Programme Interface

**BDN** Bunker Delivery Notes

**BT** Bunkering Time

**CSV** Comma-Separated Values

**DNVPS** DNV Petroleum Service

**GPS** Global Positioning System

**GT** Gross Tonnage

**IALA** International Association of Marine Aids to Navigation and Lighthouse Authorities

**IMO** International Maritime Organization

**KDE** Kernel Density Estimator

**MARPOL** International Convention for the Prevention of Pollution from Ships

**MMSI** Maritime Mobile Service Identity

**MPA** Marine Port Authority

**MT** Metric Tonnes

**NM** Nautical Miles

**OPEC** Operational Costs

**PBT** Post Bunkering Time

**PDF** Probability Density Function

**S-AIS** Satellite Automatic Identification System

**SOG** Speed over ground

**SOLAS** International Convention of Safety of Life at Sea

**SQL** Structured Query Language

**VHF** Very High Frequency

**VPS** Veritas Petroleum Services

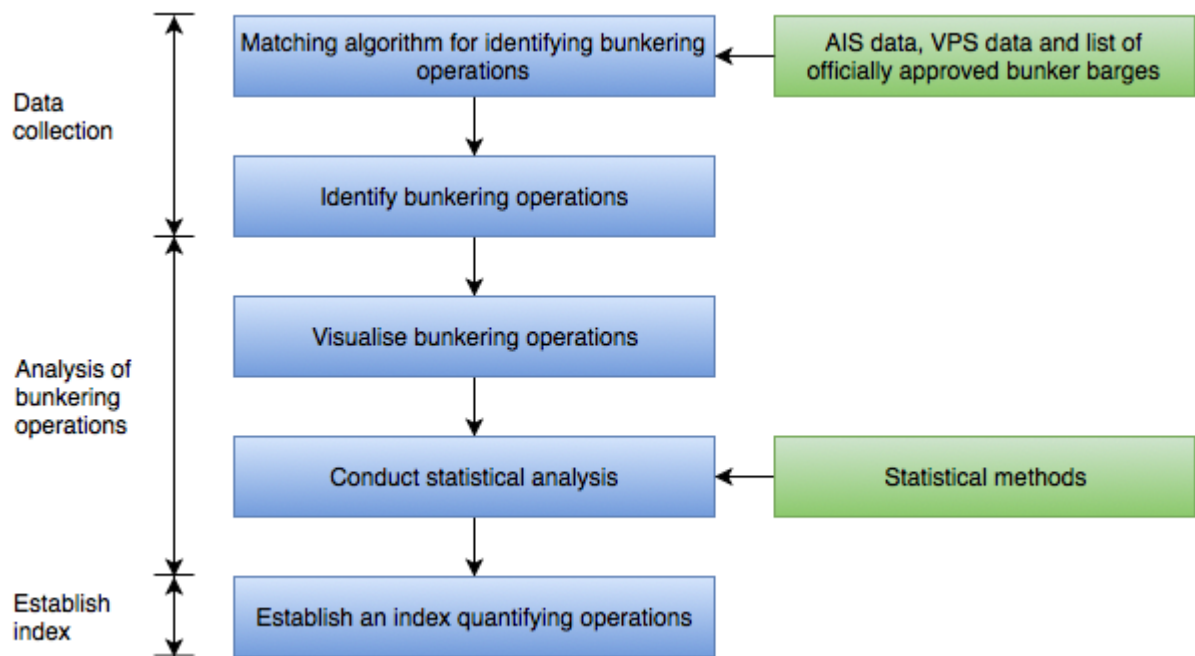
**WFS** World Fuel Services

**WT** Waiting Time



# Appendix B

## Flow Chart of the Study









# Appendix D

## Grouping of Literature

Paper title	Topic	Type of vessel	Methods
Exploring AIS Data for Intelligent Maritime Navigation: A comprehensive survey From Data to Methodology	Review of methods	All	Route planning
	Safety		Collision prediction
Are AIS-based trade volume estimates reliable? The case of crude oil exports	Economy	Tanker	Time series
An AIS data Visualization Model for Assessing Maritime Traffic Situation and its Applications	Safety	All	Collision prediction
	Data handling		
A Comprehensive Inventory of the Ship Traffic Exhaust Emissions in the Baltic Sea from 2006 to 2009	Environment/Emissions	All	Emission model
Study of Automatic Anomalous Behaviour Detection Techniques for Maritime Vessels	Safety	All	Collision prediction
		Cargo	
The complex network of global cargo ship movements	Environment/Emissions	Tanker	Traffic Estimation
Dynamic speed choice in bulk shipping	Economy	All	Speed selection
Missing in Action? Speed optimization and slow steaming in maritime shipping	Economy	All	Speed selection
		All	
Unsupervised learning of maritime traffic patterns for anomaly detection	Safety	All	Collision prediction
Energy efficiency with the application of Virtual Arrival policy	Economy	All	Speed selection
	Environment/Emissions		
Mapping Global Shipping Density from AIS Data	Data handling	Cargo	Other
Spacial and temporal allocation of ship exhaust emissions in Australian coastal waters using AIS data: Analysis and treatment of data gaps	Environment/Emissions	All	Other
Comparison study on AIS data of ship traffic behavior	Safety	Tanker	Pattern recognition
The determinants of vessel capacity utilization: The case of Brazilian Iron Ore exports	Environment/Emissions	Bulk	Other
Maritime traffic analysis of the Strait of Istanbul based on AIS data	Safety	All	Pattern recognition
Geospatial data stream processing in Python using FOSS4G components	Data handling	All	Other
MSARI: A database for large volume storage and utilisation of Maritime Data	Data handling	All	Data handling model
The application of database techniques in the integrated vessel information service system	Data handling	All	Data handling model
Expanding the Possibilities of AIS Data with Heuristics	Data handling	All	Heuristics
Teasing out the detail: How our understanding of marine AIS data can better inform industries, developments, and planning	Environment/Emissions	All	Data handling model
Empirical Ship Domain based on AIS Data	Safety	Others	Route planning
Discovering Knowledge from AIS Database for Application in VTS	Data handling	All	Traffic Estimation
Environmental accounting for Arctic shipping	Environment/Emissions	All	Emission model
A big data driven approach to extracting global trade patterns	Data handling	All	Traffic Estimation
	Economy		
Ship collision risk assessment for the Singapore Strait	Safety	All	
Investigating the Predictive Ability of AIS-data shipping in the digital age: how feasible is the application of big data to the maritime shipping industry, and under what conditions can it be developed to become an integral part of its future?	Economy	Tanker	Traffic Estimation
	General information	Bulk	Traffic Estimation
Data handling			
Trajectory Compression-Guided Visualization of Spatio-Temporal AIS Vessel Density	Data handling	All	Data handling model
A distributed approach to estimating sea port operational regions from lots of AIS data	Economy	All	Route planning
Mapping Global Shipping Density from AIS Data	Safety	All	Traffic Estimation
			Traffic Estimation
Automatic generation of geographical networks for maritime traffic surveillance	Safety	All	Route planning
Scalable and distributed sea port operational areas estimation from AIS data	Economy	All	Traffic Estimation



# Appendix E

## Python code

Python codes used in this thesis are listed in this appendix. Each script is independent of other scripts. All scripts involving reading of the CSV file from the matching algorithm have the following head:

```
1 Import packages
2
3 Read CSV file
4
5 Reduce dataframe to specified columns
6
7 Rename columns
8
9 Remove operations with certainty measure below 0.7
10 Reduce data frame to specified ship types
11
12 Specify plot style
```

Histograms and line plots are created by using Python packages *Seaborn* and *Matplotlib*, which *Seaborn* is built upon. Number of bins in histograms for a parameter is defined defined by retrieving maximum and minimum recorded values of the parameter, and dividing their difference with a specified bin size. KDE lines are plotted by specifying *KDE = True* in *Seaborn* histograms.

Normalised plots of a parameter are constructed by sorting the data frame according to the parameter in ascending order. A new column with normalised values is created with length equal to the dataframe. Each entry in the column is equal to the entry number divided by the length of the dataframe - transforming each entry to the percentage of total length. The sorted parameter and the normalised columns are plotted in a line plot.

## E.1 Distribution of Confidence Mean

### Confidence\_Mean\_Plotting.py

```

1  """
2  Plot histogram and normalised distribution of Confidence Mean ...
   (certainty measure)
3  Created: 29. April 2018
4  Author: Marion Aarsnes
5  """
6
7  import pandas as pd           # Data analysis tools
8  import seaborn as sns        # Library of plots built upon ...
   matplotlib
9  import matplotlib.pyplot as plt # Plots
10 import matplotlib.ticker as mtick # Formatting plots
11 import numpy as np           # Scientific computation
12
13 # Read CSV file
14 AIS_dataframe = pd.read_csv('filedirectory/filename.csv.csv')
15 print(AIS_dataframe.columns)
16
17 # Clean up dataframe
18 AIS_dataframe = AIS_dataframe[['imo.b', 'shipType.s', 'length.s', ...
   'bunkeringLength_hrs', 'anchoringLength_hrs', 'confidence_mean']]
19 AIS_dataframe = AIS_dataframe.rename(columns={"imo.b": "IMO",
20                                             "shipType.s": "ShipType",
21                                             "length.s": "ShipLength",
22                                             "anchoringLength_hrs": ...
   "AnchoringLengthHrs",
23                                             "bunkeringLength_hrs": ...
   "BunkeringLengthHrs",
24                                             "confidence_mean": ...
   "ConfMean"})
25
26 AIS_dataframe = ...
   AIS_dataframe.fillna(value="nan")[AIS_dataframe.ShipType != "nan"]
27
28 # Set plot variables
29 sns.set(style="whitegrid", color_codes=True)
30 sns.despine(left=True)
31
32 # Plot histogram of Confidence Mean from initial algorithm output
33 ax = sns.distplot(AIS_dataframe.ConfMean,

```



```
34         bins=25, kde=False, hist_kws=dict(edgecolor='b', ...
           linewidth=2))
35 ax.set(ylabel="Number of operations", xlabel="Confidence mean", ...
        title="Distribution of Confidence Mean")
36 plt.legend()
37 plt.show()
38
39 # Plot histogram of Confidence Mean  $\geq$  0.7
40
41 # Remove rows with Confidence Mean  $<$  0.7 from dataframe
42 AIS_dataframe = AIS_dataframe[AIS_dataframe.ConfMean  $\geq$  0.7]
43
44 # Plot certainty measure
45 ax = sns.distplot(AIS_dataframe.ConfMean,
46                 bins=25, kde=False, hist_kws=dict(edgecolor='b', ...
           linewidth=2))
47 ax.set(ylabel="Number of operations", xlabel="Confidence mean", ...
        title="Distribution of Confidence Mean")
48 plt.legend()
49 plt.show()
```

## E.2 Distribution of Ship Types

### Distribution\_Shiptypes.py

```

1  """
2  Plot distribution of shiptypes in histograms
3  Created: 02. April 2018
4  Author: Marion Aarsnes
5  """
6
7  # Importing packages
8  import pandas as pd                # Read CSV
9  import seaborn as sns              # Library of plots built upon ...
    matplotlib
10 import matplotlib.pyplot as plt    # Plots
11 import matplotlib.ticker as mtick  # Formatting plots
12 import numpy as np                 # Scientific computation
13 from matplotlib.pyplot import xticks # Rotate plot
14
15 # Read CSV file
16 AIS_dataframe = pd.read_csv('filedirectory/filename.csv')
17 print(AIS_dataframe.columns)
18
19 # Clean up dataframe
20 AIS_dataframe = AIS_dataframe[['imo.b', 'shipType.s', 'imo.s', ...
    'length.s', 'length.b', 'bunkeringLength_hrs', ...
    'anchoringLength_hrs', 'bunkeringType', 'confidence_mean']]
21 AIS_dataframe = AIS_dataframe.rename(columns={"imo.b": "IMO",
22                                             "shipType.s": "ShipType",
23                                             "imo.s": "IMOship",
24                                             "length.s": "ShipLength",
25                                             "length.b": ...
26                                             "BargeLength",
27                                             "anchoringLength_hrs": ..
28                                             "AnchoringLengthHrs",
29                                             "bunkeringLength_hrs": ..
30                                             "BunkeringLengthHrs",
31                                             "bunkeringType": ...
32                                             "BunkeringType",
33                                             "confidence_mean": ...
34                                             "ConfMean"})
35
36 AIS_dataframe = ...
37     AIS_dataframe.fillna(value="nan")[AIS_dataframe.ShipType != "nan"]

```

```
32 # Rename ship types
33 AIS_dataframe = AIS_dataframe.replace({"FSO (Floating, Storage, ...
    Offloading)": "FSO", "FPSO (Floating, Production, Storage, ...
    Offloading)": "FPSO"})
34 # Require a minimum confidence mean of 0.7
35 AIS_dataframe = AIS_dataframe[AIS_dataframe.ConfMean ≥ 0.7]
36
37 # Set plot style
38 sns.set(style="whitegrid", color_codes=True)
39
40 # Plot number of operations per ship type
41 ax = sns.countplot(AIS_dataframe.ShipType, ...
    order=AIS_dataframe['ShipType'].value_counts().index)
42 xticks(rotation=90)
43 plt.show()
44
45 # Count number of registered operations per ship type
46 for shiptype in AIS_dataframe['ShipType'].unique():
47     print(shiptype)
48     print(AIS_dataframe[AIS_dataframe.ShipType == ...
    shiptype]['ShipType'].count())
```

## E.3 Distribution of Ship Lengths

### Ships\_Length\_Distribution.py

```

1  """
2  Plot histograms and distribution of ship lengths and laying time
3  Created: 11. April 2018
4  Author: Marion Aarsnes
5  """
6
7  import pandas as pd                # Read CSV
8  import seaborn as sns             # Library of plots built ...
   upon matplotlib
9  import matplotlib.pyplot as plt   # Plots
10 import matplotlib.ticker as mtick # Formatting plots
11 import numpy as np                # Scientific computing
12 from matplotlib.pyplot import xticks # Rotate plot
13
14 # Read CSV file
15 AIS_dataframe = pd.read_csv('filedirectory/filename.csv')
16 print(AIS_dataframe.columns)
17
18 # Clean up dataframe
19 AIS_dataframe = AIS_dataframe[['imo.b', 'shipType.s', 'imo.s', ...
   'length.s', 'length.b', 'bunkeringLength_hrs', ...
   'anchoringLength_hrs', 'bunkeringType', 'confidence_mean']]
20 AIS_dataframe = AIS_dataframe.rename(columns={"imo.b": "IMO",
21                                             "shipType.s": "ShipType",
22                                             "imo.s": "IMOship",
23                                             "length.s": "ShipLength",
24                                             "length.b": ...
   "BargeLength",
25                                             "anchoringLength_hrs": ...
   "AnchoringLengthHrs",
26                                             "bunkeringLength_hrs": ...
   "BunkeringLengthHrs",
27                                             "bunkeringType": ...
   "BunkeringType",
28                                             "confidence_mean": ...
   "ConfMean"})
29
30 AIS_dataframe = ...
   AIS_dataframe.fillna(value="nan")[AIS_dataframe.ShipType != "nan"]
31 # Require a minimum confidence mean of 0.7

```

```

32 AIS_dataframe = AIS_dataframe[AIS_dataframe.ConfMean ≥ 0.7]
33
34 # Reduce dataframe to shiptypes with minimum 100 registrations
35 AIS_dataframe = AIS_dataframe.query("ShipType in ('Bulk ...
    Carrier', 'Container Ship', 'Crude Oil Tanker', 'Chemical/Oil ...
    Products Tanker', "
36
    'Vehicles Carrier', 'LPG ...
    Tanker', 'General Cargo ...
    Ship', 'Oil Products Tanker', "
37
    'Ore Carrier', 'LNG Tanker',)")
38
39 # Set plot style
40 sns.set(style="whitegrid", color_codes=True)
41
42 # ----- Distribution of ship lengths -----
43
44 # Define number of bins
45 # Bin size
46 res_len = 10
47
48 # Calculate bin number based on bin size
49 min_slen = min(AIS_dataframe['ShipLength'])
50 max_slen = max(AIS_dataframe['ShipLength'])
51 bsize_slen = int((max_slen-min_slen)/res_len) # bin number must be ...
    an integer
52
53 # Plot histogram
54 ax = sns.distplot(AIS_dataframe['ShipLength'],
55
    bins=bsize_slen, kde=False, ...
    hist_kws=dict(edgecolor="b", linewidth=1))
56 ax.set(ylabel="Number of registered ship lengths", xlabel="Ship ...
    length")
57 plt.xlim(0, 450)
58 plt.legend()
59 plt.show()
60
61 # Plot histogram with Kernel Density Line
62 ax = sns.distplot(AIS_dataframe['ShipLength'],
63
    bins=bsize_slen, hist_kws=dict(edgecolor="b", ...
    linewidth=1))
64 ax.set(ylabel="Density", xlabel="Ship length")
65 plt.xlim(0, 450)
66 plt.legend()
67 plt.show()

```

## E.4 Distribution of Lay Time for Ships

### Ships\_Time\_Distribution.py

```

1  """
2  Plot distribution of anchorage and bunkering time lengths for ships,
3  and truncated plots normalised over number of operations.
4  Created: 13. April 2018
5  Author: Marion Aarsnes
6  """
7
8  import pandas as pd                # Read CSV
9  import seaborn as sns             # Library of plots built ...
    upon matplotlib
10 import matplotlib.pyplot as plt   # Plots
11 import matplotlib.ticker as mtick # Formatting plots
12 import numpy as np                # Scientific computing
13 from matplotlib.pyplot import xticks # Get or set the current ...
    tick locations and labels of the x-axis
14
15 # Read CSV file
16 AIS_dataframe = pd.read_csv('filedirectory/filename.csv')
17
18 # Clean up dataframe
19 AIS_dataframe = AIS_dataframe[['imo.b', 'shipType.s', 'length.s', ...
    'bunkeringLength_hrs', 'anchoringLength_hrs', 'bunkeringType']]
20 AIS_dataframe = AIS_dataframe.rename(columns={"imo.b": "IMO",
21                                             "shipType.s": "ShipType",
22                                             "length.s": "ShipLength",
23                                             "anchoringLength_hrs": ..
    "AnchoringLengthHrs",
24                                             "bunkeringLength_hrs": ..
    "BunkeringLengthHrs",
25                                             "bunkeringType": ...
    "BunkeringType"})
26
27 AIS_dataframe = ...
    AIS_dataframe.fillna(value="nan")[AIS_dataframe.ShipType != "nan"]
28 # Require a minimum confidence mean of 0.7
29 print(AIS_dataframe['ShipType'].unique())
30
31 # Reduce dataframe to shiptypes with minimum 100 registrations
32 AIS_dataframe = AIS_dataframe.query("ShipType in ('Bulk ...
    Carrier', 'Container Ship', 'Crude Oil Tanker', 'Chemical/Oil ...

```

```

    Products Tanker',"
33         "'Vehicles Carrier','LPG ...
           Tanker','General Cargo ...
           Ship','Oil Products Tanker',"
34         "'Ore Carrier','LNG Tanker',)")
35
36 # Set plot style
37 sns.set(style="whitegrid", color_codes=True)
38
39 # ----- Anchorage lasting longer than value a -----
40
41 # Create new dataframe for longer lasting anchorages
42 a = 150
43 b = 200 # Not used
44
45 # Create one value for all operations lasting longer than value a
46 def fill_values(a):
47     anch_len_x = []
48     bunk_len_x = []
49     for i in range(len(AIS_dataframe.AnchoringLengthHrs)):
50         anch_len_x.append(a)
51         bunk_len_x.append(b)
52     return anch_len_x, bunk_len_x
53
54 anch_len_x, bunk_len_x = fill_values(a)
55
56 # Create column index to dataframe
57 AIS_dataframe['AnchorLengthx'] = AIS_dataframe.index
58 AIS_dataframe['BunkLengthx'] = AIS_dataframe.index
59 # Append new data to dataframe
60 AIS_dataframe.loc[:, 'AnchorLengthx'] = pd.Series(anch_len_x, ...
           index=AIS_dataframe.index)
61 AIS_dataframe.loc[:, 'BunkLengthx'] = pd.Series(bunk_len_x, ...
           index=AIS_dataframe.index)
62
63 # ----- Length grouping -----
64
65 # Create new dataframe for operations lasting longer than value a
66 AIS_dataframe_x = AIS_dataframe[AIS_dataframe.AnchoringLengthHrs >= a]
67
68 # Create new dataframe for operations lasting less than value a
69 AIS_dataframe_n = AIS_dataframe[AIS_dataframe.AnchoringLengthHrs < a]
70
71 # ----- Lay time for all ships -----
72
73 # Define number of bins

```

```

74 # Bin size
75 res_all = 4
76
77 # Calculate bin number based on bin size
78 min_as = min(AIS_dataframe_n['AnchoringLengthHrs'])
79 max_as = max(AIS_dataframe_n['AnchoringLengthHrs'])
80 bsize_as = int((max_as-min_as)/res_all)
81
82 # Plot histogram
83 plt.figure()
84 ax3 = sns.distplot(AIS_dataframe_n['AnchoringLengthHrs'],
85                   bins=bsize_as, hist_kws=dict(edgecolor="b", ...
86                                               linewidth=1), label='Lay time < 150 hours')
87 axb = sns.distplot(AIS_dataframe_x['AnchorLengthx'],
88                   kde = False, color = '.2', ...
89                   hist_kws=dict(edgecolor="k", linewidth=3), ...
90                   label='Lay time > 150 hours')
91 ax3.set(ylabel="Density", xlabel="Lay time [hours]", title = 'All ...
92         ships')
93 plt.xlim(0,160)
94 plt.legend()
95 plt.show()
96
97 # Laying length normalised over number of operations
98 # Create new column with normalised values
99 def fill_value_conf():
100     norm = []
101     for i in range(len(AIS_dataframe_n.AnchoringLengthHrs)):
102         norm.append((i+1)/ len(AIS_dataframe_n.AnchoringLengthHrs)*100)
103     return norm
104 norm = fill_value_conf()
105
106 # Sort by increasing certainty
107 AIS_dataframe_sortlen = ...
108     AIS_dataframe_n.sort_values('AnchoringLengthHrs')
109
110 # Append normalised column
111 AIS_dataframe_sortlen['norm'] = norm
112
113 # Plot
114 plt.plot(AIS_dataframe_sortlen.AnchoringLengthHrs, ...
115         AIS_dataframe_sortlen.norm)
116 plt.title('Lay length normalised over number of operations')
117 plt.xlabel("Lay time [hours]")
118 plt.ylabel("Percentage")

```



```

114 plt.xlim(0,160)
115 plt.show()
116
117 # ----- Laying time for anchorage and berth -----
118
119 # Create new dataframes for bunkering type
120 # Anchorage
121 AIS_dataframe_anchor = ...
122     AIS_dataframe_n[AIS_dataframe_n.BunkeringType == 'Anchorage']
123 # Berth
124 AIS_dataframe_berth = AIS_dataframe_n[AIS_dataframe_n.BunkeringType ...
125     == 'Berth']
126
127 # Define number of bins
128 # Bin size
129 res_anch = 4 # bin size at anchorage
130 res_bunk = 1 # bin size at berth
131
132 # Calculate bin number based on bin size for anchorage
133 min_asm = min(AIS_dataframe_anchor['AnchoringLengthHrs'])
134 max_asm = max(AIS_dataframe_anchor['AnchoringLengthHrs'])
135 bsize_asm = int((max_asm-min_asm)/res_anch)
136
137 # Calculate bin number based on bin size for berth
138 min_bsm = min(AIS_dataframe_berth['AnchoringLengthHrs'])
139 max_bsm = max(AIS_dataframe_berth['AnchoringLengthHrs'])
140 bsize_bsm = int((max_bsm-min_bsm)/res_anch)
141
142 # ----- Plot time at anchor/berth for all ships -----
143
144 # ----- Anchorage -----
145 # Plot histogram for time at anchorage
146 plt.figure()
147 ax = sns.distplot(AIS_dataframe_anchor['AnchoringLengthHrs'],
148     bins=bsize_asm, hist_kws=dict(edgecolor="b", ...
149     linewidth=1), label='Lay time < 150 hours')
150 axa = sns.distplot(AIS_dataframe_x[AIS_dataframe_x.BunkeringType == ...
151     'Anchorage']['AnchorLengthx'],
152     kde = False, color = '.2', ...
153     hist_kws=dict(edgecolor="k", linewidth=3), ...
154     label='Lay time > 150 hours')
155 ax.set(ylabel="Density", xlabel="Anchorage time [hours]", title ...
156     ='At anchorage')
157 plt.xlim(0,160)
158 plt.legend()
159 plt.show()

```

```

153
154 # Normalised laying time at anchorage
155
156 # Deine new dataframe
157 AIS_dataframe_anchor_len = AIS_dataframe_anchor
158
159 # Sort by increasing certainty
160 AIS_dataframe_anchor_len = ...
        AIS_dataframe_anchor_len.sort_values('AnchoringLengthHrs')
161
162 # Create new column with normalised values
163 def fill_value_anch():
164     norm_anch = []
165     for i in range(len(AIS_dataframe_anchor)):
166         norm_anch.append((i+1)/ ...
                            len(AIS_dataframe_anchor.AnchoringLengthHrs)*100)
167     return norm_anch
168
169 norm_anch = fill_value_anch()
170
171 # Append new column to dataframe
172 AIS_dataframe_anchor_len['norm'] = norm_anch
173
174 # Plot
175 plt.plot(AIS_dataframe_anchor_len.AnchoringLengthHrs, ...
           AIS_dataframe_anchor_len.norm)
176 plt.title('Lay time at anchorage normalised over number of operations')
177 plt.xlabel("Anchoring time")
178 plt.ylabel("Percentage")
179 plt.xlim(0,160)
180 plt.show()
181
182 # ----- Berth -----
183
184 # Plot histogram for time at berth
185 plt.figure()
186 ax3 = sns.distplot(AIS_dataframe_berth['AnchoringLengthHrs'],
187                   bins=bsize_bsm, hist_kws=dict(edgecolor="b", ...
188                   linewidth=1), label='Berth time < 150 hours')
189 axb = sns.distplot(AIS_dataframe_x[AIS_dataframe_x.BunkeringType == ...
190                   'Berth']['AnchorLengthx'],
191                   kde = False, color = '.2', ...
192                   hist_kws=dict(edgecolor="k", linewidth=3), ...
193                   label='Berth time > 150 hours')
194 ax3.set(ylabel="Density", xlabel="Berth time [hours]", title = 'At ...
195         berth')

```

```
191 plt.xlim(0,160)
192 plt.legend()
193 plt.show()
194
195 # Normalised lay time at berth
196
197 # Create new dataframe
198 AIS_dataframe_berth_len = AIS_dataframe_berth
199
200 # Sort by increasing certainty
201 AIS_dataframe_berth_len = ...
    AIS_dataframe_berth_len.sort_values('AnchoringLengthHrs')
202
203 # Create new column with normalised values
204 def fill_value_berth():
205     norm_berth = []
206     for i in range(len(AIS_dataframe_berth)):
207         norm_berth.append((i+1)/ ...
            len(AIS_dataframe_berth.AnchoringLengthHrs)*100)
208     return norm_berth
209
210 norm_berth = fill_value_berth()
211
212 # Append new column to dataframe
213 AIS_dataframe_berth_len['norm'] = norm_berth
214
215 # Plot
216 plt.plot(AIS_dataframe_berth_len.AnchoringLengthHrs, ...
    AIS_dataframe_berth_len.norm)
217 plt.title('Anchoring Length at Berth normalised over number of ...
    operations')
218 plt.xlabel("Berth time [hours]")
219 plt.ylabel("Percentage")
220 plt.xlim(0,160)
221 plt.show()
222
223 # Plot anchorage and berth
224 plt.plot(AIS_dataframe_berth_len.AnchoringLengthHrs, ...
    AIS_dataframe_berth_len.norm, label='Berth time')
225 plt.plot(AIS_dataframe_anchor_len.AnchoringLengthHrs, ...
    AIS_dataframe_anchor_len.norm, '--', color='grey', ...
    label='Anchorage time')
226 plt.title('Anchoring Length at Berth normalised over number of ...
    operations')
227 plt.xlabel("Berth time [hours]")
228 plt.ylabel("Percentage")
```

```
229 plt.xlim(0,160)
230 plt.legend()
231 plt.show()
```

## E.5 Distribution of Bunker Barge Lengths and Time

### BunkerBarges\_Length\_and\_Time\_Distribution.py

```

1  """
2  Plot histograms and distribution of barge lengths and bunkering time
3  Created: 11. April 2018
4  Author: Marion Aarsnes
5  """
6
7  import pandas as pd                # Read CSV
8  import seaborn as sns             # Library of plots built upon ...
   matplotlib
9  import matplotlib.pyplot as plt   # Plots
10 import matplotlib.ticker as mtick # Formatting plots
11 import numpy as np                # Scientific computing
12 from matplotlib.pyplot import xticks # Rotate plot
13
14 # Read CSV file
15 AIS_dataframe = pd.read_csv('filedirectory/filename.csv')
16 print(AIS_dataframe.columns)
17
18 # Clean up dataframe
19 AIS_dataframe = AIS_dataframe[['imo.b', 'shipType.s', 'imo.s', ...
   'length.s', 'length.b', 'bunkeringLength_hrs', ...
   'anchoringLength_hrs', 'bunkeringType', 'confidence_mean']]
20 AIS_dataframe = AIS_dataframe.rename(columns={"imo.b": "IMO",
21                                             "shipType.s": "ShipType",
22                                             "imo.s": "IMOship",
23                                             "length.s": "ShipLength",
24                                             "length.b": ...
   "BargeLength",
25                                             "anchoringLength_hrs": ...
   "AnchoringLengthHrs",
26                                             "bunkeringLength_hrs": ...
   "BunkeringLengthHrs",
27                                             "bunkeringType": ...
   "BunkeringType",
28                                             "confidence_mean": ...
   "ConfMean"})
29
30 AIS_dataframe = ...
   AIS_dataframe.fillna(value="nan")[AIS_dataframe.ShipType != "nan"]
31 # Rename ship types

```

```

32 AIS_dataframe = AIS_dataframe.replace({"FSO (Floating, Storage, ...
    Offloading)": "FSO", "FPSO (Floating, Production, Storage, ...
    Offloading)": "FPSO"})
33 # Requiring a minimum confidence mean of 0.7
34 AIS_dataframe = AIS_dataframe[AIS_dataframe.ConfMean ≥ 0.7]
35
36 # Setting plot style
37 sns.set(style="whitegrid", color_codes=True)
38
39 for shiptype in AIS_dataframe['ShipType'].unique():
40     print(shiptype)
41     print(AIS_dataframe[AIS_dataframe.ShipType == ...
        shiptype]['ShipType'].count())
42
43 # Reduce dataframe to shiptypes with minimum 100 registrations
44 AIS_dataframe = AIS_dataframe.query("ShipType in ('Bulk ...
    Carrier', 'Container Ship', 'Crude Oil Tanker', 'Chemical/Oil ...
    Products Tanker', "
45                                     'Vehicles Carrier', 'LPG ...
                                        Tanker', 'General Cargo ...
                                        Ship', 'Oil Products Tanker', "
46                                     'Ore Carrier', 'LNG Tanker',)")
47
48 # ----- Distribution of bunker barge lengths -----
49
50 # Define number of bins
51 # Bin size
52 res_blen = 2
53
54 # Calculate bin number based on bin size
55 min_blen = min(AIS_dataframe['BargeLength'])
56 max_blen = max(AIS_dataframe['BargeLength'])
57 bsize_blen = int((max_blen-min_blen)/res_blen)
58
59 # Plot histogram
60 ax = sns.distplot(AIS_dataframe['BargeLength'],
61                  bins=(bsize_blen+3), kde=False, ...
                    hist_kws=dict(edgecolor="b", linewidth=1))
62 ax.set(ylabel="Number of registered barge lengths", xlabel="Barge ...
    length [meters]")
63 plt.xlim(0,130)
64 plt.legend()
65 plt.show()
66
67 # Plot histogram with Kernel Density Line
68 ax = sns.distplot(AIS_dataframe['BargeLength'],

```

```

69         bins=bsize_blen, hist_kws=dict(edgecolor="b", ...
           linewidth=1))
70 ax.set(ylabel="Density", xlabel="Barge length")
71 plt.xlim(0,130)
72 plt.legend()
73 plt.show()
74
75 # ----- Distribution of bunkering time for all bunkering types -----
76 # Requiring a maximum bunkering time of 24 hours
77 AIS_dataframe = AIS_dataframe[AIS_dataframe.BunkeringLengthHrs ≤ 24]
78
79 #Define bin size
80 res_blenh = 1
81 min_blenh = min(AIS_dataframe['BunkeringLengthHrs'])
82 max_blenh = max(AIS_dataframe['BunkeringLengthHrs'])
83 bsize_blenh = int((max_blenh-min_blenh)/res_blenh) # bin number ...
           must be an integer
84
85 # Plot histogram
86 ax = sns.distplot(AIS_dataframe['BunkeringLengthHrs'],
87                   bins=bsize_blenh, kde=False, ...
88                   hist_kws=dict(edgecolor="b", linewidth=1))
89 ax.set(ylabel="Number of registered time lengths", ...
90        xlabel="Bunkering time [hours]")
91 plt.xlim(0,35)
92 plt.legend()
93 plt.show()
94
95 # Plot histogram with Kernel Density Line
96 ax = sns.distplot(AIS_dataframe['BunkeringLengthHrs'],
97                   bins=bsize_blenh, hist_kws=dict(edgecolor="b", ...
98                   linewidth=1))
99 ax.set(ylabel="Density", xlabel="Bunkering time [hours]")
100 plt.xlim(0,35)
101 plt.legend()
102 plt.show()
103
104 # ----- Distribution of bunkering time at anchorage and berth -----
105
106 # Creating new dataframes
107 # Anchorage
108 AIS_dataframe_anchor = AIS_dataframe[AIS_dataframe.BunkeringType == ...
109         'Anchorage']
110 # Berth

```

```
108 AIS_dataframe_berth = AIS_dataframe[AIS_dataframe.BunkeringType == ...
    'Berth']
109
110 # Define number of bins
111 # Bin size
112 res_blenab = 1
113
114 # Number of bins at anchor
115 min_blena = min(AIS_dataframe_anchor['BunkeringLengthHrs'])
116 max_blena = max(AIS_dataframe_anchor['BunkeringLengthHrs'])
117 bsize_blena = int(((max_blena-min_blena)/res_blenab))
118
119 # Number of bins at berth
120 min_blenb = min(AIS_dataframe_berth['BunkeringLengthHrs'])
121 max_blenb = max(AIS_dataframe_berth['BunkeringLengthHrs'])
122 bsize_blenb = int(((max_blenb-min_blenb)/res_blenab))
123
124 ax = sns.distplot(AIS_dataframe_anchor['BunkeringLengthHrs'],
125                  bins=bsize_blena, hist_kws=dict(edgecolor="b", ...
                  linewidth=1))
126 ax.set(ylabel="Density", xlabel="Bunkering time at anchorage [hours]")
127 plt.xlim(0,35)
128 plt.legend()
129 plt.show()
130
131 ax = sns.distplot(AIS_dataframe_berth['BunkeringLengthHrs'],
132                  bins=bsize_blenb, hist_kws=dict(edgecolor="b", ...
                  linewidth=1))
133 ax.set(ylabel="Density", xlabel="Bunkering time at berth [hours]")
134 plt.xlim(0,35)
135 plt.legend()
136 plt.show()
```



## E.6 Distribution of Lay and Bunkering Time for All Operations

### Time\_Distribution\_AllOperations.py

```

1  """
2  Plot distribution of anchorage and bunkering time lengths for all ...
   operations,
3  and truncated plots normalised over number of operations.
4  Created: 13. April 2018
5  Author: Marion Aarsnes
6  """
7
8  import pandas as pd                # Read CSV
9  import seaborn as sns             # Library of plots built ...
   upon matplotlib
10 import matplotlib.pyplot as plt    # Plots
11 import matplotlib.ticker as mtick  # Formatting plots
12 import numpy as np                 # Scientific computing
13 from matplotlib.pyplot import xticks # Get or set the current ...
   tick locations and labels of the x-axis
14
15 # Read CSV file
16 AIS_dataframe = pd.read_csv('filedirectory/filename.csv')
17
18 # Clean up dataframe
19 AIS_dataframe = AIS_dataframe[['imo.b', 'shipType.s', 'length.s', ...
   'bunkeringLength_hrs', 'anchoringLength_hrs', 'bunkeringType', ...
   'confidence_mean']]
20 AIS_dataframe = AIS_dataframe.rename(columns={"imo.b": "IMO",
21                                             "shipType.s": "ShipType",
22                                             "length.s": "ShipLength",
23                                             "anchoringLength_hrs": ...
   "AnchoringLengthHrs",
24                                             "bunkeringLength_hrs": ...
   "BunkeringLengthHrs",
25                                             "bunkeringType": ...
   "BunkeringType",
26                                             "confidence_mean": ...
   "ConfMean"})
27
28 AIS_dataframe = ...
   AIS_dataframe.fillna(value="nan")[AIS_dataframe.ShipType != "nan"]

```

```

29 # Require a minimum confidence mean of 0.7
30 AIS_dataframe = AIS_dataframe[AIS_dataframe.ConfMean ≥ 0.7]
31 print(AIS_dataframe['ShipType'].unique())
32
33 # Reduce dataframe to shiptypes with minimum 100 registrations
34 AIS_dataframe = AIS_dataframe.query("ShipType in ('Bulk ...
    Carrier', 'Container Ship', 'Crude Oil Tanker', 'Chemical/Oil ...
    Products Tanker', "
35                                     "'Vehicles Carrier', 'LPG ...
    Tanker', 'General Cargo ...
    Ship', 'Oil Products Tanker', "
36                                     "'Ore Carrier', 'LNG Tanker',)")
37
38 # Set plot style
39 sns.set(style="whitegrid", color_codes=True)
40
41 # ----- Anchorage lasting longer than value a -----
42
43 # Create new dataframe for longer lasting anchorages
44 a = 150
45 b = 200 # Not used
46
47 # Create one value for all operations lasting longer than value a
48 def fill_values(a):
49     anch_len_x = []
50     bunk_len_x = []
51     for i in range(len(AIS_dataframe.AnchoringLengthHrs)):
52         anch_len_x.append(a)
53         bunk_len_x.append(b)
54     return anch_len_x, bunk_len_x
55
56 anch_len_x, bunk_len_x = fill_values(a)
57
58 # Create column index to dataframe
59 AIS_dataframe['AnchorLengthx'] = AIS_dataframe.index
60 AIS_dataframe['BunkLengthx'] = AIS_dataframe.index
61 # Append new data to dataframe
62 AIS_dataframe.loc[:, 'AnchorLengthx'] = pd.Series(anch_len_x, ...
    index=AIS_dataframe.index)
63 AIS_dataframe.loc[:, 'BunkLengthx'] = pd.Series(bunk_len_x, ...
    index=AIS_dataframe.index)
64
65
66
67 # ----- Create new dataframes -----
68

```

```

69 # Create new dataframe for operations lasting longer than value a
70 AIS_dataframe_x = AIS_dataframe[AIS_dataframe.AnchoringLengthHrs >= a]
71
72 # Create new dataframe for operations lasting less than value a
73 AIS_dataframe_n = AIS_dataframe[AIS_dataframe.AnchoringLengthHrs < a]
74
75 # Create new dataframes for bunkering type
76 # Anchorage
77 AIS_dataframe_anchor = ...
78     AIS_dataframe_n[AIS_dataframe_n.BunkeringType == 'Anchorage']
79 # Berth
80 AIS_dataframe_berth = AIS_dataframe_n[AIS_dataframe_n.BunkeringType ...
81     == 'Berth']
82
83 # ----- Scatter plot -----
84
85 plt.scatter(AIS_dataframe_n.AnchoringLengthHrs, ...
86     AIS_dataframe_n.BunkeringLengthHrs)
87 #plt.xlim(-10,200)
88 #plt.ylim(-10,200)
89 plt.title('Laying Time vs Bunkering Time')
90 plt.ylabel("Bunkering Time [hours]")
91 plt.xlabel("Lay Time [hours]")
92 plt.show()
93
94 grid = sns.jointplot(AIS_dataframe_n.AnchoringLengthHrs, ...
95     AIS_dataframe_n.BunkeringLengthHrs, kind="reg", ...
96     joint_kws={'line_kws':{'color':'y'}})
97 grid.set_axis_labels("Lay time [hours]", "Bunkering time [hours]")
98 plt.xlim(0,160)
99 plt.ylim(0,35)
100 plt.show()
101
102 # ----- Laying time and bunkering time -----
103
104 # Define number of bins
105 # Bin size
106 res_all = 4
107
108 # Calculate bin number based on bin size
109 min_as = min(AIS_dataframe_n['AnchoringLengthHrs'])
110 max_as = max(AIS_dataframe_n['AnchoringLengthHrs'])
111 bsize_as = int((max_as-min_as)/res_all)
112
113 # Define number of bins for bunkering time
114 # Bin size

```

```

110 res_bll = 4
111
112 # Calculate bin number based on bin size
113 min_bs = min(AIS_dataframe_n['BunkeringLengthHrs'])
114 max_bs = max(AIS_dataframe_n['BunkeringLengthHrs'])
115 bsize_bs = int((max_bs-min_bs)/res_bll)
116
117
118 # Plot histogram
119 plt.figure()
120 ax3 = sns.distplot(AIS_dataframe_n['AnchoringLengthHrs'],
121                   bins=bsize_as, hist_kws=dict(edgecolor="b", ...
122                   linewidth=1), label='Laying time < 150 hours')
123 ax4 = sns.distplot(AIS_dataframe_n['BunkeringLengthHrs'],
124                   bins=bsize_bs, color='r', ...
125                   hist_kws=dict(edgecolor="r", linewidth=1), ...
126                   label='Bunkering time')
127 #axb = sns.distplot(AIS_dataframe_x['AnchorLengthx'],
128 #                   kde = False, color = '.2', ...
129 #                   hist_kws=dict(edgecolor="k", linewidth=3), label='Laying time > ...
130 #                   150 hours')
131 #axb = sns.distplot(AIS_dataframe_x['BunkLengthx'],
132 #                   kde = False, color = '.2', ...
133 #                   hist_kws=dict(edgecolor="k", linewidth=3), label='Bunkering time')
134 ax3.set(ylabel="Density", xlabel="Time [hours]", title ='All ships')
135 plt.xlim(0,160)
136 plt.legend()
137 plt.show()
138
139
140
141 # ----- At anchorage -----
142
143 # Define number of bins
144 # Bin size
145 res_anch = 4 # bin size at anchorage
146
147 # Calculate bin number based on bin size for anchorage
148 min_anch = min(AIS_dataframe_anchor['AnchoringLengthHrs'])
149 max_anch = max(AIS_dataframe_anchor['AnchoringLengthHrs'])
150 bsize_anch = int((max_anch-min_anch)/res_anch)
151
152 min_abunk = min(AIS_dataframe_anchor['BunkeringLengthHrs'])
153 max_abunk = max(AIS_dataframe_anchor['BunkeringLengthHrs'])
154 bsize_abunk = int((max_abunk-min_abunk)/res_anch)
155
156

```

```
150
151 # Plot histogram
152 plt.figure()
153 ax3 = sns.distplot(AIS_dataframe_anchor['AnchoringLengthHrs'],
154                   bins=bsize_anch, hist_kws=dict(edgecolor="b", ...
155                   linewidth=1), label='Laying time < 150 hours')
156 ax4 = ax3 = sns.distplot(AIS_dataframe_anchor['BunkeringLengthHrs'],
157                   bins=bsize_abunk, color='r', ...
158                   hist_kws=dict(edgecolor="r", linewidth=1), ...
159                   label='Bunkering time')
160 ax3.set(ylabel="Density", xlabel="Time [hours]", title ='At anchorage')
161 plt.xlim(0,160)
162 plt.legend()
163 plt.show()
164
165 # ----- At berth -----
166 # Define number of bins
167 # Bin size
168 res_ber = 4 # bin size at berth
169
170 # Calculate bin number based on bin size for berth
171 min_ber = min(AIS_dataframe_berth['AnchoringLengthHrs'])
172 max_ber = max(AIS_dataframe_berth['AnchoringLengthHrs'])
173 bsize_ber = int((max_ber-min_ber)/res_ber)
174
175 min_bbunk = min(AIS_dataframe_berth['BunkeringLengthHrs'])
176 max_bbunk = max(AIS_dataframe_berth['BunkeringLengthHrs'])
177 bsize_bbunk = int((max_bbunk-min_bbunk)/res_ber)
178
179 # Plot histogram
180 plt.figure()
181 ax3 = sns.distplot(AIS_dataframe_berth['AnchoringLengthHrs'],
182                   bins=bsize_ber, hist_kws=dict(edgecolor="b", ...
183                   linewidth=1), label='Laying time < 150 hours')
184 ax4 = ax3 = sns.distplot(AIS_dataframe_berth['BunkeringLengthHrs'],
185                   bins=bsize_bbunk, color='r', ...
186                   hist_kws=dict(edgecolor="r", linewidth=1), ...
187                   label='Bunkering time')
188 ax3.set(ylabel="Density", xlabel="Time [hours]", title ='At berth')
189 plt.xlim(0,160)
190 plt.legend()
191 plt.show()
192
```

```

190
191
192 # ----- Time normalised over number of operations -----
193
194 # All operations
195 # Create new column with normalised values
196 def fill_value_all():
197     norm = []
198     for i in range(len(AIS_dataframe_n.AnchoringLengthHrs)):
199         norm.append((i+1)/len(AIS_dataframe_n.AnchoringLengthHrs)*100)
200     return norm
201
202 norm = fill_value_all()
203
204 # Create dataframe for laying length
205 AIS_dataframe_norml = AIS_dataframe_n
206 # Sort by increasing certainty
207 AIS_dataframe_norml = ...
208     AIS_dataframe_norml.sort_values('AnchoringLengthHrs')
209 # Append normalised column
210 AIS_dataframe_norml['norm'] = norm
211
212 # Create dataframe for bunkering length
213 # Sort by increasing certainty
214 AIS_dataframe_normb = AIS_dataframe_n.sort_values('BunkeringLengthHrs')
215 # Append normalised column
216 AIS_dataframe_normb['norm'] = norm
217
218 # Plot
219 plt.plot(AIS_dataframe_norml.AnchoringLengthHrs, ...
220     AIS_dataframe_norml.norm, label='Laying time')
221 plt.plot(AIS_dataframe_normb.BunkeringLengthHrs, ...
222     AIS_dataframe_normb.norm, label='Bunkering time', color='r')
223 plt.title('Lay time and bunkering time normalised over number of ...
224     operations')
225 plt.xlabel("Time [hours]")
226 plt.ylabel("Percentage")
227 plt.legend()
228 plt.xlim(0,160)
229 plt.show()
230
231 # Anchorage
232 # Create new column with normalised values
233 def fill_value_anch():
234     norm_anch = []
235     for i in range(len(AIS_dataframe_anchor.AnchoringLengthHrs)):

```

```

232         norm_anch.append((i+1)/len(AIS_dataframe_anchor.AnchoringLengthHrs)*100)
233     return norm_anch
234
235 norm_anch = fill_value_anch()
236
237 # Create dataframe for laying length
238 AIS_dataframe_normla = AIS_dataframe_anchor
239 # Sort by increasing certainty
240 AIS_dataframe_normla = ...
        AIS_dataframe_normla.sort_values('AnchoringLengthHrs')
241
242 print(len(AIS_dataframe_normla))
243 print(len(norm_anch))
244 # Append normalised column
245 AIS_dataframe_normla['norm_anch'] = norm_anch
246
247 # Create dataframe for bunkering length
248 # Sort by increasing certainty
249 AIS_dataframe_normbl = ...
        AIS_dataframe_anchor.sort_values('BunkeringLengthHrs')
250 # Append normalised column
251 AIS_dataframe_normbl['norm_anch'] = norm_anch
252
253 # Plot
254 plt.plot(AIS_dataframe_normla.AnchoringLengthHrs, ...
        AIS_dataframe_normla.norm_anch, label='Anchorage time')
255 plt.plot(AIS_dataframe_normbl.BunkeringLengthHrs, ...
        AIS_dataframe_normbl.norm_anch, label='Bunkering time', color='r')
256 plt.title('Lay time and bunkering time normalised over number of ...
        operations at anchorage')
257 plt.xlabel("Time [hours]")
258 plt.ylabel("Percentage")
259 plt.legend()
260 plt.xlim(0,160)
261 plt.show()
262
263
264 # Berth
265 # Create new column with normalised values
266 def fill_value_ber():
267     norm_ber = []
268     for i in range(len(AIS_dataframe_berth.AnchoringLengthHrs)):
269         norm_ber.append((i+1)/len(AIS_dataframe_berth.AnchoringLengthHrs)*100)
270     return norm_ber
271
272 norm_ber = fill_value_ber()

```

```

273
274 # Create dataframe for laying length
275 AIS_dataframe_normlb = AIS_dataframe_berth
276 # Sort by increasing certainty
277 AIS_dataframe_normlb = ...
    AIS_dataframe_normlb.sort_values('AnchoringLengthHrs')
278
279 print(len(AIS_dataframe_normlb))
280 print(len(norm_ber))
281 # Append normalised column
282 AIS_dataframe_normlb['norm_ber'] = norm_ber
283
284 # Create dataframe for bunkering length
285 # Sort by increasing certainty
286 AIS_dataframe_normblb = ...
    AIS_dataframe_berth.sort_values('BunkeringLengthHrs')
287 # Append normalised column
288 AIS_dataframe_normblb['norm_ber'] = norm_ber
289
290 # Plot
291 plt.plot(AIS_dataframe_normlb.AnchoringLengthHrs, ...
    AIS_dataframe_normlb.norm_ber, label='Berth time')
292 plt.plot(AIS_dataframe_normblb.BunkeringLengthHrs, ...
    AIS_dataframe_normblb.norm_ber, label='Bunkering time', color='r')
293 plt.title('Lay time and bunkering time normalised over number of ...
    operations at berth')
294 plt.xlabel("Time [hours]")
295 plt.ylabel("Percentage")
296 plt.legend()
297 plt.xlim(0,160)
298 plt.show()
299
300 # ----- Less than 60 hours lay time -----
301
302 AIS_dataframe_nn = AIS_dataframe[AIS_dataframe.AnchoringLengthHrs<60]
303
304 # Define number of bins
305 # Bin size
306 res_nn = 4
307
308 # Calculate bin number based on bin size
309 min_nn = min(AIS_dataframe_nn['AnchoringLengthHrs'])
310 max_nn = max(AIS_dataframe_nn['AnchoringLengthHrs'])
311 bsize_nn = int((max_nn-min_nn)/res_nn)
312
313 # Define number of bins

```



```

314 # Bin size
315 res_bnn = 4
316
317 # Calculate bin number based on bin size
318 min_bnn = min(AIS_dataframe_nn['BunkeringLengthHrs'])
319 max_bnn = max(AIS_dataframe_nn['BunkeringLengthHrs'])
320 bsize_bnn = int((max_bnn-min_bnn)/res_bnn)
321
322
323 # Plot histogram
324 plt.figure()
325 ax3 = sns.distplot(AIS_dataframe_nn['AnchoringLengthHrs'],
326                   bins=bsize_nn, hist_kws=dict(edgecolor="b", ...
327                   linewidth=1), label='Lay time < 60 hours')
328 ax4 = sns.distplot(AIS_dataframe_nn['BunkeringLengthHrs'],
329                   bins=bsize_bnn, color='r', ...
330                   hist_kws=dict(edgecolor="r", linewidth=1), ...
331                   label='Bunkering time')
332 ax3.set(ylabel="Density", xlabel="Time [hours]", title = 'All ships')
333 plt.xlim(0,160)
334 plt.legend()
335 plt.show()
336
337
338 # Normalised graphs
339 # Berth
340 # Create new column with normalised values
341 def fill_value_nn():
342     norm_nn = []
343     for i in range(len(AIS_dataframe_nn.AnchoringLengthHrs)):
344         norm_nn.append((i+1)/len(AIS_dataframe_nn.AnchoringLengthHrs)*100)
345     return norm_nn
346
347 norm_nn = fill_value_nn()
348
349 # Sort by increasing increasing lay time
350 AIS_dataframe_nnlay = ...
351     AIS_dataframe_nn.sort_values('AnchoringLengthHrs')
352 # Sort by increasing increasing bunkering time
353 AIS_dataframe_nnbun = ...
354     AIS_dataframe_nn.sort_values('BunkeringLengthHrs')
355
356 # Append normalised column
357 AIS_dataframe_nnlay['norm_nn'] = norm_nn
358 AIS_dataframe_nnbun['norm_nn'] = norm_nn
359

```

```
355 # Scatter plot
356 plt.plot(AIS_dataframe_nnlay.AnchoringLengthHrs, ...
           AIS_dataframe_nnlay.norm_nn, label='Lay time')
357 plt.plot(AIS_dataframe_nnbun.BunkeringLengthHrs, ...
           AIS_dataframe_nnbun.norm_nn, label='Bunkering time', color='r')
358 plt.title('Lay time and bunkering time normalised over number of ...
           operations')
359 plt.xlabel("Time [hours]")
360 plt.ylabel("Percentage")
361 plt.legend()
362 plt.xlim(0,160)
363 plt.show()
```

## E.7 Distribution of Time for Ship Types

### Time\_Distribution\_ShipTypes.py

```
1  """
2  Plot different shiptypes in histograms
3  Created: 11. April 2018
4  Author: Marion Aarsnes
5  """
6
7  # Import packages
8  import pandas as pd                # Read CSV
9  import seaborn as sns              # Library of plots built upon ...
    matplotlib
10 import matplotlib.pyplot as plt    # Plots
11 import matplotlib.ticker as mtick  # Formatting plots
12 import numpy as np                 # Scientific calculation
13
14 # Read CSV file
15 AIS_dataframe = pd.read_csv('filedirectory/filename.csv')
16 print(AIS_dataframe.columns)
17
18 # Clean up dataframe
19 AIS_dataframe = AIS_dataframe[['imo.b', 'shipType.s', 'length.s', ...
    'bunkeringLength_hrs', 'anchoringLength_hrs', 'confidence_mean']]
20 AIS_dataframe = AIS_dataframe.rename(columns={"imo.b": "IMO",
21                                             "shipType.s": "ShipType",
22                                             "length.s": "ShipLength",
23                                             "anchoringLength_hrs": ...
24                                             "AnchoringLengthHrs",
25                                             "bunkeringLength_hrs": ...
26                                             "BunkeringLengthHrs",
27                                             "confidence_mean": ...
28                                             "ConfMean"})
29
30 AIS_dataframe = ...
31     AIS_dataframe.fillna(value="nan")[AIS_dataframe.ShipType != "nan"]
32 # Reduce dataframe to shiptypes with minimum 100 registrations
33 AIS_dataframe = AIS_dataframe[AIS_dataframe.ConfMean >= 0.7]
34
35 # Print top 5 lines of dataframe
36 print(AIS_dataframe.head(5))
37 # List all unique shiptypes
38 print(AIS_dataframe['ShipType'].unique())
```

```
35
36 # Set plot variables
37 sns.set(style="whitegrid", color_codes=True)
38 sns.despine(left=True)
39
40 # ----- Anchorage lasting longer than value a -----
41
42 # Create new dataframe for longer lasting anchorages
43 a = 150
44 b = 35
45
46 # Create column with same value for all operations lasting longer ...
    than value a
47 def fill_values(a):
48     anch_len_x = []
49     bunk_len_x = []
50     for i in range(len(AIS_dataframe.AnchoringLengthHrs)):
51         anch_len_x.append(a)
52         bunk_len_x.append(b)
53     return anch_len_x, bunk_len_x
54
55 anch_len_x, bunk_len_x = fill_values(a)
56
57
58 # Create column index to dataframe
59 AIS_dataframe['AnchorLengthx'] = AIS_dataframe.index
60 AIS_dataframe['BunkLengthx'] = AIS_dataframe.index
61 # Insert new data to dataframe
62 AIS_dataframe.loc[:, 'AnchorLengthx'] = pd.Series(anch_len_x, ...
    index=AIS_dataframe.index)
63 AIS_dataframe.loc[:, 'BunkLengthx'] = pd.Series(bunk_len_x, ...
    index=AIS_dataframe.index)
64
65 # Create new dataframe for operations lasting longer than value a
66 AIS_dataframe_x = AIS_dataframe[AIS_dataframe.AnchoringLengthHrs > a]
67
68 # Plot tankships, oil products tanker, bunkering tanker and cargo ...
    ships in one figure
69
70 # Reduce dataframe to contain only operations lasting shorter than ...
    value a
71 AIS_dataframe = AIS_dataframe[AIS_dataframe.AnchoringLengthHrs < a]
72 # Reduce dataframe to contain specified shiptypes
73 AIS_dataframe = AIS_dataframe.query("ShipType in ('Container Ship', ...
    'Bulk Carrier', 'Crude Oil Tanker')")
74
```

```

75
76 # Specify plotting colors
77 colors = ['b', 'g', 'r', 'k']
78
79
80 # Plot laying time for shiptypes in one figure
81 i = 0
82 for shiptype in AIS_dataframe['ShipType'].unique():
83     ax = sns.distplot(AIS_dataframe[AIS_dataframe.ShipType == ...
84         shiptype]['AnchoringLengthHrs'],
85         hist_kws=dict(edgecolor=colors[i], ...
86             linewidth=2), label=str(shiptype))
87     i +=1
88 axa = sns.distplot(AIS_dataframe_x['AnchorLengthx'],
89     kde = False, color = '.2', ...
90     hist_kws=dict(edgecolor="k", linewidth=3), ...
91     label='Anchorage of more than 180 hours')
92 ax.set(ylabel="Ship density", xlabel="Anchoring time [hours]", ...
93     title="Anchorage time")
94 plt.legend()
95 plt.show()
96
97 # ----- Plot bunkering time for ship types in one figure -----
98
99 # Laying time
100 # Bin size
101 bin_width1 = 5
102
103 # Define number of bins
104 min_opt1 = min(AIS_dataframe[AIS_dataframe.ShipType == 'Container ...
105     Ship']['AnchoringLengthHrs'])
106 max_opt1 = max(AIS_dataframe[AIS_dataframe.ShipType == 'Container ...
107     Ship']['AnchoringLengthHrs'])
108 bsize_opt1 = int((max_opt1-min_opt1)/bin_width1) # Number of ...
109     bins must be an integer
110 min_gcs1 = min(AIS_dataframe[AIS_dataframe.ShipType == 'Bulk ...
111     Carrier']['AnchoringLengthHrs'])
112 max_gcs1 = max(AIS_dataframe[AIS_dataframe.ShipType == 'Bulk ...
113     Carrier']['AnchoringLengthHrs'])
114 bsize_gcs1 = int((max_gcs1-min_gcs1)/bin_width1) # Number of ...
115     bins must be an integer
116 min_ltl = min(AIS_dataframe[AIS_dataframe.ShipType == 'Crude Oil ...
117     Tanker']['AnchoringLengthHrs'])
118 max_ltl = max(AIS_dataframe[AIS_dataframe.ShipType == 'Crude Oil ...
119     Tanker']['AnchoringLengthHrs'])

```

```

108 bsize_ltl = int((max_ltl-min_ltl)/bin_widthl)           # Number of ...
      bins must be an integer
109
110
111 # Plot laying time
112 ax = sns.distplot(AIS_dataframe[AIS_dataframe.ShipType == ...
      'Container Ship']['AnchoringLengthHrs'],
113                   bins=bsize_optl, color='g', ...
                        hist_kws=dict(edgecolor='g', linewidth=2), ...
                        label='Oil Products Tanker')
114 ax1 = sns.distplot(AIS_dataframe[AIS_dataframe.ShipType == 'Bulk ...
      Carrier']['AnchoringLengthHrs'],
115                   bins=bsize_gcsl, color='r', ...
                        hist_kws=dict(edgecolor='r', linewidth=2), ...
                        label='General Cargo Ship')
116 ax2 = sns.distplot(AIS_dataframe[AIS_dataframe.ShipType == 'Crude ...
      Oil Tanker']['AnchoringLengthHrs'],
117                   bins=bsize_ltl, color='b', ...
                        hist_kws=dict(edgecolor='b', linewidth=2), ...
                        label='LNG Tanker')
118 axb = sns.distplot(AIS_dataframe_x['AnchorLengthx'],
119                   kde = False, color = '.2', ...
                        hist_kws=dict(edgecolor="k", linewidth=3), ...
                        label='Anchorage of more than 180 hours')
120 ax.set(ylabel="Ship density", xlabel="Lay time [hours]", title="Lay ...
      time")
121 plt.legend()
122 plt.xlim(0,160)
123 plt.show()
124
125 # Bunkering time
126 # Bin size
127 bin_width = 1
128
129 # Define number of bins
130 min_opt = min(AIS_dataframe[AIS_dataframe.ShipType == 'Container ...
      Ship']['BunkeringLengthHrs'])
131 max_opt = max(AIS_dataframe[AIS_dataframe.ShipType == 'Container ...
      Ship']['BunkeringLengthHrs'])
132 bsize_opt = int((max_opt-min_opt)/bin_width)           # Number of bins ...
      must be an integer
133 min_gcs = min(AIS_dataframe[AIS_dataframe.ShipType == 'Bulk ...
      Carrier']['BunkeringLengthHrs'])
134 max_gcs = max(AIS_dataframe[AIS_dataframe.ShipType == 'Bulk ...
      Carrier']['BunkeringLengthHrs'])

```

```

135 bsize_gcs = int((max_gcs-min_gcs)/bin_width)      # Number of bins ...
        must be an integer
136 min_lt = min(AIS_dataframe[AIS_dataframe.ShipType == 'Crude Oil ...
        Tanker']['BunkeringLengthHrs'])
137 max_lt = max(AIS_dataframe[AIS_dataframe.ShipType == 'Crude Oil ...
        Tanker']['BunkeringLengthHrs'])
138 bsize_lt = int((max_lt-min_lt)/bin_width)        # Number of bins ...
        must be an integer
139
140
141 # Plot bunkering length
142 ax = sns.distplot(AIS_dataframe[AIS_dataframe.ShipType == ...
        'Container Ship']['BunkeringLengthHrs'],
143                 bins=bsize_opt, color='g', ...
                    hist_kws=dict(edgecolor='g', linewidth=2), ...
                    label='Oil Products Tanker')
144 ax1 = sns.distplot(AIS_dataframe[AIS_dataframe.ShipType == 'Bulk ...
        Carrier']['BunkeringLengthHrs'],
145                 bins=bsize_gcs, color='r', ...
                    hist_kws=dict(edgecolor='r', linewidth=2), ...
                    label='General Cargo Ship')
146 ax2 = sns.distplot(AIS_dataframe[AIS_dataframe.ShipType == 'Crude ...
        Oil Tanker']['BunkeringLengthHrs'],
147                 bins=bsize_lt, color='b', ...
                    hist_kws=dict(edgecolor='b', linewidth=2), ...
                    label='LNG Tanker')
148 #axb = sns.distplot(AIS_dataframe_x['BunkLengthx'],
149                 #kde = False, color = '.2', ...
                    hist_kws=dict(edgecolor="k", linewidth=3), ...
                    label='Anchorage of more than 180 hours')
150 ax.set(ylabel="Ship density", xlabel="Bunkering time [hours]", ...
        title="Bunkering time")
151 plt.legend()
152 plt.xlim(0,35)
153 plt.show()
154
155 # Separate plots
156 # Container ship
157 ax = sns.distplot(AIS_dataframe[AIS_dataframe.ShipType == ...
        'Container Ship']['BunkeringLengthHrs'],
158                 bins=bsize_opt, color='b', ...
                    hist_kws=dict(edgecolor='b', linewidth=2), ...
                    label='Container Ship')
159 ax.set(ylabel="Ship density", xlabel="Bunkering time [hours]", ...
        title="Bunkering time Container Ship")
160 plt.xlim(0,30)

```

```

161 plt.legend()
162 plt.show()
163
164 # Bulk carrier
165 ax = sns.distplot(AIS_dataframe[AIS_dataframe.ShipType == 'Bulk ...
    Carrier'] ['BunkeringLengthHrs'],
166                 bins=bsize_gcs, color='b', ...
    hist_kws=dict(edgecolor='b', linewidth=2), ...
    label='Bulk Carrier')
167 ax.set(ylabel="Ship density", xlabel="Bunkering time [hours]", ...
    title="Bunkering time Bulk Carrier")
168 plt.xlim(0,30)
169 plt.legend()
170 plt.show()
171
172 # Crude oil tanker
173 ax = sns.distplot(AIS_dataframe[AIS_dataframe.ShipType == 'Crude ...
    Oil Tanker'] ['BunkeringLengthHrs'],
174                 bins=bsize_lt, color='b', ...
    hist_kws=dict(edgecolor='b', linewidth=2), ...
    label='Crude Oil Tanker')
175 ax.set(ylabel="Ship density", xlabel="Bunkering time [hours]", ...
    title="Bunkering time Crude Oil Tanker")
176 plt.xlim(0,30)
177 plt.legend()
178 plt.show()
179
180
181 # ----- Plot shiplengths per shiptypes -----
182
183 # Bin size
184 bin_widthl = 10
185
186 # Define number of bins
187 min_optl = min(AIS_dataframe[AIS_dataframe.ShipType == 'Container ...
    Ship'] ['ShipLength'])
188 max_optl = max(AIS_dataframe[AIS_dataframe.ShipType == 'Container ...
    Ship'] ['ShipLength'])
189 bsize_optl = int((max_optl-min_optl)/bin_widthl) # Number of ...
    bins must be an integer
190 min_gcsl = min(AIS_dataframe[AIS_dataframe.ShipType == 'Bulk ...
    Carrier'] ['ShipLength'])
191 max_gcsl = max(AIS_dataframe[AIS_dataframe.ShipType == 'Bulk ...
    Carrier'] ['ShipLength'])
192 bsize_gcsl = int((max_gcsl-min_gcsl)/bin_widthl) # Number of ...
    bins must be an integer

```



```

193 min_ltl = min(AIS_dataframe[AIS_dataframe.ShipType == 'Crude Oil ...
    Tanker']['ShipLength'])
194 max_ltl = max(AIS_dataframe[AIS_dataframe.ShipType == 'Crude Oil ...
    Tanker']['ShipLength'])
195 bsize_ltl = int((max_ltl-min_ltl)/bin_widthl)           # Number of ...
    bins must be an integer
196
197
198 # Plot histograms
199
200 # Container ship
201 ax = sns.distplot(AIS_dataframe[AIS_dataframe.ShipType == ...
    'Container Ship']['ShipLength'],
202                 bins=bsize_optl, color='b', ...
    hist_kws=dict(edgecolor='b', linewidth=2), ...
    label='Container Ship')
203 ax.set(ylabel="Ship density", xlabel="Ship length [meters]", ...
    title="Shiplength Container Ship")
204 plt.xlim(100,450)
205 plt.legend()
206 plt.show()
207
208 # Bulk carrier
209 ax = sns.distplot(AIS_dataframe[AIS_dataframe.ShipType == 'Bulk ...
    Carrier']['ShipLength'],
210                 bins=bsize_gcsl, color='b', ...
    hist_kws=dict(edgecolor='b', linewidth=2), ...
    label='Bulk Carrier')
211 ax.set(ylabel="Ship density", xlabel="Ship length [meters]", ...
    title="Shiplength Bulk Carrier")
212 plt.xlim(100,450)
213 plt.legend()
214 plt.show()
215
216 # Crude oil tanker
217 ax = sns.distplot(AIS_dataframe[AIS_dataframe.ShipType == 'Crude ...
    Oil Tanker']['ShipLength'],
218                 bins=bsize_ltl, color='b', ...
    hist_kws=dict(edgecolor='b', linewidth=2), ...
    label='Crude Oil Tanker')
219 ax.set(ylabel="Ship density", xlabel="Ship length [meters]", ...
    title="Shiplength Crude Oil Tanker")
220 plt.xlim(100,450)
221 plt.legend()
222 plt.show()

```

## E.8 Distributions Divided by Ship Lengths

### Dividing\_ShipLengths\_150m.py

```

1  """
2  Script for gathering values exceeding a certain length
3  E.g. All anchorage length of more than 150 hours.
4  Plot lay time and bunkering time.
5  Created: 23.april 2018
6  Author: Marion Aarsnes
7  """
8
9  import pandas as pd                # Read CSV
10 import seaborn as sns              # Library of plots built upon ...
    matplotlib
11 import matplotlib.pyplot as plt    # Plots
12
13 # Read CSV file
14 AIS_dataframe = pd.read_csv('filedirectory/filename.csv')
15 print(AIS_dataframe.columns)
16
17 # Clean up dataframe
18 AIS_dataframe = AIS_dataframe[['imo.b', 'shipType.s', 'length.s', ...
    'bunkeringLength_hrs', 'anchoringLength_hrs', 'bunkeringType']]
19 AIS_dataframe = AIS_dataframe.rename(columns={"imo.b": "IMO",
20                                             "shipType.s": "ShipType",
21                                             "length.s": "ShipLength",
22                                             "anchoringLength_hrs": "...
    "AnchoringLengthHrs",
23                                             "bunkeringLength_hrs": "...
    "BunkeringLengthHrs",
24                                             "bunkeringType": "...
    "BunkeringType"})
25
26 AIS_dataframe = ...
    AIS_dataframe.fillna(value="nan")[AIS_dataframe.ShipType != "nan"]
27
28
29 # ----- Anchorage lasting longer than value a -----
30
31 # Creating new dataframe for longer lasting anchorages
32 a = 80
33 b = 24
34

```

```

35 # Creating one value for all operations lasting longer than value a
36 def fill_values(a):
37     anch_len_x = []
38     bunk_len_x = []
39     for i in range(len(AIS_dataframe.AnchoringLengthHrs)):
40         anch_len_x.append(a)
41         bunk_len_x.append(b)
42     return anch_len_x, bunk_len_x
43
44 anch_len_x, bunk_len_x = fill_values(a)
45
46
47 # Creating column index to dataframe
48 AIS_dataframe['AnchorLengthx'] = AIS_dataframe.index
49 AIS_dataframe['BunkLengthx'] = AIS_dataframe.index
50 # Inserting new data to dataframe
51 AIS_dataframe.loc[:, 'AnchorLengthx'] = pd.Series(anch_len_x, ...
         index=AIS_dataframe.index)
52 AIS_dataframe.loc[:, 'BunkLengthx'] = pd.Series(bunk_len_x, ...
         index=AIS_dataframe.index)
53
54
55 # ----- Length grouping -----
56
57 # Require a minimum confidence mean of 0.7
58 print(AIS_dataframe['ShipType'].unique())
59
60 # Reduce dataframe to shiptypes with minimum 100 registrations
61 AIS_dataframe = AIS_dataframe.query("ShipType in ('Bulk ...
         Carrier', 'Container Ship', 'Crude Oil Tanker', 'Chemical/Oil ...
         Products Tanker', "
62                                     "'Vehicles Carrier', 'LPG ...
         Tanker', 'General Cargo ...
         Ship', 'Oil Products Tanker', "
63                                     "'Ore Carrier', 'LNG Tanker',)")
64
65 # Creating new dataframe for operations lasting longer than value a
66 AIS_dataframe_x = AIS_dataframe[AIS_dataframe.AnchoringLengthHrs ≥ a]
67
68 AIS_dataframe_n = AIS_dataframe[AIS_dataframe.AnchoringLengthHrs < a]
69
70 # Creating new dataframes for bunkering type
71 # Anchorage
72 AIS_dataframe_anchor = ...
         AIS_dataframe_n[AIS_dataframe_n.BunkeringType == 'Anchorage']
73 # Berth

```

```

74 AIS_dataframe_berth = AIS_dataframe_n[AIS_dataframe_n.BunkeringType ...
    == 'Berth']
75
76
77 # ----- Define number of bins -----
78
79 # Bin size
80 res_anch = 2
81 res_bunk = 1
82 # Ship length grouping
83 length = 150
84
85
86 # Anchor
87 min_asm = min(AIS_dataframe_anchor[AIS_dataframe_anchor.ShipLength ≤ ...
    length]['AnchoringLengthHrs'])
88 max_asm = max(AIS_dataframe_anchor[AIS_dataframe_anchor.ShipLength ≤ ...
    length]['AnchoringLengthHrs'])
89 bsize_asm = int((max_asm-min_asm)/res_anch) # bin size must be an ...
    integer
90 min_ala = min(AIS_dataframe_anchor[AIS_dataframe_anchor.ShipLength ...
    > length]['AnchoringLengthHrs'])
91 max_ala = max(AIS_dataframe_anchor[AIS_dataframe_anchor.ShipLength ...
    > length]['AnchoringLengthHrs'])
92 bsize_ala = int((max_ala-min_ala)/res_anch) # bin size must be an ...
    integer
93
94 # Berth
95 min_bsm = min(AIS_dataframe_berth[AIS_dataframe_berth.ShipLength ≤ ...
    length]['AnchoringLengthHrs'])
96 max_bsm = max(AIS_dataframe_berth[AIS_dataframe_berth.ShipLength ≤ ...
    length]['AnchoringLengthHrs'])
97 bsize_bsm = int((max_bsm-min_bsm)/res_anch) # bin size must be an ...
    integer
98 min_bla = min(AIS_dataframe_berth[AIS_dataframe_berth.ShipLength > ...
    length]['AnchoringLengthHrs'])
99 max_bla = max(AIS_dataframe_berth[AIS_dataframe_berth.ShipLength > ...
    length]['AnchoringLengthHrs'])
100 bsize_bla = int((max_bla-min_bla)/res_anch) # bin size must be an ...
    integer
101
102 # Bunkering at anchorage
103 min_asmb = min(AIS_dataframe_anchor[AIS_dataframe_anchor.ShipLength ...
    ≤ length]['BunkeringLengthHrs'])
104 max_asmb = max(AIS_dataframe_anchor[AIS_dataframe_anchor.ShipLength ...
    ≤ length]['BunkeringLengthHrs'])

```

```

105 bsize_asmb = int((max_asmb-min_asmb)/res_bunk) # bin size must be ...
        an integer
106 min_alab = min(AIS_dataframe_anchor[AIS_dataframe_anchor.ShipLength >
        > length]['BunkeringLengthHrs'])
107 max_alab = max(AIS_dataframe_anchor[AIS_dataframe_anchor.ShipLength >
        > length]['BunkeringLengthHrs'])
108 bsize_alab = int((max_alab-min_alab)/res_bunk) # bin size must be ...
        an integer
109
110 # Bunkering at berth
111 min_bsmb = min(AIS_dataframe_berth[AIS_dataframe_berth.ShipLength ≤ ...
        length]['BunkeringLengthHrs'])
112 max_bsmb = max(AIS_dataframe_berth[AIS_dataframe_berth.ShipLength ≤ ...
        length]['BunkeringLengthHrs'])
113 bsize_bsmb = int((max_bsmb-min_bsmb)/res_bunk) # bin size must be ...
        an integer
114 min_blab = min(AIS_dataframe_berth[AIS_dataframe_berth.ShipLength > ...
        length]['BunkeringLengthHrs'])
115 max_blab = max(AIS_dataframe_berth[AIS_dataframe_berth.ShipLength > ...
        length]['BunkeringLengthHrs'])
116 bsize_blab = int((max_blab-min_blab)/res_bunk) # bin size must be ...
        an integer
117
118
119 # ----- Plot laying and bunkering time for operations -----
120
121 #sns.set()
122 sns.set(style="whitegrid", color_codes=True)
123
124 # Anchorage
125 plt.figure()
126 ax = ...
        sns.distplot(AIS_dataframe_anchor[AIS_dataframe_anchor.ShipLength ...
        ≤ 150]['AnchoringLengthHrs'],
127                 bins=bsize_asmb, color='b', ...
                    hist_kws=dict(edgecolor="b", linewidth=1), ...
                    label='Shiplength ≤ ' + str(length) + ' ...
                    meters') #kde=True, color="b",
128 ax2 = ...
        sns.distplot(AIS_dataframe_anchor[AIS_dataframe_anchor.ShipLength ...
        > 150]['AnchoringLengthHrs'],
129                 bins=bsize_ala, color="r", ...
                    hist_kws=dict(edgecolor="r", linewidth=1), ...
                    label='Shiplength > ' + str(length) + ' ...
                    meters') # kde=True, color="r",

```

```
130 axa = sns.distplot(AIS_dataframe_x[AIS_dataframe_x.BunkeringType == ...
    'Anchorage'] ['AnchorLengthx'],
131                 kde = False, color = '.2', ...
                    hist_kws=dict(edgecolor="k", linewidth=3), ...
                    label='Anchorage of more than ' + str(a) + ...
                        ' hours')
132 #ax.axvline(3, color="k", linestyle="--")
133 ax.set(ylabel="Ship density", xlabel="Anchorage time [hours]", ...
        title = 'Anchorage time')
134 plt.xlim(0,a+5)
135 #plt.ylim(0,0.5)
136 plt.legend()
137 plt.show()
138
139 # Berth
140 plt.figure()
141 ax3 = ...
    sns.distplot(AIS_dataframe_berth[AIS_dataframe_berth.ShipLength ...
        ≤ 150] ['AnchoringLengthHrs'],
142                 bins=bsize_bsm, color='b', ...
                    hist_kws=dict(edgecolor="b", linewidth=1), ...
                    label='Shiplength ≤ ' + str(length) + ' ...
                        meters') # kde=False, color="b",
143 ax5 = ...
    sns.distplot(AIS_dataframe_berth[AIS_dataframe_berth.ShipLength ...
        > 150] ['AnchoringLengthHrs'],
144                 bins=bsize_bla, color="r", ...
                    hist_kws=dict(edgecolor="r", linewidth=1), ...
                    label='Shiplength > ' + str(length) + ' ...
                        meters') # kde=False, color="r",
145 axb = sns.distplot(AIS_dataframe_x[AIS_dataframe_x.BunkeringType == ...
    'Berth'] ['AnchorLengthx'],
146                 kde = False, color = '.2', ...
                    hist_kws=dict(edgecolor="k", linewidth=3), ...
                    label='Anchorage of more than ' + str(a) + ...
                        ' hours')
147 #ax3.axvline(3, color="k", linestyle="--")
148 ax3.set(ylabel="Ship density", xlabel="Berth time [hours]", title ...
        ='Berth time')
149 plt.xlim(0,a+5)
150 plt.legend()
151 plt.show()
152
153
154 # Bunkering time at anchorage
155 plt.figure()
```

```

156 ax0 = ...
    sns.distplot(AIS_dataframe_anchor[AIS_dataframe_anchor.ShipLength ...
    ≤ 150]['BunkeringLengthHrs'],
157             bins=bsize_asmb, color='b', ...
                hist_kws=dict(edgecolor="b", linewidth=1), ...
                label='Shiplength ≤ ' + str(length) + ' ...
                meters') # kde=False, color="b",
158 ax2 = ...
    sns.distplot(AIS_dataframe_anchor[AIS_dataframe_anchor.ShipLength ...
    > 150]['BunkeringLengthHrs'],
159             bins=bsize_alab, color="r", ...
                hist_kws=dict(edgecolor="r", linewidth=1), ...
                label='Shiplength > ' + str(length) + ' ...
                meters') # kde=False, color="r",
160 #axc = sns.distplot(AIS_dataframe_x[AIS_dataframe_x.BunkeringType ...
    == 'Anchorage']['BunkLengthx'],
161                 #kde = False, color = '.2', ...
                hist_kws=dict(edgecolor="k", linewidth=3), ...
                label='Anchorage of more than 180 hours')
162 #ax0.axvline(3, color="k", linestyle="--")
163 ax0.set(ylabel="Ship density", xlabel="Bunkering time [hours]", ...
        title = 'Bunkering time at anchorage')
164 plt.xlim(0,b+5)
165 plt.legend()
166 plt.show()
167
168
169 # Bunkering time at berth
170 plt.figure()
171 ax3 = ...
    sns.distplot(AIS_dataframe_berth[AIS_dataframe_berth.ShipLength ...
    ≤ 150]['BunkeringLengthHrs'],
172             bins=bsize_bsmb, color='b', ...
                hist_kws=dict(edgecolor="b", linewidth=1), ...
                label='Shiplength ≤ ' + str(length) + ' ...
                meters') # kde=False, color="b",
173 ax5 = ...
    sns.distplot(AIS_dataframe_berth[AIS_dataframe_berth.ShipLength ...
    > 150]['BunkeringLengthHrs'],
174             bins=bsize_blab, color="r", ...
                hist_kws=dict(edgecolor="r", linewidth=1), ...
                label='Shiplength > ' + str(length) + ' ...
                meters') # kde=False, color="r",
175 #axd = sns.distplot(AIS_dataframe_x[AIS_dataframe_x.BunkeringType ...
    == 'Berth']['BunkLengthx'],

```

```

176             #kde = False, color = '.2', ...
                hist_kws=dict(edgecolor="k", linewidth=3), ...
                label='Anchorage of more than 180 hours')
177 #ax3.axvline(3, color="k", linestyle="--")
178 ax5.set(ylabel="Ship density", xlabel="Bunkering time [hours]", ...
          title = 'Bunkering time at berth')
179 plt.xlim(0,b+5)
180 plt.legend()
181 plt.show()
182
183
184 # ----- Normalised plots -----
185
186 # Creating new dataframe for operations lasting longer than value a
187 AIS_dataframe_x = AIS_dataframe[AIS_dataframe.AnchoringLengthHrs > a]
188
189 AIS_dataframe_n = AIS_dataframe[AIS_dataframe.AnchoringLengthHrs < a]
190
191 # Creating new dataframes for bunkering type
192 # Anchorage
193 AIS_dataframe_anchor = ...
        AIS_dataframe_n[AIS_dataframe_n.BunkeringType == 'Anchorage']
194 # Berth
195 AIS_dataframe_berth = AIS_dataframe_n[AIS_dataframe_n.BunkeringType ...
        == 'Berth']
196
197 # ----- All operations -----
198
199 # Create new dataframes
200 AIS_dataframe_above150 = AIS_dataframe_n[AIS_dataframe_n.ShipLength ...
        > 150]
201 AIS_dataframe_below150 = AIS_dataframe_n[AIS_dataframe_n.ShipLength ...
        ≤ 150]
202
203 # Sort dataframes
204 AIS_dataframe_above150l = ...
        AIS_dataframe_above150.sort_values('AnchoringLengthHrs')
205 AIS_dataframe_below150l = ...
        AIS_dataframe_below150.sort_values('AnchoringLengthHrs')
206 AIS_dataframe_above150b = ...
        AIS_dataframe_above150.sort_values('BunkeringLengthHrs')
207 AIS_dataframe_below150b = ...
        AIS_dataframe_below150.sort_values('BunkeringLengthHrs')
208
209 # Create list numerating operations
210 # Create new column with normalised values

```



```

211 def fill_value_allabove150l():
212     norm_allabove150l = []
213     for i in range(len(AIS_dataframe_above150l.ShipLength)):
214         norm_allabove150l.append((i+1)/ ...
                len(AIS_dataframe_above150l.ShipLength)*100)
215     return norm_allabove150l
216
217 norm_allabove150l = fill_value_allabove150l()
218
219 def fill_value_allbelow150l():
220     norm_allbelow150l = []
221     for i in range(len(AIS_dataframe_below150l.ShipLength)):
222         norm_allbelow150l.append((i+1)/ ...
                len(AIS_dataframe_below150l.ShipLength)*100)
223     return norm_allbelow150l
224
225 norm_allbelow150l = fill_value_allbelow150l()
226
227 def fill_value_allabove150b():
228     norm_allabove150b = []
229     for i in range(len(AIS_dataframe_above150b.ShipLength)):
230         norm_allabove150b.append((i+1)/ ...
                len(AIS_dataframe_above150b.ShipLength)*100)
231     return norm_allabove150b
232
233 norm_allabove150b = fill_value_allabove150b()
234
235 def fill_value_allbelow150b():
236     norm_allbelow150b = []
237     for i in range(len(AIS_dataframe_below150b.ShipLength)):
238         norm_allbelow150b.append((i+1)/ ...
                len(AIS_dataframe_below150b.ShipLength)*100)
239     return norm_allbelow150b
240
241 norm_allbelow150b = fill_value_allbelow150b()
242
243 # Append normalised column
244 AIS_dataframe_above150l['norm'] = norm_allabove150l
245 AIS_dataframe_below150l['norm'] = norm_allbelow150l
246 AIS_dataframe_above150b['norm'] = norm_allabove150b
247 AIS_dataframe_below150b['norm'] = norm_allbelow150b
248
249
250 # Plot laying time
251 plt.plot(AIS_dataframe_above150l.AnchoringLengthHrs, ...
          AIS_dataframe_above150l.norm, color='r', label='Shiplength > ' ...

```

```

    + str(length) + ' meters')
252 plt.plot(AIS_dataframe_below150l.AnchoringLengthHrs, ...
    AIS_dataframe_below150l.norm, color='b', label='Shiplength ≤ ' ...
    + str(length) + ' meters')
253 plt.title('Laying time normalised over number of operations')
254 plt.xlabel("Laying time [hours]")
255 plt.ylabel("Percentage")
256 plt.xlim(0,a+5)
257 plt.legend()
258 plt.show()
259
260 # Plot bunkering time
261 plt.plot(AIS_dataframe_above150b.BunkeringLengthHrs, ...
    AIS_dataframe_above150b.norm, color='r', label='Shiplength > ' ...
    + str(length) + ' meters')
262 plt.plot(AIS_dataframe_below150b.BunkeringLengthHrs, ...
    AIS_dataframe_below150b.norm, color='b', label='Shiplength ≤ ' ...
    + str(length) + ' meters')
263 plt.title('Bunkering time normalised over number of operations')
264 plt.xlabel("Bunkering time [hours]")
265 plt.ylabel("Percentage")
266 plt.xlim(0,b+5)
267 plt.legend()
268 plt.show()
269
270
271
272 # ----- At anchorage -----
273
274 AIS_dataframe_anchor_above150 = ...
    AIS_dataframe_anchor[AIS_dataframe_anchor.ShipLength > 150]
275 AIS_dataframe_anchor_below150 = ...
    AIS_dataframe_anchor[AIS_dataframe_anchor.ShipLength ≤ 150]
276
277 # Sort dataframes
278 AIS_dataframe_anchor_above150l = ...
    AIS_dataframe_anchor_above150.sort_values('AnchoringLengthHrs')
279 AIS_dataframe_anchor_below150l = ...
    AIS_dataframe_anchor_below150.sort_values('AnchoringLengthHrs')
280 AIS_dataframe_anchor_above150b = ...
    AIS_dataframe_anchor_above150.sort_values('BunkeringLengthHrs')
281 AIS_dataframe_anchor_below150b = ...
    AIS_dataframe_anchor_below150.sort_values('BunkeringLengthHrs')
282
283 # Create list numerating operations
284 # Create new column with normalised values

```

```

285 def fill_value_anchor_allabove150l():
286     norm_anchor_allabove150l = []
287     for i in range(len(AIS_dataframe_anchor_above150l.ShipLength)):
288         norm_anchor_allabove150l.append((i+1)/ ...
                len(AIS_dataframe_anchor_above150l.ShipLength)*100)
289     return norm_anchor_allabove150l
290
291 norm_anchor_allabove150l = fill_value_anchor_allabove150l()
292
293 def fill_value_anchor_allbelow150l():
294     norm_anchor_allbelow150l = []
295     for i in range(len(AIS_dataframe_anchor_below150l.ShipLength)):
296         norm_anchor_allbelow150l.append((i+1)/ ...
                len(AIS_dataframe_anchor_below150l.ShipLength)*100)
297     return norm_anchor_allbelow150l
298
299 norm_anchor_allbelow150l = fill_value_anchor_allbelow150l()
300
301 def fill_value_anchor_allabove150b():
302     norm_anchor_allabove150b = []
303     for i in range(len(AIS_dataframe_anchor_above150b.ShipLength)):
304         norm_anchor_allabove150b.append((i+1)/ ...
                len(AIS_dataframe_anchor_above150b.ShipLength)*100)
305     return norm_anchor_allabove150b
306
307 norm_anchor_allabove150b = fill_value_anchor_allabove150b()
308
309 def fill_value_anchor_allbelow150b():
310     norm_anchor_allbelow150b = []
311     for i in range(len(AIS_dataframe_anchor_below150b.ShipLength)):
312         norm_anchor_allbelow150b.append((i+1)/ ...
                len(AIS_dataframe_anchor_below150b.ShipLength)*100)
313     return norm_anchor_allbelow150b
314
315 norm_anchor_allbelow150b = fill_value_anchor_allbelow150b()
316
317 # Append normalised column
318 AIS_dataframe_anchor_above150l['norm'] = norm_anchor_allabove150l
319 AIS_dataframe_anchor_below150l['norm'] = norm_anchor_allbelow150l
320 AIS_dataframe_anchor_above150b['norm'] = norm_anchor_allabove150b
321 AIS_dataframe_anchor_below150b['norm'] = norm_anchor_allbelow150b
322
323
324 # Plot anchorage time
325 plt.plot(AIS_dataframe_anchor_above150l.AnchoringLengthHrs, ...
          AIS_dataframe_anchor_above150l.norm, color='r', ...

```

```

        label='Shiplength > ' + str(length) + ' meters')
326 plt.plot(AIS_dataframe_anchor_below150l.AnchoringLengthHrs, ...
            AIS_dataframe_anchor_below150l.norm, color='b', ...
            label='Shiplength ≤ ' + str(length) + ' meters')
327 plt.title('Anchorage time normalised over number of operations')
328 plt.xlabel("Laying time [hours]")
329 plt.ylabel("Percentage")
330 plt.xlim(0,a+5)
331 plt.legend()
332 plt.show()
333
334 # Plot bunkering time
335 plt.plot(AIS_dataframe_anchor_above150b.BunkeringLengthHrs, ...
            AIS_dataframe_anchor_above150b.norm, color='r', ...
            label='Shiplength > ' + str(length) + ' meters')
336 plt.plot(AIS_dataframe_anchor_below150b.BunkeringLengthHrs, ...
            AIS_dataframe_anchor_below150b.norm, color='b', ...
            label='Shiplength ≤ ' + str(length) + ' meters')
337 plt.title('Bunkering time normalised over number of operations at ...
            anchorage')
338 plt.xlabel("Bunkering time [hours]")
339 plt.ylabel("Percentage")
340 plt.xlim(0,b+5)
341 plt.legend()
342 plt.show()
343
344
345 # ----- At berth -----
346
347 AIS_dataframe_berth_above150 = ...
            AIS_dataframe_berth[AIS_dataframe_berth.ShipLength > 150]
348 AIS_dataframe_berth_below150 = ...
            AIS_dataframe_berth[AIS_dataframe_berth.ShipLength ≤ 150]
349
350 # Sort dataframes
351 AIS_dataframe_berth_above150l = ...
            AIS_dataframe_berth_above150.sort_values('AnchoringLengthHrs')
352 AIS_dataframe_berth_below150l = ...
            AIS_dataframe_berth_below150.sort_values('AnchoringLengthHrs')
353 AIS_dataframe_berth_above150b = ...
            AIS_dataframe_berth_above150.sort_values('BunkeringLengthHrs')
354 AIS_dataframe_berth_below150b = ...
            AIS_dataframe_berth_below150.sort_values('BunkeringLengthHrs')
355
356 # Create list numerating operations
357 # Create new column with normalised values

```

```

358 def fill_value_berth_allabove150l():
359     norm_berth_allabove150l = []
360     for i in range(len(AIS_dataframe_berth_above150l.ShipLength)):
361         norm_berth_allabove150l.append((i+1)/ ...
362             len(AIS_dataframe_berth_above150l.ShipLength)*100)
363     return norm_berth_allabove150l
364 norm_berth_allabove150l = fill_value_berth_allabove150l()
365
366 def fill_value_berth_allbelow150l():
367     norm_berth_allbelow150l = []
368     for i in range(len(AIS_dataframe_berth_below150l.ShipLength)):
369         norm_berth_allbelow150l.append((i+1)/ ...
370             len(AIS_dataframe_berth_below150l.ShipLength)*100)
371     return norm_berth_allbelow150l
372 norm_berth_allbelow150l = fill_value_berth_allbelow150l()
373
374 def fill_value_berth_allabove150b():
375     norm_berth_allabove150b = []
376     for i in range(len(AIS_dataframe_berth_above150b.ShipLength)):
377         norm_berth_allabove150b.append((i+1)/ ...
378             len(AIS_dataframe_berth_above150b.ShipLength)*100)
379     return norm_berth_allabove150b
380 norm_berth_allabove150b = fill_value_berth_allabove150b()
381
382 def fill_value_berth_allbelow150b():
383     norm_berth_allbelow150b = []
384     for i in range(len(AIS_dataframe_berth_below150b.ShipLength)):
385         norm_berth_allbelow150b.append((i+1)/ ...
386             len(AIS_dataframe_berth_below150b.ShipLength)*100)
387     return norm_berth_allbelow150b
388 norm_berth_allbelow150b = fill_value_berth_allbelow150b()
389
390 # Append normalised column
391 AIS_dataframe_berth_above150l['norm'] = norm_berth_allabove150l
392 AIS_dataframe_berth_below150l['norm'] = norm_berth_allbelow150l
393 AIS_dataframe_berth_above150b['norm'] = norm_berth_allabove150b
394 AIS_dataframe_berth_below150b['norm'] = norm_berth_allbelow150b
395
396
397 # Plot berth time
398 plt.plot(AIS_dataframe_berth_above150l.AnchoringLengthHrs, ...
399         AIS_dataframe_berth_above150l.norm, color='r', ...)

```

```
    label='Shiplength > ' + str(length) + ' meters')
399 plt.plot(AIS_dataframe_berth_below150l.AnchoringLengthHrs, ...
    AIS_dataframe_berth_below150l.norm, color='b', ...
    label='Shiplength ≤ ' + str(length) + ' meters')
400 plt.title('Berth time normalised over number of operation')
401 plt.xlabel("Laying time [hours]")
402 plt.ylabel("Percentage")
403 plt.xlim(0,a+5)
404 plt.legend()
405 plt.show()
406
407 # Plot bunkering time
408 plt.plot(AIS_dataframe_berth_above150b.BunkeringLengthHrs, ...
    AIS_dataframe_berth_above150b.norm, color='r', ...
    label='Shiplength > ' + str(length) + ' meters')
409 plt.plot(AIS_dataframe_berth_below150b.BunkeringLengthHrs, ...
    AIS_dataframe_berth_below150b.norm, color='b', ...
    label='Shiplength ≤ ' + str(length) + ' meters')
410 plt.title('Bunkering time normalised over number of operations at ...
    berth')
411 plt.xlabel("Bunkering time [hours]")
412 plt.ylabel("Percentage")
413 plt.xlim(0,b+5)
414 plt.legend()
415 plt.show()
```

## E.9 Timeline Plot of Three Operations

### Timelineplot\_Bunker\_Operations.py

```

1  """
2  Plot anchoring and bunkering for several barges in one plot
3  Created: 12. April 2018
4  Author: Marion Aarsnes
5  """
6
7  # Import packages
8  import pandas as pd                # Read CSV
9  import numpy as np                 # Scientific computation
10 import matplotlib.pyplot as plt    # Plotting
11 import matplotlib.ticker as mtick  # Formatting plots
12 import seaborn as sns              # Library of plots built upon ...
    matplotlib
13
14 # Read CSV file
15 AIS_dataframe = pd.read_csv('filedirectory/filename.csv')
16 print(AIS_dataframe.columns)
17
18 # Clean up dataframe
19 AIS_dataframe = AIS_dataframe[['imo.b', 'shipType.b', 'length.b', ...
    'shipType.s', 'name.s', 'length.s', 'bunkeringLength_hrs', ...
    'anchoringLength_hrs', 'BunkUnixStart',
20     'BunkUnixEnd', 'AnchorUnixStart', 'AnchorUnixEnd']]
21 AIS_dataframe = AIS_dataframe.rename(columns={"imo.b": "IMO",
22     "shipType.b": ...
23     "BargeType",
24     "length.b": ...
25     "BargeLength",
26     "shipType.s": "ShipType",
27     "length.s": "ShipLength",
28     "name.s": "ShipName",
29     "anchoringLength_hrs": ..
30     "AnchoringLengthHrs",
31     "bunkeringLength_hrs": ..
32     "BunkeringLengthHrs"})
33
34 AIS_dataframe = ...
35     AIS_dataframe.fillna(value="nan") [AIS_dataframe.ShipType != "nan"]
36
37 # Set plot variables

```

```

33 sns.set(style="whitegrid", color_codes=True)
34 sns.despine(left=True)
35
36 # Define operation numbers to plot
37 op_list = [3, 4, 10]
38
39 # Create empty matrices for x and y values
40 x_list = np.zeros(shape=(len(op_list),4))
41 y_list = np.zeros(shape=(len(op_list),2))
42
43 # Calculate matrix values - start and end of anchorage and ...
    bunkering for each operation
44 for j in range(len(op_list)):
45     i = op_list[j]
46     x_list[j,0] = (AIS_dataframe.AnchorUnixStart[i] - ...
        AIS_dataframe.AnchorUnixStart[i])/3600
47     x_list[j,1] = (AIS_dataframe.AnchorUnixEnd[i] - ...
        AIS_dataframe.AnchorUnixStart[i])/3600
48     x_list[j,2] = (AIS_dataframe.BunkUnixStart[i] - ...
        AIS_dataframe.AnchorUnixStart[i])/3600
49     x_list[j,3] = (AIS_dataframe.BunkUnixEnd[i] - ...
        AIS_dataframe.AnchorUnixStart[i])/3600
50     y_list[j,0] = j
51     y_list[j,1] = j+0.1
52
53 # Define colors and line styles
54 colors = ['r', 'b', 'y', 'g', 'm', 'k']
55 linestyle = ['-', '--', ':', '-.']
56
57 # Plot
58 for k in range(len(op_list)):
59     type_s = AIS_dataframe.ShipType[op_list[k]]
60     length_s = AIS_dataframe.ShipLength[op_list[k]]
61     type_b = AIS_dataframe.BargeType[op_list[k]]
62     length_b = AIS_dataframe.BargeLength[op_list[k]]
63     plt.plot([x_list[k,0], x_list[k,1]], [y_list[k,0], ...
        y_list[k,0]], 'l', color='b')
64     plt.plot([x_list[k,2], x_list[k,3]], [y_list[k,1], ...
        y_list[k,1]], 'l', color='r')
65     plt.plot([x_list[k,0], x_list[k,1]], [y_list[k,0], ...
        y_list[k,0]], linestyle[k], color='b', label='Anchorage ' + ...
        str(type_s) + ' ' + str(length_s) + 'm')
66     plt.plot([x_list[k,2], x_list[k,3]], [y_list[k,1], ...
        y_list[k,1]], linestyle[k], color='r', label='Bunkering ' + ...
        str(type_b) + ' ' + str(length_b) + 'm')

```



```
67     plt.plot([x_list[k,0], x_list[k,0]], [y_list[k,0]+0.1, ...
            y_list[k,0]-0.1], '-', color='b')
68     plt.plot([x_list[k,1], x_list[k,1]], [y_list[k,0]+0.1, ...
            y_list[k,0]-0.1], '-', color='b')
69     plt.plot([x_list[k, 2], x_list[k, 2]], [y_list[k, 1] + 0.1, ...
            y_list[k, 0]], '-', color='r')
70     plt.plot([x_list[k, 3], x_list[k, 3]], [y_list[k, 1] + 0.1, ...
            y_list[k, 0]], '-', color='r')
71     plt.text(9.9, 2.2, 'Operation 3')
72     plt.text(14, 1.2, 'Operation 2')
73     plt.text(3.5, 0.2, 'Operation 1')
74
75     plt.title('Time plot anchorage and bunkering')
76     plt.xlabel("Time [hours]")
77     plt.legend()
78     plt.show()
```

## E.10 Normalised Plots of Various Parameters

### Normalised\_Plots.py

```
1  """
2  Create normalised plots
3  Created: 02. May 2018
4  Author: Marion Aarsnes
5  """
6
7  import pandas as pd          # Read CSV
8  import seaborn as sns       # Library of plots built upon ...
9                                matplotlib
10 import matplotlib.pyplot as plt # Plots
11                                # Formatting plots
12
13 # Read CSV file
14 AIS_dataframe = pd.read_csv('filedirectory/filename.csv')
15 print(AIS_dataframe.columns)
16
17 # Clean up dataframe
18 AIS_dataframe = AIS_dataframe[['imo.b', 'length.b', 'shipType.s', ...
19                                'length.s', 'bunkeringLength_hrs', 'anchoringLength_hrs', ...
20                                'confidence_mean', "bunkeringType"]] #, 'confidence_mean']
21 AIS_dataframe = AIS_dataframe.rename(columns={"imo.b": "IMO",
22                                             "shipType.s": "ShipType",
23                                             "length.b": ...
24                                             "BargeLength",
25                                             "length.s": "ShipLength",
26                                             "anchoringLength_hrs": ...
27                                             "AnchoringLengthHrs",
28                                             "bunkeringLength_hrs": ...
29                                             "BunkeringLengthHrs",
30                                             "bunkeringType": ...
31                                             "BunkeringType",
32                                             "confidence_mean": ...
33                                             "ConfMean"})
34
35 AIS_dataframe = ...
36     AIS_dataframe.fillna(value="nan") [AIS_dataframe.ShipType != "nan"]
37 # Removing rows with Confidence Mean < 0.7
38 AIS_dataframe = AIS_dataframe[AIS_dataframe.ConfMean >= 0.7]
39
40
41
```

```

32 # Reducing dataframe to shiptypes with minimum 100 registrations
33 AIS_dataframe = AIS_dataframe.query("ShipType in ('Bulk ...
    Carrier', 'Container Ship', 'Crude Oil Tanker', 'Chemical/Oil ...
    Products Tanker', "
34                                     "'Vehicles Carrier', 'LPG ...
    Tanker', 'General Cargo ...
    Ship', 'Oil Products Tanker', "
35                                     "'Ore Carrier', 'LNG Tanker',)")
36
37 # Set plotting style
38 sns.set(style="whitegrid", color_codes=True)
39
40 # Create new column with normalised values
41 def fill_value_conf():
42     norm = []
43     for i in range(len(AIS_dataframe.ConfMean)):
44         norm.append((i+1)/len(AIS_dataframe.ConfMean)*100)
45     return norm
46
47 norm = fill_value_conf()
48
49 # ----- Normalised certainty -----
50
51 AIS_dataframe_conf = AIS_dataframe
52
53 # Sort by increasing certainty
54 AIS_dataframe_conf = AIS_dataframe_conf.sort_values('ConfMean')
55
56 # Append normalised column
57 AIS_dataframe_conf['norm'] = norm
58
59 # Plot
60 plt.plot(AIS_dataframe_conf.ConfMean, AIS_dataframe_conf.norm)
61 plt.title('Confidence Mean normalised over number of operations')
62 plt.xlabel("Confidence Mean")
63 plt.ylabel("Percentage")
64 plt.show()
65
66 # ----- Normalised barge lengths -----
67
68 AIS_dataframe_blen = AIS_dataframe
69
70 # Sort by increasing certainty
71 AIS_dataframe_blen = AIS_dataframe_blen.sort_values('BargeLength')
72
73 # Append new column to dataframe

```

```

74 AIS_dataframe_blen['norm'] = norm
75 print(max(AIS_dataframe_blen.BargeLength))
76
77 # Plot
78 plt.plot(AIS_dataframe_blen.BargeLength, AIS_dataframe_blen.norm)
79 plt.title('Barge Length normalised over number of operations')
80 plt.xlabel("Barge Length [meters]")
81 plt.ylabel("Percentage of operations")
82 plt.xlim(0,130)
83 plt.show()
84
85 # ----- Normalised ship lengths -----
86
87 AIS_dataframe_slen = AIS_dataframe
88
89 # Sort by increasing certainty
90 AIS_dataframe_slen = AIS_dataframe_slen.sort_values('ShipLength')
91
92 # Append new column to dataframe
93 AIS_dataframe_slen['norm'] = norm
94 print(max(AIS_dataframe_slen.ShipLength))
95
96 # Plot
97 plt.plot(AIS_dataframe_slen.ShipLength, AIS_dataframe_slen.norm)
98 plt.title('Ship Length normalised over number of operations')
99 plt.xlabel("Ship Length [meters]")
100 plt.ylabel("Percentage")
101 plt.show()
102
103 # ----- Normalised lay time -----
104
105 # Create new dataframes for bunkering type
106 # Anchorage
107 AIS_dataframe_anchor = AIS_dataframe[AIS_dataframe.BunkeringType == ...
    'Anchorage']
108 # Berth
109 AIS_dataframe_berth = AIS_dataframe[AIS_dataframe.BunkeringType == ...
    'Berth']
110
111 # ----- Normalised time at anchorage -----
112
113 AIS_dataframe_anchor_len = AIS_dataframe_anchor
114
115 # Sort by increasing certainty
116 AIS_dataframe_anchor_len = ...
    AIS_dataframe_anchor_len.sort_values('AnchoringLengthHrs')

```

```

117
118 # Create new column with normalised values
119 def fill_value_anch():
120     norm_anch = []
121     for i in range(len(AIS_dataframe_anchor)):
122         norm_anch.append((i+1)/ len(AIS_dataframe_anchor.ConfMean)*100)
123     return norm_anch
124
125 norm_anch = fill_value_anch()
126
127 # Append new column to dataframe
128 AIS_dataframe_anchor_len['norm'] = norm_anch
129
130 # Plot
131 plt.plot(AIS_dataframe_anchor_len.AnchoringLengthHrs, ...
          AIS_dataframe_anchor_len.norm)
132 plt.title('Anchoring time normalised over number of operations')
133 plt.xlabel("Anchoring time [hours]")
134 plt.ylabel("Percentage")
135 plt.show()
136
137 # ----- Normalised time at berth -----
138
139 AIS_dataframe_berth_len = AIS_dataframe_berth
140
141 # Sort by increasing certainty
142 AIS_dataframe_berth_len = ...
          AIS_dataframe_berth_len.sort_values('AnchoringLengthHrs')
143
144 # Create new column with normalised values
145 def fill_value_berth():
146     norm_berth = []
147     for i in range(len(AIS_dataframe_berth)):
148         norm_berth.append((i+1)/ len(AIS_dataframe_berth.ConfMean)*100)
149     return norm_berth
150
151 norm_berth = fill_value_berth()
152
153 # Append new column to dataframe
154 AIS_dataframe_berth_len['norm'] = norm_berth
155
156 # Plot
157 plt.plot(AIS_dataframe_berth_len.AnchoringLengthHrs, ...
          AIS_dataframe_berth_len.norm)
158 plt.title('Berth time normalised over number of operations')
159 plt.xlabel("Anchoring time [hours]")

```

```
160 plt.ylabel("Percentage")
161 plt.show()
162
163 # ----- Normalised bunkering time -----
164
165 # All operations
166 # Create new dataframe and sort by increasing certainty
167 AIS_dataframe_bunk_all= AIS_dataframe.sort_values('BunkeringLengthHrs')
168 # Add new column to dataframe
169 AIS_dataframe_bunk_all['norm'] = norm
170
171 # Plot
172 plt.plot(AIS_dataframe_bunk_all.BunkeringLengthHrs, ...
           AIS_dataframe_bunk_all.norm)
173 plt.title('Bunkering time normalised over number of operations')
174 plt.xlabel("Bunkering Time [hours]")
175 plt.ylabel("Percentage")
176 plt.xlim(0,35)
177 plt.show()
178
179
180 # Anchorage
181 # Create new dataframe
182 AIS_dataframe_bunk_anchor = ...
           AIS_dataframe[AIS_dataframe.BunkeringType == 'Anchorage']
183 # Sort by increasing certainty
184 AIS_dataframe_bunk_anchor = ...
           AIS_dataframe_bunk_anchor.sort_values('BunkeringLengthHrs')
185 # Add new column to dataframe
186 AIS_dataframe_bunk_anchor['norm'] = norm_anch
187
188 # Plot
189 plt.plot(AIS_dataframe_bunk_anchor.BunkeringLengthHrs, ...
           AIS_dataframe_bunk_anchor.norm)
190 plt.title('Bunkering time at anchorage normalised over number of ...
           operations')
191 plt.xlabel("Bunkering Time [hours]")
192 plt.ylabel("Percentage")
193 plt.xlim(0,35)
194 plt.show()
195
196 # Berth
197 # Create new dataframe
198 AIS_dataframe_bunk_berth = ...
           AIS_dataframe[AIS_dataframe.BunkeringType == 'Berth']
199 # Sort by increasing certainty
```

```

200 AIS_dataframe_bunk_berth = ...
    AIS_dataframe_bunk_berth.sort_values('BunkeringLengthHrs')
201 # Add new column to dataframe
202 AIS_dataframe_bunk_berth['norm'] = norm_berth
203
204 # Plot
205 plt.plot(AIS_dataframe_bunk_berth.BunkeringLengthHrs, ...
    AIS_dataframe_bunk_berth.norm)
206 plt.title('Bunkering time at berth normalised over number of ...
    operations')
207 plt.xlabel("Bunkering Time [hours]")
208 plt.ylabel("Percentage")
209 plt.xlim(0,35)
210 plt.show()
211
212 # ----- Bunkering time less than 24 hours -----
213
214 # Create new dataframe for operations with bunkering of less than ...
    24 hours
215 AIS_dataframe_24 = AIS_dataframe[AIS_dataframe.BunkeringLengthHrs ≤ 24]
216
217 # Sort dataframe by increasing bunkering time
218 AIS_dataframe_bunk_all_24= ...
    AIS_dataframe_24.sort_values('BunkeringLengthHrs')
219
220 # Create new column with normalised values
221 def fill_value_all_24():
222     norm_24 = []
223     for i in range(len(AIS_dataframe_bunk_all_24)):
224         norm_24.append((i+1)/len(AIS_dataframe_bunk_all_24.ConfMean)*100)
225     return norm_24
226
227 norm_24 = fill_value_all_24()
228
229 # Append new column to dataframe
230 AIS_dataframe_bunk_all_24['norm'] = norm_24
231
232 # Plot normalised plot for all operations
233 plt.plot(AIS_dataframe_bunk_all_24.BunkeringLengthHrs, ...
    AIS_dataframe_bunk_all_24.norm)
234 plt.title('Bunkering time normalised over number of operations')
235 plt.xlabel("Bunkering Time [hours]")
236 plt.ylabel("Percentage")
237 plt.xlim(0,27)
238 plt.show()
239

```

```
240
241 # At anchorage
242 # Create new dataframe and sort by increasing bunkering time
243 AIS_dataframe_bunk_all_24a = ...
    AIS_dataframe_24[AIS_dataframe_24.BunkeringType == 'Anchorage']
244 AIS_dataframe_bunk_all_24a = ...
    AIS_dataframe_bunk_all_24a.sort_values('BunkeringLengthHrs')
245
246 # Create new column with normalised values
247 def fill_value_all_24a():
248     norm_24a = []
249     for i in range(len(AIS_dataframe_bunk_all_24a)):
250         norm_24a.append((i+1)/len(AIS_dataframe_bunk_all_24a.ConfMean)*100)
251     return norm_24a
252
253 norm_24a = fill_value_all_24a()
254
255 # Append new column to dataframe
256 AIS_dataframe_bunk_all_24a['norm'] = norm_24a
257
258 # Plot normalised plot for bunkering at anchorage
259 plt.plot(AIS_dataframe_bunk_all_24a.BunkeringLengthHrs, ...
    AIS_dataframe_bunk_all_24a.norm)
260 plt.title('Bunkering time normalised over number of operations')
261 plt.xlabel("Bunkering Time at anchorage [hours]")
262 plt.ylabel("Percentage")
263 plt.xlim(0,27)
264 plt.show()
265
266 # At berth
267 # Create new dataframe and sort by increasing bunkering time
268 AIS_dataframe_bunk_all_24b = ...
    AIS_dataframe_24[AIS_dataframe_24.BunkeringType == 'Berth']
269 AIS_dataframe_bunk_all_24b = ...
    AIS_dataframe_bunk_all_24b.sort_values('BunkeringLengthHrs')
270
271 # Create new column with normalised values
272 def fill_value_all_24b():
273     norm_24b = []
274     for i in range(len(AIS_dataframe_bunk_all_24b)):
275         norm_24b.append((i+1)/len(AIS_dataframe_bunk_all_24b.ConfMean)*100)
276     return norm_24b
277
278 norm_24b = fill_value_all_24b()
279
280 # Append new column to dataframe
```



```
281 AIS_dataframe_bunk_all_24b['norm'] = norm_24b
282
283 # Plot normalised plot of bunkerings at berth
284 plt.plot(AIS_dataframe_bunk_all_24b.BunkeringLengthHrs, ...
          AIS_dataframe_bunk_all_24b.norm)
285 plt.title('Bunkering time normalised over number of operations')
286 plt.xlabel("Bunkering Time at berth [hours]")
287 plt.ylabel("Percentage")
288 plt.xlim(0,27)
289 plt.show()
```

## E.11 Histogram and Scatter Plot of Waiting Time and Post Bunkering Time

### Histogram\_WT\_PBT.py

```

1  """
2  Plot histograms of Waiting Time and Post Bunkering Time
3  Created: 26. April 2018
4  Author: Marion Aarsnes
5  """
6
7  import pandas as pd                # Read CSV
8  import seaborn as sns             # Library of plots built upon ...
   matplotlib
9  import matplotlib.pyplot as plt   # Plots
10 import matplotlib.ticker as mtick # Formatting plots
11
12 # Read CSV file
13 AIS_dataframe = pd.read_csv('filedirectory/filename.csv')
14 print(AIS_dataframe.columns)
15
16 # Clean up dataframe
17 AIS_dataframe = AIS_dataframe[['imo.b', 'shipType.s', 'length.s', ...
   'bunkeringLength_hrs', 'anchoringLength_hrs', 'BunkUnixStart',
18   'BunkUnixEnd', 'AnchorUnixStart', 'AnchorUnixEnd', ...
   'confidence_mean', 'bunkeringType', ...
   'timeBeforeBunkering_hrs', 'timeAfterBunkering_hrs']]
19 AIS_dataframe = AIS_dataframe.rename(columns={"imo.b": "IMO",
20   "shipType.s": "ShipType",
21   "length.s": "ShipLength",
22   "anchoringLength_hrs": "...
   "AnchoringLengthHrs",
23   "bunkeringLength_hrs": "...
   "BunkeringLengthHrs",
24   "confidence_mean": ...
   "ConfMean",
25   "bunkeringType": ...
   "BunkeringType",
26   "timeBeforeBunkering_hrs": ...
   "WaitingTime",
27   "timeAfterBunkering_hrs": ...
   "PBTime"})
28

```

```

29 AIS_dataframe = ...
    AIS_dataframe.fillna(value="nan")[AIS_dataframe.ShipType != "nan"]
30 # Removing rows with Confidence Mean < 0.7
31 AIS_dataframe = AIS_dataframe[AIS_dataframe.ConfMean ≥ 0.7]
32
33 # Reduce dataframe to shiptypes with minimum 100 registrations
34 AIS_dataframe = AIS_dataframe.query("ShipType in ('Bulk ...
    Carrier','Container Ship','Crude Oil Tanker','Chemical/Oil ...
    Products Tanker',"
35
    "'Vehicles Carrier','LPG ...
    Tanker','General Cargo ...
    Ship','Oil Products Tanker',"
36
    "'Ore Carrier','LNG Tanker',)")
37
38 # Create new dataframes for bunkering type
39 # Anchorage
40 AIS_dataframe_anchor = AIS_dataframe[AIS_dataframe.BunkeringType == ...
    'Anchorage']
41 # Berth
42 AIS_dataframe_berth = AIS_dataframe[AIS_dataframe.BunkeringType == ...
    'Berth']
43
44 # Set plot stylr
45 sns.set(style="whitegrid", color_codes=True)
46
47 # ----- Plot WT vs PBT -----
48
49 # Scatter plot showing waiting time vs anchorage time after bunkering
50 plt.scatter(AIS_dataframe.WaitingTime, AIS_dataframe.PBTime)
51 plt.xlim(-10,200)
52 plt.ylim(-10,200)
53 #plt.title('Waiting time vs time after bunkering')
54 plt.ylabel("Post Bunkering Time [hours]")
55 plt.xlabel("Waiting Time [hours]")
56 plt.show()
57
58 grid2 = sns.jointplot(AIS_dataframe.WaitingTime, ...
    AIS_dataframe.PBTime, kind="reg", ...
    joint_kws={'line_kws':{'color':'y'}})
59 grid2.set_axis_labels("Waiting Time [hours]","Post Bunkering Time ...
    [hours]")
60 plt.xlim(-10,200)
61 plt.ylim(-10,200)
62 plt.show()
63
64 # ----- Define bin size and number of bins -----

```

```

65
66 # All operations
67 b_len = 2.5 # Bin size
68 min_wall = min(AIS_dataframe['WaitingTime'])
69 max_wall = max(AIS_dataframe['WaitingTime'])
70 bsize_wall = int((max_wall-min_wall)/b_len) # Number of bins must ...
        be an integer
71
72 # At anchorage
73 b_len = 2.5 # Bin size
74 min_wan = min(AIS_dataframe_anchor['WaitingTime'])
75 max_wan = max(AIS_dataframe_anchor['WaitingTime'])
76 bsize_wan = int((max_wan-min_wan)/b_len) # Number of bins must be ...
        an integer
77
78 # At berth
79 b_len = 2.5 # Bin size
80 min_wber = min(AIS_dataframe_berth['WaitingTime'])
81 max_wber = max(AIS_dataframe_berth['WaitingTime'])
82 bsize_wber = int((max_wber-min_wber)/b_len) # Number of bins must ...
        be an integer
83
84
85 # ----- Waiting Time -----
86
87 # Plot Waiting Time for all operations
88 plt.figure()
89 ax = sns.distplot(AIS_dataframe['WaitingTime'],
90                  bins=bsize_wall,kde=False, ...
91                  hist_kws=dict(edgecolor="b", linewidth=1)) ...
92                  # kde=False, color="b", 112
93 ax.set(ylabel="Number of operations", xlabel="Waiting time ...
94        [hours]", title ='All')
95
96 # Plot Waiting Time at anchorage
97 plt.figure()
98 ax = sns.distplot(AIS_dataframe_anchor['WaitingTime'],
99                  bins=bsize_wan,kde=False, ...
100                  hist_kws=dict(edgecolor="b", linewidth=1)) ...
101                  # kde=False, color="b", 112
102 ax.set(ylabel="Number of operations", xlabel="Waiting time ...
103        [hours]", title ='At anchorage')
104 plt.xlim(0,25)

```

```

102 plt.legend()
103 plt.show()
104
105 # Plot Waiting Time at berth
106 plt.figure()
107 ax = sns.distplot(AIS_dataframe_berth['WaitingTime'],
108                  bins=bsize_wber,kde=False, ...
109                  hist_kws=dict(edgecolor="b", linewidth=1)) ...
110                  # kde=False, color="b",
111 ax.set(ylabel="Number of operations", xlabel="Waiting time ...
112        [hours]", title ='At berth')
113
114
115 # ----- Post Bunkering Time -----
116
117 # Define bin size and number of bins
118
119 # All operations
120 b_lenp = 2.5 # Bin size
121 min_pall = min(AIS_dataframe['PBTime'])
122 max_pall = max(AIS_dataframe['PBTime'])
123 bsize_pall = int((max_pall-min_pall)/b_lenp) # Number of bins must ...
124         be an integer
125
126 # At anchorage
127 b_lenp = 2.5 # Bin size
128 min_pan = min(AIS_dataframe_anchor['PBTime'])
129 max_pan = max(AIS_dataframe_anchor['PBTime'])
130 bsize_pan = int((max_pan-min_pan)/b_lenp) # Number of bins must be ...
131         an integer
132
133 # At berth
134 b_lenp = 2.5 # Bin size
135 min_pber = min(AIS_dataframe_berth['PBTime'])
136 max_pber = max(AIS_dataframe_berth['PBTime'])
137 bsize_pber = int((max_pber-min_pber)/b_lenp) # Number of bins must ...
138         be an integer
139
140 # Plot Post Bunkering Time for all operations
141 plt.figure()
142 ax = sns.distplot(AIS_dataframe['PBTime'],
143                  bins=bsize_pall,kde=False, ...
144                  hist_kws=dict(edgecolor="b", linewidth=1)) ...

```

```

# kde=False, color="b", 112
141 ax.set(ylabel="Number of operations", xlabel="Post Bunkering Time ...
    [hours]", title ='All')
142 plt.xlim(0,25)
143 plt.legend()
144 plt.show()
145
146 # Plot Post Bunkering Time at anchorage
147 plt.figure()
148 ax = sns.distplot(AIS_dataframe_anchor['PBTime'],
149                 bins=bsize_pan,kde=False, ...
                    hist_kws=dict(edgecolor="b", linewidth=1)) ...
                    # kde=False, color="b", 112
150 ax.set(ylabel="Number of operations", xlabel="Post Bunkering Time ...
    [hours]", title ='At anchorage')
151 plt.xlim(0,25)
152 plt.legend()
153 plt.show()
154
155 # Plot Post Bunkering Time at berth
156 plt.figure()
157 ax = sns.distplot(AIS_dataframe_berth['PBTime'],
158                 bins=bsize_pber,kde=False, ...
                    hist_kws=dict(edgecolor="b", linewidth=1)) ...
                    # kde=False, color="b",
159 ax.set(ylabel="Number of operations", xlabel="Post Bunkering Time ...
    [hours]", title ='At berth')
160 plt.xlim(0,25)
161 plt.legend()
162 plt.show()
```

## E.12 Normalise Waiting Time and Post Bunkering Time

### Normalise\_WT\_PBT.py.py

```

1  """
2  Plot Waiting Time and Post Bunkering Time normalised over number of ...
   bunker operations
3  Created: 05. May 2018
4  Author: Marion Aarsnes
5  """
6
7  # Import packages
8  import pandas as pd           # Read CSV
9  import numpy as np           # Scientific calculation
10 import seaborn as sns        # Library of plots built upon ...
   matplotlib
11 import matplotlib.pyplot as plt # Plots
12 import matplotlib.ticker as mtick # Format plots
13
14 # Read CSV file
15 AIS_dataframe = pd.read_csv('filedirectory/filename.csv')
16 print(AIS_dataframe.columns)
17
18 # Clean up dataframe
19 AIS_dataframe = AIS_dataframe[['imo.b', 'shipType.s', 'length.s', ...
   'bunkeringLength_hrs', 'anchoringLength_hrs', 'BunkUnixStart',
20   'BunkUnixEnd', 'AnchorUnixStart', 'AnchorUnixEnd', ...
   'confidence_mean', 'bunkeringType', ...
   'timeBeforeBunkering_hrs', 'timeAfterBunkering_hrs']]
21 AIS_dataframe = AIS_dataframe.rename(columns={"imo.b": "IMO",
22   "shipType.s": "ShipType",
23   "length.s": "ShipLength",
24   "anchoringLength_hrs": "...
   "AnchoringLengthHrs",
25   "bunkeringLength_hrs": "...
   "BunkeringLengthHrs",
26   "confidence_mean": ...
   "ConfMean",
27   "bunkeringType": ...
   "BunkeringType",
28   "timeBeforeBunkering_hrs": ...
   "WaitingTime",
29   "timeAfterBunkering_hrs": ...
   "PBTime"})

```

```

30
31 AIS_dataframe = ...
    AIS_dataframe.fillna(value="nan")[AIS_dataframe.ShipType != "nan"]
32 # Removing rows with Confidence Mean < 0.7
33 AIS_dataframe = AIS_dataframe[AIS_dataframe.ConfMean ≥ 0.7]
34
35 # Reduce dataframe to shiptypes with minimum 100 registrations
36 AIS_dataframe = AIS_dataframe.query("ShipType in ('Bulk ...
    Carrier', 'Container Ship', 'Crude Oil Tanker', 'Chemical/Oil ...
    Products Tanker', "
37                                     "'Vehicles Carrier', 'LPG ...
    Tanker', 'General Cargo ...
    Ship', 'Oil Products Tanker', "
38                                     "'Ore Carrier', 'LNG Tanker',)")
39
40 # Set plot style
41 sns.set(style="whitegrid", color_codes=True)
42
43 # Create new dataframes for bunkering type
44 # Anchorage
45 AIS_dataframe_anchor = AIS_dataframe[AIS_dataframe.BunkeringType == ...
    'Anchorage']
46 # Berth
47 AIS_dataframe_berth = AIS_dataframe[AIS_dataframe.BunkeringType == ...
    'Berth']
48
49 # Create new column with normalised values
50 def fill_value():
51     norm = []
52     for i in range(len(AIS_dataframe.ConfMean)):
53         norm.append((i+1)/len(AIS_dataframe.ConfMean)*100)
54     return norm
55
56 norm = fill_value()
57
58 # ----- Normalised Waiting Time -----
59
60 AIS_dataframe_wt = AIS_dataframe
61
62 # Sorting by increasing certainty
63 AIS_dataframe_wt = AIS_dataframe_wt.sort_values('WaitingTime')
64
65 # Appending normalised column
66 AIS_dataframe_wt['norm'] = norm
67
68 # Plot

```



```
69 plt.plot(AIS_dataframe_wt.WaitingTime, AIS_dataframe_wt.norm)
70 plt.title('Waiting Time normalised over number of operations')
71 plt.xlabel("Waiting Time [hours]")
72 plt.ylabel("Percentage")
73 #plt.legend()
74 plt.xlim(0,700)
75 plt.show()
76
77
78 # ----- Normalised Post Bunkering Time -----
79
80 AIS_dataframe_pbt = AIS_dataframe
81
82 # Sort by increasing certainty
83 AIS_dataframe_pbt = AIS_dataframe_pbt.sort_values('PBTime')
84
85 # Append normalised column
86 AIS_dataframe_pbt['norm'] = norm
87
88 # Plot
89 plt.plot(AIS_dataframe_pbt.PBTime, AIS_dataframe_pbt.norm)
90 plt.title('Post Bunkering Time normalised over number of operations')
91 plt.xlabel("Post Bunkering Time [hours]")
92 plt.ylabel("Percentage")
93 #plt.legend()
94 plt.xlim(0,700)
95 plt.show()
96
97
98 # ----- Normalised WT and PBT -----
99
100 plt.plot(AIS_dataframe_pbt.PBTime, AIS_dataframe_pbt.norm, ...
          label="WT", color = "r")
101 plt.plot(AIS_dataframe_wt.WaitingTime, AIS_dataframe_wt.norm, ...
          label="PBT", color = "b")
102 plt.title('WT and PBT normalised over number of operations')
103 plt.xlabel("Time [hours]")
104 plt.ylabel("Percentage")
105 plt.xlim(0,200)
106 plt.legend()
107 plt.show()
108
109 # Mark 75% quantile
110 plt.plot(AIS_dataframe_pbt.PBTime, AIS_dataframe_pbt.norm, ...
          label="PBT", color = "b")
```

```
111 plt.plot(AIS_dataframe_wt.WaitingTime, AIS_dataframe_wt.norm, ...
           label="WT", color = "r")
112 plt.axhline(y=75, color="grey")
113 plt.axhline(y=25, color="grey")
114 plt.plot(5.57,75, 'o', color='grey')
115 plt.plot(10.58,75, 'o', color='grey')
116 plt.plot([5.55,5.55],[0,75], '-', color='grey')
117 plt.plot([10.58,10.58],[0,75], '-', color='grey')
118 plt.text(3.5, 77, 'x = 5.55')
119 plt.text(11, 70, 'x = 10.58')
120 plt.text(32, 76, 'y = 75', color='grey')
121 plt.plot(1,25, 'o', color='grey')
122 plt.plot(2.1,25, 'o', color='grey')
123 plt.plot([1,1],[0,25], '-', color='grey')
124 plt.plot([2.1,2.1],[0,25], '-', color='grey')
125 plt.text(0.1, 26, 'x = 1')
126 plt.text(2.5, 26, 'x = 2.1')
127 plt.text(32, 26, 'y = 25', color='grey')
128 plt.title('WT and PBT normalised over number of operations')
129 plt.xlabel("Time [hours]")
130 plt.ylabel("Percentage")
131 plt.xlim(0,35)
132 plt.ylim(0,105)
133 plt.legend()
134 plt.show()
```

## E.13 Create Index Plot

### Plot\_Quantiles.py

```
1  """
2  Plot quantiles and correlation between quantiles (index plot)
3  Created: 24. May 2018
4  Author: Marion Aarsnes
5  """
6
7  # Import packages
8  import pandas as pd          # Read CSV
9  import numpy as np          # Scientific computation
10 import matplotlib.pyplot as plt # Plotting
11 import matplotlib.ticker as mtick # Formatting plots
12 import seaborn as sns        # Library of plots built upon ...
    matplotlib
13
14 # Set plot style
15 sns.set(style="whitegrid", color_codes=True)
16
17 # Define quantiles
18 WTquant_0 = 2                # Maximum accepted WT
19 WTquant_75 = 5.55
20 PBTquant_0 = 2.1            # Minimum accepted PBT
21 PBTquant_75 = 10.58
22
23 # Plot quantiles
24 plt.plot(WTquant_0,1,'o',color='r', label='lower bound WT')
25 plt.plot(WTquant_75,0,'o',color='coral', label='upper bound WT')
26 plt.plot(PBTquant_0,0,'o',color='b', label='lower bound PBT')
27 plt.plot(PBTquant_75,1,'o',color='lightblue', label='upper bound PBT')
28 plt.title('Quantile plot')
29 plt.xlabel('Time [hours]')
30 plt.ylabel('Score')
31 plt.legend()
32 plt.show()
33
34 # Plot linearised relationship between quantiles
35 plt.plot(WTquant_0,1,'o',color='r')
36 plt.plot(WTquant_75,0,'o',color='r')
37 plt.plot(PBTquant_0,0,'o',color='b')
38 plt.plot(PBTquant_75,1,'o',color='b')
39 plt.plot([0,WTquant_0],[1,1], '-', color='r')
```

```
40 plt.plot([WTquant_0,WTquant_75],[1,0], '-', color='r', label='WT')
41 plt.plot([WTquant_75,15],[0,0], '-', color='r')
42 plt.plot([0,PBTquant_0],[0,0], '-', color='b')
43 plt.plot([PBTquant_0,PBTquant_75],[0,1], '-', color='b', label='PBT')
44 plt.plot([PBTquant_75,15],[1,1], '-', color='b')
45 plt.title('Index plot')
46 plt.xlabel('Time [hours]')
47 plt.ylabel('TW/TPBT')
48 plt.xlim(0,12)
49 plt.legend()
50 plt.show()
```

## E.14 Calculate Index

### Calculate\_Index.py

```

1  """
2  Calculate index for bunkering operations and plot results
3  Created: 03. June 2018
4  Author: Marion Aarsnes
5  """
6
7  # Import packages
8  import pandas as pd                # Read CSV
9  import numpy as np                # Scientific computation
10 import seaborn as sns              # Library of plots built upon ...
    matplotlib
11 import matplotlib.pyplot as plt    # Plots
12
13 # Read CSV file
14 AIS_dataframe = pd.read_csv('filedirectory/filename.csv')
15 print(AIS_dataframe.columns)
16
17 # Clean up dataframe
18 AIS_dataframe = AIS_dataframe[['Unnamed: 0', 'imo.b', 'shipType.s', ...
    'length.s', 'bunkeringLength_hrs', 'anchoringLength_hrs', ...
    'BunkUnixStart',
19     'BunkUnixEnd', 'AnchorUnixStart', 'AnchorUnixEnd', ...
    'confidence_mean', 'bunkeringType', ...
    'timeBeforeBunkering_hrs', 'timeAfterBunkering_hrs']]
20 AIS_dataframe = AIS_dataframe.rename(columns={"Unnamed: 0": ...
    "OperationNr",
21
22                                     "imo.b": "IMO",
23                                     "shipType.s": "ShipType",
24                                     "length.s": "ShipLength",
25                                     "anchoringLength_hrs": ...
26                                     "AnchoringLengthHrs",
27                                     "bunkeringLength_hrs": ...
28                                     "BunkeringLengthHrs",
29                                     "confidence_mean": ...
30                                     "ConfMean",
31                                     "bunkeringType": ...
32                                     "BunkeringType",
33                                     "timeBeforeBunkering_hrs": ...
34                                     "WTime",
35                                     "timeAfterBunkering_hrs": ...
36                                     "WTime",
37                                     "BunkUnixStart": "BunkUnixStart",
38                                     "BunkUnixEnd": "BunkUnixEnd",
39                                     "AnchorUnixStart": "AnchorUnixStart",
40                                     "AnchorUnixEnd": "AnchorUnixEnd"}).drop("Unnamed: 0", axis=1)
41
42 # Plot results
43 # Create figure
44 fig = plt.figure(figsize=(10, 10))
45 # Create subplots
46 # Create a grid of subplots
47 # Create a grid of subplots
48 # Create a grid of subplots
49 # Create a grid of subplots
50 # Create a grid of subplots
51 # Create a grid of subplots
52 # Create a grid of subplots
53 # Create a grid of subplots
54 # Create a grid of subplots
55 # Create a grid of subplots
56 # Create a grid of subplots
57 # Create a grid of subplots
58 # Create a grid of subplots
59 # Create a grid of subplots
60 # Create a grid of subplots
61 # Create a grid of subplots
62 # Create a grid of subplots
63 # Create a grid of subplots
64 # Create a grid of subplots
65 # Create a grid of subplots
66 # Create a grid of subplots
67 # Create a grid of subplots
68 # Create a grid of subplots
69 # Create a grid of subplots
70 # Create a grid of subplots
71 # Create a grid of subplots
72 # Create a grid of subplots
73 # Create a grid of subplots
74 # Create a grid of subplots
75 # Create a grid of subplots
76 # Create a grid of subplots
77 # Create a grid of subplots
78 # Create a grid of subplots
79 # Create a grid of subplots
80 # Create a grid of subplots
81 # Create a grid of subplots
82 # Create a grid of subplots
83 # Create a grid of subplots
84 # Create a grid of subplots
85 # Create a grid of subplots
86 # Create a grid of subplots
87 # Create a grid of subplots
88 # Create a grid of subplots
89 # Create a grid of subplots
90 # Create a grid of subplots
91 # Create a grid of subplots
92 # Create a grid of subplots
93 # Create a grid of subplots
94 # Create a grid of subplots
95 # Create a grid of subplots
96 # Create a grid of subplots
97 # Create a grid of subplots
98 # Create a grid of subplots
99 # Create a grid of subplots
100 # Create a grid of subplots

```

```

29                                     "timeAfterBunkering_hrs": ...
                                     "PBTime"})
30
31 AIS_dataframe = ...
    AIS_dataframe.fillna(value="nan")[AIS_dataframe.ShipType != "nan"]
32
33 # Set plot style
34 sns.set(style="whitegrid", color_codes=True)
35
36 print(AIS_dataframe.head(4))
37 print(AIS_dataframe.ShipLength[4])
38 print(AIS_dataframe.ShipLength[0])
39
40 # Number of operations to be calculated for
41 num = len(AIS_dataframe.ShipLength) # Replace 20 with ...
    len(AIS_dataframe.ShipLength) to calculate index for all operations
42
43 # Create empty matrix to store index data
44 index_list = np.zeros(shape=(num,6))
45 index_list_berth = []
46 index_list_anchor = []
47
48 # Define quantiles
49 WTquant_0 = 2 # Maximum accepted WT
50 WTquant_75 = 5.55
51 PBTquant_0 = 2 # Minimum accepted PBT
52 PBTquant_75 = 10.58
53
54 # Calculate index
55 # Array = [Operation nr, Waiting Time, Post Bunkering Time, Score ...
    WT, score PBT, index]
56 def calc_index():
57     for i in range(0,num):
58         index_list[i,0] = int(i+1) #int(AIS_dataframe.OperationNr[i])
59         index_list[i,1] = AIS_dataframe.WTime[i]
60         index_list[i,2] = AIS_dataframe.PBTime[i]
61         if AIS_dataframe.WTime[i] <= WTquant_0:
62             TWT = 1
63         elif AIS_dataframe.WTime[i] > WTquant_0 and ...
            AIS_dataframe.WTime[i] < WTquant_75:
64             TWT = (111/71) - ((20*AIS_dataframe.WTime[i])/71)
65         else:
66             TWT = 0
67         if AIS_dataframe.PBTime[i] <= PBTquant_0:
68             TPBT = 0

```

```

69     elif AIS_dataframe.PBTime[i] > PBTquant_0 and ...
70         AIS_dataframe.PBTime[i] < PBTquant_75:
71         TPBT = (-105/424) + ((50*AIS_dataframe.PBTime[i])/424)
72     else:
73         TPBT = 1
74     index_list[i,3] = TWT
75     index_list[i,4] = TPBT
76     index_list[i,5] = TWT + ((1-TWT)*TPBT)
77     if str(AIS_dataframe.BunkeringType[i]) == "Anchorage" and ...
78         AIS_dataframe.ConfMean[i] ≥ 0.7:
79         index_list_anchor.append(TWT + ((1-TWT)*TPBT))
80     elif str(AIS_dataframe.BunkeringType[i]) == "Berth" and ...
81         AIS_dataframe.ConfMean[i] ≥ 0.7:
82         index_list_berth.append(TWT + ((1 - TWT) * TPBT))
83     return index_list, index_list_anchor, index_list_berth
84
85 index_list, index_list_anchor, index_list_berth = calc_index()
86
87 print(index_list)
88
89 # Print array of calculate indexes to .txt-file
90 np.savetxt('maximums.txt', index_list)
91
92 # Plot histogram with Kernel Density Line
93 ax = sns.distplot(index_list_berth,
94                 bins=50, color='r', hist_kws=dict(edgecolor="r", ...
95                 linewidth=1), label='At berth')
96 ax = sns.distplot(index_list_anchor,
97                 bins=50, color='b', hist_kws=dict(edgecolor="b", ...
98                 linewidth=1), label='At anchorage')
99 ax.set(ylabel="Density", xlabel="QBT")
100 plt.xlim(0,1.05)
101 plt.legend()
102 plt.show()
103
104 # Normalised plot
105
106 # Berth
107 # Sort list by increasing QBT
108 index_list_berth.sort()
109
110 # Create normalising column
111 def fill_value_b():
112     norm_b = []
113     for i in range(len(index_list_berth)):

```

```
110         norm_b.append((i+1)/len(index_list_berth)*100)
111     return norm_b
112
113 norm_b = fill_value_b()
114
115 # Anchorage
116 # Sort list by increasing QBT
117 index_list_anchor.sort()
118
119 # Create normalising column
120 def fill_value_a():
121     norm_a = []
122     for i in range(len(index_list_anchor)):
123         norm_a.append((i+1)/len(index_list_anchor)*100)
124     return norm_a
125
126 norm_a = fill_value_a()
127
128 # Plot normalised QBT for anchorage and berth operations
129 plt.plot(index_list_anchor, norm_a, color='b', label='At anchorage')
130 plt.plot(index_list_berth, norm_b, color='r', label='At berth')
131 plt.title('QBT normalised over number of operations')
132 plt.xlabel("QBT")
133 plt.ylabel("Percentage")
134 plt.legend()
135 plt.xlim(0,1.05)
136 plt.show()
```



## E.15 List of Electronic Appendages

- Confidence\_Mean\_Plotting.py
- Distribution\_Shiptypes.py
- Ships\_Length\_Distribution.py
- Ships\_Time\_Distribution.py
- BunkerBarges\_Length\_and\_Time\_Distribution.py
- Time\_Distribution\_AllOperations.py
- Time\_Distribution\_ShipTypes.py
- Dividing\_ShipLengths\_150m.py
- Timelineplot\_Bunker\_Operations.py
- Histogram\_WT\_PBT.py
- Normalise\_WT\_PBT.py
- Normalised\_Plots.py
- Plot\_Quantiles.py
- Calculate\_Index.py