Andreas Matre

Relative Variable Importance Approaches for Linear Models with Random Intercepts

Master's thesis in Mathematical Sciences (MSMNFMA)
Supervisor: Stefanie Muff
June 2022

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

Andreas Matre

Relative Variable Importance Approaches for Linear Models with Random Intercepts

Master’s thesis in Mathematical Sciences (MSMNFMA)
Supervisor: Stefanie Muff
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

Abstract

Interpreting linear regression models is a common desire in research. A common question researchers are interested in is how important a predictor is to the model. Knowing the individual importance of each predictor can give the researchers a better understanding of their data, from which their results are derived, and therefore lead to better research. The probably most commonly used tool to determine the importance of a predictor to a model is the p-value. The p-value is involved when testing the null-hypothesis that the coefficient of a predictor is zero, against the alternative hypothesis, that the coefficient is not zero. The p-value of that hypothesis test, however, is not suited to determining the importance of a predictor and does not give any information about the impact the predictor has on the model. To get better information on the importance of a predictor to a regression model, other, supplementary, tools are used. We will consider tools based on on the coefficient of determination (R^2) because of it's ease of interpretation.

This thesis proposes extensions of two popular methods based on R^2 for linear regression models, the LMG and relative weights methods, such that they work on linear random intercept models. Such models are commonly used in fields like biology, epidemiology and the social sciences. The LMG method considers the mean increase in R^2 when the predictors are added to the model in different orderings, which is computationally expensive. The relative weights method takes advantage of the fact that the squared coefficients are meaningful when the predictors are uncorrelated to be more computationally efficient than the LMG method. To use the fact that the squared coefficients are meaningful for uncorrelated predictors, the relative weights method transforms the data to get uncorrelated predictors, gives each of these an importance using the squared coefficients, and then transforms the importances back to the original form of the data. The transformation of the data requires that all the predictors are numerical, so the relative weights method does not work with categorical predictors.

The extended LMG method works by considering the random intercepts the same as fixed effects and looks at the mean increase in R^2 when they are added to the model. The extended relative weights method works by combining the LMG method and the relative weights method, where the numerical fixed effects are transformed as usual in relative weights, and then are always either all in the model or none are in the model. The increase in R^2 when the transformed numerical fixed effects are added to the model can then be distributed to each original fixed effect.

The two proposed extensions are applied in a simulation study while the extended relative weights method is also applied on an example with real data. The simulation study shows that the extended relative weights method is a useful approximation of the extended LMG method while the application on real data shows the extended relative weights methods usefulness by comparing the calculated importances to other measures of importance, such as the p-value and squared coefficients. Finally, the R-package `decompR2` is developed, which implements the proposed methods such that they are easy to use. Having the proposed methods in an easy to use package will hopefully make it more likely that they will be used and thus lead to researchers having a more robust understandings of their data and results.

Sammendrag

Å tolke lineære regresjonsmodeller er et vanlig ønske innen forskning. Et vanlig spørsmål forskere er interessert i er hvor viktig en prediktor er for modellen. Å vite den individuelle betydningen av hver prediktor kan bidra å øke forskernes forståelse av dataen som resultatene deres er utledet fra og derfor føre til bedre forskning. Det sannsynligvis mest brukte verktøyet for å bestemme betydningen av en prediktor for en modell er p-verdien. Mer spesifikt så er p-verdien som er involvert den som er tilknyttet nullhypotesen om at koeffisienten til en prediktor er null, mot den alternative hypotesen om at koeffisienten ikke er null. Denne p-verdien er imidlertid ikke egnet til å bestemme viktigheten av en prediktor og gir ingen informasjon om hvilken innvirkning prediktoren har på modellen. For å få bedre informasjon om betydningen av en prediktor for en regresjonsmodell, brukes andre, supplerende, verktøy. Vi vil bruke verktøy basert på bestemmelseskoeffisienten (R^2) på grunn av dens enkle tolkning.

Denne oppgaven foreslår utvidelser av to populære metoder basert på R^2 for lineære regresjonsmodeller, LMG og relative weights metodene, slik at de fungerer på lineære stokastiske skjæringspunktmodeller. Slike modeller er ofte brukt innen felt som biologi, epidemiologi og samfunnsvitenskap. LMG-metoden vurderer gjennomsnittsøkningen i R^2 når prediktorene legges til modellen i forskjellige rekkefølger, noe som er beregningsmessig kostbart. Relative weights-metoden bruker det faktum at de kvadrerte koeffisientene gir nyttig informasjon når prediktorene er ukorrelerte, som gjør metoden mer beregningsmessig effektiv enn LMG-metoden. For å bruke det faktum at de kvadrerte koeffisientene gir nyttig informasjon for ukorrelerte prediktorer, transformerer relative weights-metoden dataene for å få ukorrelerte prediktorer, gir hver av disse en viktighet ved å bruke de kvadrerte koeffisientene, og transformerer deretter viktighetene tilbake til den opprinnelige formen av dataene.

Den utvidede LMG-metoden fungerer ved å behandle de stokastiske skjæringspunktene på samme måte som fikserte effekter og ser på gjennomsnittsøkningen i R^2 når de legges til modellen. Den utvidede relative weights-metoden fungerer ved å kombinere LMG-metoden og relative weights-metoden, hvor de kontinuerlige fikserte effektene transformeres som vanlig i relative weights-metoden, og deretter er enten alle i modellen eller ingen i modellen. Økningen i R^2 når de transformerte kontinuerlige fikserte effektene legges til modellen kan deretter distribueres til hver originale fikserte effekt.

De to foreslåtte utvidelsene brukes i en simuleringsstudie, mens den utvidene relative weights-metoden i tillegg brukes på et eksempel med ordentlige data. Simuleringsstudien viser at den utvidede relative weights-metoden er en nyttig tilnærming til den utvidede LMG-metoden, mens applikasjonen på ordentlige data viser nyttigheten av utvidede relative vekter-metoder ved å sammenligne de beregnede betydningene med andre viktighetsmål, som for eksempel p-verdien og de kvadrerte koeffisientene. Til slutt utvikles R-pakken `decompR2`, som implementerer de foreslåtte metodene slik at de er enkle å bruke. Å ha de foreslåtte metodene i et brukervennlig format vil forhåpentligvis gjøre det mer sannsynlig at de vil bli brukt og dermed føre til at forskere får en mer robust forståelse av deres data og resultater.

Preface

This thesis is the final part of a Master of Science in the program Mathematical Sciences at the Norwegian University of Science and Technology (NTNU). The thesis is a 45 credit individual project worked on in fall 2021 and spring 2022.

I want to give a large thanks to my supervisor Stefanie Muff for excellent guidance.

Contents

List of Figures

List of Tables

1 Introduction

When interpreting statistical models it is often interesting to know how important each predictor is to the model, or how much information each predictor explains in the response. This is useful for model interpretation, *i.e.*, trying to understand the relationships in the data.

Probably the most common way to determine how important a predictor is to a model is to use p-values, which tests the null-hypothesis that a predictor's coefficient is zero. Interpreting p-values in such a way is based on a fundamental misunderstanding of what p-values are, but is nevertheless prevalent. The p-value is the probability of observing the given observations, or observations more extreme, assuming that the null-hypothesis is true. The concept of p-values and null-hypotheses testing dates back to the 18^{th} century but was popularized by Ronald Fisher (Arbuthnot, 1710; Fisher, 1925). When using p-values, an arbitrary significance level is usually set, often 0.05, and if the p-value of a hypothesis test is smaller than the significance level the null-hypothesis is rejected under that significance level. More specifically, in regression models, the null-hypothesis that the coefficient of a predictor is equal to 0 is tested against the alternative hypothesis that the coefficient is not equal to 0. If the p-value is smaller than the significance level, then the null-hypothesis, that the coefficient is equal to 0, is rejected and the predictor is considered "statistically significant".

Unfortunately, p-values are difficult to understand properly and are therefore often misused and misinterpreted, especially in the applied sciences. A common mistake when interpreting p-values is thinking that a smaller p-value means that a predictor has a larger effect in the model, or that the predictor has a larger scientific significance. This is generally not true since any predictor with any kind of "effect" on the response can get an arbitrarily small p-value if there are enough observations (Simmons et al., 2011; Head et al., 2015). Goodman (2008) lists some other common mistakes in interpreting p-values. In addition to the problems in the interpretability of p-values, another common critique on p-values is that there is a sharp cut-off, where a result is considered statistically significant if the p-value is smaller than the significance value and not statistically significant if the p-value is larger than the significant value. The sharp cut-off can lead to what is called p-hacking, which is the practice of redoing statistical analyses with small modifications until a statistical significant result is found. The practice of p-hacking, combined with the tendency of journals to only publish papers with "statistically significant" results, has led to a large amount of false positives in literature and what is called a "reproducibility crisis" (Ioannidis, 2005; Gelman and Loken, 2014). The difficulty in interpreting p-values correctly along with the reproducibility crisis has caused controversy and has led to debate on how p-values should be used, or even if they should be used at all (Nuzzo, 2014; Claridge-Change and Assam, 2016; Goodman, 2016; Wasserstein and Lazar, 2016; Ioannidis, 2018; Wasserstein, Schirm et al., 2019). All this is not to say that p-values are not useful or that they do not give valuable information about a model, only that that information is more limited than what many researchers believe and that care should be used when interpreting them.

In light of the problems that p-values have, scientists have started to look for supplementary tools which give additional information about statistical models and which can help make the models easier to interpret. Some of the most common supplements that are often used in conjunction with or instead of p-values are

(i) **The absolute- or squared value of the coefficients of the (standardized) predictors.** These are useful to get an idea of the size of the effect of the predictor on the response if the predictors are uncorrelated. If the predictors are correlated, however, then some predictors might get artificially large coefficients, while other predictors would correspondingly get artificially small coefficients which makes the coefficients difficult to interpret.

(ii) **Confidence intervals of the coefficients of the predictors.** Confidence intervals of the coefficients give more information than the pure coefficient and the p-value and gives a range of values for the coefficient which is compatible with the data. Unfortunately, the way confidence intervals are often used is by checking if 0 is included in the 95% confidence interval of a coefficient estimate, which is equivalent to checking if the p-value is less than 0.05. Such a use of confidence intervals therefore inherits most of the problems that p-values have. Confidence intervals of the coefficients also have the same problem as looking at the coefficients, *i.e.*, if the coefficients are correlated, predictors might get artificially large or small coefficients, which causes the center of the confidence intervals to change.

(iii) **The correlation between the predictors and the response** (Darlington, 1968; Grömping, 2015). The advantage of looking at the correlation between a predictor and the response is that it is not influenced by the predictors being correlated. The correlation will give information on how strong the linear relationship between the predictor and the response is, which is not influenced by the other predictors in the model. The disadvantage is that if a model has two strongly correlated predictors, X_1 and X_2, then it might be difficult to interpret the result. In this case, by just using the sum of the correlations between the predictors and the response, it would seem like a model with both X_1 and X_2 would contain roughly twice the amount of information as a model with only one of the predictors. But since the predictors are strongly correlated, very little new information would be added to the model when adding a predictor, so the model with both predictors would be only slightly better.

If the pairwise correlation between all the predictors and the correlation between the predictors and response are analyzed carefully, the correlations could give a lot of information about the model, but analyzing the correlations properly can be challenging, especially as the number of correlations increases with the square of the number of predictors in the model.

(iv) **Information criteria such as AIC or BIC** (Akaike, 1973; Schwarz, 1978). Information criteria have been considered as a useful alternative to p-values and are especially useful for variable selection in statistical models (Burnham and Anderson, 2002; Johnson and Omland, 2004; Claeskens and Hjort, 2008; Burnham and Anderson, 2014). Information criteria are not as useful when interpreting models, however, since the criteria are usually used to look at the unique information added to the model by a predictor. This can be useful information, but it can be difficult to interpret, since predictors will not get "credit" for information shared with other predictor. This causes the same problem as in the above methods, where it can be difficult to interpret importances of correlated predictors.

(v) **Look at the difference in R^2 when the predictor of interest is removed from the full model.** This method gives the proportion of variance in the response uniquely explained by the predictor.

(vi) **Look at the R^2 of the model with only the predictor of interest.** This method gives the total proportion of variance in the response explained by the predictor.

(vii) **Add the predictors to the model in some order, and look at the increase in R^2 when each predictor is added to the model.** This method gives the amount of new contribution each predictor gives to the model when they are added to the model in some order.

Where (v) - (vii) are naive approaches based on the coefficient of determination, commonly called R^2. The R^2 gives the proportion of variance in the response explained by the model, which is a value scientists often have an understanding of. Since the R^2 is an intuitive value, a simple to understand supplementary statistic that can be used to understand the model could be the contribution of a predictor to the R^2 of the model. Because the R^2 is the proportion of variance in the response explained by the model, the contribution of a predictor to the R^2 can be interpreted as the proportion of variance in the response explained by the predictor. As long as the scientist has an understanding of the R^2, then the interpretation of the predictors contribution to R^2 should be relatively simple to understand. Since metrics based on R^2 are simpler to interpret than the p-value these metrics will hopefully reduce the misuse and misunderstandings when interpreting models.

The methods listed above all have problems when the predictors are correlated, which is usually the case in real world applications. Since p-values are based on the coefficients in the model they have the same problems as the coefficients when there are correlated predictors. The methods based on R^2, (v) - (vii), will all give the same result when the predictors are uncorrelated, but when the predictors are correlated the methods each give different information about the predictors. The problems caused by correlated predictors means that the methods listed above are difficult to use in practice. Methods to determine the relative importance of predictors, which we will call relative variable importance, which handle correlated predictors are therefore needed. The methods which will be focused on in this thesis are ones which are based on R^2. The prior work described below is developed for linear regression models.

Some of the earliest of the work on relative variable importance methods based on R^2 which handle correlated predictors dates back to the 1960s (Hoffman, 1960; Hoffman, 1962; Ward, 1962; Johnson, 1966). The first methods were based on the fact the squared standardized coefficients are meaningful for uncorrelated predictors, and scientists therefore worked on transforming the observed data to get uncorrelated predictors. Using the coefficients of the uncorrelated predictors it is possible to determine the contribution of each original predictor to the model R^2. The work by Johnson (1966) was later improved on by Fabbris (1980), Genizi (1993) and Johnson (2000), who all, independently, proposed the same improved technique to relate the coefficients of the uncorrelated predictors back to the original predictors, which will be called relative weights in this thesis (Nimon and Oswald, 2013).

Lindeman et al. (1980) created a variation of method (vii), which is commonly called the LMG method according to the last names of the authors, Lindeman, Merenda and Gold, where instead of just considering one ordering of the predictors, look at the mean increase in R^2 over all orderings of the predictors when the predictor of interest is added to the model. The LMG method was independently discovered by Kruskal (1987). A variation of the LMG method, called the PMVD method, where the orderings are weighted differently was introduced by Feldman (2005). Chevan and Sutherland (1991) used a similar concept to the one used in the LMG method in what they call hierarchical partitioning. The LMG method was extended into dominance analysis by Budescu (1993), Azen and Budescu (2003) and Budescu and Azen (2004), which in addition to giving an importance to each individual predictor also considers groups of predictors. Stufken (1992) and Lipovetsky and Conklin (2001) related the LMG method to game-theory, by noting that the method is equivalent to the Shapley value (Shapley, 1953). As the relative weights method is less computationally demanding than the LMG method it has been considered an ap-

	$\log(Hg_{soil})$	age	$smoking$	$\sqrt{amalgam}$	$\sqrt{fish}$	Total R^2
Coefficient	0.048	0.014	0.265	0.286	0.139	
p-value	0.38	0.06	0.003	< 0.001	< 0.001	
Rel. imp.	0.002	0.038	0.023	0.316	0.064	0.444

Table 1: Coefficients, p-values and relative variable importances based on the LMG method. The values are from a model modeling the mercury in urine with the predictors: $\log(Hg_{soil})$ which is the mercury concentration in the soil where the person lives, age which is the age of the person, $smoking$ which is whether the person smokes or not, $\sqrt{amalgam}$ which is the number of amalgam fillings the person has in their teeth and $\sqrt{fish}$ which is the number of fish meals consumed by the person each month. The data is from Imo et al. (2017)

proximation of the LMG method that can be used when the LMG method is not computationally viable (Grömping, 2015). This thesis will focus on the LMG method and the relative weights method. Both the LMG method and the relative weights method will be considered because we believe that the LMG method will give more accurate decompositions than the relative weights method, but the relative weights method is needed when too many predictors in the model makes the LMG method computationally unviable.

The approaches discussed above based on the proportion of variance explained by each predictor gives complementary information to p-values. The p-value gives information on whether the predictor has any effect on the response, but it says nothing about how large the effect is or how much information the predictor carries. The relative variable importance measures, however, do not try to tackle the problem of testing the null-hypothesis that each predictor has no effect on the model, it only focuses on the amount of information the predictor contributes to the model. The difference is illustrated in Table 1, which shows coefficients, p-values and the relative variable importances from the LMG method of a model with 5 predictors. The model has the concentration of mercury in urine as the response and the predictors are the amount of mercury in the soil, the age of the person, whether the person is smoking or not, the number of teeth with amalgam fillings and the number of fish meals consumed per month respectively (Imo et al., 2017). We can see that $\sqrt{amalgam}$ and $\sqrt{fish}$ both have p-values < 0.001, which could at first glance give the impression that they are equally important to the model and they explain roughly the same amount of the mercury concentration. Looking at the relative importance of the two predictors, however, we see that $\sqrt{amalgam}$ has a relative importance of 0.316 while $\sqrt{fish}$ has a relative importance of only 0.064. This means that, according to the LMG method, $\sqrt{amalgam}$ explains roughly 5 times more of the variance in the response than $\sqrt{fish}$ and that the number of amalgam fillings is therefore likely more important to the concentration of mercury than the amount of fish eaten a month.

All the methods discussed so far are based on linear regression models. In many applications, such as biology, epidemiology and the social sciences, standard linear regression models are not sufficient, instead, random intercept models are often used. Probably the most common method used to determine the importance of a random intercept in a model is to look at the random intercepts variance in the model. A larger variance means there is more information contained in the random intercept. Looking at variances, however, has the same problem of looking at coefficients of fixed effects, if the random intercepts are correlated, either with each other or with fixed effects, then the random intercepts might absorb information from the other predictors in the model or other predictors might absorb information from the random intercepts. The absorption of information can cause the variances to change and not represent the correct amount of information contained in the random intercept. This thesis will therefore focus on expanding the concept of relative variable importance to random intercept models. More specifically, we want to expand the LMG method and the relative weights method to give relative importances to both the fixed effects and the random intercepts in random intercept models.

Some work on relative variable importance measures for random intercept models has recently been done by Stoffel et al. (2021), where they attempt to calculate relative variable importances in generalized linear mixed-effect models (GLMMs). Their approach has some weaknesses, however. First, importances are only given to the fixed effects in the model, not the random effects. Second, little attempt at taking the correlations between the predictors into account is made. The idea behind the two approaches Stoffel et al. (2021) end up with, part R^2 and inclusive R^2, will be considered and criticized in Section 2.5.1. Simply explained, only the uniquely explained variance, similar to method (v), and the total explained variance, similar to method (vi), by each predictor, respectively, were considered. A different attempt at extending the concept of relative variable importance to random intercept models was done by Byhring (2020). That work, however, was only able to give importances to the predictors in a model with only one random intercept, which is limiting. Additionally, it did not properly take the correlations between the fixed effects and the random intercept into account.

The approaches proposed in this thesis will expand the LMG and relative weights method to random intercept models with an arbitrary number of random intercepts and fixed effects. The approaches will, simply speaking, distribute the "credit" for information contained in correlated predictors to all predictors containing that information. Such a distribution of the credit will give shares which add up to the model R^2, which makes the shares more interpretable. The shares can be interpreted as the amount of R^2 each predictor contributes, or, equivalently, as the proportion of information in the response the predictor explains.

In addition to proposing methods to calculate relative variable importance of the predictors in random intercept models, a focus of this thesis is the development of an R package implementing these methods. We believe an R package will be a useful tool for statistical analyses, since usage of relative variable importances has been increasing and the R package `relaimpo`, which implements several of the mentioned methods for linear regression models, is quite popular with over 260 000 downloads since it's introduction (Grömping, 2006; MetaCRAN, 2022). The end goal of this thesis is therefore to create a simple to use tool that researchers can use to help interpret their models which is easier to interpret than the tools available today. Such a tool will hopefully let researchers who are not statisticians better understand their results which can lead to better research. The R-package can be found at https://gitlab.com/elonus/decompr2.

Section 2 will contain background theory and an introduction to relative variable importance metrics. The proposed new methods which give relative variable importances to random intercept models will be in Section 3. Section 4 contains a simulation study showing some of the properties of the proposed methods while Section 5 shows the new methods applied on a real dataset. A discussion of the results will be in Section 6. Finally, a vignette showing the use of the R-package along with the implementation will be in Appendix A and Appendix B.

2 Theory

We will here cover existing theory and methods that will be used in this thesis. Linear regression, linear mixed-effect models and the coefficient of determination for linear regression models are assumed known to the reader. This section will therefore only contain a brief review of the respective theory. Then, calculation of the coefficient of determination for linear mixed-effect models will be discussed before introducing relative importance measures that will be generalized.

2.1 Linear regression models

A very commonly used technique in data modeling is linear regression. Linear regression models the relationship between the response, $y_i,\ i \in \{1, 2, \ldots, n\}$, and the predictors, $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where n is the number of observations and $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,p})^T$ is one observation, where p is now the number of predictors, for $i \in \{1, 2, \ldots, n\}$. The data is assumed to be in the form

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_p x_{i,p} + \epsilon_i \ , \tag{1}$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^T$, called the coefficients of the model, describes the deterministic part of the relationship between y_i and $\mathbf{x}_i$. The stochastic part of the relationship between y_i and $\mathbf{x}_i$ is described by $\epsilon_i \sim N(0, \sigma_\epsilon^2)$, which is the error term of the model and called the residual. The β_0 in $\boldsymbol{\beta}$ is called the intercept, which makes the model centered on the mean of the response.

If a predictor is categorical, some special care needs to be taken when adding it to the model. The encoding of the categories should not matter, meaning that if the different categories are identified by different numbers, then changing these numbers should not affect the model. This means that even if the encoding is numeric, it should not just be added to the model as any other predictor, since changing the encoding would change the model. Instead, the typical way to add categorical predictors to a linear regression model is through *dummy encoding.* If the categorical predictor has l levels, then dummy encoding adds $l - 1$ new binary "predictors", $x_{i,k_1}, x_{i,k_2}, \ldots, x_{i,k_{l-1}}$, where

$$\begin{aligned} x_{i,k_1} &= I(\mathit{Observation\ i\ has\ level\ k_1\ of\ the\ categorical\ predictor}) \\ x_{i,k_2} &= I(\mathit{Observation\ i\ has\ level\ k_2\ of\ the\ categorical\ predictor}) \\ &\vdots \\ x_{i,k_{l-1}} &= I(\mathit{Observation\ i\ has\ level\ k_{l-1}\ of\ the\ categorical\ predictor}) \ , \end{aligned}$$

where $I()$ is the indicator function. A column for when observation i has level k_l is not needed, since this case is equivalent to all of $x_{i,k_1}, x_{i,k_2}, \ldots, x_{i,k_{l-1}}$ being zero.

The relationship in equation (1) is usually rewritten in the more compact form

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \ ,$$

where $\boldsymbol{\epsilon} \sim N_n\left(0, \sigma_\epsilon^2 \mathbf{I}_n\right)$ and $\mathbf{X}$ is the $n \times (p+1)$ matrix

$$\mathbf{X} = \begin{bmatrix} 1 & \mathbf{x}_1 \\ 1 & \mathbf{x}_2 \\ \vdots & \vdots \\ 1 & \mathbf{x}_n \end{bmatrix} \ .$$

The 1's are added to the matrix to represent the intercept.

Generally, the true coefficients are not known. Only the response, $\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$, and the predictors, $\mathbf{X}$, are observed. The goal of fitting the model is therefore to find estimates for the coefficients, $\hat{\boldsymbol{\beta}}$, to attempts to understand the deterministic part of the relationship between y and $\mathbf{x}$. Fitting the model is usually done by minimizing the error sum of squares

$$\mathrm{SS}_E = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \left(y_i - \mathbf{x}_i^T \hat{\boldsymbol{\beta}}\right)^2 ,$$

which gives the estimate

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^T \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{y} .$$

The derivation of the estimate can be found in, *e.g.*, Fahrmeir et al. (2013, Chapter 3.2).

2.2 Linear random intercept models

A common scenario when modeling real world data is that observations might be clustered in some way. Either in time, *e.g.*, by repeated measurements of the same individual, or in space, *e.g.*, measurements of several individuals in the same geographic area, like an island. Since observations within the same cluster are often not statistically independent, the clustering needs to be taken into account in the model. If there are few clusters compared to the number of observations, this can be solved by categorical predictors. If the number of clusters are large compared to the number of observations, however, this approach will give a model with a large amount of predictors estimated by relatively few observations. This means that the predictors will be very sensitive to slight changes in the observed data, which gives a model with a large variance.

Linear Mixed-effect Models (LMMs) attempt to solve this problem by imposing a regularization assumption on the coefficients explaining the cluster effects. This assumption restricts the estimates of these coefficients, which makes them less sensitive to small changes in the observed data, which causes the variance in the model to be reduced. The assumption will, however, have a trade-off. It will cause the model to not fit as closely to the observed data, since the restriction will prevent the coefficients from taking the value which makes the predicted values as close to the observed values as possible. This is called the bias in the model. Generally, decreasing the variance will increase the bias in the model, and vice versa. This tendency is called the bias-variance trade-off, see *e.g.*, Hastie et al. (2009, Chapter 2.9).

The simplest type of a LMM is the random intercept model, which assumes the data can be described by

$$y_{i,j} = \beta_0 + \mathbf{x}_{i,j}^T \boldsymbol{\beta} + \alpha_j + \epsilon_{i,j} , \tag{2}$$

where $y_{i,j}$ is the i'th observation of cluster j, β_0 is the mean of the response, while $\alpha_j \sim N\left(0, \sigma_\alpha^2\right)$ is the random effect, also called the cluster-specific effect, which describes how the mean of the cluster deviates from the population-wide mean. Further, $\epsilon_{i,j} \sim N\left(0, \sigma_\epsilon^2\right)$ is the residual for the i'th observation for cluster j. Finally, σ_α^2 is called the between-cluster variance and σ_ϵ^2 is called the within-cluster variance (Nakagawa and Schielzeth, 2013).

Adding the cluster-specific effects, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)^T$, where m is the number of clusters, to the model means that the model explains the dependence caused by the observations being in the same cluster, since α_j explains the dependent part. A similar effect of removing the dependence could be achieved by using a categorical predictor in a normal linear regression model. However, this results in a slightly different model, since the assumed normal distribution of the cluster-specific effects works as a regularization assumption. This regularization assumption reduces the variance of the model by introducing bias through the bias-variance trade-off. More concretely, the assumed normal distribution causes a shrinking effect, which causes the cluster-specific effects to be closer to zero than they would otherwise be.

The model in equation (2) can be written in a more general form

$$\mathbf{y} = \beta_0 + \mathbf{X}\boldsymbol{\beta} + \mathbf{R}\mathbf{u} + \boldsymbol{\epsilon} \ , \tag{3}$$

where β_0 is the mean of the response and $\mathbf{X}$ is a $n \times p$ matrix describing the fixed effects. Note that $\mathbf{X}$ does not have an intercept column since the intercept is added as a separate term. Further, $\boldsymbol{\beta}$ is a vector of length p describing the relationship between the fixed effects and the response and $\mathbf{R}$ describes the random effects (in this case it is just a dummy encoding of the categorical clustering predictor). Finally $\mathbf{u} \sim N_m\left(0, \sigma_\alpha^2 \mathbf{I}_m\right)$ is the same as $\boldsymbol{\alpha}$, and $\boldsymbol{\epsilon}$ is the same as the residuals described above.

More complicated random intercept models can be constructed with several different clustering predictors, where the coefficients describing the relationship of the clustering predictors with the response have a dependency structure. Then, $\mathbf{R}$ will be a matrix containing the dummy encoding of all the clustering predictors, while $\mathbf{u} \sim N(\mathbf{0}, \mathbf{A})$ describes the relationship between the clustering predictors and the response, with $\mathbf{A}$ describing the dependency structure of the coefficients. For the simple case where there are two independent clustering predictors $\mathbf{A}$ will simply be

$$\begin{bmatrix} \sigma_{r_1}^2 \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \sigma_{r_2}^2 \mathbf{I} \end{bmatrix} ,$$

where $\sigma_{r_1}^2$ is the variance of the coefficients for the first clustering predictor and $\sigma_{r_2}^2$ is the variance of the coefficients for the second clustering predictor. But $\mathbf{A}$ can be any valid covariance matrix, which describes more complex data relationships. More complex models where $\mathbf{R}$ does not just contain dummy encodings of categorical predictors are also possible, but they will not be discussed in this thesis. For more information of these more complex models as well as details of how LMMs are fitted, see, *e.g.*, Fahrmeir et al. (2013, Chapter 7).

2.3 Coefficient of Determination (R^2)

2.3.1 Linear regression

The coefficient of determination, usually called R^2, is a value often used in statistical modeling, since it gives information regarding how much of the information in the response the model explains. More precisely, it is the proportion between the variance of the fitted values and the variance of the response, or equivalently, one minus the proportion between the variance of the residuals and the variance of the response. In linear regression it is defined through sum of squares. By defining

$$
\begin{aligned}
\mathrm{SS}_R &= \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 \\
\mathrm{SS}_{tot} &= \sum_{i=1}^{n} (y_i - \bar{y})^2 \\
\mathrm{SS}_E &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \hat{\epsilon}_i^2 \ ,
\end{aligned}
$$

R^2 can be defined by

$$
\begin{aligned}
R^2 &= \frac{\mathrm{SS}_R}{\mathrm{SS}_{tot}} \\
&= 1 - \frac{\mathrm{SS}_E}{\mathrm{SS}_{tot}} \ ,
\end{aligned}
$$

where the last equivalence comes from the fact that $\mathrm{SS}_{tot} = \mathrm{SS}_E + \mathrm{SS}_R$ in linear regression models. SS_R, SS_{tot} and SS_E can be thought of as $\widehat{\mathrm{Var}}\left(\hat{\mathbf{y}}\right)$, $\widehat{\mathrm{Var}}\left(\mathbf{y}\right)$ and $\widehat{\mathrm{Var}}\left(\boldsymbol{\epsilon}\right)$ respectively.

An R^2 close to 1 means that the model explains a lot of the variance in the response while an R^2 close to 0 means that there is a lot of variance in the response that the model does not explain. Different fields and applications have different criteria for what is considered a good R^2 value. If there is a lot of noise in the response that can not be modeled by the available predictors, then even a relatively small R^2 might be considered good.

2.3.2 Linear Mixed-effect Models

To generalize the concept of R^2 to LMMs it is useful to rewrite the variance of the response. If the fixed effects are assumed independent from the random effects and the residuals and the random effects are assumed independent from the residuals, then the variance of the response can be written as

$$
\mathrm{Var}\left(y\right) = \sigma_f^2 + \sigma_r^2 + \sigma_\epsilon^2 \ , \tag{4}
$$

where $\sigma_f^2 = \mathrm{Var}\left(\mathbf{x}^T \boldsymbol{\beta}\right)$ is the variance of the fixed effects, σ_r^2 is the variance of the random effects and σ_ϵ^2 is the variance of the residuals. Then two R^2 definitions can be made

$$
R^2_{(m)} = \frac{\sigma_f^2}{\sigma_f^2 + \sigma_r^2 + \sigma_\epsilon^2} \tag{5}
$$

$$
R^2_{(c)} = \frac{\sigma_f^2 + \sigma_r^2}{\sigma_f^2 + \sigma_r^2 + \sigma_\epsilon^2} \ , \tag{6}
$$

where $R^2_{(m)}$ is called the marginal R^2, which gives the proportion of variance the fixed effects explain, and $R^2_{(c)}$ is called the conditional R^2, which gives the proportion of variance the whole model explains.

The variance of the fixed effects, σ_f^2, can be estimated by

$$\sigma_f^2 = \text{Var}\left(\mathbf{x}^T\boldsymbol{\beta}\right) = \frac{1}{n-1}\sum_{i=1}^{n}\left(\hat{y}_{f,i} - \bar{\hat{y}}_f\right) \ ,$$

where $\hat{\mathbf{y}}_f = \mathbf{X}\hat{\boldsymbol{\beta}}$. The other variances, σ_r^2 and σ_ϵ^2, are estimated as part of the model fit. More details regarding this extension of R^2 to LMMs can be found in Nakagawa and Schielzeth (2013).

2.4 Relative variable importance based on R^2 decomposition

As mentioned in Section 1, many approaches have been proposed to find the relative importance of a predictor in linear regression models. A common approach is to base the relative importance on the coefficient of determination (R^2), since the R^2 of a model gives information regarding how much of the variance in the response the model explains (Grömping, 2015). For uncorrelated predictors, it is trivial to give a relative importance to a predictor; the decomposition can be based on the coefficients in the model. To see this, consider that according to the assumptions of a linear regression model, the variance of y is,

$$\begin{aligned}\text{Var}\left(y\right) &= \text{Var}\left(\mathbf{x}^T\boldsymbol{\beta} + \epsilon\right) \\ &= \sum_{i=1}^{p}\beta_i^2\text{Var}\left(x_i\right) + \sum_{\substack{i,j=1 \\ i\neq j}}^{p}\beta_i\beta_j\text{Cov}\left(x_i, x_j\right) + \sigma_\epsilon^2 \ .\end{aligned}$$

If the predictors are uncorrelated, the above equation simplifies to

$$\text{Var}\left(y\right) = \sum_{i=1}^{p}\beta_i^2\text{Var}\left(x_i\right) + \sigma_\epsilon^2 \ ,$$

meaning that the contribution of each predictor is simply $\beta_i^2\text{Var}\left(x_i\right)$, which simplifies to β_i^2 if $\mathbf{X}$ is standardized. It is not obvious, however, how to distribute the R^2 to each predictor in the model when the predictors are correlated.

The literature agrees that when considering relative importance metrics there are some conditions that should be satisfied such that decompositions can be interpreted in a useful way (Grömping, 2007; Grömping, 2015; Feldman, 2005). While Grömping (2015) lists 12 criteria that can be used when evaluating relative importance measures, we will focus on

i. **Proper decomposition**: The R^2 of the model is to be decomposed into shares, that is, the sum of all shares has to be the R^2 of the model.

ii. **Non-negativity**: All shares have to be non-negative.

iii. **Inclusion**: A regressor X_j with $\beta_j \neq 0$ should receive a nonzero share.

which are the ones focused on in Grömping (2007). We consider these three the most important criteria, especially proper decomposition is considered important at it is essential for the intuitive understanding of the relative importances. When evaluating the relative importance measures we will be checking whether they satisfy the proper decomposition, non-negativity and inclusion criteria.

2.5 Relative variable importance for linear regression

In this section we will discuss approaches used to decompose the R^2 of linear regression models with correlated predictors.

2.5.1 Naive approaches that do not work

Before looking at possible approaches to handle correlated predictors, some notation is needed. There is a response, y, along with p predictors, note that this is a different p from the p-value, $(1, 2, \ldots, p)$, to which a linear regression model is fitted. Let the relative importance of predictor x_i be $RI(i)$. Since we want to compare the R^2 of models with different subsets of predictors in them, let $R^2(S)$, where $S = \{h_1, h_2, \ldots, h_r\} \subseteq \{1, 2, \ldots, p\}$, be the R^2 of the linear regression model fitted with only the predictors $\{h_1, h_2, \ldots, h_r\}$. Three easy solutions will now be considering as well as showing why they do not work.

The most obvious approach to see how much predictor i contributes to the model R^2 would be to look at the difference in R^2 between the full model and the model with predictor i removed, *i.e.*, $RI(i) = R^2(\{1, 2, \ldots, p\}) - R^2(\{1, 2, \ldots, p\} \setminus i)$. However, this approach has problems when the predictors are correlated. To see why, consider the simple case where $Y = X_1 + X_2$ and

$$\text{Var}\left((X_1, X_2)\right) = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix} .$$

Then $R^2(\{1, 2\}) = 1$, since the relationship between Y and (X_1, X_2) is deterministic and all the variance in Y is explained by the predictors. In this case, it would be expected that $RI(\{1\}) = RI(\{2\}) = 0.5$ since X_1 and X_2 explain the same amount of variance due to symmetry. However, looking at the difference in R^2 when the predictor of interest is removed from the full model would instead give

$$RI(1) = R^2(\{1, 2\}) - R^2(\{2\}) = 1 - \frac{\hat{\beta}_2^2}{\text{Var}(Y)} = 1 - \frac{1.9^2}{3.8} \approx 0.05 ,$$

where, in the second equality, the known result that, for linear regression with one predictor, the coefficient is simply the correlation between the response and the predictor is used.

Symmetry gives $RI(\{2\}) = RI(\{1\})$. Hence, looking at the difference in R^2 when the predictor of interest is removed from the full model violates the proper decomposition criterion in this example, since

$$RI(\{1\}) + RI(\{2\}) = 0.05 + 0.05 = 0.1 < 1 .$$

More generally, the same problem occurs whenever there is correlation between the predictors, the relative importance of each predictor will be too small, since each predictor only gets "credit" for the information uniquely in that predictor, not the information in the overlap with other predictors in the model (Grömping, 2015). Looking at the difference in R^2 when the predictor of interest is removed from the full model is what Stoffel et al. (2021) calls part R^2.

Another simple approach is to instead compare the model where predictor i is the only predictor with the empty model with only an intercept, *i.e.*, $RI(\{i\}) = R^2(\{i\}) - R^2(\{\})$. The R^2 of a model with only an intercept is always 0, however, so $RI(i)$ simplifies to $RI(i) = R^2(\{i\})$. The same example used when discussing the previous approach gives

$$RI(\{1\}) = R^2(\{1\}) = \frac{\hat{\beta}_1^2}{\mathrm{Var}(Y)} = \frac{1.9^2}{3.8} \approx 0.95 \ ,$$

where symmetry again gives $RI(\{2\}) = RI(\{1\})$. This means

$$RI(\{1\}) + RI(\{2\}) \approx 0.95 + 0.95 = 1.9 > 1 \ .$$

Such an approach generally gives a too large share when there are correlated predictors, since each predictor gets “credit” for both the information uniquely explained by that predictor, but also all the information it explains jointly with other correlated predictors. Just looking at the R^2 of a model only the predictor of interest is a scaled version of what Stoffel et al. (2021) calls Inclusive R^2.

Finally, consider linear regression models fitted with the sets of predictors

$$\{\}, \{1\}, \{1,2\}, \dots, \{1,2,\dots,p\} \ .$$

Then, give predictor 1 the share $R^2(\{1\}) - R^2(\{\})$, predictor 2 the share $R^2(\{1,2\}) - R^2(\{1\})$, and so on until predictor p would get the share $R^2(\{1,2,\dots,p\}) - R^2(\{1,2,\dots,p-1\})$. However, this is not a very robust approach either, since if two predictors are correlated, the first one added will get the contribution of the common part. Thus, changing the ordering of the predictors will likely change the shares, *e.g.*, if the predictors are added to the model in the reverse order, then predictor p will likely get a larger share than when it is added last.

2.5.2 The LMG method

To address the shortcomings of the naive methods presented in Section 2.5.1, Lindeman et al. (1980) proposed to consider all possible orderings of the predictors, and give each predictor the mean increase in R^2 when the predictor is added to the model as its share. If $P = \{\pi_1, \pi_2, \dots, \pi_{p!}\}$ is the set of all permutations of $(1,2,\dots,p)$, and $S_i(\pi_h)$ be the set of predictors appearing before i in π_h, then the share given to predictor i is the mean increase of R^2 when predictor i is added to the model, defined as

$$RI(i) = LMG(i) = \frac{1}{p!} \sum_{\pi \in P} \left(R^2(S_i(\pi) \cup i) - R^2(S_i(\pi)) \right) \ . \tag{7}$$

Equation 7 can be rewritten in the more computationally efficient form

$$RI(i) = LMG(i) = \frac{1}{p!} \sum_{S \subseteq \{1,2,\dots,p\} \setminus i} |S|!\,(p - |S| - 1)! \left(R^2(S \cup i) - R^2(S) \right) \ , \tag{8}$$

since the order of predictors before and after predictor i does not change the compared models (Grömping, 2007). The reformulation in equation (8) reduces the computational complexity from $O(p!)$ to $O(2^{p-1})$.

The LMG method has no problem working with categorical predictors. The categorical predictor can be permuted the same way as a numerical predictor. Note, however, that it is not the individual columns in the dummy encoding that is permuted, it is the whole categorical predictor which is added or removed from the model.

Importantly, the criteria listed in Section 2.4 are satisfied by the LMG approach:

- **Proper decomposition**: Feldman (2005) gives a proof showing that the LMG method, called the averaging decomposition in the paper, will always give a proper decompositions. This proof is complex, however, so a simpler proof, which also applies to the proposed new methods of the thesis, will be shown in Section 3.1.
- **Non-negativity**: R^2 never decreases when a predictor is added to a model, see *e.g.*, Fahrmeir et al. (2013, Chapter 3.2.3). This means that $R^2\left(S \cup i\right) \geq R^2(S)\ \forall i \in \{1, 2, \dots, p\}, S \subseteq \{1, 2, \dots, p\} \setminus i$, from which $RI(i) \geq 0$ trivially follows.
- **Inclusion**: Let $h \in \{1, 2, \dots, p\}$ be some predictor. If $\hat{\beta}_h \neq 0$ and $\mathbf{y}$ and $\mathbf{x}_h$ have a non-zero variance, then

  $$R^2(\{h\}) - R^2(\{\}) = \frac{\hat{\beta}_h^2 \widehat{\mathrm{Var}}\left(\mathbf{x_h}\right)}{\widehat{\mathrm{Var}}\left(\mathbf{y}\right)} - 0 > 0\ . \tag{9}$$

  This, combined with the fact that none of the terms of the sum in equation (8) can be negative, means that $RI(h) > 0$.

Thus, the LMG method is a robust approach that gives useful shares to each predictor.

2.5.3 Relative weights for linear regression

The high computational complexity of the LMG method means that it can be difficult to calculate the relative variable importances when the number of predictors is large. This section will therefore introduce an alternative approach which is significantly less computationally complex. The alternative method, denoted as *relative weights*, will give an approximation of the LMG method and was independently discovered by Fabbris (1980), Genizi (1993) and Johnson (2000), see Nimon and Oswald (2013). Here, the formulation and name used by Johnson (2000) will be used. The relative weights method relies on the fact that relative importances are easy to calculate for uncorrelated predictors, in which case they simply are the square of the standardized coefficients of the linear regression model (Section 2.4). We can assume, without loss of generality, that the predictors and the response are standardized and centered such that they have zero mean and unit variance. This makes it possible to remove the intercept from the $n \times p$ model-matrix, $\mathbf{X}$. The relative weights method works by using the singular value decomposition of the model-matrix

$$\mathbf{X} = \mathbf{P}\mathbf{D}\mathbf{Q}^T\ ,$$

where $\mathbf{P}$ is an $n \times p$ orthonormal matrix containing the eigenvectors of $\mathbf{X}\mathbf{X}^\mathbf{T}$, $\mathbf{D}$ is a $p \times p$ diagonal matrix with the singular values of $\mathbf{X}$ on its diagonal while $\mathbf{Q}$ is a $p \times p$ orthonormal matrix containing the eigenvectors of $\mathbf{X}^\mathbf{T}\mathbf{X}$, see, *e.g.*, Friedberg et al. (2003). The $n \times p$ matrix

$$\mathbf{Z} = \sqrt{n-1}\mathbf{P}\mathbf{Q}^\mathbf{T}$$

can then be created, which is the closest orthogonal matrix to $\mathbf{X}$ in the least-square sense, meaning $\mathbf{Z} = \mathbf{P}\mathbf{Q}^{\mathbf{T}}$ is the orthogonal matrix minimizing

$$tr\left((\mathbf{X} - \mathbf{Z})^T (\mathbf{X} - \mathbf{Z})\right) \ ,$$

where $tr\,()$ means the trace of a matrix (Johnson, 1966). The $\sqrt{n-1}$ factor is used to keep $\mathbf{Z}$ standardized, since each column of $\mathbf{Z}$ would otherwise have a variance of $\frac{1}{n-1}$.

Since the columns in $\mathbf{Z}$ are orthogonal, they are also uncorrelated, meaning that the relative importance of each column in $\mathbf{Z}$ is the square of their coefficients. These coefficients can easily be calculated by

$$\begin{aligned}
\hat{\boldsymbol{\beta}}_{\mathbf{Z}} &= \left(\mathbf{Z}^{\mathbf{T}}\mathbf{Z}\right)^{-1}\mathbf{Z}^{\mathbf{T}}\mathbf{y} \\
&= \left((n-1)\,\mathbf{Q}\mathbf{P}^T\mathbf{P}\mathbf{Q}^T\right)^{-1}\sqrt{n-1}\mathbf{Q}\mathbf{P}^T\mathbf{y} \\
&= \frac{1}{\sqrt{n-1}}\mathbf{Q}\mathbf{P}^{\mathbf{T}}\mathbf{y} \ ,
\end{aligned}$$

as $\mathbf{P}^T\mathbf{P} = \mathbf{Q}\mathbf{Q}^T = \mathbf{I}$ and where $\mathbf{y}$ is the standardized and centered response. An intercept column in $\mathbf{Z}$ is not needed since $\mathbf{y}$ is centered and hence has mean 0. The columns of $\mathbf{Z}$ thus have relative importances $\hat{\boldsymbol{\beta}}_{\mathbf{Z}}^{[2]}$, where $\hat{\boldsymbol{\beta}}_{\mathbf{Z}}^{[2]}$ means that all elements of $\hat{\boldsymbol{\beta}}_{\mathbf{Z}}$ are squared. It is the importances of the columns of $\mathbf{X}$ which are of interest, however, not the importances of the columns of $\mathbf{Z}$. The calculated importances therefore need to be related back to the columns of interest. Johnson (2000) proposes using the regression coefficients when regressing $\mathbf{Z}$ on $\mathbf{X}$ for this purpose, which can be calculated by

$$\begin{aligned}
\boldsymbol{\Lambda} &= \left(\mathbf{Z}^{\mathbf{T}}\mathbf{Z}\right)^{-1}\mathbf{Z}^{\mathbf{T}}\mathbf{X} \\
&= \left((n-1)\,\mathbf{Q}\mathbf{P}^T\mathbf{P}\mathbf{Q}^T\right)^{-1}\sqrt{n-1}\mathbf{Q}\mathbf{P}^T\mathbf{P}\mathbf{D}\mathbf{Q}^T \\
&= \frac{1}{\sqrt{n-1}}\mathbf{Q}\mathbf{D}\mathbf{Q}^T \ ,
\end{aligned}$$

Since $\mathbf{X}$ is a linear combination of $\mathbf{Z}$, and vice versa, $\boldsymbol{\Lambda}$ will give a "perfect" regression, with an R^2 of 1. The "perfect" regression can be seen by

$$\begin{aligned}
\mathbf{Z}\boldsymbol{\Lambda} &= \sqrt{n-1}\mathbf{P}\mathbf{Q}^T\frac{1}{\sqrt{n-1}}\mathbf{Q}\mathbf{D}\mathbf{Q}^T \\
&= \mathbf{P}\mathbf{D}\mathbf{Q}^T \\
&= \mathbf{X} \ .
\end{aligned}$$

Additionally, the columns of $\mathbf{Z}$ are uncorrelated, meaning that the relative importance of each column of $\mathbf{Z}$ to each column of $\mathbf{X}$ is just the square of the corresponding regression coefficient. Combined, the sum of the squared elements of each column of $\boldsymbol{\Lambda}$ is 1.

The importances of the columns of $\mathbf{X}$, *i.e.*, the original predictors, are calculated by relating the relative importance of each column of $\mathbf{Z}$ to the columns of $\mathbf{X}$, by

$$RI(i) = [\boldsymbol{\Lambda}^{[2]}\hat{\boldsymbol{\beta}}_{\mathbf{Z}}^{[2]}]_i \ ,$$

where $\boldsymbol{\Lambda}^{[2]}$ means that all elements of $\boldsymbol{\Lambda}$ are squared.

It is computationally challenging to calculate $\mathbf{P}$ when there are many observations, since calculating $\mathbf{P}$ means calculating the eigenvectors of the $n \times n$ matrix $\mathbf{X}\mathbf{X}^T$. Therefore, in practice, another approach to calculate $\mathbf{Z}$ is used. Let

$$\begin{aligned}\mathbf{R}_{xx} &= \frac{1}{n-1}\mathbf{X}^T\mathbf{X} \\ &= \frac{1}{n-1}\mathbf{Q}\mathbf{D}\mathbf{P}^T\mathbf{P}\mathbf{D}\mathbf{Q}^T \\ &= \frac{1}{n-1}\mathbf{Q}\mathbf{D}\mathbf{D}\mathbf{Q}^T \\ &= \frac{1}{n-1}\mathbf{Q}\mathbf{D}^{[2]}\mathbf{Q}^T\end{aligned}$$

be the correlation matrix of the columns of $\mathbf{X}$, where $\mathbf{Q}$ and $\mathbf{D}$ are the same as above (Johnson, 2000). Next, let

$$\mathbf{R}_{xx}^{-\frac{1}{2}} = \sqrt{n-1}\mathbf{Q}\mathbf{D}^{[-1]}\mathbf{Q}^T \ .$$

Then $\mathbf{Z}$ can be calculated by

$$\begin{aligned}\mathbf{X}\mathbf{R}_{xx}^{[-\frac{1}{2}]} &= \mathbf{P}\mathbf{D}\mathbf{Q}^T\sqrt{n-1}\mathbf{Q}\mathbf{D}^{[-1]}\mathbf{Q}^T \\ &= \sqrt{n-1}\mathbf{P}\mathbf{D}\mathbf{D}^{[-1]}\mathbf{Q}^T \\ &= \sqrt{n-1}\mathbf{P}\mathbf{Q}^T \\ &= \mathbf{Z} \ ,\end{aligned}$$

thus explicitly calculating $\mathbf{P}$ is not needed. Only $\mathbf{Q}$ and $\mathbf{D}^{[2]}$ need to be calculated, but they only contain the eigenvectors and eigenvalues of the $p \times p$ matrix $\mathbf{X}^T\mathbf{X}$ respectively.

The relative weights method satisfies the criteria listed in Section 2.4, which means that it is useful as a more computationally efficient approximation of the LMG method.

- **Proper decomposition**: Since $\mathbf{Z}$ is a linear combination of $\mathbf{X}$, and vice versa, we know that the sum of the squared elements of the columns of $\mathbf{\Lambda}$ are equal to 1, *i.e.*, $\sum_{i=1}^{p} \lambda_{i,j}^2 = 1$ for $j = 1, 2, \ldots, p$. Thus,

$$\begin{aligned}\sum_{i=1}^{p} RI(i) &= \sum_{i=1}^{p} [\mathbf{\Lambda}^{[2]}\hat{\boldsymbol{\beta}}_{\mathbf{Z}}^{[2]}]_i \\ &= \sum_{i=1}^{p}\sum_{j=1}^{p} \lambda_{i,j}^2\hat{\beta}_{Z,j}^2 \\ &= \sum_{j=1}^{p} \hat{\beta}_{Z,j}^2 \sum_{i=1}^{p} \lambda_{i,j}^2 \\ &= \sum_{j=1}^{p} \hat{\beta}_{Z,j}^2 \cdot 1 \\ &= R^2\left(Z_1, Z_2, \ldots, Z_p\right) \\ &= R^2\left(X_1, X_2, \ldots, X_p\right) \ ,\end{aligned}$$

where the last equality comes from the fact that since $\mathbf{Z}$ is a linear combination of $\mathbf{X}$, the linear regression models with the columns of $\mathbf{Z}$ and columns of $\mathbf{X}$ are equivalent, in the sense that they explain the same information. This means that relative weights gives a proper decomposition.

- **Non-negativity**: The relative weights method will satisfy non-negativity, since the relative importances, $\mathbf{\Lambda}^{[2]}\hat{\boldsymbol{\beta}}_{\mathbf{Z}}^{[2]}$, are a matrix with non-negative elements multiplied with a vector with non-negative elements.
- **Inclusion**: The relative weights method satisfies the inclusion criteria (Grömping, 2015).

It would be useful if the relative weights method would work with categorical predictors in addition to numerical predictors. To use the relative weights method on categorical predictors the most obvious solution is to use the same method as described above, but add the column saying which level of the categorical predictor each observation is in to $\mathbf{X}$, alternatively, add the dummy encodings of the categorical predictors to $\mathbf{X}$. The problem, however, is that when $\mathbf{Z} = \mathbf{P}\mathbf{Q}^T$ is created, the nice properties of the added columns to $\mathbf{X}$ will likely be lost. If one column is added for each categorical predictor, then two observations with the same level in the categorical predictor will likely get different values in that column, meaning that the information that they are in the same "group" is lost. Alternatively, if dummy encoding of the categorical predictors is used, then the properties of the dummy encoding, binary values with only one or none 1's in each row, will also likely be lost. Hence, the model will have no way of knowing the structure of the categorical predictors.

3 Methods

While Section 2 introduced existing theory that is needed to discuss relative variable importance, the current section represents my contribution to the topic. First, a proof of the proper decomposition criteria of the LMG method is shown before an extension of the LMG method to random intercept models will be presented. Then, the main result of this thesis will be introduced, an extension of the relative weights method that works for random intercept models that will make the calculations computationally viable in real models.

3.1 Proof of proper decomposition

This section will give a simple proof that the LMG method will always give a proper decomposition of the R^2 of the model it is applied to. It is a simple result coming from the fact that most of the R^2 terms in equation (7) will cancel.

The same notation used in Section 2.5.2 will be used in this proof. Let $P = \{\pi_1, \pi_2, \ldots, \pi_{p!}\}$ be the set of all permutations of $(1, 2, \ldots, p)$, and $S_i(\pi_h)$ be the set of predictors appearing before i in π_h. The LMG method giving a proper decomposition means that

$$\begin{aligned} R^2(\{1,2,\ldots,p\}) &= \sum_{i=1}^{p} RI(i) \\ &= \sum_{i=1}^{p} \frac{1}{p!} \sum_{\pi \in P} \left(R^2\left(S_i(\pi) \cup i\right) - R^2(S_i(\pi)) \right) \\ &= \frac{1}{p!} \sum_{\pi \in P} \sum_{i=1}^{p} \left(R^2\left(S_i(\pi) \cup i\right) - R^2(S_i(\pi)) \right) \ , \end{aligned}$$

where the second equality simply comes from inserting equation (7). The order of the sums can be changed as they are finite.

Let $\pi = (h_1, h_2, \ldots, h_p)$ be some permutation of $(1, 2, \ldots, p)$ and let

$$d(\pi, i) = R^2\left(S_i(\pi) \cup i\right) - R^2(S_i(\pi)) \ ,$$

where $S_i(\pi)$ is the predictors appearing before i in π. It is then possible to prove that

$$\sum_{i=1}^{p} d(\pi, i) = R^2(\{1, 2, \ldots, p\}) \ .$$

To see this consider $d(\pi, h_1) + d(\pi, h_2)$ which simplifies to

$$\begin{aligned} d(\pi, h_1) + d(\pi, h_2) &= R^2\left(S_{h_1}(\pi) \cup h_1\right) - R^2(S_{h_1}(\pi)) + R^2\left(S_{h_2}(\pi) \cup h_2\right) - R^2(S_{h_2}(\pi)) \\ &= R^2\left(\{\} \cup h_1\right) - R^2\left(\{\}\right) + R^2\left(\{h_1\} \cup h_2\right) - R^2\left(\{h_1\}\right) \\ &= R^2\left(\{h_1, h_2\}\right) - R^2\left(\{\}\right) \ . \end{aligned}$$

Thus $d(\pi, h_1) + d(\pi, h_2) + d(\pi, h_3)$ further simplifies to

$$\begin{aligned} d(\pi, h_1) + d(\pi, h_2) + d(\pi, h_3) &= R^2\left(\{h_1, h_2\}\right) - R^2\left(\{\}\right) + R^2\left(S_{h_3}(\pi) \cup h_3\right) - R^2(S_{h_3}(\pi)) \\ &= R^2\left(\{h_1, h_2\}\right) - R^2\left(\{\}\right) + R^2\left(\{h_1, h_2\} \cup h_3\right) - R^2\left(\{h_1, h_2\}\right) \\ &= R^2\left(\{h_1, h_2, h_3\}\right) - R^2\left(\{\}\right) \ . \end{aligned}$$

This is simple to extend to

$$\sum_{i=1}^{p} d(\pi, i) = R^2(\{h_1, h_2, \dots, h_p\}) - R^2\left(\{\}\right) = R^2(\{1, 2, \dots, p\})$$

since $R^2\left(\{\}\right) = 0$.

Further, since $\sum\limits_{i=1}^{p} d(\pi, h_i) = \sum\limits_{i=1}^{p} d(\pi, i)$,

$$\begin{aligned}
\sum_{i=1}^{p} RI(i) &= \frac{1}{p!} \sum_{\pi \in P} \sum_{i=1}^{p} \left(R^2\left(S_i(\pi) \cup i\right) - R^2(S_i(\pi))\right) \\
&= \frac{1}{p!} \sum_{\pi \in P} R^2(\{1, 2, \dots, p\}) \\
&= R^2(\{1, 2, \dots, p\}) \frac{1}{p!} p! \\
&= R^2(\{1, 2, \dots, p\}) \ ,
\end{aligned}$$

which is what we wanted to show.

This proof is not specific to the use of R^2 in equation (7), it actually holds for any deterministic function, F, satisfying $F(\{\}) = 0$. This means that the definition of R^2 does not matter, as long as $R^2(\{\}) = 0$.

3.2 Extending the LMG method

In our proposed extension of the LMG method to random intercept models, the same approach of permuting the predictors that is described in Section 2.5.2 and shown in equation (8) can be used. The random intercepts can be treated the same as categorical fixed effects, that is, the relative importance given to a random intercept is the mean increase of model R^2 when the random intercept is added to the model. The only difference with a random intercept compared to a categorical predictor is that R^2 needs to be extended to work for random intercept models. Such an extension has been proposed by Nakagawa and Schielzeth (2013) and Johnson (2014), who showed that a meaningful R^2 extension can be calculated for random intercept models, shown in equation (6). The extension works by decomposing the variance of the model into the variance of the fixed effects, random intercept and residuals. Section 2.3.2 gives more details regarding the extension of R^2 to random intercept models. The method which we propose as the *extended LMG (ELMG)* method is

$$RI(i) = \frac{1}{(p+q)!} \sum_{\substack{S \subseteq \{1,2,\dots,p, \\ p+1,\dots,p+q\} \setminus i}} |S|!\,((p+q) - |S| - 1)! \left(R^2\left(S \cup i\right) - R^2(S)\right) \ , \qquad (10)$$

where there are p fixed effects and q random intercepts. Notice that equation (10) is identical to equation (8) other than the fact that there are more predictors.

A potential challenge with this extension of the LMG method comes from the extension of the R^2 to random intercept models. For linear regression models it is known that if $S_1, S_2 \subseteq \{1, 2, \dots, p\}$

with $S_1 \subset S_2$ then $R^2(S_1) \leq R^2(S_2)$, see *e.g.*, Fahrmeir et al. (2013, Chapter 3.2.3). This property is not guaranteed for the extension of R^2 to random intercept models by Nakagawa and Schielzeth (2013) and Johnson (2014) however. Thus, it is theoretically possible that if $S_1, S_2 \subseteq \{1, 2, \dots, p, p+1, \dots, p+q\}$ with $S_1 \subset S_2$ then $R^2(S_1) > R^2(S_2)$, which can cause the non-negativity and inclusion criteria to be violated. This possibility will be explored further in a simulation study in Section 4. When discussing the criteria listed in Section 2.4 with regards to the ELMG method we will, for now, assume that it is true that if $S_1, S_2 \subseteq \{1, 2, \dots, p, p+1, \dots, p+q\}$ with $S_1 \subset S_2$ then $R^2(S_1) \leq R^2(S_2)$.

- **Proper decomposition**: This method will always give a proper decomposition, as the R^2 terms in the sums will cancel out, see Section 3.1 for details.
- **Non-negativity**: The definition of conditional R^2, equation (6), is a fraction where both the numerator and denominator are sums of non-negative values, hence it is always non-negative. Additionally, $|S|!\,(p-|S|-1)!$ is non-negative. This means that each term of equation (10) is $|S|!\,(p-|S|-1)!$, which is non-negative, multiplied with $R^2(S \cup i) - R^2(S)$, which is assumed non-negative, giving a non-negative result. Hence, equation (10) is a sum of non-negative elements, giving a non-negative result.
- **Inclusion**: The definition of the inclusion criterion is:
  "A regressor X_j with $\beta_j \neq 0$ should receive a nonzero share.".
  This definition needs to be modified slightly to make it fit with random intercept models as only the fixed effects have coefficients. The equivalent statement for a random intercept is that its variance is larger that zero. The following definition will therefore be used instead: "A fixed-effect regressor X_j with $\beta_j \neq 0$ or a random intercept regressor with estimated variance > 0 should receive a nonzero share.".

  The fixed effect part follows from the same argument as in Section 2.5.2. Let $h \in \{1, 2, \dots, p\}$ be a fixed effect. If $\hat{\beta}_h \neq 0$ and both $\mathbf{y}$ and $\mathbf{x}_h$ have a non-zero variance, then

  $$R^2(\{h\}) - R^2(\{\}) = \frac{\hat{\beta}_h^2 \widehat{\mathrm{Var}}\,(\mathbf{x_h})}{\widehat{\mathrm{Var}}\,(\mathbf{y})} - 0 > 0\ .$$

  Since $R^2(\{h\}) - R^2(\{\})$ corresponds to $R^2\,(S \cup i) - R^2(S)$ in equation (10) when $S = \{\}$ and $i = h$, the above inequality, combined with the assumption that none of the terms of the sum in equation (10) can be negative, means that $RI(h) > 0$.

  The random intercept part follows from a similar argument. If the variance of a random intercept x_j is non-zero, then $R^2(\{j\}) > 0$, since the numerator of R^2 is the sum of $\sigma_f^2 >= 0$ and the variances of the random intercepts, which is assumed larger than zero. The denominator is also larger than zero, since it is a sum of variances. Combined, this means that $R^2(\{i\}) > 0$, which means that one of the terms of $RI(i)$ will be $(p-1)!\left(R^2(\{i\}) - R^2(\{\})\right) = (p-1)!R^2(\{i\}) > 0$. Using the assumption that each term of equation (10) is in practice non-negative, this means that $RI(i) > 0$.

To calculate the relative variable importance of one predictor, 2^{p+q-1} subsets have to be considered. This means that to calculate the relative importance of all $p+q$ predictors, $(p+q)\,2^{p+q-1}$ subsets have to be considered. The computational complexity therefore increases very quickly and can quickly make this method difficult to use when the number of predictors increases. The next section proposes a solution to this problem.

3.3 Extending the relative weights method

The previous subsection described an extension of the LMG method to random intercept models. It works and gives results that make sense and mostly satisfy the criteria Grömping (2015) outlined and which are listed in Section 2.4. The ELMG method has the drawback of being computationally expensive, which makes it unviable for models with many predictors. Here, we therefore propose an extension of the relative weights method as an alternative. The extended relative weights method will calculate an approximation of the relative variable importance given by the ELMG method in the same way the standard relative weights method in Section 2.5.3 calculated an approximation of the relative variable importance given by the LMG method in Section 2.5.2.

First, a simplified method with a problem will be introduced before the more correct method without the problem will be shown. The reason for starting with the simplified method is that the simplified method is useful to show the intuition behind the more correct method.

The main idea is to first use the same techniques as in the relative weights method to get uncorrelated fixed effects. Then, use the same approach as in the LMG method, that is, consider the mean increase in R^2 when a predictor is added to the model. Here, however, the fixed effects are treated as one "block" which are either all in the model or not in the model. This approach gives a share to each random intercept as well as a joint share to the fixed effects. The joint share given to the fixed effects can then be distributed to each individual fixed effect by using their coefficients in the model along with the fact that they are uncorrelated, as in the relative weights method.

The first step is therefore to create $\mathbf{Z} = \sqrt{n-1}\mathbf{P}\mathbf{Q}^T$ by using $\mathbf{X} = \mathbf{P}\mathbf{D}\mathbf{Q}^T$, which contains the observed fixed effects. Recall that the random intercepts are not in $\mathbf{X}$ but are contained in a separate matrix, $\mathbf{R}$, see equation (3). The columns of $\mathbf{Z}$ creates new fixed effects which are pairwise uncorrelated. Having uncorrelated columns means that, if we are able to give a joint relative importance to the fixed effects, it is possible to distribute that in a meaningful and efficient way to each fixed effect. This distribution can be done using the same techniques as in the relative weights method shown in Section 2.5.3.

To get the joint relative importance of the fixed effects along with the relative importance of the random intercepts it is possible to use the same approach of looking at the average increase in R^2 as in the LMG method

$$RI(i) = \frac{1}{(1+q)!} \sum_{S \subseteq \{f, p+1, \ldots, p+q\} \setminus i} |S|! \left((1+q) - |S| - 1\right)! \left(R^2\left(S \cup i\right) - R^2(S)\right) ,$$

$\forall i \in \{f, p+1, \ldots, p+1\}$. The difference here is that the transformed fixed effects, f, will be considered as one "block", where they are either all in the model or not. Considering the fixed effects as a "block" means that many fewer subsets need to be considered. Without considering the fixed effects as a block, 2^{p+q-1} subsets need to be considered for each predictor, while only 2^{1+q-1} need to be considered when the fixed effects are considered as a block. Since the number of fixed effects in the model is usually larger than the number of random intercepts, *i.e.*, $p > q$, the computational complexity is dramatically decreased.

The shares given to the random intercepts, *i.e.*, $RI(p+1), RI(p+2), \ldots, RI(p+q)$, can be used as they are, but the joint share given to the block of fixed effects, $RI(f)$, must still be distributed to each individual effect. To distribute $RI(f)$ to each individual fixed effect, fit the full model, *i.e.*, a model with the columns of $\mathbf{Z}$ along with all the random intercepts. The model gives the columns of $\mathbf{Z}$ coefficients, $\boldsymbol{\beta}_{\mathbf{Z}}$, which can be used to distribute $RI(f)$ to the original fixed effects, by creating $\boldsymbol{\Lambda} = \frac{1}{\sqrt{n-1}}\mathbf{QDQ}^T$ and then

$$\mathbf{r} = \boldsymbol{\Lambda}^{[2]}\hat{\boldsymbol{\beta}}_{\mathbf{Z}}{}^{[2]} \ .$$

The values in $\mathbf{r}$ give importances to the fixed effects. These importances, however, are only valid if the fixed effects are all uncorrelated with the random intercepts. Correlation between the fixed effects and the random intercepts can cause two problems: First, the decomposition might violate the proper decomposition criterion, since, if a fixed effect is correlated with a random intercept, their coefficient might not be representative of their importance. This causes the same problems as described in Section 2.5.1. And second, if fixed effects are correlated with random intercepts, their relative importance will depend on which random intercepts are in the model. For example, if Z_g and R_h, a fixed effect and random intercept respectively, contain some of the same information, then Z_g will have a lower importance compared to the other fixed effects when R_h is also in the model compared to when R_h is not in the model, since R_h would absorb some of the information in Z_g. This absorption would therefore give Z_g a smaller coefficient when R_h is in the model and a larger coefficient when R_h is not in the model. The method described above, however, will only look at the coefficients of the full model, and will therefore not take such relationships into account and give results that are intuitively wrong.

The first problem is simple to solve, $\mathbf{r}$ just needs to be scaled such that it sums to $RI(f)$, *i.e.*, create

$$\mathbf{r}^* = \frac{\mathbf{r}}{\sum\limits_{i=1}^{p} r_i} RI(f) \ .$$

The scaling ensures that the proper decomposition criteria is satisfied, since

$$RI(f), RI(p+1), RI(p+2), \ldots, RI(p+q)$$

is a proper decomposition, because the decompositions are created using the LMG method which means that the proof in Section 3.1 is valid. Thus,

$$r_1^*, r_2^*, \ldots, r_p^*, RI(p+1), RI(p+2), \ldots, RI(p+q) \ ,$$

where $\mathbf{r^*} = \left(r_1^*, r_2^*, \ldots, r_p^*\right)$, is also a proper decomposition, since $\sum\limits_{i=1}^{p} r_i^* = RI(f)$.

Solving the second problem, however, is more challenging. Currently, the method first looks at the increase in R^2 across all subsets of the random intercepts when adding the fixed effects to the model and then distributes this increase to each fixed effect using the coefficients of the fixed effects in the full model. A better approach is to instead look at each subset by itself and find the increase in R^2 when adding the fixed effects to this subset. The respective increase can then be distributed to each fixed effect using the coefficients of the fixed effects in this model. More precisely, for a subset of the random intercepts, S, fit the model with the random intercepts in S together with the fixed effects in $\mathbf{Z}$. Let $\hat{\boldsymbol{\beta}}_{\mathbf{Z}}(S)$ be the coefficients of the fixed effects in the respective model. Then, the relative contribution of each of the original fixed effects to this model can be calculated by

$$\mathbf{r}(S) = \boldsymbol{\Lambda}^{[2]}\hat{\boldsymbol{\beta}}_{\mathbf{Z}}(S)^{[2]} \ ,$$

where $\mathbf{\Lambda} = \frac{1}{\sqrt{n-1}}\mathbf{QDQ}^T$ as earlier. We still need to scale $\mathbf{r}(S)$, however, to ensure that the relative contribution of each of the fixed effects sum to the increase in R^2 for this subset such that a proper decomposition is created, for the same reason as explained earlier,

$$\mathbf{R}(S) = \frac{\mathbf{r}(S)}{\sum\limits_{i=1}^{p} r(S)_i} \left(R^2\left(S \cup f\right) - R^2\left(S\right)\right) \ .$$

Finally, the relative importance of each fixed effect is

$$(RI(1), RI(2), \ldots, RI(p)) = \frac{1}{(1+q)!} \sum_{S \subseteq \{p+1,\ldots,p+q\}} |S|!\left((1+q) - |S| - 1\right)! \mathbf{R}(S) \ . \tag{11}$$

The resulting method is what we propose as the *extended relative weights (ERW)* method. A compact description of the method is shown in Algorithm 1.

To calculate the relative variable importance of one predictor using the ERW method, $2^{1+q-1} = 2^q$ subsets need to be considered. To calculate the importance of all $p+q$ predictors, $(1+q)\,2^q$ subsets need to be considered, as the relative variable importance of the p fixed effects are calculated at the same time. Thus, the complexity of the ERW method scales only with the number of random intercepts in the model, not the number of fixed effects. The computational complexity scaling with only the number of random intercepts is a significant improvement compared to the $(p+q)\,2^{p+q-1}$ subsets that need to be considered when using the ELMG method, since p, the number of fixed effects, will generally be large compared to q, the number of random intercepts.

When considering whether the ERW method satisfies the criteria listed in Section 2.4 the same problem as for the ELMG method where there is no guarantee that $R^2(S_1) \leq R^2(S_2)$ when $S_1, S_2 \subseteq \{1, 2, \ldots, p, p+1, \ldots, p+q\}$ with $S_1 \subset S_2$ occurs. We will again assume that $R^2(S_1) \leq R^2(S_2)$ when discussing the criteria listed in Section 2.4 before discussing the problem more in Section 4.

- **Proper decomposition**: The proposed method will give a proper decompositions because for each subset S, then

  $$\sum_{i=1}^{p} R\left(S\right)_i = R^2\left(S \cup f\right) - R^2\left(S\right) \ .$$

  This means that

  $$\begin{aligned}\sum_{i=1}^{p} RI(i) &= \sum_{i=1}^{p} \sum_{S \subseteq \{p+1,\ldots,p+q\}} |S|!\left((1+q) - |S| - 1\right)! R\left(S\right)_i \\ &= \sum_{S \subseteq \{p+1,\ldots,p+q\}} |S|!\left((1+q) - |S| - 1\right)! \sum_{i=1}^{p} R\left(S\right)_i \\ &= \sum_{S \subseteq \{p+1,\ldots,p+q\}} |S|!\left((1+q) - |S| - 1\right)! R^2\left(S \cup f\right) - R^2\left(S\right) \ .\end{aligned}$$

Algorithm 1 Algorithm describing the calculations of the relative variable importance of each predictor when using the extended relative weights method.

```
# Preparation
$\mathbf{Z} \leftarrow \sqrt{n-1}\mathbf{PQ}^T$
$\boldsymbol{\Lambda} \leftarrow \frac{1}{\sqrt{n-1}}\mathbf{QDQ}^T$

# Relative importance of random intercepts
for $i \in \{p+1, p+2, \dots, p+q\}$ do
    $RI(i) \leftarrow \frac{1}{(1+q)!} \sum\limits_{S \subseteq \{f, p+1, \dots, p+q\} \setminus i} |S|!\,((1+q) - |S| - 1)!\,\left(R^2\left(S \cup i\right) - R^2(S)\right)$
end for

# Relative importance of fixed effects
$(RI(1), RI(2), \dots, RI(p)) \leftarrow (0, 0, \dots, 0)$
for $S \subseteq \{p+1, p+2, \dots, p+q\}$ do
    $m \leftarrow$ fitted model with the fixed effects in $\mathbf{Z}$ and the random intercepts in $S$
    $\hat{\boldsymbol{\beta}}_{\mathbf{Z}}(S) \leftarrow$ coefficients for fixed effects in $m$
    $\mathbf{r}(S) \leftarrow \boldsymbol{\Lambda}^{[2]}\hat{\boldsymbol{\beta}}_{\mathbf{Z}}^{[2]}$
    $\mathbf{R}(S) \leftarrow \frac{\mathbf{r}(S)}{\sum\limits_{i=1}^{p} r(S)_i} \left(R^2\left(S \cup f\right) - R^2\left(S\right)\right)$
    $(RI(1), RI(2), \dots, RI(p)) \leftarrow (RI(1), RI(2), \dots, RI(p)) + |S|!\,((1+q) - |S| - 1)!\,\mathbf{R}(S)$
end for
$(RI(1), RI(2), \dots, RI(p)) \leftarrow \frac{1}{(1+q)!}\left(RI(1), RI(2), \dots, RI(p)\right)$
```

Thus,

$$\sum_{i=1}^{p+q} RI(i) = \sum_{i \in \{f, p+1, \dots, p+q\}} \; \sum_{S \subseteq \{f, p+1, \dots, p+q\} \setminus i} |S|!\,((1+q) - |S| - 1)! R^2\left(S \cup i\right) - R^2\left(S\right) \ ,$$

which means that the proof in Section 3.1 can be used.

- **Non-negativity**: We know from Section 2.5.3 that

$$\mathbf{r}(S) = \boldsymbol{\Lambda}^{[2]}\hat{\boldsymbol{\beta}}_{\mathbf{Z}}(S)^{[2]} \ ,$$

will be non-negative, since both $\boldsymbol{\Lambda}^[2]$ and $\hat{\boldsymbol{\beta}}_{\mathbf{Z}}^{[2]}$ only have non-negative values. Further, since $R^2\left(S \cup f\right) - R^2(S)$ is assumed to be non-negative,

$$\mathbf{R}(S) = \frac{\mathbf{r}(S)}{\sum\limits_{i=1}^{p} r(S)_i} \left(R^2\left(S \cup f\right) - R^2\left(S\right)\right) \ ,$$

is non-negative. Finally, each element of

$$(RI(1), RI(2), \dots, RI(p)) = \frac{1}{(1+q)!} \sum_{S \subseteq \{p+1, \dots, p+q\}} |S|!\,((1+q) - |S| - 1)!\mathbf{R}(S)$$

will be a sum of non-negative values, which means that it is also non-negative.

The values $RI(p+1), RI(p+2), \dots, RI(p+q)$ will be non-negative, since they are created with the same approach as in the ELMG method.

- **Inclusion**: We will again use the modified definition of the inclusion criteria used in Section 3.2 which takes random intercept models into account, *i.e.*, "A fixed-effect regressor X_j with $\beta_j \neq 0$ or a random intercept regressor with estimated variance > 0 should receive a nonzero share.".

  First we will consider fixed effects, *i.e.*, let the predictor X_j have its coefficient $\beta_j \neq 0$. This assumption means that for $S = \{p+1, p+2, \ldots, p+q\}$, *i.e.*, S includes all the random intercepts, we can use the fact that the standard relative weights method satisfies the inclusion criteria to get that $\mathbf{r}\left(S\right)_j > 0$. Further, $\mathbf{R}\left(S\right)$ is simply a rescaling of $\mathbf{r}\left(S\right)$ which means that $\mathbf{R}\left(S\right)_j > 0$. Thus, $|S|!\left((1+q) - |S| - 1\right)!\mathbf{R}(S)_j > 0$ and since all the other terms of the sum in equation (11) are assumed to be non-negative we get that $RI(j) > 0$.

  Inclusion is satisfied for random intercepts because $RI(p+1), RI(p+2), \ldots, RI(p+q)$ are calculated using the same approach as in the LMG method, thus the same argument as in Section 2.5.2 can be applied.

In addition to being able to handle random intercepts, the ERW method will also work for categorical predictors, in contrast to the normal relative weights method. To get relative importances for categorical predictors simply treat them the same as the random intercepts. Transform the numerical predictors and then use the LMG approach to give a relative importance to each categorical predictor as well as a joint share to the numerical predictors. The joint share can then be distributed to each numerical predictor as described above.

Note that the ERW method is not a perfect approximation of the LMG method, as will be discussed in Section 4 and Section 6, meaning that the ELMG method should still be preferred for models with relatively few predictors. A comparison between the time needed to run the methods will be shown in Section 5.

3.4 An illustrative example

To make the novel method more accessible to the reader, consider a simple case where there are two fixed effects, X_1, X_2, and a random intercept, R_1. To calculate the relative variable importance of the predictors using the ERW method the steps will be as follows.

First, create new fixed effects which are uncorrelated, Z_1 and Z_2. Since the fixed effects are to be kept together in one "block", there are four possible sets of predictors: $\{\}$, $\{Z_1, Z_2\}$, $\{R_1\}$ and $\{Z_1, Z_2, R_1\}$. The relative variable importance given to R_1 will be the mean increase in R^2 when R_1 is added to model, *i.e.*,

$$RI(R_1) = 0.5\left(\left(R^2\left(\{R_1\}\right) - R^2\left(\{\}\right)\right) + \left(R^2\left(\{Z_1, Z_2, R_1\}\right) - R^2\left(\{Z_1, Z_2\}\right)\right)\right) \ .$$

To find the relative importance of the fixed effects, we need to consider the two subsets without the fixed effects, *i.e.*, $\{\}$ and $\{R_1\}$. Then, for each of these subsets, find the coefficients of Z_1 and Z_2 when they are added to the model and use their squared values to give an importance to X_1 and X_2, denoted as $\left(r_1\left(S\right), r_2\left(S\right)\right)$, where S is the respective subset. These importances then need to be scaled to the increase in R^2 when Z_1 and Z_2 are added to the model such that we get a proper decomposition, *i.e.*,

$$\mathbf{R}\left(S\right) = \frac{\left(r_1\left(S\right), r_2\left(S\right)\right)}{r_1\left(S\right) + r_2\left(S\right)}\left[R^2\left(S \cup \{Z_1, Z_2\}\right) - R^2\left(S\right)\right] \ .$$

Predictor	ELMG	ERW
$RI(R_1)$	$\{\}$, $\{X_1\}$, $\{X_2\}$, $\{X_1, X_2\}$	$\{\}$, $\{X_1, X_2\}$
$RI(X_1)$	$\{\}$, $\{X_2\}$, $\{R_1\}$, $\{X_2, R_1\}$	$\{\}$, $\{R_1\}$
$RI(X_2)$	$\{\}$, $\{X_1\}$, $\{R_1\}$, $\{X_1, R_1\}$	
Total subsets	12	4

Table 2: Comparison of the the number of subsets that need to be considered for the ELMG method and the ERW method. Here there are two fixed effects, X_1 and X_2, and one random intercept, R_1.

Finally, the share given to the fixed effects is the mean of the $\mathbf{R}$'s

$$(RI(X_1), RI(X_2)) = 0.5\left(\mathbf{R}\left(\{\}\right) + \mathbf{R}\left(\{R_1\}\right)\right) \ .$$

Table 2 compares the number of subsets that need to be considered in this example for the ELMG method and the ERW method. We see that there are many more subsets that need to be considered for the ELMG method. The difference increases quickly when the number of predictors in the model increases.

4 Simulation study

We will do a simulation study to explore the properties of the ELMG and ERW methods. In the simulation study we will compare the shares given to the predictors in random intercept models, where we will vary the coefficients and the correlation structure of the fixed effects. The data will therefore be of the form

$$\mathbf{y} = \beta_0 + \mathbf{X}\boldsymbol{\beta} + \mathbf{R}\mathbf{u} + \boldsymbol{\epsilon} \ ,$$

where $\mathbf{X}$ is a $n \times p$ design matrix containing the fixed effects, $\boldsymbol{\beta}$ is a p vector containing the coefficients of the fixed effects, $\mathbf{R} = \begin{bmatrix}\mathbf{R}_1 & \mathbf{R}_2 & \dots & \mathbf{R}_m\end{bmatrix}$ is a $n \times (l_1 + l_2 + \cdots + l_m)$ matrix, where $\mathbf{R}_i$ is a $n \times l_i$ matrix containing the dummy encoding of the clusters of random intercept i. Further, $\mathbf{u}$ is a $(l_1 + l_2 + \cdots + l_m)$ vector containing the coefficients for each level of random intercepts with distribution

$$\mathbf{u} \sim N\left(\mathbf{0}, \begin{bmatrix} \sigma_1^2\mathbf{I}_{l_1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \sigma_2^2\mathbf{I}_{l_2} & \dots & \mathbf{0} \\ \vdots & & \ddots & \\ \mathbf{0} & \mathbf{0} & \dots & \sigma_m^2\mathbf{I}_{l_m} \end{bmatrix}\right) ,$$

where $\sigma_1, \sigma_2, \dots, \sigma_m$ are the variances of the m random intercepts and $\boldsymbol{\epsilon}$ a n vector which is the error term with distribution $\boldsymbol{\epsilon} \sim N\left(\mathbf{0}, \sigma_\epsilon^2\mathbf{I}_n\right)$. Details regarding how the data was technically sampled can be found in Appendix C.

If all the predictors in the model, both fixed effects and random intercepts, are pairwise independent of each other, it is possible to calculate the theoretical correct decomposition of the model R^2 by using that the variance of y will be

$$\operatorname{Var}(y) = \sum_{i=1}^{p} \beta_i^2 + \sum_{j=1}^{q} \sigma_j^2 + \sigma_\epsilon^2 \ .$$

Then, the proportion of variance in the response explained by each fixed effect is

$$\frac{\beta_i^2}{\operatorname{Var}(y)}$$

for $i \in \{1, 2, \dots, p\}$ and the proportion explained by each random intercept is

$$\frac{\sigma_j^2}{\operatorname{Var}(y)}$$

for $j \in \{1, 2, \dots, q\}$. The proportions can be calculated using the parameters used to simulate the data. The theoretical correct decompositions can be used to examine the performance of the ELMG and ERW methods.

The shares given by the ELMG and ERW methods are compared using violin plots, which can be thought of as estimated density curves which are rotated such that they are vertical and then mirrored over their vertical axis. The width of the "violin" shows how many observations were observed in the corresponding area. The three horizontal lines inside each "violin" show, from top to bottom, the 90%, 50% and 10% quantile respectively.

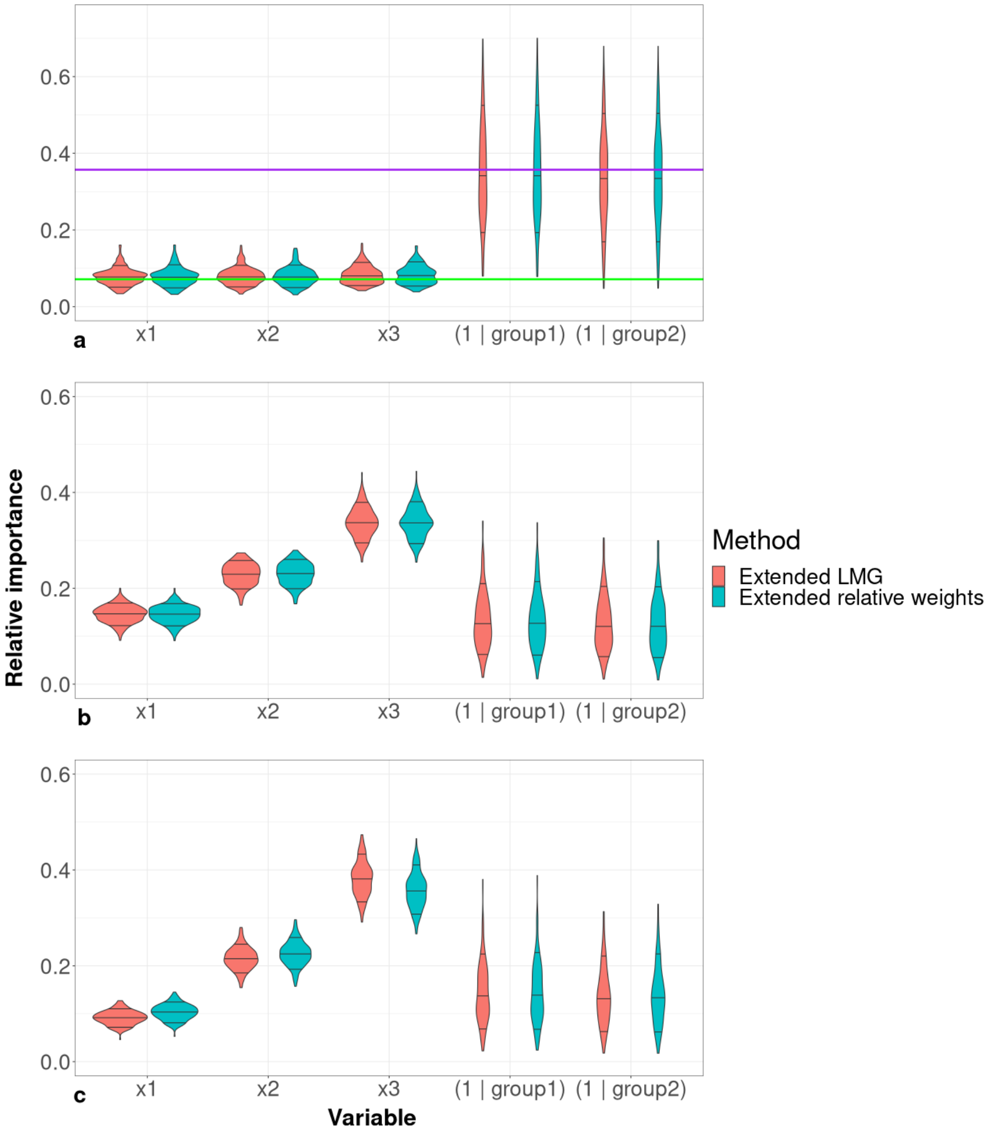

Figure 1: Violin plots showing decompositions of the ELMG method and the ERW method based on 200 simulations. The predictors x_1, x_2 and x_3 are fixed effects while group1 and group2 are random intercepts. The horizontal lines inside each "violin" show, from top to bottom, the 90%, 50% and 10% quantile respectively. The random intercepts have variance $\sigma_1^2 = \sigma_2^2 = 5$. a. Fixed effect have coefficients $\boldsymbol{\beta} = (1, 1, 1)^T$ and are pairwise independent. b. Fixed effect have coefficients $\boldsymbol{\beta} = (1, 2, 3)^T$ and covariances $\text{Cov}\,(x_1, x_2) = \text{Cov}\,(x_1, x_3) = \text{Cov}\,(x_2, x_3) = 0.5$. c. Fixed effect have coefficients $\boldsymbol{\beta} = (1, 2, 3)^T$ and covariances $\text{Cov}\,(x_1, x_2) = -0.2$ and $\text{Cov}\,(x_1, x_3) = \text{Cov}\,(x_2, x_3) = 0.5$. The green and purple horizontal line in a. shows the theoretically correct share for the fixed effects and random intercepts respectively.

Figure 1 shows violin plots representing the distribution of the decompositions of three fixed effects and two random intercepts in 200 simulated datasets. They differ in the coefficients and covariances of the fixed effects. In Figure 1a the fixed effects have coefficients $\boldsymbol{\beta} = (1, 1, 1)^T$ and are all pairwise independent, *i.e.*, $\text{Cov}(x_1, x_2) = \text{Cov}(x_1, x_3) = \text{Cov}(x_2, x_3) = 0$. In Figures 1b and c the fixed effects have coefficients $\boldsymbol{\beta} = (1, 2, 3)^T$ and the covariances in Figure 1b are $\text{Cov}(x_1, x_2) = \text{Cov}(x_1, x_3) = \text{Cov}(x_2, x_3) = 0.5$ while the covariances in Figure 1c are $\text{Cov}(x_1, x_2) = -0.2$ and $\text{Cov}(x_1, x_3) = \text{Cov}(x_2, x_3) = 0.5$. Since all the predictors are pairwise independent in Figure 1a the theoretical contributions of each predictor are shown as horizontal lines. Figures 1b and c do not have these lines, since the theoretical contributions are challenging to calculate when the coefficients of the fixed effects vary and the fixed effects are not independent.

We can see that the median of the decompositions on the simulated data of the fixed effects are slightly larger than the theoretical proportions explained by the fixed effects and shown by the horizontal green line (Figure 1a). Similarly, the median of the decompositions on the simulated data of the random intercepts are slightly smaller than the theoretical proportions explained by the random and shown by the horizontal purple line.

Generally, the "violins" of the ELMG and ERW methods are very similar indicating that the ERW method is a good computationally efficient approximation of the ELMG method. There are some differences, however, especially in Figures 1b and c. The two figures are created from simulated data using the same parameters except for the covariance between X_1 and X_2, in Figure 1b $\text{Cov}(x_1, x_2) = 0.5$ while in Figure 1c $\text{Cov}(x_1, x_2) = -0.2$. We see that in Figure 1b the ELMG method and the ERW method gives almost identical decompositions. In Figure 1c, however, the ERW method gives slightly larger shares to x_1 and x_2 than the ELMG method, while the ERW method gives a slightly larger share to x_3 than the ELMG method does. This shows that small changes in the covariance structure of the fixed effects can influence how similar the two methods are.

To further explore how changes in the covariance structure of the fixed effects influence the similarity between the ELMG method and the ERW method we calculated the difference between the ELMG method and the ERW method when the covariance between x_1 and x_2 varies (Figure 2). The data is simulated the same way as in Figure 1. There are three fixed effects and two random intercepts, but only the shares given to the fixed effects are shown in Figure 2, as the shares given to the random intercepts by the ELMG and ERW methods were almost identical.

The difference between the decompositions given by the two methods varies with $\text{Cov}(x_1, x_2)$. In the left column, where $\text{Cov}(x_2, x_3) = 0.5$, the two methods seem to give identical results for $\text{Cov}(x_1, x_2) \approx 0.5$. Otherwise, the difference between the two methods increases when $\text{Cov}(x_1, x_2)$ goes further away from 0.5. In the right column, where $\text{Cov}(x_2, x_3) = -0.2$, the relationship between the two methods seems to vary for the three fixed effects. For x_1, the two methods meet at $\text{Cov}(x_1, x_2) \approx 0$ and $\text{Cov}(x_1, x_2) \approx -0.75$, with the difference increasing as the distance from those two points grow. When looking at x_2, the two methods seem to agree only at $\text{Cov}(x_1, x_2) \approx 0$ while for x_3, the two methods meet at $\text{Cov}(x_1, x_2) \approx -0.4$. The largest difference between the two methods seems to be for x_1 when $\text{Cov}(x_1, x_2) = 0.7$, where the difference is ~ 0.06.

The lines have 95% confidence intervals around them, but these intervals are very tight, it's clearest in the lower right panel for $\text{Cov}(x_1, x_2) = -0.4$. This shows that the shares given to the fixed effects are robust with regard to small changes in the data.

The difference between the theoretical proportion of y's variance explained by the predictors and

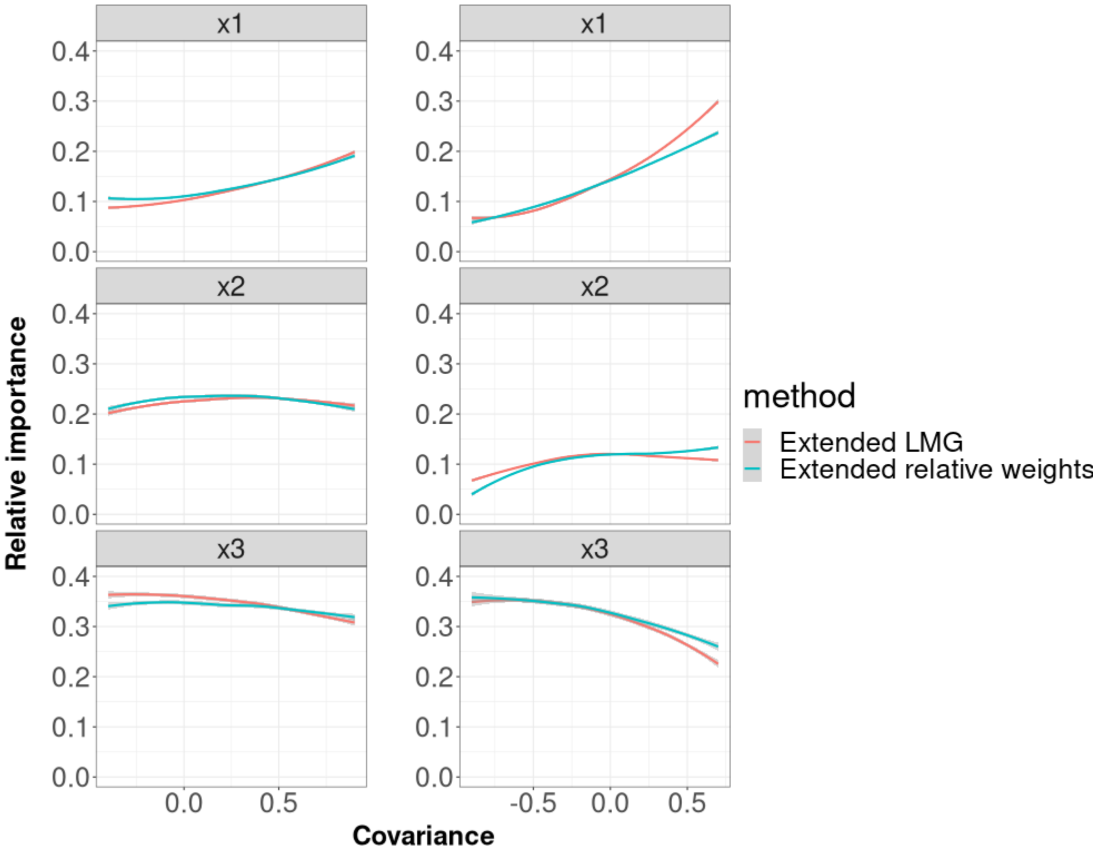

Figure 2: Difference between the ELMG method and the ERW method when the covariance between x_1 and x_2 varies. $\mathrm{Cov}\,(x_1, x_3) = 0.5$. In the left column $\mathrm{Cov}\,(x_2, x_3) = 0.5$ while in the right column $\mathrm{Cov}\,(x_2, x_3) = -0.2$. In the left column the covariance between x_1 and x_2 varies between -0.4 and 0.9 while in the right column the covariance between x_1 and x_2 varies between -0.9 and 0.7. The reason for the difference is that a valid covariance matrix is needed. For each covariance value 50 simulations are done with the following parameters. The coefficients of the fixed effects are $\boldsymbol{\beta} = (1, 2, 3)^T$ and the fixed effects have unit variance. The random intercepts have variance $\sigma_1^2 = \sigma_2^2 = 5$. The lines shown are the mean of the 50 simulations, while the gray area around shows a 95% confidence interval.

	Proper decomposition	Non-negativity	Inclusion
The LMG method	X	X	X
The relative weights method	X	X	X
The ELMG method	X	X*	X*
The ERW method	X	X*	X*

Table 3: Overview of which of the three criteria listed in Section 2.4 the different methods considered in this thesis satisfy. The X* means that the method usually satisfies the criteria in practice, however the fact that it is possible, for random intercept models, for $R^2(S_1) > R^2(S_2)$ when $S_1 \subseteq S_2$ means that it is theoretically possible for the criteria to be violated.

the calculated decompositions in Figure 1a is due to LMMs imposing a shrinkage effect on the random intercepts. The shrinkage effect causes the estimated coefficients of each level of the random intercept to be closer to zero than they would otherwise be, which causes the estimated variance of each random intercept to be slightly smaller than the data suggests. Thus, the shares given to the random intercepts are slightly decreased. Since both the ELMG method and the ERW method satisfy the proper decomposition criteria, this causes the share given to the fixed effects to increase by the same amount. The shrinkage effect is a known property of LMMs, see *e.g.*, Clark and Linzer (2015) or Fahrmeir et al. (2013, Page 355). When decomposing the model R^2 there are two possible approaches regarding this shrinkage, try to explain the relationship of the data or try to explain the properties of the model. Since relative variable importance is meant as a tool to help interpret models, the choice was made to focus on explaining the properties of the model. The fact that the theoretical contribution of the predictors and the decompositions do not exactly match is therefore expected and desired. The difference between the theoretical contributions and the decompositions thus means that the shrinkage in LMMs is taken into account.

Simulations have shown that, for random intercept models, it is possible for the R^2 of a model to decrease when adding a predictor. This happened when data was simulated using the same parameters as in Figure 1b. To make it happen, first a model, $model_1$, was fitted with the three fixed effects and one of the random intercepts. Then, a second model, $model_2$, was fitted with the same random intercept as a categorical predictor in addition to the predictors in $model_1$, *i.e.*, the random intercept was in $model_2$ both as a random intercept and as a categorical predictor. For some simulations of the data, the R^2 of $model_1$ was larger than the R^2 of $model_2$. Thus, it is theoretically possible for the ELMG method and ERW method to violate the non-negativity and inclusion criteria, as the proofs that the criteria are valid rely on the fact that the model R^2 can not decrease when adding a predictor to the model. However, it is not likely for the criteria to be violated in practice, for two reasons. First, it is not likely that someone will add the same predictor as both a random intercept and a categorical predictor in the same model. Second, even if the same predictor was added as both a random intercept and as a categorical predictor, the R^2 decreasing only happens rarely. The R^2 only decreased in $\sim 5\%$ of the simulations. Taking the simulations into account, Table 3 shows which of the criteria outlined in Section 2.4 the LMG method, the relative weights method, the ELMG method and the ERW method satisfy.

5 Basketball example

Value	Description
player_name	Name of player (categorical)
age	Age of player (years) (numerical)
player_height	Height of player (cm) (numerical)
player_weight	Weight of player (kg) (numerical)
college	Name of the college the player attended (categorical)
country	Country player was born in (categorical)
draft_round	The draft round the player was picked (categorical)
draft_number	The number at which the player was picked in his draft round (categorical)
gp	Number of games played in the season (numerical)
pts	Average number of points scored in a game (numerical)
reb	Average number of rebounds grabbed in a game (numerical)
ast	Average number of assists in a game (numerical)
net_rating	Team's point differential per 100 possessions while the player is on the court (numerical)
oreb_pct	Percentage of available offensive rebounds the player grabbed while he was on the floor (numerical)
dreb_pct	Percentage of available defensive rebounds the player grabbed while he was on the floor (numerical)
usg_pct	Percentage of team plays used by the player while he was on the floor (numerical)
ts_pct	A shooting percentage that factors in the value of three-point field goals and free throws in addition to conventional two-point field goals (numerical)
ast_pct	Percentage of teammate field goals the player assisted while he was on the floor (numerical)
season	NBA season (categorical)

Table 4: Description of the variables in the example data set about basketball.

To illustrate the ERW method an example on real data will be shown. The data is about player-statistics in basketball, compiled by Cirtautas (2022) and sourced from NBA (2022). There are 11700 observations and 21 variables, described in Table 4. A model is fitted with the average number of points scored in a game, *pts*, as the response, with all other variables as predictors. There has been no attempt at creating a good model, as it is only for illustration purposes. The only transformation done on the data is to standardize and center all numerical variables, which is done to make it possible to compare standardized coefficients with other relative importance methods. All categorical predictors are added to the model as random intercepts.

Table 5 shows the coefficient, squared coefficient, p-value and relative importance from the ERW method of each numerical predictor in the created model. The p-values are difficult to calculate for random intercept models, as it is not obvious how many degrees of freedom to use for the t-values when testing the null-hypothesis that the coefficients are equal to 0. A conservative estimate of the degrees of freedom, however, can be acquired by taking the number of observations and then subtracting the number of estimated values in the model, *i.e.*, one coefficient for each numeric fixed effect, a variance estimate for each random intercept and one coefficient for each level of the random intercepts and categorical predictors. A random intercept model will generally have more degrees of freedom than this, since it will not "use" a degree of freedom for each level

	$\hat{\beta}$	$\hat{\beta}^2$	p	RI
usg_pct	0.309	0.096	< 0.001	0.146
ast_pct	-0.199	0.040	< 0.001	0.098
player_height	0.009	< 0.001	0.170	0.093
reb	0.513	0.263	< 0.001	0.041
ast	0.473	0.224	< 0.001	0.036
player_weight	0.001	< 0.001	0.441	0.028
ts_pct	0.074	0.006	< 0.001	0.024
dreb_pct	-0.145	0.021	< 0.001	0.013
oreb_pct	-0.115	0.013	< 0.001	0.011
net_rating	0.019	< 0.001	< 0.001	0.009
gp	0.051	0.003	< 0.001	0.004
age	-0.010	< 0.001	0.002	0.001

Table 5: Coefficient, squared coefficient, t-value, p-value and the relative importance from the ERW method of the numerical predictors in the created model of the basketball example data. The predictors are shown in decreasing order of the relative importances.

	$\hat{\sigma}$	RI
player_name	0.237	0.242
draft_number	0.048	0.051
country	0.017	0.048
draft_round	0.029	0.040
college	0.024	0.036
season	0.024	0.010

Table 6: Standard deviation and relative importance from the ERW method of the categorical predictors in the created model of the basketball example data. The predictors are shown in decreasing order of the relative importances.

of each random intercept, because the regularization assumption imposes extra structure on the model. The true p-values will therefore likely be slightly smaller than the ones shown in Table 5. Table 6 shows the standard deviation and relative importance from the ERW method of each of the random intercepts in the created model.

In Table 5 we can compare the information given by the coefficients in the model, the p-values and the relative variable importances given by the ERW method when considering the numerical predictors. Almost all of the predictors have a p-value which is less than 0.001, which is normal when there are many observations in a data-set, since the many observations causes the standard deviation of the coefficients to shrink, which causes the p-value to shrink. The squared coefficients seem to agree with the p-values in the sense that the three numerical predictors with the largest p-values are three of the four numerical predictors with the smallest squared coefficients. The relative importances given by the ERW method, however, ranks the numerical predictors differently. The two numerical predictors with the largest squared coefficients only have the fourth and fifth largest relative importances. When looking at the random intercepts in Table 6 we can see that the ranking of the random intercepts by the standard deviations and the relative importances are mostly similar, with the exception of *country* which has the third largest relative importance but the smallest standard deviation.

	player_height	player_weight
player_height	1.000	0.827
player_weight	0.827	1.000
age	-0.012	0.052
gp	-0.005	0.012
reb	0.422	0.437
ast	-0.457	-0.387
net_rating	-0.009	0.001
oreb_pct	0.591	0.604
dreb_pct	0.615	0.608
usg_pct	-0.110	-0.072
ts_pct	0.070	0.062
ast_pct	-0.626	-0.541

Table 7: Correlations between *player_height* and *player_weight* and the other numeric fixed effects in the basketball example.

A good illustration of the difficulties of using p-values and squared coefficients to determine the relative importance of correlated predictors is shown when looking at the predictors *player_height* and *player_weight*. We can see that both *player_height* and *player_weight* have large p-values and their squared coefficients are < 0.001, which could be interpreted as *player_height* and *player_weight* being unimportant to the model and not containing much information about *pts*. The relative variable importances of the two predictors are relatively large, however, third and fifth largest respectively. The reason for this discrepancy is likely that the height and weight of a player influences many other statistics which are used as predictors in the model. We can easily imagine that a taller player performs better than a short player, for example. Table 7 shows that *player_height* and *player_weight* are strongly correlated with the other numerical fixed effects, which influences the coefficients of *player_height* and *player_weight*. The coefficients further influences the p-values. If we look at a model where only *player_height* and *player_weight* are included as predictors, the p-values are both < 0.001. This discrepancy is a good example of the extra information given by the relative variable importance methods, which is not available from

p-values and coefficients in the full model. Just looking at the full model where *player_height* and *player_weight* have small coefficients and large p-values could easily lead to the conclusion that *player_height* and *player_weight* do not contain any useful information to model the average number of points scored in a game, but the relative variable importance gives more information.

An important detail to be aware of when using the ELMG and ERW methods (and also the original LMG and relative weights methods), is that the relative variable importance given to a predictor is dependent on the other predictors in the model. Here, *player_height* and *player_weight* are strongly correlated, with a correlation of 0.827. The strong correlation means that *player_height* and *player_weight* contain much of the same information, which the ERW method distributes the credit for. Thus, if *player_weight* were to be removed from the model and the relative importances recalculated, then we would expect *player_height* to get credit for all the information it shared with *player_weight*. When *player_weight* was removed from the model *player_height* got a relative importance of 0.108, which is 0.015 more than in the full model. When *player_weight* is not in the model *player_height* has a larger relative importance than *ast_pct*, while it had less when *player_weight* was in the model. When looking at relative importances we should therefore think about the property we are interested in, for example physical characteristics in this case for *player_height* and *player_weight*, and sum the relative importances for the predictors in that property. The sum of these related shares should give a more robust importance for the property we are interested in.

To get the relative importances, the accompanied R-package is used. This example is a good illustration of the difference in computational complexity between the ELMG method and the ERW method. The ELMG method ran for over 16 hours before it was canceled, while the ERW method finished in just 2 minutes. This is expected, as the ELMG method has to consider $18 \cdot 2^{17} = 2359296$ subsets while the ERW method only has to consider $7 \cdot 2^{6} = 448$. This means that a very rough estimate of the time it would take to complete the calculations of the ELMG method is $\frac{2359296\ subsets}{448\ subsets} \cdot \frac{2\ minutes}{60 \frac{minutes}{hour}} \approx 175\ hours$.

6 Discussion and conclusion

6.1 What is accomplished?

In this thesis we have looked at how to extend the concept of relative variable importance in linear regression to random intercept models. This extension has been accomplished by extending the LMG method, which is quite computationally demanding, and the relative weights method, which is not as accurate but more efficient. Extending the LMG and relative weights methods has required the use of recent developments in expanding the R^2 concept to random intercept models before decomposing the R^2 of the model such that each predictor is given a share (Nakagawa and Schielzeth, 2013; Johnson, 2014). We have seen through a simulation study that the shares given to the random intercepts are very similar for the LMG and relative weights methods. The two methods vary slightly for the shares of the fixed effects, however. When the fixed effects are pairwise independent, the two methods give very similar shares, but when the fixed effects have more complex correlation structures, the two methods differ more. Finally, we have seen that the main result of the thesis, the ERW method, gives useful results, by looking at a simulation study and a real world data set.

The ELMG and ERW methods are new tools that researchers can use when interpreting random intercept models. Currently, it can be difficult to understand the properties of statistical models, where the p-value, which is probably the most used statistic when interpreting models, is difficult to understand and prone to misinterpretation. The ELMG and ERW methods are hopefully simple enough to interpret, so that less misinterpretations occur when the methods are used in practice. Importantly, the sum of the shares given to the predictors in the model is the model R^2, which means that the shares can simply be interpreted as the predictor's respective contribution to the model R^2. The ELMG method satisfies the goals for this thesis, that is, create a relative variable importance measure for random intercept models that is easy to interpret and which gives useful information about the model. It decomposes the R^2 of the model, which is an intuitive value for most scientists, and gives a share to each predictor which can be interpreted as that predictors contribution to the model R^2.

Unfortunately, the ELMG method is quite computationally demanding, which means that for many applications the approximation given by the less computationally demanding ERW method will need to be used instead. The ERW method is much more efficient, and will usually give similar results as the ELMG method. The ERW method can be sensitive to the correlation structure of the data however, where some correlation structures might cause the ERW method to give less accurate results. Thus, while the shares given by the ERW method sum to the model R^2, and are therefore easy to interpret, the sensitivity of the method to the correlations between the fixed effects means that the approximations given by the ERW method might not be as easy to trust as the ELMG method and as hoped. The fact that the correlation structure of the predictors in the model affects the calculated shares in a not understood way means that it can be difficult to trust the results. Luckily, in the simulation study the difference between the ELMG and ERW methods was relatively small, the largest absolute difference was ~ 0.06 (0.235 - 0.3), but it can be enough to slightly change the results. Some care should therefore be used when using the shares given by the ERW method and if computationally feasible, the ELMG method should be preferred.

As a step in making the proposed ELMG and ERW methods easy to use, effort has been made in creating the R-package `decompR2`, which is in the process of being submitted to CRAN.

decompR2 makes it simple to use the ELMG and ERW methods, by simply requiring a fitted model as input to calculate the relative importances of the predictors in the model. Making it easy for the user to calculate the relative importance of predictors in created models will hopefully make it more likely for relative importance measures to be used in research as an additional tool to help in interpreting modeling output. The source code of decompR2 can be found at https://gitlab.com/elonus/decompr2 and in Appendix B while an accompanying vignette showing how to use decompR2 can be found in Appendix A.

Unfortunately, it is likely that any statistic used to interpret models and results will develop the same malpractice as the p-value, where a cut-off value is chosen, and if the statistic is smaller (or larger in some cases) than this limit the predictor is considered "significant", and if not, the predictor is not considered significant. Such a practice is difficult to prevent through technical or scientific means, since it stems from people wanting simple rules they can use in practice. Instead, better education is needed to prevent researchers from using such simple rules and instead use more nuanced approaches and care when interpreting results.

Finally, both Chevan and Sutherland (1991) and Grömping (2007) have warned that relative importance measures give more limited information than we might hope. First, as with other statistics calculated when interpreting models, if the theoretical assumptions of the model is not fulfilled by the data, then the calculated statistics will not be correct, and can give misleading values. Relative variable importance and other statistics rely on the model being properly specified and can otherwise give incorrect information. Second, if the relative importance measures are to be used to prioritize intervention to influence the response, care should be taken, since an intervention might not only influence the predictor but also the correlation structure among the predictor which can change the relationships in the data. To plan interventions and understand causality, theory-driven explanatory models will give more robust information. A good model should be based on a good understanding of the data, calculated statistics like relative variable importance is no substitute for domain knowledge.

6.2 Further work

The work done in this thesis can be expanded in several ways. An obvious next step that would be useful is to further explore when the ERW method does not give the same results as the ELMG method. Having a better understanding of the estimates given by the ERW method would let researchers be able to use the ERW method with more confidence in the cases that it gives more correct results and avoid using the ERW method when it does not give correct results.

Another possibility is to work on further extending the ELMG and ERW methods to GLMMs (Figure 3). The first step in such an expansion is creating relative variable importance measures that calculate importances for general LMMs, *i.e.*, models with random slopes and interactions between predictors. Expanding the ELMG method to handle interactions has the challenge that it makes no sense to have a model including the interaction between two predictors if the model does not contain the individual predictors. That means that it is not possible to simply consider all subsets of predictors as is done now. A solution to give a relative importance to an interaction could be to only consider the increase in R^2 when looking at the subsets including the individual predictors of the interaction and not consider the subsets without the individual predictors of the interaction. This approach is used in the relaimpo package (Grömping, 2006). Random slopes have a similar problem, as random slopes can be interpreted as the interaction between a numerical fixed effect and a random intercept. It does not make sense to have a random slope

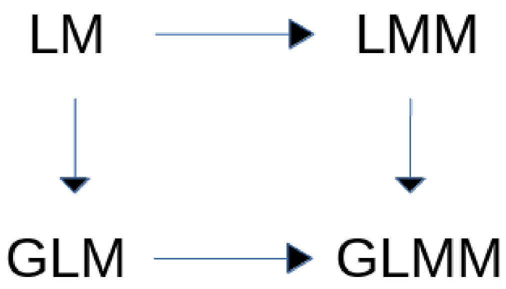

Figure 3: How to work towards relative variable importance for GLMMS.

in a model if the fixed effect and random intercept which are a part of the random slope are not in the model. A similar solution might work as for interactions, where the increase in R^2 when adding the random slope to the model is only considered when the corresponding fixed effect and random intercept are already in the model.

Before looking at GLMMs it would be useful to understand how to work with relative variable importances for general GLMs. Chevan and Sutherland (1991) worked on generalizing the concept behind the LMG method to GLMs, but since there was no good definition of R^2 for GLMs at that time, they instead decomposed a chi-square statistic. Decomposing the R^2 has the advantage of being easier to interpret, so we believe that it would be useful to take a new look at this problem by using the work done by Nakagawa and Schielzeth (2013) and Johnson (2014) on expanding the R^2 concept to GLMs. Challenges in expanding the LMG method to GLMs include the fact that there are different scales that can be used to measure the variance in the model. It is possible to either measure variance on the scale of the response or it is possible to measure variance on the scale of the linear predictor, *i.e.*, before the inverse link function is used. After approaches that work for LMMs and GLMs are created, it is hopefully possible to combine these approaches to get an approach that works for GLMMs.

A different aspect that would be interesting to explore is the uncertainty in the shares given to each predictor. A straightforward way to do this would be through bootstrapping, but that can quickly become computationally expensive (Efron, 1992). In the example using the basketball data in Section 5 the ERW method takes roughly 2 minutes to run. A bootstrapping procedure with 100 iterations would therefore take roughly $2\ minutes \cdot 100 \approx 3\ hours$. A more theoretical approach of considering the distribution of the shares and then creating confidence intervals would therefore be useful. Finding the distribution could be difficult, however, since that would require understanding the statistical properties of the conditional R^2 in equation (6), which is not obvious.

6.3 Conclusion

We hope that the work done in this thesis will be a useful tool when interpreting linear random intercept models. To prevent the problems caused by misinterpreting the p-value when interpreting models we believe a simple to use and easy to interpret tool, like `decompR2`, is helpful. Hopefully `decompR2` will cause some researchers to diversify the techniques they use in model interpretation which will give more robust results.

References

Akaike, H. (1973). "Information Theory and an Extension of the Maximum Likelihood Principle". In: *2nd International Symposium of Information Theory.* Ed. by B. N. Petrov and F. Csaki. Budapest, Akedemia Kiado, pp. 267–281.

Arbuthnot, John (1710). "II. An argument for divine providence, taken from the constant regularity observ'd in the births of both sexes. By Dr. John Arbuthnott, Physitian in Ordinary to Her Majesty, and Fellow of the College of Physitians and the Royal Society". In: *Philosophical Transactions of the Royal Society of London* 27.328, pp. 186–190.

Azen, Razia and David V Budescu (2003). "The dominance analysis approach for comparing predictors in multiple regression." In: *Psychological methods* 8.2, p. 129.

Budescu, David V (1993). "Dominance analysis: a new approach to the problem of relative importance of predictors in multiple regression." In: *Psychological bulletin* 114.3, p. 542.

Budescu, David V and Razia Azen (2004). "Beyond global measures of relative importance: Some insights from dominance analysis". In: *Organizational Research Methods* 7.3, pp. 341–350.

Burnham, K. P. and D. R. Anderson (2002). *Model selection and multimodel inference : a practical information-theoretic approach.* New York: Springer.

– (2014). "P values are only an index to evidence: 20th- vs. 21st-century statistical science". In: *Ecology* 95, pp. 627–630.

Byhring, Oliver (2020). *Relative variable importance in linear regression models with random intercept term.*

Chevan, Albert and Michael Sutherland (1991). "Hierarchical partitioning". In: *The American Statistician* 45.2, pp. 90–96.

Cirtautas, Justinas (2022). *NBA Players.* URL: https://www.kaggle.com/datasets/justinas/nba-players-data (visited on 11th Apr. 2022).

Claeskens, G. and N. L. Hjort (2008). *Model Selection and Model Averaging.* Campridge: Cambridge University Press.

Claridge-Change, A. and P. N. Assam (2016). "Estimation Statistics Should Replace Significance Testing". In: *Nature* 13, pp. 108–109.

Clark, Tom S and Drew A Linzer (2015). "Should I use fixed or random effects?" In: *Political science research and methods* 3.2, pp. 399–408.

Darlington, Richard B (1968). "Multiple regression in psychological research and practice." In: *Psychological bulletin* 69.3, p. 161.

Efron, Bradley (1992). "Bootstrap methods: another look at the jackknife". In: *Breakthroughs in statistics.* Springer, pp. 569–593.

Fabbris, Luigi (1980). "Measures of predictor variable importance in multiple regression: An additional suggestion". In: *Quality and Quantity* 14.6, pp. 787–792.

Fahrmeir, Ludwig et al. (2013). "Regression Models". In: *Regression: Models, Methods and Applications.* Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-642-34333-9. DOI: 10.1007/978-3-642-34333-9_2. URL: https://doi.org/10.1007/978-3-642-34333-9_2.

Feldman, Barry (Mar. 2005). "Relative Importance and Value". In: *SSRN Electronic Journal.* DOI: 10.2139/ssrn.2255827.

Fisher, RA (1925). *Statistical methods for research workers.* Edinburgh: Oliver and Boyd.

Friedberg, Stephen H., Arnold J. Insel and Lawrence E. Spence (2003). "Linear Algebra". In: 4th ed. Pearson. Chap. 6.7. ISBN: 9780130084514.

Gelman, A. and E. Loken (2014). "The Statistical Crisis in Science". In: *American Scientist* 102, pp. 460–465.

Genizi, Abraham (1993). "Decomposition of R 2 in multiple regression with correlated regressors". In: *Statistica Sinica*, pp. 407–420.

Goodman, S. N. (2016). "Aligning statistical and scientific reasoning". In: *Science* 352, pp. 1180–1182.

Goodman, Steven (2008). "A Dirty Dozen: Twelve P-Value Misconceptions". In: *Seminars in Hematology* 45.3. Interpretation of Quantitative Research, pp. 135–140. ISSN: 0037-1963. DOI: https://doi.org/10.1053/j.seminhematol.2008.04.003. URL: https://www.sciencedirect.com/science/article/pii/S0037196308000620.

Grömping, Ulrike (2006). "Relative Importance for Linear Regression in R: The Package relaimpo". In: *Journal of Statistical Software* 17.1, pp. 1–27.

– (2007). "Estimators of relative importance in linear regression based on variance decomposition". In: *The American Statistician* 61.2, pp. 139–147.

– (2015). "Variable importance in regression models". In: *WIREs Computational Statistics* 7, pp. 137–152.

Hastie, Trevor et al. (2009). *The elements of statistical learning: data mining, inference, and prediction.* Vol. 2. Springer.

Head, Megan L et al. (2015). "The extent and consequences of p-hacking in science". In: *PLoS biology* 13.3, e1002106.

Hoffman, Paul J (1960). "The paramorphic representation of clinical judgment." In: *Psychological bulletin* 57.2, pp. 116–131.

– (1962). "Assessment of the independent contributions of predictors." In: *Psychological bulletin* 59.1, pp. 77–80.

Imo, D. et al. (2017). *Risk assessment for children and mothers in a mercury contaminated area using biomonitoring and individual soil measurements: a cross-sectional study.* Tech. rep. University of Zurich.

Ioannidis, John PA (2005). "Why Most Published Research Findings are False". In: *PLoS Medicine* 2, e124.

– (2018). "The proposal to lower P value thresholds to. 005". In: *Jama* 319.14, pp. 1429–1430.

Johnson, J. B. and K. S. Omland (2004). "Model selection in ecology and evolution". In: *Trends in Ecology and Evolution* 19, pp. 101–107.

Johnson, Jeff W (2000). "A heuristic method for estimating the relative weight of predictor variables in multiple regression". In: *Multivariate behavioral research* 35.1, pp. 1–19.

Johnson, Paul CD (2014). "Extension of Nakagawa & Schielzeth's R2GLMM to random slopes models". In: *Methods in ecology and evolution* 5.9, pp. 944–946.

Johnson, Richard M (1966). "The minimal transformation to orthonormality". In: *Psychometrika* 31.1, pp. 61–66.

Kruskal, William (1987). "Relative importance by averaging over orderings". In: *The American Statistician* 41.1, pp. 6–10.

Lindeman, R. H., P. F. Merenda and R. Z Gold (1980). *Introduction to Bivariate and Multivariate Analysis.* Glenview: IL: Scott, Foreman, p. 119.

Lipovetsky, Stan and Michael Conklin (2001). "Analysis of regression in game theory approach". In: *Applied Stochastic Models in Business and Industry* 17.4, pp. 319–330.

MetaCRAN (2022). *relaimpo package downloads.* https://cranlogs.r-pkg.org/badges/grand-total/relaimpo. Accessed: 2022-01-20.

Nakagawa, Shinichi and Holger Schielzeth (2013). "A general and simple method for obtaining R2 from generalized linear mixed-effects models". In: *Methods in ecology and evolution* 4.2, pp. 133–142.

NBA (2022). *NBA Advanced Stats.* URL: https://www.nba.com/stats/ (visited on 11th Apr. 2022).

Nimon, Kim F and Frederick L Oswald (2013). “Understanding the results of multiple linear regression: Beyond standardized regression coefficients”. In: *Organizational Research Methods* 16.4, pp. 650–674.

Nuzzo, R. (2014). “Statistical Errors”. In: *Nature* 506, pp. 150–152.

Schwarz, G. (1978). “Estimating the dimension of a model”. In: *Annals of Statistics* 6, pp. 461–464.

Shapley, Lloyd S (1953). “Stochastic games”. In: *Proceedings of the national academy of sciences* 39.10, pp. 1095–1100.

Simmons, Joseph P, Leif D Nelson and Uri Simonsohn (2011). “False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant”. In: *Psychological science* 22.11, pp. 1359–1366.

Stoffel, Martin A, Shinichi Nakagawa and Holger Schielzeth (2021). “partR2: Partitioning R2 in generalized linear mixed models”. In: *PeerJ* 9, e11414.

Stufken, John (1992). “On hierarchical partitioning”. In: *The American Statistician* 46.1, pp. 70–71.

Ward, Joe H Jr (1962). “Comments on” The paramorphic representation of clinical judgment.””. In: *Psychological bulletin* 59.1, pp. 74–76.

Wasserstein, Ronald L. and N. A. Lazar (2016). “The ASA’s statement on p-values: context, process, and purpose”. In: *The American Statistician* 70, pp. 129–133.

Wasserstein, Ronald L., Allen L. Schirm and Nicole A. Lazar (2019). “Moving to a World Beyond “p $<$ 0.05””. In: *The American Statistician* 73.sup1, pp. 1–19. DOI: 10.1080/00031305.2019.1583913. eprint: https://doi.org/10.1080/00031305.2019.1583913. URL: https://doi.org/10.1080/00031305.2019.1583913.

A Vignette

A.1 Relative variable importance using R^2

A.1.1 Background

Relative variable importance is the concept of determining how important each predictor is to a model, or similarly, how much information there is in each predictor about the response in the model. There are several approaches to calculate such importances, where likely the most common, but erroneous, way is to use the p-value of the hypothesis test which is testing that the coefficient of a predictor is 0. This is not a valid method to determine the amount of information in a predictor, since the p-value only looks for an effect, it does not take the size of the effect into account.

Some other common approaches are looking at the squared coefficients of the standardized predictors and the confidence intervals of the coefficients of the predictors. Both of these are useful and make sense if the predictors are uncorrelated. If the predictors are correlated, however, it is not clear how to interpret these metrics because the size of the coefficient no longer necessarily indicates the amount of information explained by the predictor. If there are two strongly correlated preditors in the model which contain information about the response, then one of the predictors might get a larger coefficient than the other, even if they contain the same information about the response (Grömping, 2007).

Methods which handle correlated predictors are therefore needed. Basing such relative variable importance methods on the coefficient of determination (R^2) is common because of it's simple interpretation. The R^2 of the model is the proportion of variance in the response explained by the model. If a relative variable importance method manages to decompose the R^2 of the model to the predictors, then the respective importances can be interpreted as the proportion of variance in the response explained by the respective predictor.

Two methods used for linear regression models are the LMG method and the relative weights method (Lindeman et al., 1980; Johnson, 2000). The LMG method works by looking at all the permutations of the predictors, and then considering the mean increase in R^2 when the predictor of interest is added to the model. The relative weights method instead takes advantage of the fact that the squared coefficients are meaningful for uncorrelated predictors. Therefore, the data is transformed such that the columns are uncorrelated, then the squared coefficients for these uncorrelated columns are calculated, before the coefficients are distributed back to the original predictors by doing the reverse transformation. Because of the transformation of the data the relative weights method only works for numerical predictors. The LMG method gives good results but is computationally expensive, so the more computationally efficient relative weights method is often used as an approximation. These methods are implemented in other packages as well, such as the `relaimpo` package. The new development in this package is an extension of the LMG method and the relative weights method to linear random intercept models.

The extended LMG method works by simply treating the random intercepts the same as the predictors of a normal linear regression model, *i.e.*, look at the mean increase in R^2 when each predictor is added to the model. The extended relative weights method works by combining the LMG method and the relative weights method. First, transform the numerical fixed effects the same way as in normal relative weights. Then, when doing the permutations, let either all the transformed fixed effects be in the model or none of them. This reduces the computational complexity since there are many fewer permutation to consider. To get the importance of a random intercept or a categorical predictor, simply do the same as in the LMG method, except that the transformed fixed effects are always together, either all in the model or not. For the numerical fixed effects, consider all permutations and add all the transformed fixed effects to the models together. Then, using the squared coefficients of the transformed fixed effects in the model, distribute the increase in R^2 to each transformed fixed effect. Then, transform these shares back to the original numerical fixed effect using the same transformation as in normal relative weights. Finally, take the mean of these shares to get an importance for each numerical fixed effect.

Some nice properties of the above methods based on R^2 is that

1. The sum of the importances of the predictors of a model will always sum to the R^2 of the model.
2. Each importance is always non-negative.
3. If a predictor has an effect on the model, *i.e.*, a fixed effect has a coefficient not equal to zero or a random intercept has variance larger than zero, then that predictor will have an importance larger than zero.

which makes the importances easier to interpret as the amount of R^2 contributed by the predictor to the model.

A.1.2 Limitations

Currently, `decompR2` has some limitations:

1. It can only handle random intercepts, not random slopes.
2. It can not handle interactions between predictors.
3. It can not handle GLMs

And some bugs which are being worked on:

1. It can not handle terms in the model formula which expand into several predictors in the model, like `poly()`.
   As a workaround, instead add the terms manually to the formula, *e.g.*, use `y ~ x1 + I(x1^2)` instead of `y ~ poly(x1, 2)`.

Additionaly, both Chevan and Sutherland (1991) and Grömping (2007) have warned that relative importance measures give more limited information than we might hope. First, as with other statistics calculated when interpreting models, if the theoretical assumptions of the model is not fulfilled by the data, then the calculated statistics will not be correct, and can give misleading values. Relative variable importance and other statistics rely the model being properly specified and can otherwise give incorrect information. Second, if the relative importance measures are to be used to prioritize intervention to influence the response, care should be taken, since an intervention might not only influence the predictor but also the correlation structure among the predictor which can change the relationships in the data. To plan interventions and understand causality, theory-driven explanatory models will give more robust information. A good model should be based on a good understanding of the data, calculated statistics like relative variable importance is no substitute for domain knowledge.

A.1.3 Warnings about convergence

There might be some warnings about models failing to converge when there are random intercepts in the model. This likely comes from the fact that for some combinations of predictors there is not enough difference between the different clusters, which gives `lme4::lmer()` problems when fitting. This failed convergence, luckily, only seems to happen for few of the fitted models, which means that it should not affect the results too much, since the result is an average of many values.

A.2 How to use `decompR2`

The only function needed to calculate the relative importances of the predictors of a model is `decompR2()`. This function takes as input either a model output, from either `stats::lm()` or `lme4::lmer()`, or a model formula along with the data to fit the model. Additionally, the method to be used to calculate the relative importances can be provided, although if no method is given, then the extended LMG method will be used if there are less than 15 numerical fixed effects and the extended relative weights method otherwise. This is to make the calculations not take too long.

Since `decompR2` is not on CRAN yet, it can not be installed with `install.packages(decompR2)`, but `devtools::install_git()` must be used instead. This installs the latest version of the package on GitLab.

```
devtools::install_git("https://gitlab.com/elonus/decompr2.git")
library(decompR2)
```

A.2.1 Simple example

```
# Load other packages
library(lme4)
```

We will show a simple example illustrating how to use `decompR2()`. The use of relative variable importance will be illustrated on a data set containing player statistics in the National Basketball Association (NBA). The data set contains information on physical characteristics and game statistics of each player in each season from 1996-1997 until 2020-2021. The goal of this example will be to determine the proportion of variance in the average number of points scored in a game that is explained by the physical characteristics of a player. The data is included in the `decompR2` package and can be loaded by

```r
data(basketball)
```

To not make the example too complex, we will only use a subset of the variables in the data set, where we will standardize the numerical variables. The physical characteristics included are *age*, *player_height* and *player_weight*.

```r
basketball <- basketball[,c("pts", "age", "player_height", "player_weight",
                            "gp", "ast_pct", "season", "player_name")]

basketball[,c("pts", "age", "player_height",
              "player_weight", "gp", "ast_pct")] <-
  scale(basketball[,c("pts", "age", "player_height",
                      "player_weight", "gp", "ast_pct")])
```

The descriptions of the variables we will consider are shown in Table 8.

Variable	Description
pts	Average number of points scored in a game (the response)
age	Age of player (years)
player_height	Height of player (cm)
player_weight	Weight of player (kg)
gp	Number of games played in the season
ast_pct	Percentage of teammate field goals the player assisted while he was on the floor
season	NBA season
player_name	Name of player

Table 8: Description of the variables in the example data set about basketball.

Figure 4 shows plots with the predictors on the x-axes and *pts* on the y-axis.

We can see that *age* seems to have a quadratic relationship with *pts*, so we will use *age*^{2} instead.

```r
basketball$agesq <- scale(basketball$age^2)
```

The correlations between the numerical predictors are

```r
cor(basketball[,c("pts", "agesq", "player_height",
                  "player_weight", "gp", "ast_pct")])

##                      pts        agesq player_height player_weight           gp
## pts            1.0000000 -0.1036971247  -0.060523999   -0.03065530  0.538366685
## agesq         -0.1036971  1.0000000000   0.044034913    0.02808822 -0.039522394
## player_height -0.0605240  0.0440349135   1.000000000    0.82730131 -0.005329397
## player_weight -0.0306553  0.0280882156   0.827301314    1.00000000  0.012496076
```

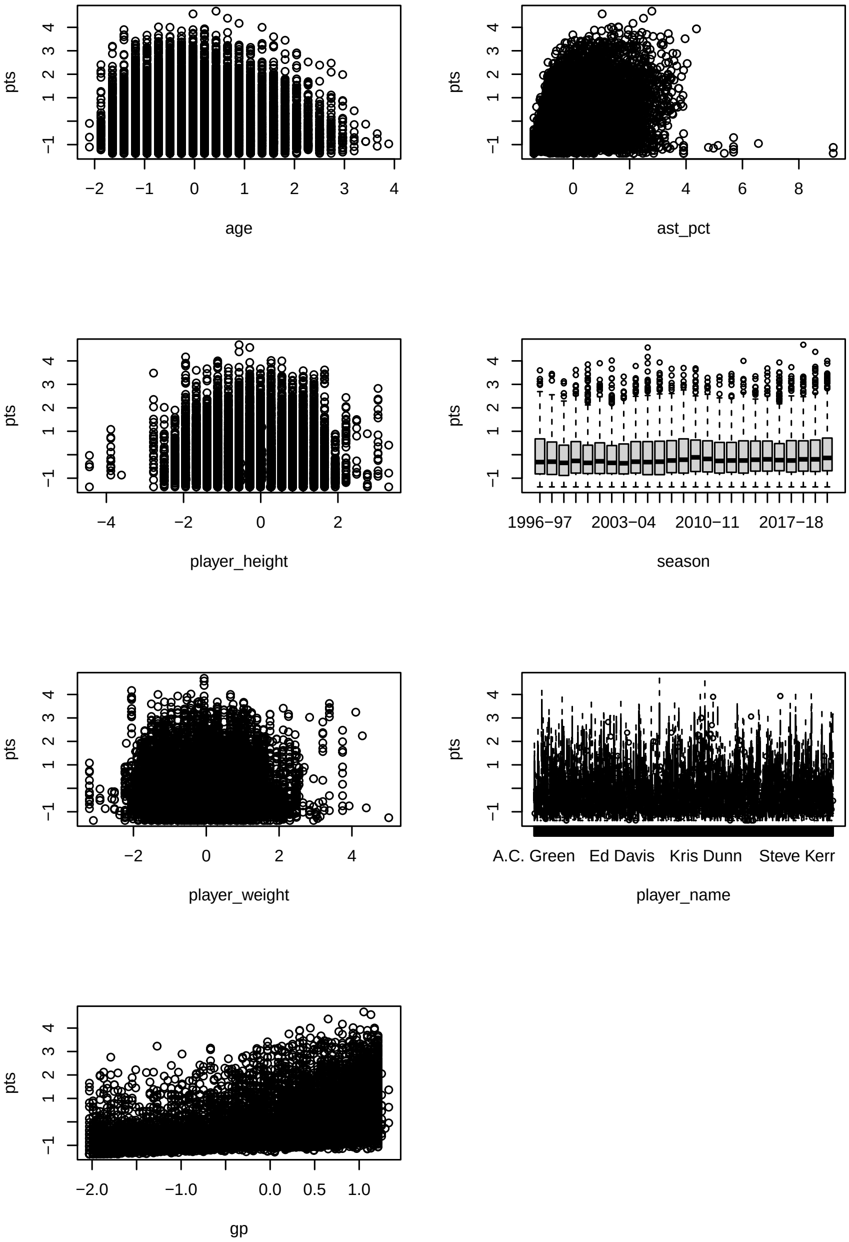

Figure 4: Relationship between the predictors and the response, *pts*.

```
## gp              0.5383667 -0.0395223944  -0.005329397    0.01249608  1.000000000
## ast_pct         0.3302695 -0.0005073946  -0.625888234   -0.54090218  0.134809543
##                      ast_pct
## pts             0.3302694537
## agesq          -0.0005073946
## player_height  -0.6258882340
## player_weight  -0.5409021831
## gp              0.1348095426
## ast_pct         1.0000000000
```

We see that *gp*, the number of games played by the player in the respective season, has a relatively high correlation with *pts*, the average number of points scored in a game, where the correlation is ~ 0.54. This makes sense as we would expect the better players to play in more games. The physical characteristics seem more weakly correlated with *pts*, with all of them having an absolute correlation less than 0.07. There is a high correlation between *player_height* and *player_weight*, however, which makes sense, as a taller person usually weighs more.

We create a random intercept model, where *player_name* and *season* are included as random intercepts. Having the name of the player as a random intercept allows us to correct for individual skill.

```
m <- lme4::lmer(pts ~ agesq + player_height + player_weight + gp + ast_pct +
                  (1 | season) + (1 | player_name), data = basketball)
summary(m)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: pts ~ agesq + player_height + player_weight + gp + ast_pct +
##     (1 | season) + (1 | player_name)
##    Data: basketball
##
## REML criterion at convergence: 19874.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.6935 -0.5484 -0.0755  0.5174  5.4549
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  player_name (Intercept) 0.341592 0.5845
##  season      (Intercept) 0.008464 0.0920
##  Residual                0.216298 0.4651
## Number of obs: 11700, groups:  player_name, 2333; season, 25
##
## Fixed effects:
##                 Estimate Std. Error t value
## (Intercept)    -0.179116   0.022987  -7.792
## agesq          -0.234886   0.005828 -40.301
## player_height   0.122824   0.019807   6.201
## player_weight  0.005146   0.017267   0.298
## gp             0.327940   0.005953  55.089
## ast_pct        0.265969   0.009166  29.017
##
## Correlation of Fixed Effects:
##              (Intr) agesq  plyr_h plyr_w gp
## agesq         0.025
## player_hght   0.002 -0.016
## player_wght   0.027  0.035 -0.697
## gp            0.090  0.150 -0.012  0.000
## ast_pct       0.045  0.101  0.253  0.012 -0.039
```

```
lme4::fixef(m)[-1]^2
```

```
##          agesq player_height player_weight            gp       ast_pct
##   5.517163e-02  1.508563e-02  2.647727e-05  1.075449e-01  7.073934e-02
```

```
sum(lme4::fixef(m)[c("agesq", "player_height", "player_weight")]^2)
```

```
## [1] 0.07028373
```

We see that the sum of the squared standardized coefficients of the physical characteristics is $\sim$ 0.07. This indicates that the physical characteristics explain $\sim$ 7% of the variance in the average number of points scored in a game.

The coefficients and squared coefficients gives information about the size of the effect each predictor has on *pts*, but they are hard to interpret because of pairwise correlations between the predictors, especially between *player_height*, *player_weight* and *ast_pct*. We therefore want to use relative variable importance to get more robust information. We can do this by using `decompR2()`. We simply input the model, `m`, and we get the relative importances as output. Here we specify the method to use to illustrate.

```
d_auto <- decompR2::decompR2(m)
d_lmg <- decompR2::decompR2(m, method = "lmg")
d_rw <- decompR2::decompR2(m, method = "rw")
```

Equivalently we could have inputted the model formula and data like this

```
d_auto <- decompR2::decompR2(pts ~ agesq + player_height + player_weight + gp +
                   (1 | season) + (1 | player_name), data = basketball)
d_lmg <- decompR2::decompR2(pts ~ agesq + player_height + player_weight + gp +
                   (1 | season) + (1 | player_name), data = basketball,
                   method = "lmg")
d_rw <- decompR2::decompR2(pts ~ agesq + player_height + player_weight + gp +
                   (1 | season) + (1 | player_name), data = basketball,
                   method = "rw")
```

```r
# Decompositions from not inputting a method:
print(d_auto)
```

```
##             agesq     player_height     player_weight                gp 
##      0.0516723424      0.0016505247      0.0004707879      0.1450274572 
##      (1 | season) (1 | player_name) 
##      0.0143853853      0.5067433253 
```

```r
# LMG decompositions:
print(d_lmg)
```

```
##             agesq     player_height     player_weight                gp 
##      0.0516723424      0.0016505247      0.0004707879      0.1450274572 
##      (1 | season) (1 | player_name) 
##      0.0143853853      0.5067433253 
```

```r
# Relative weights decompositions:
print(d_rw)
```

```
##             agesq     player_height     player_weight                gp 
##      0.0218178515      0.0016518269      0.0005092512      0.1758156137 
## (1 | player_name)      (1 | season) 
##      0.5036053358      0.0165498126 
```

We can see that in this case the decompositions from the LMG method are identical with the decompositions from not specifying a method, which is because the number of numerical fixed effects are only 4, which is less than 15. Thus, the LMG method is used by default.

The proportion explained by physical characteristics according to the LMG method is

```r
sum(d_lmg[c("agesq", "player_height", "player_weight")])
```

```
## [1] 0.05379365
```

While the proportion explained by physical characteristics according to the relative weights method is

```r
sum(d_rw[c("agesq", "player_height", "player_weight")])
```

```
## [1] 0.02397893
```

We can see that the two methods disagree slightly, but they both agree that physical characteristics does not seem to explain more than $\sim 5\%$ of the average number of points scored in a game. The name of the player, on the other hand, explains roughly 50%, which makes sense, since the name of the player is just a proxy for individual specific effects, such as the skill of the player, which is likely the most important factor.

The fact that physical characteristics have such a small importance might be explained by the fact that there is a large sampling bias, only professional players are included in the statistics. If a short player plays professionally, where being tall is considered an advantage, then they likely are exceptionally skilled or have other advantages that bridge the gap. If the basketball skills of the whole population was analyzed then we would expect that physical characteristics would play a larger role.

B Implementation of the R-package

The `decompR2` R-package can be installed and loaded by

```
devtools::install_git(https://gitlab.com/elonus/decompr2.git)
library(decompR2)
```

```
##' @title Decompose the coefficient of determination (R^2)
##'
##' @description Decompose the coefficient of determination (R^2) of a linear regression or linear random intercept model such that each predictor gets a share of the R^2 of the model.
##'              This creates a measure of relative variable importance which can be used, as a supplement to p-values, to interpret the model.
##'
##' @details Creates a decomposition of the R^2 of a model, i.e., distribute the R^2 to the predictors of the model such that each predictor gets a share corresponding to it's contribution to the model R^2. These shares are called the relative variable importances of the predictors.
##'          The relative importances have the properties that they will always sum to the model R^2 (proper decomposition), are always non-negative (non-negativity) and if a predictor has a non-zero coefficient in the model it will not get a zero share (inclusion).
##'          There are two methods currently implemented to perform the decomposition, the LMG method and the relative weights method.
##'
##'          The LMG method is the most accurate one, but also the most computationally expensive. It is therefore the default method if there are less than 15 fixed effects in the model.
##'          The LMG method works by looking at all permutations of the predictors, i.e., all the orderings of the predictors. Then, fit models with just the first predictor in the permutation, the first two predictors in the permutation, the first three predictors in the permutation, and so on.
##'          For a specific predictor, it's share is calculated by looking at the increase in R^2 of the model when that predictor is added to the model. Then take the mean of this increase when looking at all permutations of predictors.
##'
##'          The relative weights method is not as accurate, but much more computationally efficient. It is therefore the default method when there are more than 15 fixed effects in the model.
##'          The relative weights method works by first performing a linear transforming on the numeric fixed effects such that new numeric fixed effects are calculated which are uncorrelated.
##'          In a model with only uncorrelated standardized numeric fixed effects the relative importance of each predictor is simply it's squared coefficient.
##'          If there are categorical predictors or random intercepts in the model it is more complicated. Then a similar approach is used as in the LMG method, where the mean increase in R^2 is considered for each predictor, but now the transformed numerical fixed effects are considered as one "block" which is either all in the model or none are in the model.
##'          The categorical predictors and random intercepts get shares the same way is in the LMG method, the mean increase in model R^2 when they are added to the model.
##'          For the numerical fixed effect, look at the mean increase in R^2 when all the numerical fixed effects are added to the model. Then use the squared coefficients to distribute this increase to each numerical fixed effect.
##'          Having the numerical fixed effects as a "block" reduces the number of possible permutations dramatically, thus reducing the computational complexity.
```

```r
##'
##'         There might be some warnings about models failing to converge when there are random intercepts in the model. This likely comes from the fact that for some combinations of predictors there is not enough difference between the different clusters, which gives `lme4::lmer()` problems when fitting. This failed convergence, luckily, only seems to happen for few of the fitted models, which means that it should not affect the results too much, since the result is an average of many values.
##'
##' @param obj is a model object from either `stats::lm` or `lme4::lmer`
##'
##' OR
##'
##' a formula, e.g., y ~ x1 + x2 + (1 | group)
##' @param data only if `obj` is a formula. data-frame, contains data used to fit the model. All variables referred to in the formula in `obj` need to be present as columns in 'data'.
##'            The random intercepts should have it's column name on the form "*name*", not "(1 | *name*)".
##' @param subset only if `obj` is a formula. Is passed to `stats::model.frame` to subset data. A specification of the rows to be used: defaults to all rows.
##'              This can be any valid indexing vector (see [.data.frame)
##'              for the rows of `data` or if that is not supplied, a data
##'              frame made up of the variables used in formula. For more details see `?stats::model.frame`
##' @param na.action only if `obj` is a formula. Is passed to `stats::model.frame`. Is how `NA`s are treated.  The default is first, any
##'         `na.action` attribute of `data`, second a `na.action` setting
##'         of options, and third na.fail if that is unset.  The
##'         `factory-fresh` default is na.omit.  Another possible value
##'         is `NULL`. For mode details see `?stats::model.frame`
##' @param weights only if `obj` is a formula. Is passed to `stats::lm` or `lme4::lmer`, depending on if there are random intercepts in the formula. See `?stats::lm` and `?lme4::lmer` for more details.
##' @param method is a character string saying which method to use to decompose R^2.
##' Available methods are the LMG method ("lmg") and the relative weights ("rw") method. The default is to use the LMG method if there are less than 15 fixed effects in the model and otherwise use the relative weights method, as it is more computationally efficient.
##' @param ... currently not used, only here to satisfy S3 generic requirements
##' @return Named vector containing the relative variable importances of the predictors of the specified model.
##' @author Andreas Matre
##' @aliases decompR2.formula decompR2.lm decompR2.lmerMod
##'
##' @examples
##' data(cake, package = "lme4")
##'
##' decompR2(angle ~ temp + (1 | recipe) + (1 | replicate), data = cake)
##' decompR2(angle ~ temp + (1 | recipe) + (1 | replicate), data = cake, method = "rw")
##'
##' m <- lme4::lmer(angle ~ temp + (1 | recipe) + (1 | replicate), data = cake)
##' decompR2(m)
##'
##' @export
decompR2 <- function(obj, ...) {
  UseMethod("decompR2", obj)
}
```

```r
##' @rdname decompR2
##' @export
decompR2.lm <- function(obj,
             method = NULL,
             ...) {
  f <- stats::formula(obj)
  data <- stats::model.frame(obj)
  args <- list(formula = f, data = data, method = method)
  if(!is.null(stats::weights(obj))) {
    args[["weights"]] <- stats::weights(obj)
  }
  return(do.call(decompR2_internal, args))
  #return(decompR2.formula(obj = f, data = data, weights = weights, na.action = na.action, method = method))
}
```

```r
##' @rdname decompR2
##' @export
decompR2.lmerMod <- function(obj,
            method = NULL,
            ...) {
  if(!identical(stats::family(obj)$family, "gaussian") || !identical(stats::family(obj)$link, "identity")) {
    stop("'decompR2' only supports LMMs, i.e., the family of the model must be gaussian and the link function must be the identity.")
  }
  if(any(!sapply(obj@cnms, function(x) identical(x, "(Intercept)")))) {
    stop("'decompR2' only supports random intercepts as random effects")
  }

  f <- stats::formula(obj)
  data <- stats::model.frame(obj)
  weights <- stats::weights(obj) # If there are no weights the default for `lme4::lmer` is a vector of 1's, so don't need to check if it exists as for `lm` objects.
  return(decompR2_internal(formula = f, data = data, weights = weights, method = method))
}
```

```r
##' @rdname decompR2
##' @export
decompR2.formula <- function(obj,
                  data,
                  subset,
                  weights,
                  na.action,
                  method = NULL,
                  ...) {
  t <- stats::terms(obj, data = data)

  response <- as.character(attr(t, "variables")[attr(t, "response") + 1])
  predictors <- attr(t, "term.labels")
  fixed_effects <- predictors[!grepl(pattern = '|', x = predictors, fixed = TRUE)]
  if(any(grepl(pattern = '|', x = predictors, fixed = TRUE))) { # Check for random effects
    lf <- lme4::lFormula(obj, data = data)

    # Check for any random slopes
    if(any(!sapply(lf$reTrms$cnms, function(x) identical(x, "(Intercept)")))) {
      stop("'decompR2' only supports random intercepts as random effects")
    }

    random_intercept_cols <- names(lf$reTrms$cnms)
  } else {
    random_intercept_cols <- c()
  }


  # Use model.frame to do subsets and remove NA's according to na.action and create columns for the transformed variables in the formula
  model.frame_args <- list()
  model.frame_args$formula <- stats::as.formula(paste0(response, " ~ ", paste0(c(fixed_effects, random_intercept_cols), collapse = " + ")))
  model.frame_args$data <- data
  if(!missing(na.action)) {
    model.frame_args$na.action <- na.action
  }
  if(!missing(subset)) {
    model.frame_args$subset <- subset
  }

  data <- do.call(stats::model.frame, model.frame_args)

  #browser()
  if(missing(weights)) {
    weights <- NULL
  }

  return(decompR2_internal(formula = obj, data = data,
                           method = method, weights = weights))
}
```

```r
##' @title Decompose the coefficient of determination (R^2)
##'
##' @description Decompose the coefficient of determination (R^2) of the model with `response` as response and `predictors` as predictors, such that each predictor gets a share of the R^2 of the model.
##' This creates a measure of relative variable importance which can be used, in addition to P-values, to interpret the model.
##' The relative importances have the properties that they will always sum to the model R^2 (proper decomposition), are always non-negative (non-negativity) and if a predictor has a non-zero coefficient in the model it will not get a zero share (inclusion).
##'
##' @param formula is a formula specifying the model
##' @param data data-frame, contains data used to fit the model. All data referred to in 'response' and 'predictors' need to be present in 'data'.
##' The random intercepts should have it's column name on the form "*name*", not "(1 | *name*)".
##' @param weights weights vector passed to `stats::lm` and `lme4::lmer` when fitting models.
##' @param method Character string saying which method to use to decompose R^2.
##' Available methods are the LMG method ("lmg") and the relative weights ("rw") method.
##' @return Named vector containing the relative variable importances of the predictors of the specified model.
##' @author Andreas Matre
decompR2_internal <- function(formula,
                              data,
                              weights = NULL,
                              method = NULL) {
  RW_limit <- 15 # Number of fixed effects to accept before using the relative weights method
  methods_lmg <- c("lmg")
  methods_rw <- c("rw", "relative weights")

  t <- stats::terms(formula, data = data)
  if(any(attr(t, "order") > 1)) {
    stop(paste0("'decompR2' does not support interactions"))
  }
  if(attr(t, "response") == 0) {
    stop("There needs to be a response (left hand term) in the formula")
  }

  response <- as.character(attr(t, "variables")[attr(t, "response") + 1])
  predictors <- attr(t, "term.labels")

  random_intercepts <- grepl(pattern = '|', x = predictors, fixed = TRUE)
  fixed_effects <- !random_intercepts

  predictors[random_intercepts] <- paste0("(", predictors[random_intercepts], ")")

  # If there is only one predictor in the model, just return the R^2 of the model.
  if(ncol(data) == 2 & !any(random_intercepts)) {
    model <- fit_model(formula = as.formula(paste0(response, " ~ .")), data = data, weights = weights)
    R2 <- calc_R2(model)
    names(R2) <- predictors
    return(R2)
  }
  if(ncol(data) == 2 & any(random_intercepts)) {
    model <- fit_model(formula = as.formula(paste0(response, " ~ ", predictors)), data = data, weights = weights)
    R2 <- calc_R2(model)
    names(R2) <- predictors
    return(R2)
  }

  if(is.null(method)) {
    if(sum(fixed_effects) >= RW_limit) {
      method <- "rw"
    } else {
      method <- "lmg"
    }
  }

  if(tolower(method) %in% methods_lmg) {
    return(decompR2_lmg(response = response, predictors = predictors, data = data, weights = weights))
  } else if(tolower(method) %in% methods_rw) {
    return(decompR2_rw(response = response, predictors = predictors, data = data, weights = weights))
  } else {
    stop(paste0("Invalid `methods` value. Valid values are: ", methods_lmg, " for LMG and: ", methods_rw, " for relative weights."))
  }
}
```

```r
#' @title Calculate relative importance using the LMG method
#'
#' @description Internal function to calculate the relative variable importance using the LMG method.
#'
#' @param response is the response of the model, specified as a character string.
#' @param predictors is a character vector containing the names of the predictors to be used in the model.
#' @param data is a data.frame containing the data that will be used in the model.
#' @param weights is passed to `stats::lm` or `lme4::lmer`, depending on if there are random intercepts in the formula. See `?stats::lm` and `?lme4::lmer` for more details.
#'
#' @return Named vector with the relative importances of the predictors in `fixed_effects` and `random_intercepts`.
#' @author Andreas Matre
decompR2_lmg <- function(response,
                         predictors,
                         data,
                         weights = NULL) {

  # Rename the fixed effects to prevent problems from the fact that if f.ex. sin(x2) a predictor then the column in `data` for x2 is already transformed to sin(x2).
  # Thus, if trying to fit a model with for example y ~ x1 + sin(x2), the model will look for the column x2 in data and not find it.
  fixed_effects <- predictors[!grepl(pattern = '|', x = predictors, fixed = TRUE)]
  random_intercepts <- predictors[grepl(pattern = '|', x = predictors, fixed = TRUE)]
  new_predictors <- c()
  if(length(fixed_effects) > 0) {
    colnames(data)[colnames(data) %in% fixed_effects] <- paste0("f", 1:length(fixed_effects))
    new_predictors <- c(new_predictors, paste0("f", 1:length(fixed_effects)))
  }
  new_predictors <- c(new_predictors, random_intercepts)

  R2s <- list()

  res <- sapply(new_predictors, function(focus_pred) {
    other_preds <- new_predictors[new_predictors != focus_pred]
    subsets <- powerset(other_preds)

    R2_diffs <- sapply(subsets, function(subset) {
      subset_preds <- c(1, subset) # Add the intercept

      # Find R2 for small model
      sorted_preds <- sort(subset_preds) # Sort the predictors, to be able to tell if they have been calculated before
      # Check if the R2 has been calculated before
      R2_key <- paste0(sorted_preds, collapse = "")
      if (utils::hasName(R2s, R2_key)) {
        R2_small_model <- R2s[[paste0(sorted_preds, collapse = "")]]
      } else {
        #R2_small_model <- calc_R2(response = response, predictors = sorted_preds, data = data, weights = weights)
        formula_for_modelfit <- stats::as.formula(paste0(response, " ~ ",
                                                    paste0(sorted_preds, collapse = " + ")))
        model <- fit_model(formula = formula_for_modelfit, data = data, weights = weights)
        R2_small_model <- calc_R2(model = model)
        R2s[[R2_key]] <<- R2_small_model
      }

      # Find R2 for large model
      sorted_preds <- sort(c(sorted_preds, focus_pred)) # Sort the predictors, to be able to tell if they have been calculated before
      # Check if the R2 has been calculated before
      R2_key <- paste0(sorted_preds, collapse = "")
      if (utils::hasName(R2s, R2_key)) {
        R2_large_model <- R2s[[paste0(sorted_preds, collapse = "")]]
      } else {
        #R2_large_model <- calc_R2(response = response, predictors = sorted_preds, data = data, weights = weights)
        formula_for_modelfit <- stats::as.formula(paste0(response, " ~ ",
                                                    paste0(sorted_preds, collapse = " + ")))
        model <- fit_model(formula = formula_for_modelfit, data = data, weights = weights)
        R2_large_model <- calc_R2(model = model)
        R2s[[R2_key]] <<- R2_large_model
      }

      # Calculate the difference in R2 between the model with focus_pred and the one without
      diff <- R2_large_model - R2_small_model
      return(factorial(length(subset)) * factorial(length(other_preds) + 1 - length(subset) - 1) * diff)
    })
    return(1 / factorial(length(other_preds) + 1) * sum(R2_diffs))
  })
  names(res) <- predictors
  return(res)
}
```

```r
#' @title Calculate relative importance using the relative weights method
#'
#' @description Internal function to calculate the relative variable importance using the relative weights method.
#'
#' @param response is the response of the model, specified as a character string.
#' @param predictors is a character vector listing the predictors to be used in the full model.
#' @param data is a data.frame containing the data that will be used in the model.
#' @param weights is passed to `stats::lm` or `lme4::lmer`, depending on if there are random intercepts in the formula. See `?stats::lm` and `?lme4::lmer` for more details.
#' @return Named vector with the relative importances of the predictors in `fixed_effects` and `random_intercepts`.
#' @author Andreas Matre
#decompR2_rw <- function(response,
#                 fixed_effects,
#                 random_intercepts = NULL,
#                 data,
#                 weights = NULL){
decompR2_rw <- function(response,
                 predictors,
                 data,
                 weights = NULL){

  fixed_effects <- predictors[!grepl(pattern = '|', x = predictors, fixed = TRUE)]
  if(any(grepl(pattern = '|', x = predictors, fixed = TRUE))) {
    # Get the names of the columns in the random intercepts
    random_intercepts_full <- predictors[grepl(pattern = '|', x = predictors, fixed = TRUE)]
    lf <- lme4::lFormula(stats::as.formula(paste0(response, " ~ ", paste0(random_intercepts_full, collapse = " + "))), data = data)
    random_intercepts <- names(lf$reTrms$cnms)
  } else {
    random_intercepts <- NULL
  }

  data_fixed <- data[,fixed_effects, drop = FALSE]
  if(length(fixed_effects) > 0) {
    categorical_preds <- colnames(data_fixed)[which(sapply(data_fixed, is.factor))]
    numerical_preds <- setdiff(colnames(data_fixed)[which(!sapply(data_fixed, is.factor))], response)
  } else {
    categorical_preds <- character(0)
    numerical_preds <- character(0)
  }

  has_random_intercepts <- !(is.null(random_intercepts) || (length(random_intercepts) == 0))
  has_only_numerical <- (length(categorical_preds) == 0) && !has_random_intercepts

  # Restructure and scale data
  y <- data[,response]
  y <- scale(y)

  if(length(numerical_preds) > 0) {
    X <- data_fixed[,numerical_preds, drop = FALSE]
    X <- scale(X)

    # Calculate eigenvalues and eigenvectors
    e <- eigen(t(X) %*% X)
    Q <- e$vectors
    D <- diag(sqrt(e$values), nrow = nrow(Q))
    Dinv <- diag(1/sqrt(e$values), nrow = nrow(Q))

    # Calculate R_xx^(-0.5)
    R <- sqrt(nrow(X) - 1) * Q %*% Dinv %*% t(Q)

    # Calculate the transformed numerical fixed effects
    Z <- X %*% R

    lambda <- 1 / sqrt(nrow(X) - 1) * Q %*% D %*% t(Q)
  } else {
    Z <- NULL
  }

  # Create new data.frame with the transformed numerical fixed effects
  data_Z <- as.data.frame(cbind(y, data_fixed[,categorical_preds, drop = FALSE], data[,random_intercepts, drop = FALSE]))
  if(!is.null(Z)) {
    data_Z <- cbind(data_Z, Z)
  }
  if(length(numerical_preds) > 0) {
    numerical_preds_Z <- paste0("n", 1:length(numerical_preds))
  } else {
    numerical_preds_Z <- c()
  }
  if(length(categorical_preds) > 0) {
    categorical_preds_Z <- paste0("c", (length(numerical_preds) + 1):(length(numerical_preds) + length(categorical_preds)))
  } else {
    categorical_preds_Z <- c()
  }
  #colnames(data_Z) <- c(response, numerical_preds_Z, categorical_preds_Z, random_intercepts)
  colnames(data_Z) <- c(response, categorical_preds_Z, random_intercepts, numerical_preds_Z)

  if(has_only_numerical) {
    f <- stats::as.formula(paste0(response, " ~ ."))
    model <- fit_model(formula = f, data = data_Z, weights = weights)
    beta <- get_fixed_coef(model = model)[-1]
    result <- as.vector(lambda^2 %*% beta^2)
  } else {

    # The groups of predictors to look at subsets of. Same as all the predictors, except that the numerical fixed effects are always considered together.
    # This reduces the number of subsets to look at.
    groups <- list()
    if(length(numerical_preds_Z) > 0) {
      groups <- c(groups, list(numerical_preds_Z))
    }
    #groups <- list(numerical_preds_Z)
    groups <- c(groups, as.list(categorical_preds_Z))
    if(has_random_intercepts) {
      groups <- c(groups, as.list(paste0("(1 | ", random_intercepts, ")")))
    }

    # Create lists to save the calculated R2 values and coefficients.
    # This saves calculations, as models will be fitted several times.
    R2s <- list()
    coefs <- list()

```

```r
# Look at all predictors and find their importance
result <- lapply(groups, function(focus_pred) {
  other_preds <- groups[!sapply(groups, function(pred) identical(focus_pred, pred))]

  # Look at all subsets of the other predictors
  R2_diffs <- lapply(powerset(other_preds), function(subset) {
    subset_preds <- unlist(c(1, subset)) # Add intercept to set of predictors

    # Find R2 for small model
    sorted_preds <- sort(subset_preds) # Sort the predictors, to be able to tell if they have been calculated before
    # Check if the R2 has been calculated before
    key <- paste0(sorted_preds, collapse = "")
    if (utils::hasName(R2s, key)) { # R2s and coefs will always have the same keys
      R2_small_model <- R2s[[key]]
    } else {
      formula_for_modelfit <- stats::as.formula(paste0(response, " ~ ",
                                                  paste0(sorted_preds, collapse = " + ")))

      model <- fit_model(formula = formula_for_modelfit, data = data_Z, weights = weights)
      R2_small_model <- calc_R2(model = model)
      fixed_coefs <- get_fixed_coef(model = model)
      fixed_coefs <- fixed_coefs[intersect(names(fixed_coefs), numerical_preds_Z)] # Choose only the coefficients for the numerical predictors

      # Save the R2 and coefficients to use later
      R2s[[key]] <<- R2_small_model
      coefs[[key]] <<- fixed_coefs
    }

    # Find R2 for large model
    sorted_preds <- sort(c(sorted_preds, focus_pred)) # Sort the predictors, to be able to tell if they have been calculated before
    # Check if the R2 has been calculated before
    key <- paste0(sorted_preds, collapse = "")
    if (utils::hasName(R2s, key)) {
      R2_large_model <- R2s[[paste0(sorted_preds, collapse = "")]]
    } else {
      formula_for_modelfit <- stats::as.formula(paste0(response, " ~ ",
                                                  paste0(sorted_preds, collapse = " + ")))

      model <- fit_model(formula = formula_for_modelfit, data = data_Z, weights = weights)
      R2_large_model <- calc_R2(model = model)
      fixed_coefs <- get_fixed_coef(model = model)
      fixed_coefs <- fixed_coefs[intersect(names(fixed_coefs), numerical_preds_Z)] # Choose only the coefficients for the numerical predictors

      # Save the R2 and coefficients to use later
      R2s[[key]] <<- R2_large_model
      coefs[[key]] <<- fixed_coefs
    }

    # Calculate the difference in R2 between the model with focus_pred and the one without
```

```r
        diff <- R2_large_model - R2_small_model

        # Weigh the R2 difference
        share <- factorial(length(subset)) * factorial(length(other_preds) + 1 - length(subset) - 1) * diff

        # If focus_pred is the numerical fixed effects, distribute the R2-difference to each numerical fixed effect
        if(identical(focus_pred, numerical_preds_Z)){
          beta <- coefs[[key]]
          shares <- lambda^2 %*% beta^2

          share <- share * shares / sum(shares) # Weigh the shares such that we get a proper decomposition
        }
        return(share)
      })
      # If focus_pred is the numerical fixed effects R2_diffs is a list of vectors
      # In that case, we need to make it into a matrix and then take the row sums, to get the share for each numerical fixed effect
      R2_diffs <- do.call(cbind, R2_diffs)
      return(1 / factorial(length(other_preds) + 1) * rowSums(R2_diffs))
    })
  }

  result <- unlist(result) # Make all elements have the same level in the list
  name <- c(numerical_preds, categorical_preds)
  if(has_random_intercepts) {
    name <- c(name, paste0("(1 | ", random_intercepts, ")"))
  }
  names(result) <- name
  return(result)
}
```

```
##' Checks if a formula contains a term of the form "(*something* | *something else*)", which is the way to specify random effects in `lme4::lmer` models.
##' Currently, this function rather naive and only checks if "|" is in each term.
##'
##' @title Check if a formula contains a random effect
##' @param f formula
##' @return logical vector, TRUE if a term is a random effect, FALSE if non of the terms are random effects.
##' @author Andreas Matre
formula_contains_RE <- function(f) {
  # Currently naive, only checks for '|' in the formula. Might need more advanced check later.
  any(grepl(pattern = '|', x = f, fixed = TRUE))
}

##' @title Fit `stats::lm` or `lme4::lmer` model
##' @description `lme4::lmer` does not support fitting a model without random effects, so this is a wrapper function that fits a `stats::lm` model if there are no random effects in the formula and a `lme4::lmer` model if there are random effects in the formula.
##' @param formula is a formula used to fit the model
##' @param data is a data.frame used to fit the model
##' @param ... is other arguments to pass to `stats::lm` or `lme4::lmer`
##' @return the output from either `stats::lm` or `lme4::lmer`
##' @author Andreas Matre
fit_model <- function(formula, data, ...) {
  if(formula_contains_RE(formula)) {
    return(eval(substitute(lme4::lmer(formula = formula, data = data, ...)), parent.frame()))
  } else {
    return(eval(substitute(stats::lm(formula = formula, data = data, ...)), parent.frame()))
  }
}

##' @title Get the coefficients of the fixed effects
##' @description Gets the coefficients of the fixed effects from either a `stats::lm` or `lme4::lmer` model.
##' @param model is a model from either `stats::lm` or `lme4::lmer`
##' @return named vector containing the coefficients of the fixed effects of `model`.
##' @author Andreas Matre
get_fixed_coef <- function(model) {
  if(inherits(x = model, what = "merMod")) {
    return(lme4::fixef(model))
  } else if(inherits(x = model, what = "lm")) {
    return(stats::coef(model))
  } else {
    stop("Only supports lme4::lmer and stats::lm models")
  }
}

##' @title Calculate the R^2 of a random intercept model
##'
##' @description Calculates the R^2 of an 'lme4::lmer' random intercept model. This function will not work for more complicated 'lme4::lmer' models.
##'
##' @details The calculation is based on the ideas introduced in Nakagawa, Shinichi and Holger Schielzeth (2013).
##' @param m merMod object created by 'lme4::lmer'
##' @return numeric, the R^2 value of 'm'
##' @author Andreas Matre
```

```r
calc_lmer_R2 <- function(m) {
  if(inherits(m, "merMod")) {
    fixed_effects <- setdiff(names(lme4::fixef(m)), "(Intercept)") # We don't need the intercept, as adding a constant doesn't change the variance
    X <- stats::model.matrix(m)
    fixed_var <- stats::var(as.vector(lme4::fixef(m)[-1] %*% t(as.matrix(X[,fixed_effects]))))

    random_var <- do.call(sum, lme4::VarCorr(m))
    residual_var <- stats::sigma(m)^2

    return((fixed_var + random_var)/(fixed_var + random_var + residual_var))
  } else {
    stop("Only supports lme4::lmer models")
  }
}

##' @title Calculate the R^2 of a `stats::lm` or `lme4::lmer` model.
##' @description Calculates the R^2 of a `stats::lm` or `lme4::lmer` model. If the model is from `lme4::lmer` it can only handle random intercept models. Random slopes are not handled.
##' @details For a `stats::lm` model calc_R2 uses the value from `summary(model)` while for a `lme4::lmer` model the conditional R^2 introduced by Nakagawa, Shinichi and Holger Schielzeth (2013) will be calculated.
##' @param model is a model from either `stats::lm` or `lme4::lmer`
##' @return numeric
##' @author Andreas Matre
##' @export
calc_R2 <- function(model) {
  if(inherits(x = model, what = "merMod")) {
    return(calc_lmer_R2(model))
  } else if(inherits(x = model, what = "lm")) {
    return(summary(model)$r.squared)
  } else {
    stop("Only supports lme4::lmer and stats::lm models")
  }
}

##' @title Create powerset
##'
##' @description
##' Internal function creating the powerset of 'x'. The powerset of `x` is a set including all possible subsets of `x`.
##' As each element in `x` can either be or not be in a subset, the powerset of `x` will include 2^length(x) sets.
##'
##' @param x vector containing the set of elements to create a powerset of.
##'
##' @details
##' Removes duplicates in 'x' if there are any.
##'
##' @return Returns a list with 2^length(x) elements containing the powerset of 'x'.
##'
##' @examples
##' #powerset(c(1, 2, 3))
##' # Should return list(c(), c(1), c(2), c(3), c(1, 2), c(1, 3), c(2, 3), c(1, 2, 3))
##' @author Andreas Matre
powerset <- function(x) {
  if(length(x) == 0) {return(list())}

  x <- unique(x)

  res <- lapply(1:length(x), function(n) {
    m <- utils::combn(x, n)
    lapply(1:ncol(m), function(i) m[,i])
  })

  res <- unlist(res, recursive = FALSE)
  res[[length(res) + 1]] <- vector(mode = typeof(x)) # Add the empty set

  return(res)
}
```

C Simulation of data

```r
##' @title Create test data
##' @description Creates data that is used to test the functionality of the package.
##'
##' @param n.obs is numeric. Is the number of observations to create. Only used if there are no random effects.
##' @param fixed.coef is a numeric vector. Is the coefficients of the fixed effects.
##' @param fixed.cov is a numeric matrix. Is the covariance matrix of the fixed effects.
##' @param random.var is a numeric vector. Is the variance of each random intercept
##' @param random.n.levels is a numeric vector. Is the number of levels for each random intercept.
##' @param random.n.in.groups is a numeric. Is the number of times each combination of random intercepts are repeated.
##' @param residuals.var is a numeric. Is the variance of the residuals.
##' @return a data.frame with either `n.obs` or `prod(random.n.levels) * random.i.in.groups` rows and `1 + length(fixed.coef) + length(random.var)` columns. There is one column for each fixed effect and each random intercept and one column for the response.
##'         The fixed effects get column names `x1`, `x2`, ... and the random intercepts get column names `group1`, `group2`, ... while the response gets column name `y`.
##' @author Andreas Matre
create_test_data <- function(n.obs = NULL, fixed.coef, fixed.cov = diag(length(fixed.coef)),
                            random.var = NULL, random.n.levels = NULL, random.n.in.groups = 1,
                            residuals.var = 1)
  {
  if(length(fixed.cov) == 1 && !is.matrix(fixed.cov)) {fixed.cov <- matrix(fixed.cov)}
  if(length(random.var) == 1 && !is.matrix(random.var)) {random.var <- matrix(random.var)}
  if(is.null(random.var) & missing(n.obs)) {stop("`n.obs` needs to be specified if there are no random effects.")}

  if(!is.null(random.var)) {
    n.obs <- prod(random.n.levels) * random.n.in.groups
  }

  # Create fixed effects
  X <- MASS::mvrnorm(n = n.obs, mu = rep(0, length(fixed.coef)), Sigma = fixed.cov)
  X <- apply(X, 2, function(x) (x - mean(x)) / stats::sd(x)) # Standardize

  # Create response calculated by just the fixed effects and residual
  y <- X %*% fixed.coef + stats::rnorm(n = n.obs, mean = 0, sd = sqrt(residuals.var))

  # If there are random effects, add them as well
  if(!is.null(random.var) && !is.null(random.n.levels)) {
    group <- do.call(expand.grid, lapply(random.n.levels, function(n) 1:n))
    group <- as.data.frame(group[rep(1:nrow(group), each = random.n.in.groups),])
    rownames(group) <- NULL
    alphas <- lapply(1:length(random.var), function(i) {
      stats::rnorm(n = random.n.levels[i], mean = 0, sd = sqrt(random.var[i]))
    })
    # Add the random intercepts to the response
    for(i in 1:length(alphas)) {
    y <- y + alphas[[i]][group[,i]]
    }
  }

  df <- as.data.frame(X)
  colnames(df) <- paste0("x", 1:length(fixed.coef))
  if(!is.null(random.var) && !is.null(random.n.levels)) {
    if(length(random.var) == 1) {
      colnames(group) <- "group"
    } else {
      colnames(group) <- paste0("group", 1:ncol(group))
    }
    df <- cbind(df, group)
  }
  df$y <- y

  return(df)
}
```

NTNU

Norwegian University of
Science and Technology