

The variable importance metric LMG applied to Bayesian linear regression models

Master Thesis in Biostatistics (STA495)

by

Silvano Sele

09 734 492

supervised by

Dr. Stefanie Muff

Name of supervisor (with title and affiliation if external)

Zurich, September, 2018

Abstract

Quantifying the importance of predictors in regression has been an active area of research for a long time (Grömping, 2015). The variance decomposition metric *LMG* provides useful information about possible associations between variables. The LMG metric is implemented in the R packages `hier.part` (Walsh and Nally, 2015) and `relaimpo` (Grömping, 2006). Bayesian methods gained high popularity in many applied research areas in recent years.

This master thesis shows how the LMG metric can be applied to a linear regression model that is fitted in the Bayesian framework. The LMG metric requires calculation of R^2 for the possible submodels. The conditional variance formula can be applied to calculate the R^2 values of the submodels from the posterior samples of the model containing all predictors. The mutual interdependence of the submodels is then respected for each posterior sample.

The implementation of the LMG metric in the Bayesian framework is presented on simulated and on empirical data. Using weakly informative priors resulted in very similar LMG values as the values obtained by using bootstrap in `relaimpo`.

Some possible extension of the LMG metric to repeated measures studies are additionally presented. There are certain difficulties involved in quantifying the R^2 in longitudinal data. However, the extension seems to be reasonably possible for the simple random intercept model and marginal models.

Contents

Abstract	i
Preface	v
1 Introduction	1
2 Theoretical background	3
2.1 LMG variable importance metric	3
2.2 Appropriate R^2 definitions in the Bayesian framework	4
2.3 Use of conditional variance formula to get R^2 of submodels	5
2.4 Bayesian Regression	7
3 Examples	9
3.1 Simulated data	9
3.2 Empirical data	13
4 Extension to longitudinal data	17
4.1 Random intercept model	17
4.2 Marginal model	19
5 Discussion and Conclusion	21
5.1 Other variable importance metrics in the Bayesian framework	21
5.2 Conclusion	21
A Additional notes and R-codes	23
A.1 Code used in chapter 3	23
A.2 Code used in chapter 4	30
Bibliography	35

Preface

Wenns scheisse läuft, läuft's scheisse! Oliver Kahn

Max Muster
June 2018

Chapter 1

Introduction

The objective of this master thesis is to implement the variable importance measure LMG (named after the authors Lindeman, Merenda, and Gold ([Grömping, 2007](#))) in linear models estimated with Bayesian methods. Bayesian methods have gained popularity because they allow to quantify the uncertainty about parameters and they allow to include prior information.

Regression models are popular in many applied research areas ([Nimon and Oswald, 2013](#)). These models provide a tool to find an association between a response variable and a set of explanatory variables. The explanatory variables are also called predictors or covariates. Regression parameters provide information to what extent the response variable is expected to change when one predictor changes by one unit, given all other predictors in the model remain the same. Being aware of this last remark is very important for the correct interpretation of the regression parameters. It shows that the parameter value of a predictor is dependent on the other predictors in the model.

Because predictors are often correlated to some degree to each other, it is obviously not an easy task to find the most important predictors in a model. The first question is, what is meant by the importance of a predictor? There is no easy answer to this question and it is depending on the research issue. [Grömping \(2015\)](#) concludes that there may never be a unique definition of variable importance. There exist different metrics to quantify the importance of predictors. These metrics focus on different aspects and with correlated predictors they lead to different conclusions. A summary of variable importance metrics can be found in [Grömping \(2015\)](#).

A distinction should be made between the importance of predictors in regression models that are used to predict future data and in regression models, applied to find an association between predictors and the response variable. In the first case, the aim is only to reduce the error between the predicted values and the observable values. The underlying association between predictors is of minor importance. In the second case, the focus is on the strength of the relationship between the predictors and the response variable. A predictor may explain little of the response variable, given two other correlated predictors are already included in a regression model. However, this predictor that is unimportant from the regression output may be the main cause of the other two predictors. Therefore, it may be the most important predictor of this regression model ([Grömping, 2007](#)).

The causal relationship between the variables is missing in standard regression models. Studying a predictor, given other variables are already included or using models that contain only the predictor itself, provide only some parts of the bigger picture about the predictor in a model. Which are the most useful variable importance metrics is still an open debate. A convincing theoretical basis is still lacking for all of those metrics. [Grömping \(2015\)](#) recommends to use the existing best practices, until a more profound solution will be found. For variance (or generally goodness-of-fit) decomposition based importance, she recommends to use LMG enhanced with joint contributions or dominance analysis ([Grömping, 2015](#)).

Chapter 2

Theoretical background

2.1 LMG variable importance metric

The focus of this master thesis is on the LMG variable importance metric. The LMG is a metric that is based on variance decomposition. The total R^2 of a model is decomposed onto the predictors. Marginal and conditional information are incorporated (Grömping, 2015). The formulas of this section are taken from Grömping (2015), using the same mathematical notations.

The following notations for the explained variance (2.1) and sequentially added variance (2.2) simplify the notation of the LMG formula.

$$\text{evar}(S) = \text{Var}(Y) - \text{Var}(Y \mid X_j, j \in S), \quad (2.1)$$

$$\text{svar}(M \mid S) = \text{evar}(M \cup S) - \text{evar}(S), \quad (2.2)$$

where S and M denote disjoint sets of predictors.

The LMG formula is given below for the first predictor only. Because of exchangeable predictors, this is no loss of generality. $R^2(S)$ can be written as $\text{evar}(S) / \text{Var}(Y)$.

$$\text{LMG}(1) = \frac{1}{p!} \sum_{\pi \text{ permutation}} \text{svar}(\{1\} \mid S_1(\pi)), \quad (2.3)$$

$$= \frac{1}{p!} \sum_{S \subseteq \{2, \dots, p\}} n(S)! (p - n(S) - 1)! \text{svar}(\{1\} \mid S) \quad (2.4)$$

$$= \frac{1}{p} \sum_{i=0}^{p-1} \left(\sum_{\substack{S \subseteq \{2, \dots, p\} \\ n(S)=1}} \text{svar}(\{1\} \mid S) \right) \bigg/ \binom{p-1}{i} \quad (2.5)$$

where $S_1(\pi)$ is the set of predecessors of predictor 1.

The different formula writings help to better understand what the calculation is about in the LMG metric. The R^2 of the model including all predictors is decomposed. In the formula on the top (2.3), the LMG value of predictor 1 is represented as an unweighted average over all orderings of the sequential added variance contribution of predictor 1. The formula in the center (2.4), shows that the calculation can be done more efficiently. The orderings with the same set of predecessors S are combined into one summand. Instead of $p!$ summands only 2^{p-1} summands need to be calculated. The formula on the bottom (2.5) shows that the LMG metric can also be seen as the unweighted average over average explained variance improvements when adding predictor 1 to a model of size i without predictor 1 (Grömping, 2015). The LMG metric is implemented in the R package `relaimpo` (Grömping, 2006).

Chevan and Sutherland (1991) propose that, instead of only using the variances, an appropriate goodness-of-fit metric can as well be used in the LMG formula. They name their proposal *hierarchical partitioning*. The requirements are simply: an initial measure of fit when no predictor variable is present, a final measure of fit when N predictor variables are present, all intermediate models when various combinations of predictor variables are present. The LMG component of each variable is named *independent component* (I). The sum of the independent components (I) results then in the overall goodness-of-fit metric. The difference between the goodness-of-fit when only the predictor itself is included in the model, compared to its independent component (I), is named the *joint contribution* (J) (Grömping, 2015). Hierarchical partitioning is implemented in the `hier.part` package (Walsh and Nally, 2015). When R^2 is chosen as the goodness-of-fit measure, the LMG values are calculated. The hierarchical partitioning function of `hier.part` is used in this master thesis. The hierarchical partitioning function accepts a data frame with the R^2 values of all submodels as input. Of note, the partitioning function of `hier.part` is only guaranteed to work up to nine predictors and does not work at all for more than twelve predictors.

2.2 Appropriate R^2 definitions in the Bayesian framework

The focus of this master thesis is on the standard linear model. For this model, the most widely used goodness-of-fit metric is R^2 . Different formulas for R^2 exist Kvalseth (1985), all leading to the same value when an intercept is included and the model is fitted by maximum likelihood.

Two commonly used R^2 definitions are:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (2.6)$$

$$R^2 = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad i = 1, \dots, n, \quad (2.7)$$

where $\hat{y}_i = E(y | X_i, \hat{\theta})$. $\hat{\theta}$ is the maximum likelihood estimate of the regression coefficients.

When other estimation methods than maximum likelihood are used, equation (2.6) can be negative and equation (2.7) can be bigger than 1. This is not uncommon in a Bayesian regression setting when samples of the posterior parameter distribution are employed. A model that explains more than 100% of the variance is nonsense. A negative R^2 is also difficult to interpret. A negative R^2 may be interpreted as a fit that is worse than the mean of the data. This can make sense for predictive purposes, e.g. when new data from a test set is predicted by leave-one-out crossvalidation (Alexander *et al.*, 2015). For non predicting purposes, a negative R^2 does not make sense. The aim of the LMG formula is to gain some more information about the possible association between variables. A predictor can not explain less than zero variance in the population. To respect the non-negative share property of the LMG formula, the R^2 of submodels should not decrease when adding predictors. Both classical R^2 definitions seem not to be well suited for the LMG metric in the Bayesian framework.

A more reasonable R^2 definition for the LMG formula in the Bayesian framework can be found by noting that the variance of the linear model can also be written as

$$\text{Var}(y) = \text{Var}(\mathbf{X}\boldsymbol{\beta}) + \sigma^2 = \boldsymbol{\beta}^\top \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{X}} \boldsymbol{\beta} + \sigma^2, \quad (2.8)$$

where $\boldsymbol{\beta}^\top = (\beta_1 \dots \beta_p)$ are the regression parameters without the intercept. $\boldsymbol{\Sigma}_{\mathbf{X}\mathbf{X}}$ is the covariance matrix of the predictors.

By using this variance definition Gelman *et al.* (2017) propose to use

$$R_{Gelman}^2 = \frac{\text{Var}(\sum_{i=1}^n \hat{y}_i^s)}{\text{Var}(\sum_{i=1}^n \hat{y}_i^s) + \text{Var}(\sum_{i=1}^n e_i^s)}, \quad i = 1, \dots, n, \quad (2.9)$$

where $\hat{y}_i^s = E(y | X_i, \theta^s)$ and the vector of errors $e_i^s = y_i - \hat{y}_i^s$ and $\theta^s, s = 1, \dots, S$ are draws from the posterior parameter distribution. The R^2 is then guaranteed to be between 0 and 1. The R^2 can be interpreted as a data-based estimate of the proportion of variance explained for new data under the assumption that the predictors are held fixed (Gelman *et al.*, 2017).

In the Bayesian framework, the σ^2 parameter is explicitly modeled in the standard linear regression setting. Therefore, it is possible to sample the σ^2 parameter from its posterior distribution instead of defining the error as in definition (2.9), which would lead to the following definition:

$$\begin{aligned} R^2 &= \frac{\text{Var}(\sum_{i=1}^n \hat{y}_i^s)}{\text{Var}(\sum_{i=1}^n \hat{y}_i^s) + \sigma_s^2} \\ &= \frac{\beta_s^\top \Sigma_{\mathbf{X}\mathbf{X}} \beta_s}{\beta_s^\top \Sigma_{\mathbf{X}\mathbf{X}} \beta_s + \sigma_s^2}, \quad i = 1, \dots, n, \end{aligned} \quad (2.10)$$

where $\hat{y}_i^s = E(y | X_i, \theta^s)$, and $\theta^s, s = 1, \dots, S$ are draws from the posterior parameter distribution.

The predictors in definition (2.9) and definition (2.10) could also be taken as random (Gelman *et al.*, 2017). The predictors are then called stochastic predictors. Using the sample covariance estimate provides then just an estimate of the true covariance structure. With stochastic predictors, there is an additional uncertainty in the R^2 formula that can have a large influence on the R^2 and the LMG values.

In practice, definition (2.10) and definition (2.9) should lead to similar values in the standard linear model. In my opinion, it is more reasonable to take the full Bayesian route by sampling σ^2 of its posterior distribution. This approach provides the opportunity to include prior information about σ^2 directly into to R^2 calculations. The LMG calculations in the examples of this master thesis will therefore be based on definition (2.10). A benefit of definition (2.9) is that it also works for generalized linear models where we often have no separate variance parameter.

The denominator of R^2 is no longer fixed in definition (2.9) and in definition (2.10). We can therefore no longer interpret an increase in R^2 as an improved fit to a fixed target (Gelman *et al.*, 2017). The unfixed denominator seems to be problematic for the LMG formula in the Bayesian framework. However, in the linear model it is possible to calculate the R^2 of all submodels from the parameters of the model including all predictors (full-model) and the covariance matrix of the predictors. Therefore, all submodels of a posterior sample are compared to the same fixed value. A possible way to get the R^2 of the submodels from the full-model is shown in the next section.

2.3 Use of conditional variance formula to get R^2 of submodels

For two predictors, definition (2.8) simplifies to

$$\text{Var}(y) = \beta_1^2 \text{Var}(X_1) + 2\beta_1\beta_2 \text{Cov}(X_1, X_2) + \beta_2^2 \text{Var}(X_2) + \sigma^2, \quad (2.11)$$

When predictor X_1 is the only one in the model, the explained variance includes the variance of the predictor itself, the whole covariance term, and some of the contribution of the variance of X_2 in equation (2.11) additionally. In mathematical notation, that is

$$\text{svar}(X_1 \mid \emptyset) = \beta_1^2 \text{Var}(X_1) + 2\beta_1\beta_2 \text{Cov}(X_1, X_2) + \beta_2^2 \text{Var}(X_2)\rho_{12}^2.$$

The contribution of the second regressor is then simply the difference to the total explained variance (Grömping, 2007).

In the general case with p regressors, the conditional variance formula (2.12) can be used to calculate the R^2 of all submodels. For example, the conditional variance formula can be used to specify the conditional distribution of a multivariate normal distribution \mathbf{Y} .

The elements of the vector \mathbf{Y} are reordered as

$$\mathbf{Y} = \begin{pmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \end{pmatrix}, \mathbf{Y}_1 \in \mathbb{R}^q, \mathbf{Y}_2 \in \mathbb{R}^{p-q}.$$

The joint distribution is a multivariate normal distribution with elements

$$\begin{pmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}\right), \boldsymbol{\Sigma}_{21} = \boldsymbol{\Sigma}_{12}^T,$$

the conditional distribution is normally distributed again with mean

$$\mathbb{E}(\mathbf{Y}_1 \mid \mathbf{y}_2) = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{Y}_2 - \boldsymbol{\mu}_2),$$

and the conditional variance is

$$\text{Var}(\mathbf{Y}_1 \mid \mathbf{y}_2) = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}. \quad (2.12)$$

The aim is to calculate R^2 of a submodel containing the predictors $\mathbf{X}_{q\dots p}$, and the regression coefficients $\boldsymbol{\beta}^\top = (\beta_1, \dots, \beta_p)$ without the intercept. The regression coefficients are further separated in $\boldsymbol{\beta}_{1,\dots,q-1}^\top = (\beta_1, \dots, \beta_{q-1})$ and $\boldsymbol{\beta}_{q,\dots,p}^\top = (\beta_q, \dots, \beta_p)$.

As in the multivariate normal distribution example above, the covariance matrix of p predictors is written as

$$\text{Cov}(\mathbf{X}) = \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{X}} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}^{p \times p},$$

$$\begin{aligned} \text{where} \quad \boldsymbol{\Sigma}_{11} &= \text{Cov}(\mathbf{X}_{1,\dots,q-1}, \mathbf{X}_{1,\dots,q-1}), \\ \boldsymbol{\Sigma}_{12} &= \text{Cov}(\mathbf{X}_{1,\dots,q-1}, \mathbf{X}_{q,\dots,p}), \\ \boldsymbol{\Sigma}_{22} &= \text{Cov}(\mathbf{X}_{q,\dots,p}, \mathbf{X}_{q,\dots,p}). \end{aligned}$$

The conditional variance of the predictors $\mathbf{X}_{1,\dots,q-1}$ given the predictors $\mathbf{X}_{q,\dots,p}$ is then

$$\text{Cov}(\mathbf{X}_{1,\dots,q-1} \mid \mathbf{x}_{q,\dots,p}) = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}.$$

The total explained variance of the full-model containing $\mathbf{X}_{1\dots p}$ omits simply the σ^2 parameter in (2.8), which is

$$\text{evar}(\mathbf{X}_{1,\dots,p}) = \boldsymbol{\beta}^\top \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{X}} \boldsymbol{\beta}.$$

The explained variance of a submodel can be calculated by subtracting the explained variance of the not-in-the-model-included-predictors that is not explained by in-the-model-included-predictors from the total explained variance. The variance that is not explained by in-the-model-included-predictors is given by the variance of the not-in-the-model-included predictors conditional on the in-the-model-included-predictors. The explained variance of a submodel containing predictors $\mathbf{X}_{q,\dots,p}$ can therefore be written as

$$\text{evar}(\mathbf{X}_{q,\dots,p}) = \text{evar}(\mathbf{X}_{1,\dots,p}) - \boldsymbol{\beta}_{1,\dots,q-1}^\top \text{Cov}(\mathbf{X}_{1,\dots,q-1} \mid \mathbf{x}_{q,\dots,p}) \boldsymbol{\beta}_{1,\dots,q-1}. \quad (2.13)$$

To gain the R^2 value of the submodel, it is necessary to divide the explained variance by the total variance, which is

$$\text{evar}(\mathbf{X}_{q,\dots,p}) / \text{Var}(\mathbf{Y}),$$

where $\text{Var}(\mathbf{Y})$ is defined as $\boldsymbol{\beta}^\top \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{X}} \boldsymbol{\beta} + \sigma^2$.

A posterior density distribution is obtained for the regression parameters in the Bayesian regression setting. The LMG formula requires calculation of the R^2 values for all $2^p - 1$ submodels. Samples from the joint posterior parameters of the full-model are used to calculate the explained variance of the submodels. For each sample, the conditional variance formula is used to obtain the R^2 of the $2^p - 1$ submodels. The non-negative property and the dependence of the parameters from the submodels to each other is then respected for each sample.

Instead of using the conditional mean formula to get the R^2 of the submodels, it would be possible to fit a separate Bayesian model for each submodel. An R^2 distribution can easily be built for each submodel by using definition (2.9) or definition (2.10). However, the problem is how to calculate the LMG values out of these R^2 distributions. If we just sample independently from the R^2 distributions, the dependence of the parameter values of the submodels to each other is ignored. We would have many possibly true parameter values of a predictor in the same LMG comparison. It would then also be possible that the R^2 decreases when adding predictors. Another drawback is that it would be much more time-consuming to fit a separate Bayesian model for each submodel. Using the conditional variance formula on the full-model allows to calculate LMG values in the Bayesian framework in a reasonable time exposure. Depending on the number of predictors and the number of posterior samples, the calculations still take some time in the Bayesian framework. For stochastic predictors, the computation time is multiplied by the number of covariance samples.

2.4 Bayesian Regression

The following section provides a brief introduction to Bayesian regression. It further shows that assuming stochastic or non-stochastic predictors results in the same posteriors for the regression parameters under some assumptions. It is summarized from the book *Bayesian Analysis for the Social Sciences* (Jackman, 2009).

In regression analysis, we are interested in the dependence of \mathbf{y} on \mathbf{X} . The conditional mean of a continuous response variable $\mathbf{y} = (y_1, \dots, y_n)^\top$ is related to a $n \times k$ predictor matrix \mathbf{X} via a linear model,

$$\mathbf{E}(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\beta}) = \mathbf{X}\boldsymbol{\beta},$$

where $\boldsymbol{\beta}$ is a $k \times 1$ vector of unknown regression coefficients.

Under some assumptions about the density, conditional independence and homoskedastic variances, the regression setting can be written as

$$\mathbf{y} \mid \mathbf{X}, \beta, \sigma^2 \sim \mathcal{N}(\mathbf{X}\beta, \sigma^2 \mathbf{I}_n).$$

Under the assumption of weak exogeneity and conditional independence, the joint density of the data can be written as

$$p(\mathbf{y}, \mathbf{X} \mid \boldsymbol{\theta}) = p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}_{y|x}) p(\mathbf{X} \mid \boldsymbol{\theta}_x),$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_{y|x}, \boldsymbol{\theta}_x)^\top$.

The weak exogeneity assumption implicates that the whole information about \mathbf{y}_i is contained in x_i and $\boldsymbol{\theta}_{y|x}$. Knowledge of the parameters $\boldsymbol{\theta}_{x_i}$ provides no additional information about \mathbf{y}_i . The interest of regression is mostly on the posterior parameters $\boldsymbol{\theta}_{y|x}$. These posterior densities are proportional to the likelihood of the data multiplied by the prior density. The joint density $p(\mathbf{y}, \mathbf{X} \mid \boldsymbol{\theta})$ is used to learn about the posterior parameters, via Bayes Rule

$$p(\boldsymbol{\theta} \mid \mathbf{y}, \mathbf{X}) \propto p(\mathbf{y}, \mathbf{X} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}).$$

The dependence of \mathbf{y} on \mathbf{X} is captured in the parameters $\boldsymbol{\theta}_{y|x} = (\beta, \sigma^2)$. Under the assumption of independent prior densities about $\boldsymbol{\theta}_{y|x}$ and $\boldsymbol{\theta}_x$ the posterior distribution of the parameters can be written as

$$p(\beta, \sigma^2, \boldsymbol{\theta}_x \mid \mathbf{y}, \mathbf{X}) = \frac{p(\mathbf{y} \mid \mathbf{X}, \beta, \sigma^2) p(\beta, \sigma^2)}{p(\mathbf{y} \mid \mathbf{X})} \times \frac{p(\mathbf{X} \mid \boldsymbol{\theta}_x) p(\boldsymbol{\theta}_x)}{p(\mathbf{X})}. \quad (2.14)$$

The factorization in equation 2.14 shows, that under the above mentioned assumptions, the posterior inference about the parameters $\boldsymbol{\theta}_{y|x} = (\beta, \sigma^2)$ is independent from the inference about $\boldsymbol{\theta}_x$ given data \mathbf{X} . This also means that the assumptions about \mathbf{X} being non-stochastic or stochastic result in the same posterior density of $\boldsymbol{\theta}_{y|x}$. In the case of non-stochastic regressors, $p(\mathbf{X})$ and $\boldsymbol{\theta}_x$ drop out of the calculations. For stochastic predictors, it means, that given \mathbf{X} , nothing more can be gained about $\boldsymbol{\theta}_{y|x} = (\beta, \sigma^2)$ from knowing $\boldsymbol{\theta}_x$.

The focus of regression is on $\boldsymbol{\theta}_{y|x} = (\beta, \sigma^2)$, for which it does not matter whether we assume fixed or stochastic predictors under the above mentioned assumptions. The variance of the predictors is also incorporated in the LMG formula. The LMG formula may be especially interesting for continuous predictors, which often are of stochastic nature. Grömping (2006) recommends in most cases to use the non fixed regressor option when calculating bootstrap confidence intervals. Therefore, the information about $\boldsymbol{\theta}_x$ would also be relevant for stochastic regressors. As seen in equation (2.14), inference about $\boldsymbol{\theta}_x$ is independent from inference about $\boldsymbol{\theta}_{y|x}$. If there are stochastic predictors and we use the sample estimate of the covariance matrix, we do not incorporate the uncertainty of the estimate. Because the explained variance is calculated by $\beta^\top \Sigma_{\mathbf{X}\mathbf{X}} \beta$, inference about $\boldsymbol{\theta}_x$ seems to be equally important as inference about $\boldsymbol{\theta}_{y|x}$ for stochastic predictors. If the distribution of the $p(\mathbf{X})$ is known, the $\boldsymbol{\theta}_x$ could be estimated. However, the computation time is then much higher, because the whole LMG calculation need to done for each posterior covariance sample of the predictors. Depending on the number of predictors this would be very time-consuming. In most cases, the problem is that the distribution of the \mathbf{X} is unknown. As a practical solution, nonparametric bootstrapping of the covariance matrix could be used to include the uncertainty of the stochastic predictors in the LMG calculations. Again, it would be necessary to do the LMG calculations for each bootstrap sample of the covariance matrix. There exist also different covariance estimators. The shrinkage method may be an interesting estimator with some nice properties (Schäfer and Strimmer, 2005).

Chapter 3

Examples

The following chapter presents the Bayesian LMG implementation by two examples. Simulated data is used for the first example. Empirical data is used for the second example.

3.1 Simulated data

We assume a simple model for the first example:

$$\begin{aligned} Y_i &\sim \mathcal{N}(\beta_0 + x_1\beta_1 + x_2\beta_2 + x_3\beta_3 + x_4\beta_4, \sigma^2), \\ \beta_1 &= 0.5, \beta_2 = 1, \beta_3 = 2, \beta_4 = 0, \sigma^2 = 1 \\ \mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4 &\sim \mathcal{N}(0, 1) \end{aligned}$$

The values of the four predictors are sampled from a standard normal distribution. These values are then multiplied by the regression coefficients. A standard normal distributed error is added to obtain the dependent variable. Fifty observations were sampled. The data generating R-code can be found in the Appendix [A.1](#).

The model is fitted using the `rstanarm` package ([Stan Development Team, 2016](#)) with the default priors for the regression and σ^2 parameters. The exact command can be found in R-code [A.2](#). These default priors are called 'weakly informative priors', because they take into account the order of magnitude of the variables by using the variance of the observed data. Information about these priors can be found in [Stan Development Team \(2017\)](#). A burn-in period of 20000, a sample size of 20000, and a thinning of 20 were chosen, resulting in a posterior sample size of 1000. The first few posterior samples are shown in Table [3.1](#).

For each posterior sample of the parameters, the R^2 value was calculated. The R^2 of the submodels was then calculated by the conditional variance formula for each posterior sample. The R-code is found in Appendix [A.3](#). The resulting R^2 values of the first few posterior samples are shown in Table [3.2](#). The thinning is reasonable in this case to reduce the computational burden and to still obtain an appropriate posterior of the R^2 values ([Link and Eaton, 2012](#)).

The `hier.part` package was used to calculate the LMG value for each posterior sample. The LMG posteriors are shown in Table [3.3](#). The independent component (I) represents the LMG value. The joint contribution (J) represents the difference from the independent component to the explained variance of the model containing only the predictor itself (T). Assuming stochastic or non-stochastic regressors has an influence on the uncertainty of the LMG values.

At first, non-stochastic regressors were assumed. The resulting LMG values and joint contributions with a 95% credible interval are shown in Table [3.3](#). An option to display the resulting LMG distribution is shown in Figure [3.1](#). Using the default weakly informative priors, the LMG distributions obtained from the Bayesian framework were very similar to the bootstrap confidence intervals, assuming non-stochastic predictors of the LMG estimates obtained from the `relaimpo` package, as shown in Table [3.4](#).

Table 3.1: Samples from the posterior distributions of the regression parameters.

	x1	x2	x3	x4	sigma
sample 1	0.873	1.126	0.952	1.049	0.994
sample 2	0.971	1.066	0.743	1.194	1.050
sample 3	1.109	0.953	0.840	1.124	0.915
sample 4	1.190	1.117	1.016	0.896	1.114
sample 5	1.017	0.937	0.971	1.155	0.829
sample 6	0.797	0.792	1.069	1.045	0.927
sample 7	0.945	0.894	1.081	1.150	0.998
sample 8	0.783	1.121	0.792	1.192	0.774
sample 9	0.887	1.134	1.109	1.162	0.879
sample 10	0.780	0.821	0.852	1.006	0.913

Table 3.2: R^2 for all submodels for the first six posterior samples.

	sample 1	sample 2	sample 3	sample 4	sample 5	sample 6
none	0.000	0.000	0.000	0.000	0.000	0.000
x1	0.317	0.340	0.423	0.418	0.396	0.324
x2	0.191	0.161	0.131	0.162	0.129	0.106
x3	0.182	0.110	0.151	0.203	0.191	0.247
x4	0.235	0.306	0.280	0.158	0.290	0.256
x1 x2	0.497	0.491	0.544	0.568	0.515	0.422
x1 x3	0.418	0.387	0.491	0.523	0.495	0.475
x1 x4	0.483	0.564	0.616	0.511	0.600	0.507
x2 x3	0.372	0.271	0.282	0.364	0.320	0.353
x2 x4	0.437	0.479	0.422	0.328	0.429	0.371
x3 x4	0.468	0.462	0.483	0.405	0.539	0.567
x1 x2 x3	0.599	0.538	0.612	0.674	0.615	0.574
x1 x2 x4	0.673	0.725	0.746	0.669	0.728	0.613
x1 x3 x4	0.628	0.647	0.723	0.650	0.747	0.715
x2 x3 x4	0.671	0.635	0.625	0.576	0.679	0.682
all	0.821	0.810	0.855	0.810	0.878	0.823

Table 3.3: Variance decomposition for non-stochastic predictors. I = LMG values, J = joint contribution, Total = total explained variance in one-predictor model.

Variable	I	J	Total
x1	0.253 (0.163, 0.353)	0.087 (0.071, 0.099)	0.34 (0.235, 0.445)
x2	0.152 (0.084, 0.239)	-5.633×10^{-4} (-0.002, 0.001)	0.152 (0.083, 0.24)
x3	0.165 (0.091, 0.251)	0.016 (0.005, 0.028)	0.182 (0.102, 0.272)
x4	0.258 (0.163, 0.359)	0.008 (-0.003, 0.018)	0.266 (0.169, 0.368)

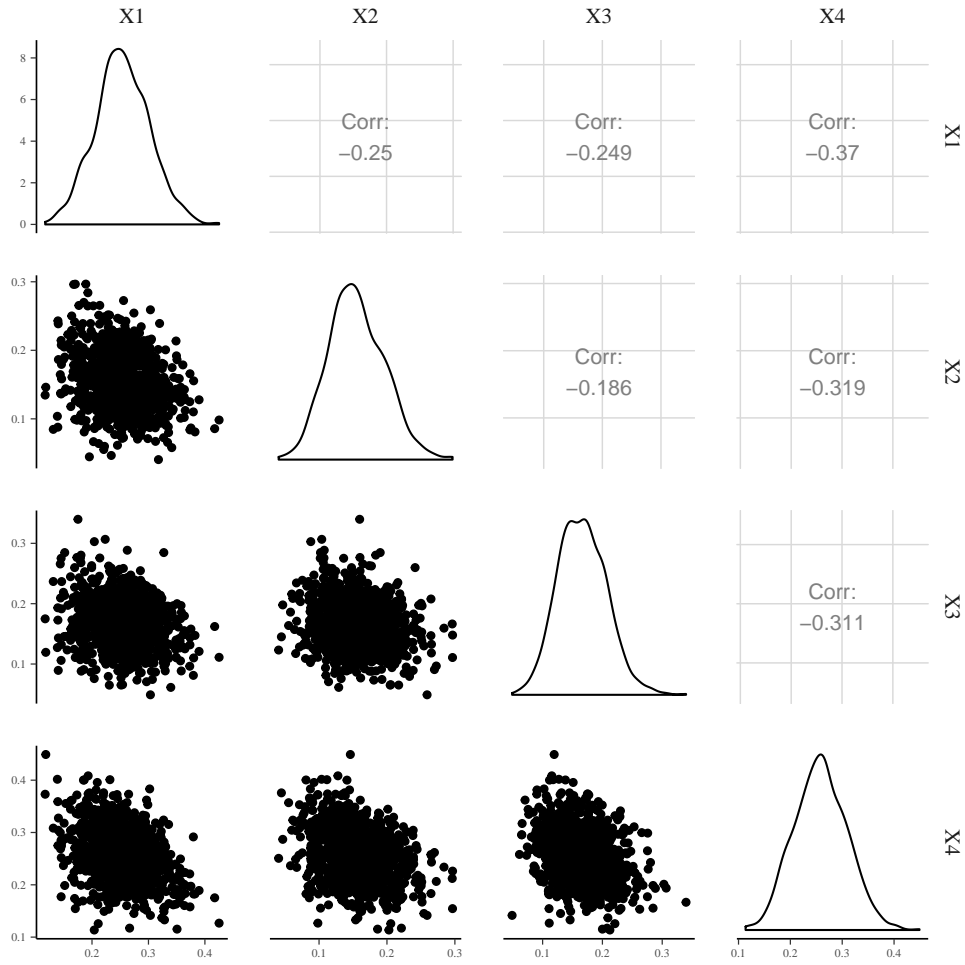


Figure 3.1: LMG distributions of the predictors.

Table 3.4: Variance decomposition for non-stochastic predictors. I = LMG values, J = joint contribution, Total = total explained variance in one-predictor only model.

Variable	LMG value (95%-CI)	
	Relaimpo	Bayesian framework
x1	0.258 (0.172, 0.353)	0.253 (0.163, 0.353)
x2	0.157 (0.09, 0.236)	0.152 (0.084, 0.239)
x3	0.169 (0.098, 0.254)	0.165 (0.091, 0.251)
x4	0.266 (0.175, 0.368)	0.258 (0.163, 0.359)

In this example, we know, that the predictor values were sampled from a normal distribution. It would therefore be more reasonable to assume stochastic predictors. Under the assumption of weak exogeneity and conditional independence, the posterior distributions of the regression parameters β are valid for non-stochastic and stochastic predictors. However, the uncertainty about the LMG values needs to include the uncertainty about the covariance matrix. If we know the distribution of the predictors we can incorporate this information and obtain the posterior distribution of the covariance matrix. The package **Jags** was used for inference about the covariance matrix in a Bayesian way. The R-code of the covariance inference can be found in Appendix A.6 and the R-code to get the R^2 values can be found in Appendix A.5. As an alternative, non-parametric bootstrap was used for inference about the covariance matrix. The R-code of the adapted LMG function can be found in Appendix A.4.

Using the bootstrap samples of the covariance matrix or samples from the posterior covariance matrix produced very similar LMG distributions. Bootstrap seems to be a valuable option for stochastic predictors when the distribution of the predictors is unknown. Even when the distribution is known, the difference seems to be tiny. A benefit of going the full Bayesian way is that we can also include prior knowledge about the covariance matrix. Using the default priors further produced very similar LMG distribution as using the non-parametric bootstrap option of the **relaimpo** package. Table 3.5 shows the LMG values of these approaches. For stochastic predictors, in contrast to non-stochastic predictors, the uncertainty about the covariance matrix is reflected in the larger credible intervals. Even when the exact regression parameters were known, there would a lot of uncertainty in the LMG values caused by the uncertainty about the covariance matrix.

Table 3.5: LMG values assuming stochastic predictors with 95% CI.

Variable	Relaimpo	Bayesian framework	
		nonparameteric bootstrap	covariance inference
x1	0.258 (0.14, 0.391)	0.25 (0.124, 0.391)	0.234 (0.126, 0.344)
x2	0.157 (0.059, 0.271)	0.138 (0.053, 0.241)	0.163 (0.062, 0.265)
x3	0.169 (0.078, 0.281)	0.152 (0.043, 0.265)	0.159 (0.068, 0.269)
x4	0.266 (0.143, 0.413)	0.257 (0.157, 0.373)	0.276 (0.158, 0.417)

Table 3.6: Variable description of empirical data set

Variable Name	Description
paragrap	scores on paragraph comprehension test
general	scores on general information test
sentence	scores on sentence completion test
wordc	scores on word classification test
wordm	scores on word meaning test

3.2 Empirical data

In the following section, the Bayesian LMG implementation is applied on an empirical dataset containing test scores of pupils ($N = 301$) from a study by [Holzinger and Swineford \(1939\)](#) available in the R package `MBESS` ([Kelley, 2017](#)). This dataset was used in [Nimon *et al.* \(2008\)](#) to present commonality analysis, which is another variance decomposition technique. Scores from a paragraph comprehension test (paragrap) were predicted by four verbal tests: general-information (general), sentence-comprehension (sentence), word-classification (wordc), and word-meaning (wordm) (Table 3.6).

The aim of the regression analysis was to determine the association between verbal ability and paragraph comprehension. An overview of the data is shown in Figure 3.3. The regression results are shown in Table 3.7). A novice researcher may wrongly conclude, that there is little association between the "non-significant" predictors (general information and word-classification) and paragraph comprehension. Given the other predictors are already included in the model, the predictors seem not to provide much information about the expected paragraph comprehension ability. However, it should not be concluded from this regression table, that the association between any of these "non-significant" predictors and the dependent variable is unimportant. As shown in Figure 3.3, the correlations between the predictors are rather high. The LMG metric may therefore provide new information about the importance of each predictor.

The Bayesian regression model was fitted in `rstanarm`. The default priors were used for the regression coefficients and the σ^2 parameter. A burn-in period of 20000, a sample size of 20000, and a thinning of 20 resulted in a posterior sample size of 1000. The first few posterior samples are shown in Table 3.8. The resulting R^2 of these posterior samples are shown in Table 3.9. The LMG values were calculated by using `hier.part`. The independent component (I), joint contribution (J), and total explained variance in a one-predictor model (T) are shown in Table 3.10. Sentence-comprehension and word-meaning seem to be the most important predictors by applying the LMG metric. However, none of the predictors seem to be unimportant. The joint contributions of each predictor were quite large.

For comparison purposes, the LMG metric was additionally calculated with the `relaimpo` package using parametric bootstrapping. The confidence intervals of `relaimpo` were almost identical to the credible intervals of the Bayesian framework (Table 3.11). Assuming stochastic or non-stochastic predictors resulted also in almost identical uncertainty estimates with such a large sample size (Table 3.12).

Table 3.7: Regression of paragraph comprehension on verbal tests.

	Coefficient	95%-confidence interval	<i>p</i> -value
Intercept	0.071	from -1.17 to 1.31	0.91
general	0.03	from -0.00 to 0.06	0.084
sentence	0.26	from 0.18 to 0.34	< 0.0001
wordc	0.047	from -0.01 to 0.11	0.14
wordm	0.14	from 0.08 to 0.19	< 0.0001

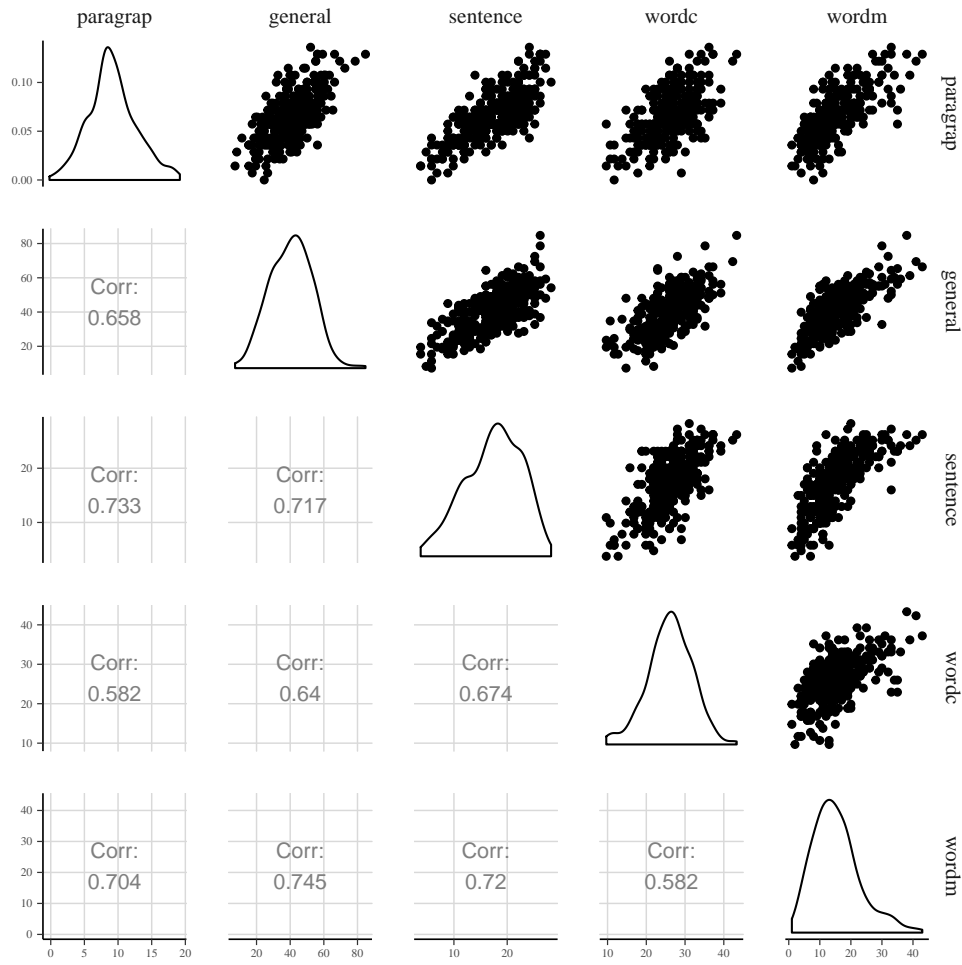


Figure 3.2: Test scores from Holzinger and Swineford's (1939) Study. $N=301$.

Table 3.8: Samples from the posterior distributions of the regression parameters.

	general	sentence	wordc	wordm	sigma
sample 1	0.017	0.275	0.072	0.136	2.15
sample 2	0.050	0.260	0.042	0.100	2.05
sample 3	0.024	0.291	0.083	0.135	2.28
sample 4	-0.003	0.259	0.017	0.195	2.31
sample 5	-0.006	0.344	0.043	0.120	2.26
sample 6	0.043	0.201	0.059	0.163	2.23
sample 7	0.057	0.230	0.015	0.138	2.25
sample 8	0.048	0.287	0.092	0.105	2.16
sample 9	0.028	0.268	0.058	0.123	2.19
sample 10	0.030	0.362	0.019	0.080	2.05

Table 3.9: R^2 for all submodels for the first six posterior samples.

	sample 1	sample 2	sample 3	sample 4	sample 5	sample 6
none	0.000	0.000	0.000	0.000	0.000	0.000
general	0.421	0.473	0.431	0.353	0.347	0.456
sentence	0.553	0.555	0.556	0.483	0.540	0.504
wordc	0.364	0.350	0.373	0.273	0.309	0.344
wordm	0.496	0.484	0.492	0.497	0.434	0.521
general sentence	0.580	0.604	0.587	0.502	0.547	0.561
general wordc	0.481	0.512	0.493	0.388	0.401	0.497
general wordm	0.531	0.549	0.532	0.508	0.456	0.564
sentence wordc	0.572	0.570	0.578	0.489	0.546	0.525
sentence wordm	0.612	0.608	0.612	0.570	0.575	0.596
wordc wordm	0.553	0.537	0.554	0.516	0.479	0.563
general sentence wordc	0.589	0.607	0.596	0.503	0.551	0.567
general sentence wordm	0.616	0.624	0.617	0.570	0.575	0.608
general wordc wordm	0.564	0.570	0.567	0.519	0.484	0.582
sentence wordc wordm	0.621	0.614	0.623	0.570	0.577	0.604
all	0.622	0.626	0.625	0.570	0.577	0.612

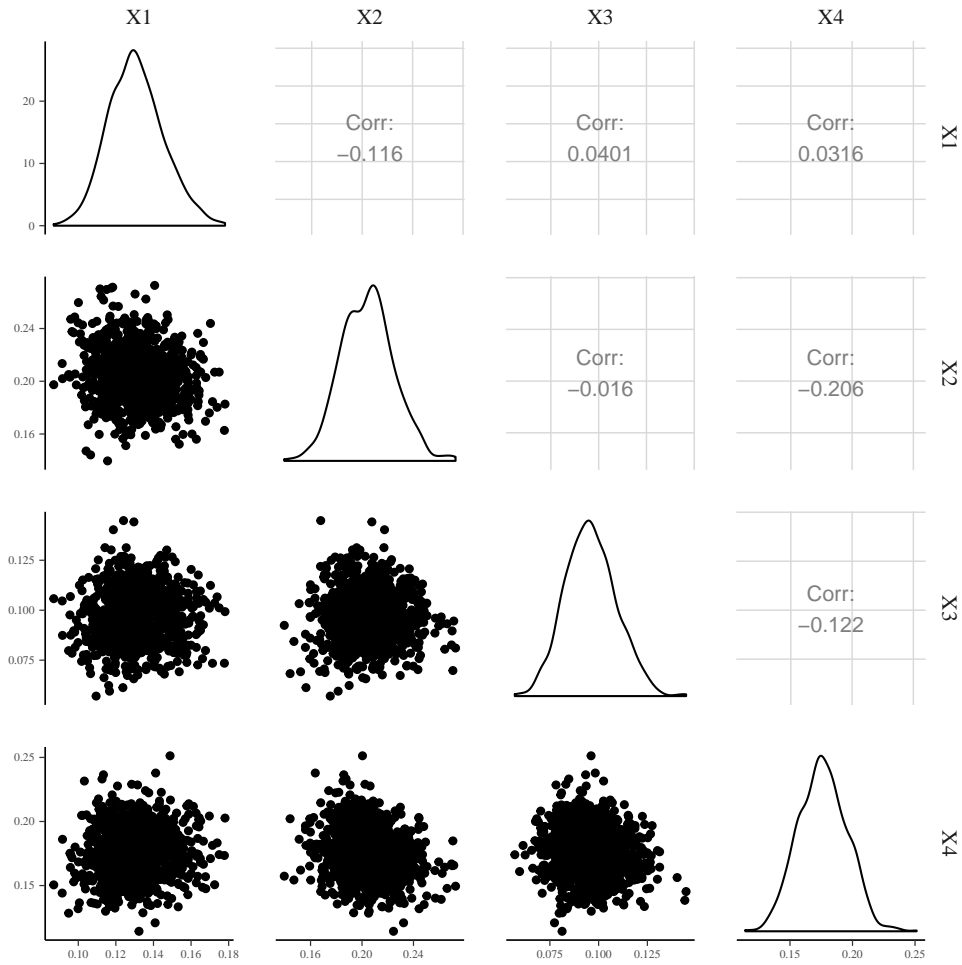
**Figure 3.3:** LMG distributions of the different verbal ability predictors.

Table 3.10: Variance decomposition for non-stochastic predictors. I = LMG values, J = joint contribution, Total = total explained variance in one-predictor only model.

Variable	I	J	Total
general	0.13 (0.104, 0.163)	0.298 (0.258, 0.336)	0.428 (0.362, 0.495)
sentence	0.205 (0.167, 0.245)	0.326 (0.294, 0.359)	0.532 (0.468, 0.593)
wordc	0.095 (0.072, 0.122)	0.239 (0.197, 0.273)	0.335 (0.269, 0.394)
wordm	0.176 (0.14, 0.213)	0.314 (0.278, 0.346)	0.491 (0.423, 0.553)

Table 3.11: Variance decomposition for non-stochastic predictors. I = LMG values, J = joint contribution, Total = total explained variance in one-predictor only model

Variable	LMG value (95%-CI)	
	Relaimpo	Bayesian framework
general	0.131 (0.107, 0.16)	0.13 (0.104, 0.163)
sentence	0.206 (0.17, 0.251)	0.205 (0.167, 0.245)
wordc	0.096 (0.073, 0.127)	0.095 (0.072, 0.122)
wordm	0.178 (0.142, 0.218)	0.176 (0.14, 0.213)

Table 3.12: Variance decomposition for stochastic predictors. I = LMG values, J = joint contribution, Total = total explained variance in one-predictor only model.

Variable	LMG value (95%-CI)	
	Relaimpo	Bayesian framework
general	0.131 (0.108, 0.164)	0.132 (0.095, 0.166)
sentence	0.206 (0.167, 0.248)	0.203 (0.164, 0.244)
wordc	0.096 (0.073, 0.128)	0.099 (0.068, 0.129)
wordm	0.178 (0.145, 0.218)	0.176 (0.139, 0.214)

Chapter 4

Extension to longitudinal data

Some extensions of the LMG formula beyond the simple linear regression model are shown in the following chapter. The focus is on repeated measures model. These models extend the simple linear regression by allowing intra-subject correlation between repeated measures.

The dependence of within-subject measurements can be modeled by including random effects (mixed model) or by assuming correlated errors within a subject (marginal model). A mixed model can be extended by including a random slope per subject, allowing for less restrictive longitudinal shapes. Different covariance matrices of the error terms allow for less restrictions in the marginal approach. An unstructured covariance matrix, where no restriction are imposed, allows for the most freedom in the error term. However, depending on the number of repeated measurements and the sample size, the covariance matrix can get too large to make reasonable inference about it ([Fitzmaurice *et al.*, 2011](#)).

The extension of the LMG formula in the Bayesian framework applied to longitudinal models is restricted to models where the conditional variance formula can be applied to get the explained variance of the submodel from the regression parameters of the full-model. Therefore, the focus is on the fixed predictors, but not on the random effects. The conditional variance formula can be used in the marginal models, because only the fixed effects are modeled anyway. In the mixed model framework, the conditional variance formula is applicable to random intercept models. For random-slope models, there are at least some difficulties involved - if it is possible at all - to get the explained variance of the submodels.

In this chapter, the Bayesian LMG Implementation is shown on a random intercept model and on a repeated measurement model with an unstructured covariance matrix.

4.1 Random intercept model

The first example concerns a simple random intercept model with time-varying predictors. Different R^2 metrics exist for linear mixed models. The variance of a random intercept model with regression parameter β can be written as

$$\text{Var}(y) = \sigma_f^2 + \sigma_\alpha^2 + \sigma_\epsilon^2, \quad (4.1)$$

where $\sigma_f^2 = \text{Var}(\mathbf{X}\beta) = \beta^\top \Sigma_{\mathbf{X}\mathbf{X}} \beta$, σ_α^2 is the variance of the random intercept, and σ_ϵ^2 represents the error variance ([Nakagawa and Schielzeth, 2013](#)).

An R^2 that is guaranteed to be positive is defined in [Nakagawa and Schielzeth \(2013\)](#) as

$$R_{\text{LMM}}^2 = \frac{\sigma_f^2}{\sigma_f^2 + \sigma_\alpha^2 + \sigma_\epsilon^2}. \quad (4.2)$$

Theoretically, it is possible that the R^2_{LMM} decreases when adding predictors (Nakagawa and Schielzeth, 2013). By adding predictors, σ_f^2 should always increase and σ_ϵ^2 should decrease. However, the σ_α^2 may also increase minimally and the total R^2 may then be minimally lower. The R^2 can not decrease by using the conditional variance formula on the full-model to calculate the R^2 of the submodels, because the total variance is fixed. Equal results should be obtained, by fitting a new model by maximum likelihood for each submodel and by comparing the explained variance of the fixed effects to the total variance of the full-model. In the Bayesian framework, the conditional variance formula is necessary to account for the mutual interdependence of the submodels. The total variance of the full-model can be calculated as $\text{Var}(y) = \text{Var}(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}b) + \sigma^2$ or by using samples of σ_α^2 as in definition (4.1). The error term could again be sampled or calculated as in definition (2.9). In the following examples, definition (4.1) was used, σ_α^2 and σ_ϵ^2 were sampled from their posterior distribution.

In repeated measures studies, the focus is often on within-subject changes. The between-subject variance, estimated with the random intercept term, is of minor importance. The important question is, how much variance the fixed predictors do explain, compared to the variance of the within subject error, which is

$$R^2_{\text{repeated}} = \frac{\sigma_f^2}{\sigma_f^2 + \sigma_\epsilon^2}, \quad (4.3)$$

The square root of this term is known under the name *correlation within subjects* in Bland and Altman (1995). Often, there are between-subject and within-subject predictors in a model. If the interest lies in the within-subject effects, a model including only the between-subject predictors can be used as the null-model.

The following example shows a simple random intercept model with time-varying predictors. The main question is which within-subject predictors are the most important ones. The between-subject variance is of minor importance.

The data are simulated from the following regression setting with $m = 4$ timepoints and $n = 20$ number of subjects (see Appendix A.7 for R-code),

$$Y_{i,j} \sim \mathcal{N}(\beta_0 + x_{1,i,j}\beta_1 + x_{2,i,j}\beta_2 + x_{3,i,j}\beta_3 + x_{4,i,j}\beta_4 + \alpha_i, \sigma^2), \quad \begin{aligned} i &= 1, \dots, n \\ j &= 1, \dots, m \end{aligned}$$

where $\beta_1 = 1$, $\beta_2 = 1$, $\beta_3 = 2$, $\beta_4 = 2$, $\sigma^2 = 1$, $\alpha_i \sim \mathcal{N}(0, \sigma_\alpha^2)$, $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$.

The random intercept effect was of minor interest. The R^2 of the models was calculated according to the formula of repeated measure correlation (4.3). The R-code of the model and the function can be found in the Appendix (A.8 and A.9). Most of the within-subject variance was explained by the predictors (Table 4.1). The credible intervals were very small. For information about the between-subject variance term, we can look at the posterior distribution of the random intercept variance term.

Next, the random intercept was directly included in the total variance calculation of the R^2 values. There was a large between-subject variance in this simulated dataset (Table 4.2). Therefore, the LMG values including the between subject variance were much lower. The credible intervals were as well much larger, because the uncertainty about the between-subject variance was included.

In my opinion, we can get more useful information from separating the between-subject variance from the within-subject variance components in this simple case. Note, that we assumed non-stochastic predictors. Otherwise, the credible intervals were larger. In general, it seems more reasonable to assume stochastic time-varying predictors. The variance could then be estimated by non-parametric bootstrap, resampling whole subjects (all repeated measurements of a subject).

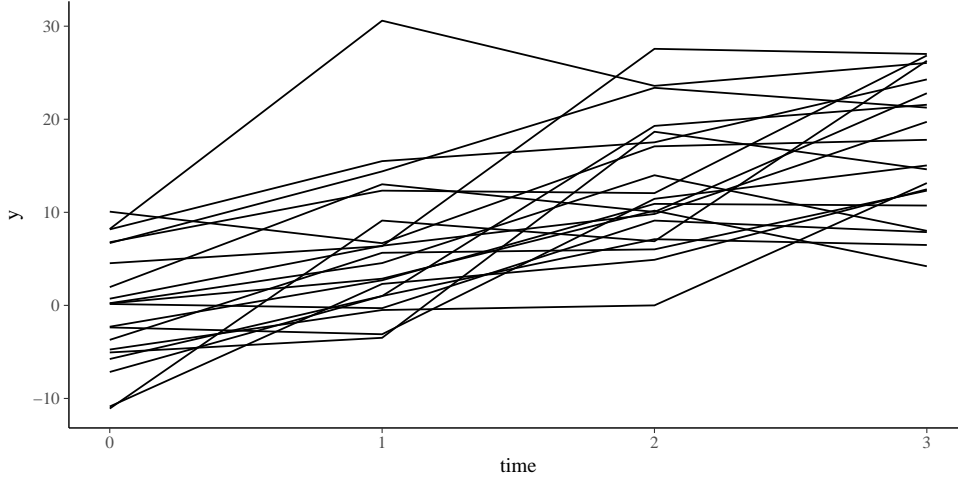


Figure 4.1: Individual trajectories of simulated random intercept model.

Table 4.1: Variance decomposition for non-stochastic predictors. I = LMG values, J = joint contribution, Total = total explained variance in one-predictor model

Variable	I	J	Total
x1	0.197 (0.195, 0.199)	0.489 (0.487, 0.491)	0.686 (0.682, 0.69)
x2	0.208 (0.206, 0.21)	0.515 (0.513, 0.517)	0.723 (0.719, 0.727)
x3	0.305 (0.303, 0.306)	0.647 (0.647, 0.647)	0.952 (0.95, 0.954)
x4	0.29 (0.288, 0.291)	0.623 (0.623, 0.624)	0.913 (0.911, 0.915)

4.2 Marginal model

The next example is about a repeated measurement model with time-varying predictors and an unstructured error covariance matrix. The data are generated from the following model

$$Y_i \sim \mathcal{N}(\mathbf{X}_i\boldsymbol{\beta}, \boldsymbol{\Sigma}), \quad i = 1, \dots, n, \quad (4.4)$$

where $\boldsymbol{\Sigma}$ represents an unstructured error covariance matrix, \mathbf{X}_i represents the predictor matrix of size $m \times p$ of subject i .

The R-code of the data generation can be found in Appendix A.10. In the variance calculation, it is necessary to take into account that there is not just one σ^2 parameter, but a covariance matrix $\boldsymbol{\Sigma}$. The diagonal elements of $\boldsymbol{\Sigma}$ represent the variance of each timepoint. The sum of the diagonal elements of $\boldsymbol{\Sigma}$ represents the variance for a whole subject. The mean can be taken of $\text{diag}(\boldsymbol{\Sigma})$ to make the formula compatible with the $\boldsymbol{\beta}^\top \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{X}} \boldsymbol{\beta}$ of (2.10), resulting in the total variance term

$$\text{Var}(\mathbf{Y}) = \boldsymbol{\beta}^\top \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{X}} \boldsymbol{\beta} + \text{mean}(\text{diag}(\boldsymbol{\Sigma})), \quad (4.5)$$

The individual trajectories are shown in Figure 4.2. The R-code for the model and LMG implementation can be found in Appendix (A.12 and A.11). The resulting LMG values of the predictors are shown in Table 4.3.

Table 4.2: Variance decomposition for non-stochastic predictors. I = LMG values, J = joint contribution, Total = total explained variance in one-predictor model

Variable	I	J	Total
x1	0.042 (0.012, 0.107)	0.112 (0.028, 0.233)	0.154 (0.04, 0.337)
x2	0.056 (0.022, 0.096)	0.129 (0.047, 0.233)	0.184 (0.069, 0.329)
x3	0.074 (0.031, 0.127)	0.158 (0.058, 0.286)	0.232 (0.089, 0.412)
x4	0.075 (0.032, 0.126)	0.154 (0.057, 0.274)	0.23 (0.091, 0.398)

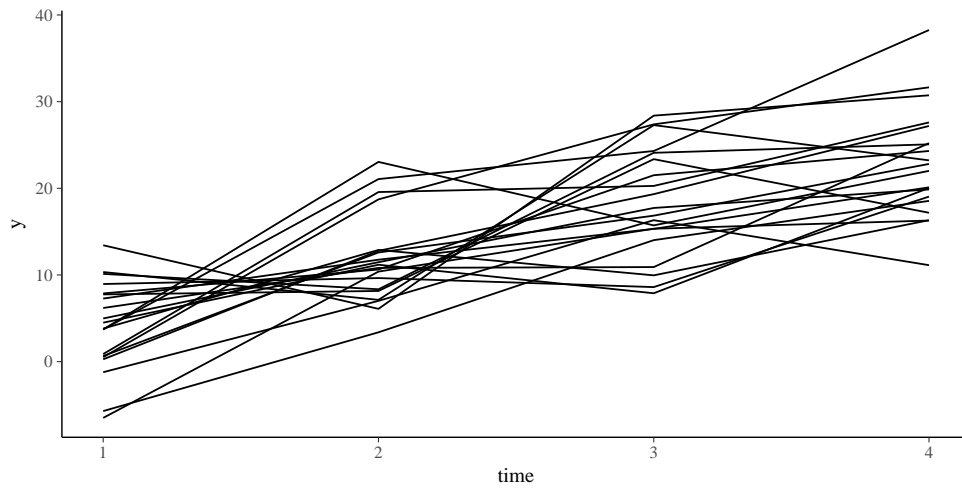


Figure 4.2: Individual trajectories of simulated data with unstructured error covariance matrix.

Table 4.3: Variance decomposition for non-stochastic predictors. I = LMG values, J = joint contribution, Total = total explained variance in one-predictor model

Variable	I	J	Total
x1	0.11 (0.082, 0.131)	0.284 (0.215, 0.329)	0.394 (0.298, 0.459)
x2	0.115 (0.086, 0.137)	0.301 (0.227, 0.348)	0.417 (0.312, 0.484)
x3	0.196 (0.149, 0.229)	0.396 (0.305, 0.453)	0.592 (0.451, 0.679)
x4	0.197 (0.149, 0.23)	0.399 (0.307, 0.456)	0.596 (0.455, 0.683)

Chapter 5

Discussion and Conclusion

5.1 Other variable importance metrics in the Bayesian framework

There exists different variable importance metrics (Grömping, 2015). The focus of this master thesis was on the LMG formula. A lot of variable importance metrics are based on the R^2 of the full model compared to the submodels. Instead of the LMG formula, another variable importance metric could as well be used on the R^2 of all submodels for each posterior sample. Commonality analysis (Nimon *et al.*, 2008) and dominance analysis (Grömping, 2015) seem to be interesting variance decomposition metrics. Both metrics provide some extensions to the LMG framework. Moreover, different options exist for the description of the LMG distribution. For example, the `relaimpo` package provides more bootstrap options, like pairwise differences, that could easily be transferred to the Bayesian framework.

5.2 Conclusion

The Bayesian framework provides the option to include the uncertainty about parameters. Using the conditional variance formula allows to calculate the R^2 of all the submodels from the posterior parameter distributions of the full model. Instead of fitting $2^p - 1$ models, only the full-model needs to be fitted. The mutual interdependence of the submodels is then automatically respected. The R^2 of the submodels do not decrease when adding predictors. The important property of non-negativity shares is then respected in the Bayesian framework.

A disadvantage about calculating the R^2 of all the submodels with the conditional variance formula seems to be the restriction to the linear model. Although, this may be a topic of further research. Another disadvantage of the Bayesian framework, compared to the classical LMG implementation, are higher computational costs. The calculations are still possible in a reasonable time period for non-stochastic predictors. Parallel computing could be used to speed up the calculations. For stochastic predictors, the computation time are much higher than in the classical framework.

Assuming non-stochastic or stochastic predictors can have a big impact in small samples on the uncertainty of the explained variance and on the LMG values. Although the posterior regression parameter distributions are the same in both cases (under the assumptions described in chapter 2), the explained variance of a model is directly dependent on the covariance matrix. Inference about the covariance of the predictors \mathbf{X} is therefore an important part when stochastic predictors are assumed. However, this does not seem to be an easy problem in general. Non-parametric bootstrap of the covariance matrix provides a practical solution in the Bayesian framework.

When the sample size is large enough, the classical and the Bayesian framework should lead to very similar values. The Bayesian framework allows including prior information, that may especially be relevant for small sample sizes. The credible interval of the Bayesian framework,

in contrast to the confidence intervals, may further be easier to interpret in the mathematically correct way.

A lot of studies are concerned with within-subject changes. The extension of the LMG formula to those kinds of problems is not straightforward. It is depending on the complexity of the data. However, the extension seems to be easily possible when the focus is on the fixed effects for the simple random intercept model. In the Bayesian framework and in the maximum likelihood framework, it seems reasonable to compare only the explained variance of the fixed effects of the submodels against the total variance, containing $\sigma_f^2 + \sigma_\alpha^2 + \sigma_\epsilon^2$. Otherwise, there may be problems with the non-negativity property of the shares.

Appendix A

Additional notes and R-codes

The error term in the definition of the R^2 proposed by [Gelman *et al.* \(2017\)](#) is defined as $\text{Var}(\sum_{i=1}^n e_n^s)$. I think we could also use $\sum (y - \hat{y}^s)^2 / (n - 1)$ as an estimate of the error. By using the maximum likelihood estimate, $\text{Var}(y_i - \hat{y}_i) = \sum (y_i - \hat{y}_i)^2 / (n - 1)$, because the mean of the residuals is 0. When samples of the posterior parameters are used, the mean of the residuals is not exactly zero. $\text{Var}(y_i - \hat{y}_i) = \sum (y_i - \hat{y}_i)^2 / (n - 1)$ is then a little bit larger than $\text{Var}(y_i - \hat{y}_i)$. In practice, the values should only differ by a very small amount. We do not expect the errors to have a systematic bias. However, the residuals are just a sample of the error. The mean of the residuals must not be exactly 0 when the samples of the posteriors are used for the regression coefficients.

A.1 Code used in chapter 3

R-Code A.1: Data generation of example 1 chapter 3

```
x1 <- rnorm(50, 0, 1)
x2 <- rnorm(50, 0, 1)
x3 <- rnorm(50, 0, 1)
x4 <- rnorm(50, 0, 1)
# b1 <- 0.5; b2 <- 1; b3 <- 2; b4 <- 0.8
b1 <- 1
b2 <- 1
b3 <- 1
b4 <- 1

y <- b1 * x1 + x2 * b2 + b3 * x3 + b4 * x4 + rnorm(50, 0, 1)

df <- data.frame(y = y, x1 = x1, x2 = x2, x3 = x3, x4 = x4)
```

R-Code A.2: Bayesian regression model for example 1 chapter 3

```
post2 <- stan_glm(y ~ 1 + x1 + x2 + x3 + x4, data = df, chains = 1, cores = 1,
  iter = 40000, thin = 20)
```

R-Code A.3: Function to get R^2 from posterior samples of the full model

```

rtwos <- function(X, post.sample) {
  # X: Predictor data as data frame post.sample: posterior sample as matrix
  # (M[sample_i,]), last position should be sigma paramater.
  X <- cov(X) #covariance matrix
  # Prepare data frame and rownames by using the combn() function
  lst <- list()
  pcan <- dim(X)[2]
  n <- (2^pcan) - 1
  for (i in 1:pcan) {
    lst[[i]] <- combn(pcan, i)
  }
  var.names <- character(length = 0)
  v <- rownames(X)
  for (i in 1:length(lst)) {
    for (j in 1:ncol(lst[[i]])) {
      cur <- lst[[i]][, j]
      name <- paste0(v[-cur])
      name <- paste(name, collapse = " ")
      var.names <- c(var.names, name)
    }
  }
  var.names <- c(rev(var.names), "all")
  var.names[1] <- "none"
  size <- nrow(post.sample) # how many samples
  sig.posi <- ncol(post.sample)
  df.Rtwos <- data.frame(matrix(0, n + 1, size))
  row.names(df.Rtwos) <- var.names
  # fill in R^2 matrix, use posterior samples and calculate submodels
  # according to the conditional variance formula.
  v <- 1:dim(X)[2]
  for (s in 1:size) {
    sample.s <- post.sample[s, -sig.posi]
    Vtot <- sample.s %*% X %*% sample.s #total explained variance
    count = n
    for (i in 1:(length(lst) - 1)) {
      for (j in 1:ncol(lst[[i]])) {
        cur <- lst[[i]][, j]
        set <- v[-cur]
        matr <- X[cur, cur] - X[cur, set] %*% solve(X[set, set]) %*%
          X[set, cur] #conditional variance formula
        var.explain <- sample.s[cur] %*% matr %*% sample.s[cur]
        df.Rtwos[count, s] <- Vtot - var.explain
        count = count - 1
      }
    }
    df.Rtwos[n + 1, s] <- Vtot
    df.Rtwos[, s] <- df.Rtwos[, s]/c(Vtot + post.sample[s, sig.posi]^2)
  }
  return(df.Rtwos)
}

```



```
}
```

R-Code A.4: R^2 function for stochastic predictors using bootstrap

```
# helper function to calculate covariance bootstrap sample
```

```
bootcov <- function(df, boot.n) {
  len <- nrow(df)
  cov.m <- cov(df)
  l <- dim(cov.m)[1]
  M.boot <- array(NA, c(l, l, boot.n))
  M.boot[, , 1] <- cov(df)
  for (i in 2:boot.n) {
    dfs <- df[sample(1:len, replace = T), ]
    M.boot[, , i] <- cov(dfs)
  }

  return(M.boot)
}
```

```
# Function that includes uncertainty about stochastic predictors by
# bootstrapping using bootcov boot.n is the number of bootstrap samples
# takes boot.n times longer to calculate.
```

```
rtwos.boot <- function(df, post.sample, boot.n) {
  # df: Predictor data as data frame post.sample: posterior sample as matrix
  # (M[sample_i,]), last position should be sigma paramater.
  X <- cov(df) #covariance matrix
  # Prepare data frame and rownames by using the combn() function
  lst <- list()
  pcan <- dim(X)[2]
  n <- (2^pcan) - 1
  for (i in 1:pcan) {
    lst[[i]] <- combn(pcan, i)
  }
  var.names <- character(length = 0)
  v <- rownames(X)
  for (i in 1:length(lst)) {
    for (j in 1:ncol(lst[[i]])) {
      cur <- lst[[i]][, j]
      name <- paste0(v[-cur])
      name <- paste(name, collapse = " ")
      var.names <- c(var.names, name)
    }
  }
  var.names <- c(rev(var.names), "all")
  var.names[1] <- "none"
  size <- nrow(post.sample) # how many samples
```

```

sig.posi <- ncol(post.sample)
df.Rtwos <- array(0, c(n + 1, size, boot.n))
dimnames(df.Rtwos)[[1]] <- var.names
boot.M <- bootcov(df, boot.n)
v <- 1:dim(X)[2]
for (b in 1:boot.n) {
  X <- boot.M[, , b]
  ##### fill in R^2 matrix, use posterior samples and calculate submodels
  ##### according to the conditional variance formula.
  for (s in 1:size) {
    sample.s <- post.sample[s, -sig.posi]
    Vtot <- sample.s %*% X %*% sample.s #total explained variance
    count = n
    for (i in 1:(length(lst) - 1)) {
      for (j in 1:ncol(lst[[i]])) {
        cur <- lst[[i]][, j]
        set <- v[-cur]
        matr <- X[cur, cur] - X[cur, set] %*% solve(X[set, set]) %*%
          X[set, cur] #conditional variance
        var.explain <- sample.s[cur] %*% matr %*% sample.s[cur]
        df.Rtwos[count, s, b] <- Vtot - var.explain
        count = count - 1
      }
    }
    df.Rtwos[n + 1, s, b] <- Vtot
    df.Rtwos[, s, b] <- df.Rtwos[, s, b]/c(Vtot + post.sample[s, sig.posi]^2)
  }
}
return(df.Rtwos)
}

```

R-Code A.5: R^2 function for stochastic predictors using covariance samples

```

# Function to include samples of the predictor covariance matrix boot.n =
# number of samples covm = posterior sample of covariance matrix

rtwos.covm <- function(df, post.sample, covm, boot.n) {
  # df: Predictor data as data frame post.sample: posterior sample as matrix
  # (M[sample_i,]), last position should be sigma paramater.
  X <- cov(df) #covariance matrix
  # Prepare data frame and rownames by using the combn() function
  lst <- list()
  pcan <- dim(X)[2]
  n <- (2^pcan) - 1
  for (i in 1:pcan) {
    lst[[i]] <- combn(pcan, i)
  }
  var.names <- character(length = 0)
  v <- rownames(X)

```

```

for (i in 1:length(lst)) {
  for (j in 1:ncol(lst[[i]])) {
    cur <- lst[[i]][, j]
    name <- paste0(v[-cur])
    name <- paste(name, collapse = " ")
    var.names <- c(var.names, name)
  }
}
var.names <- c(rev(var.names), "all")
var.names[1] <- "none"
size <- nrow(post.sample) # how many samples
sig.posi <- ncol(post.sample)
df.Rtwos <- array(0, c(n + 1, size, boot.n))
dimnames(df.Rtwos)[[1]] <- var.names
v <- 1:dim(X)[2]
for (b in 1:boot.n) {
  X <- covm[, , b]
  ##### fill in R^2 matrix, use posterior samples and calculate submodels
  ##### according to the conditional expectation formula.
  for (s in 1:size) {
    sample.s <- post.sample[s, -sig.posi]
    Vtot <- sample.s %*% X %*% sample.s #total explained variance
    count = n
    for (i in 1:(length(lst) - 1)) {
      for (j in 1:ncol(lst[[i]])) {
        cur <- lst[[i]][, j]
        set <- v[-cur]
        matr <- X[cur, cur] - X[cur, set] %*% solve(X[set, set]) %*%
          X[set, cur] #conditional variance
        var.explain <- sample.s[cur] %*% matr %*% sample.s[cur]
        df.Rtwos[count, s, b] <- Vtot - var.explain
        count = count - 1
      }
    }
    df.Rtwos[n + 1, s, b] <- Vtot
    df.Rtwos[, s, b] <- df.Rtwos[, s, b]/c(Vtot + post.sample[s, sig.posi]^2)
  }
}
return(df.Rtwos)
}

```

R-Code A.6: Jags code to make inference about covariance matrix

```

zy = df[,2:5]
#-----
# In the following example we know that the  $X$  are coming from a normal distribution. The cov
#-----
# Assemble data for sending to JAGS:
dataList = list(

```

```

    zy = zy ,
    Ntotal = nrow(zy) ,
    Nvar = ncol(zy) ,
    zRscal = ncol(zy) , # for dwish prior
    zRmat = diag(x=1,nrow=ncol(zy))
)

# Define the model:
modelString = "
model {
  for ( i in 1:Ntotal ) {
    zy[i,1:Nvar] ~ dmnorm( zMu[1:Nvar] , zInvCovMat[1:Nvar,1:Nvar] )
  }
  for ( varIdx in 1:Nvar ) { zMu[varIdx] ~ dnorm( 0 , 1/2^2 ) }
  zInvCovMat ~ dwish( zRmat[1:Nvar,1:Nvar] , zRscal )
  # Convert invCovMat to sd and correlation:
  zCovMat <- inverse( zInvCovMat )
}
" # close quote for modelString
writeLines( modelString , con="Jags-MultivariateNormal-model.txt" )

# Run the chains:
nChain = 3
nAdapt = 500
nBurnIn = 500
nThin = 10
nStepToSave = 20000
require(rjags)
jagsModel = jags.model( file="Jags-MultivariateNormal-model.txt" ,

update( jagsModel , n.iter=nBurnIn )
codaSamples = coda.samples( jagsModel ,

# Convergence diagnostics:
parameterNames = varnames(codaSamples) # get all parameter names
# Examine the posterior distribution:
mcmcMat = as.matrix(codaSamples)
chainLength = nrow(mcmcMat)
covMat <- array(NA, c(4,4,chainLength))
for (i in 1:chainLength){
  covMat[1:4,1:4,i]<-matrix(mcmcMat[i,], 4, 4)
}
covMat <- covMat[1:4,1:4,sample(1:20000, replace=F)]
df.rtwos <-rtwos.covm(df[,2:5], post.sample, covMat, 10)
n.boot = 10
LMG.Vals.I.covm<-array(0, c(4,dim(df.rtwos)[2], n.boot))
LMG.Vals.J.covm<-array(0, c(4,dim(df.rtwos.boot)[2], n.boot))
LMG.Vals.T.covm<-array(0, c(4,dim(df.rtwos.boot)[2], n.boot))
for (b in 1:n.boot){

```

```
for(i in 1:dim(df.rtwos)[2]){  
  
  gofn<-df.rtwos[,i,b]  
  
  obj.Gelman<-partition(gofn, pcan = 4, var.names = names(df[,2:5]))  
  
  LMG.Vals.I.covm[,i, b]=obj.Gelman$IJ[,1]  
  LMG.Vals.J.covm[,i, b]=obj.Gelman$IJ[,2]  
  LMG.Vals.T.covm[,i, b]=obj.Gelman$IJ[,3]  
}  
}  
  
quantile(c(LMG.Vals.I.covm[1,1:1000,]), c(0.025, 0.5, 0.975))  
quantile(c(LMG.Vals.I.covm[2,1:1000,]), c(0.025, 0.5, 0.975))  
quantile(c(LMG.Vals.I.covm[3,1:1000,]), c(0.025, 0.5, 0.975))  
quantile(c(LMG.Vals.I.covm[4,1:1000,]), c(0.025, 0.5, 0.975))
```

A.2 Code used in chapter 4

R-Code A.7: Data generating code random intercept model

```
sub <- 1:20
subi <- rnorm(20, 0, 4)
subi <- rep(subi, 4)
t <- c(0, 1, 2, 3)
t <- c(rep(0, 20), rep(1, 20), rep(2, 20), rep(3, 20))

mu <- rep(0, 4)
sig <- matrix(0.4, 4, 4)
diag(sig) <- 1
sig[3, 4] <- 0.9
sig[4, 3] <- 0.9
sig[1, 2] <- 0.3
sig[2, 1] <- 0.3

rawvars <- mvrnorm(n = 80, mu = mu, Sigma = sig)

x1 <- t + rawvars[, 1]
x2 <- t + rawvars[, 2]
x3 <- t + rawvars[, 3]
x4 <- t + rawvars[, 4]

b1 <- b2 <- 1
b3 <- b4 <- 2

y <- x1 * b1 + x2 * b2 + x3 * b3 + x4 * b4 + subi + rnorm(80, 0, 0.1)
df <- data.frame(y = y, x1 = x1, x2 = x2, x3 = x3, x4 = x4, sub = rep(sub,
  4))
```

R-Code A.8: Marginal model LMG implementation

```
fit <- stan_glmer(y ~ x1 + x2 + x3 + x4 + (1 | sub), data = df, chains = 4,
  cores = 4)
```

R-Code A.9: Rndom intercept LMG implementation

```
# function to calculate R2 data frame from random intercept model rtws for
# repeated measurement how much variance is explained by the within subject
# predictors (repeated measurement correlations, Bland-Altman 1995)
# var(fixed) / (var(fixed)+residual)
rtws.ri.r <- function(X, post.sample) {
  # X: Predictor data as data frame post.sample: posterior sample as matrix
  # (M[sample_i,]), second last position should be sigma parameter, last
```

```

# position random intercept sd.
X <- cov(X) #covariance matrix
# Prepare data frame and rownames by using the combn() function
lst <- list()
pcan <- dim(X)[2]
n <- (2^pcan) - 1
for (i in 1:pcan) {
  lst[[i]] <- combn(pcan, i)
}
var.names <- character(length = 0)
v <- rownames(X)
for (i in 1:length(lst)) {
  for (j in 1:ncol(lst[[i]])) {
    cur <- lst[[i]][, j]
    name <- paste0(v[-cur])
    name <- paste(name, collapse = " ")
    var.names <- c(var.names, name)
  }
}
var.names <- c(rev(var.names), "all")
var.names[1] <- "none"
size <- nrow(post.sample) # how many samples
sig.posi <- ncol(post.sample) - 1 # second last position sigma
df.Rtwos <- data.frame(matrix(0, n + 1, size))
row.names(df.Rtwos) <- var.names
##### fill in R^2 matrix, use posterior samples and calculate submodels
##### according to the conditional variance formula.
v <- 1:dim(X)[2]
for (s in 1:size) {
  sample.s <- post.sample[s, 1:ncol(X)]
  Vtot <- sample.s %*% X %*% sample.s #total explained variance
  count = n
  for (i in 1:(length(lst) - 1)) {
    for (j in 1:ncol(lst[[i]])) {
      cur <- lst[[i]][, j]
      set <- v[-cur]
      matr <- X[cur, cur] - X[cur, set] %*% solve(X[set, set]) %*%
        X[set, cur] #conditional variance formula
      var.explain <- sample.s[cur] %*% matr %*% sample.s[cur] #multiply by parameter
      df.Rtwos[count, s] <- Vtot - var.explain
      count = count - 1
    }
  }
  df.Rtwos[n + 1, s] <- Vtot
  df.Rtwos[, s] <- df.Rtwos[, s]/c(Vtot + post.sample[s, sig.posi]^2) #total variance +
}
return(df.Rtwos)
}

```

```

sub <- 1:20
subi <- rnorm(20, 0, 1)
subi <- rep(subi, 4)
mu <- rep(0, 4)
sig <- matrix(0.4, 4, 4)
diag(sig) <- 1
sig[3, 4] <- 0.9
sig[4, 3] <- 0.9
sig[1, 2] <- 0.3
sig[2, 1] <- 0.3
rawvars <- mvrnorm(n = 80, mu = mu, Sigma = sig)
cov(rawvars)
t <- c(rep(1, 20), rep(2, 20), rep(3, 20), rep(4, 20))
x1 <- t + rawvars[, 1]
x2 <- t + rawvars[, 2]
x3 <- t + rawvars[, 3]
x4 <- t + rawvars[, 4]
Sig <- matrix(3, 4, 4)
diag(Sig) <- 10
u <- rep(0, 4)
Sig[1, 1] <- 5
Sig[2, 2] <- 7
Sig[3, 4] <- 8
Sig[4, 3] <- 8
Sig[1, 2] <- 4
Sig[2, 1] <- 4
error <- mvrnorm(20, u, Sig)
y <- x1 * b1 + x2 * b2 + x3 * b3 + x4 * b4 + c(error)
t <- c(rep(1, 20), rep(2, 20), rep(3, 20), rep(4, 20))
df <- data.frame(y = y, x1 = x1, x2 = x2, x3 = x3, x4 = x4, sub = rep(sub,
  4), t = t)

# Prepare for Bayesian framework

Y <- matrix(df[, "y"], 20, 4, byrow = F)
x1 <- matrix(df[, "x1"], 20, 4, byrow = F)
x2 <- matrix(df[, "x2"], 20, 4, byrow = F)
x3 <- matrix(df[, "x3"], 20, 4, byrow = F)
x4 <- matrix(df[, "x4"], 20, 4, byrow = F)

N = 20 #subjects
M = 4 # repeated measures

```

R-Code A.11: Marginal model LMG implementation

```

rtwos.marg <- function(X, post.sample, m) {
  # X: Predictor data as data frame post.sample: posterior sample as matrix
  # (M[sample_i,]), the last positions are filled with diag(SIGMA) inferences

```



```

# (number of repeated measures)
X <- cov(X) #covariance matrix
# Prepare data frame and rownames by using the combn() function
lst <- list()
pcan <- dim(X)[2]
n <- (2^pcan) - 1
for (i in 1:pcan) {
  lst[[i]] <- combn(pcan, i)
}
var.names <- character(length = 0)
v <- rownames(X)
for (i in 1:length(lst)) {
  for (j in 1:ncol(lst[[i]])) {
    cur <- lst[[i]][, j]
    name <- paste0(v[-cur])
    name <- paste(name, collapse = " ")
    var.names <- c(var.names, name)
  }
}
var.names <- c(rev(var.names), "all")
var.names[1] <- "none"
size <- nrow(post.sample) # how many samples
sig.posi <- ((ncol(post.sample) - (m - 1)):ncol(post.sample)) # second last position sigma
df.Rtwos <- data.frame(matrix(0, n + 1, size))
row.names(df.Rtwos) <- var.names
##### fill in R^2 matrix, use posterior samples and calculate submodels
##### according to the conditional variance formula.
v <- 1:dim(X)[2]
for (s in 1:size) {
  sample.s <- post.sample[s, 1:ncol(X)]
  Vtot <- sample.s %*% X %*% sample.s #total explained variance
  count = n
  for (i in 1:(length(lst) - 1)) {
    for (j in 1:ncol(lst[[i]])) {
      cur <- lst[[i]][, j]
      set <- v[-cur]
      matr <- X[cur, cur] - X[cur, set] %*% solve(X[set, set]) %*%
        X[set, cur] #conditional variance formula
      var.explain <- sample.s[cur] %*% matr %*% sample.s[cur]
      df.Rtwos[count, s] <- Vtot - var.explain
      count = count - 1
    }
  }
  df.Rtwos[n + 1, s] <- Vtot
  df.Rtwos[, s] <- df.Rtwos[, s]/c(Vtot + sum(post.sample[s, sig.posi]))
  # total explained variance + diag(SIGMA)
}
return(df.Rtwos)
}

```

R-Code A.12: Marginal model LMG implementation

```

#-----
modelString <- "model{

# Likelihood
for(i in 1:N){
Y[i,1:M] ~ dmnorm(mu[i,1:M],Omega[1:M,1:M])
for(j in 1:M){
mu[i,j] <- beta0 + beta1*x1[i,j]+ beta2*x2[i,j]+ beta3*x3[i,j] + beta4*x4[i,j]
}}

# Priors

Omega[1:M, 1:M] ~dwish(zRmat[1:M,1:M] , zRscal)
Sigma[1:M, 1:M] <- inverse(Omega)

beta0      ~ dnorm(0,0.001)
beta1      ~ dnorm(0,0.001)
beta2      ~ dnorm(0,0.001)
beta3      ~ dnorm(0,0.001)
beta4      ~ dnorm(0,0.001)

}"

writeLines(modelString, con = "Jags-MultivariateNormal-model.txt")

model <- jags.model(textConnection(modelString), data = list(Y = Y, N = N,
  M = M, x1 = x1, x2 = x2, x3 = x3, x4 = x4, zRscal = ncol(Y), zRmat = diag(x = 1,
  nrow = ncol(Y))), n.chains = 3)

samp <- coda.samples(model, variable.names = c("beta1", "beta2", "beta3", "beta4",
  "Sigma"), n.iter = 20000, progress.bar = "none")

# summary(samp)

# L
calculations

samp <- coda.samples(model, variable.names = c("beta1", "beta2", "beta3", "beta4",
  "Sigma[1,1]", "Sigma[2,2]", "Sigma[3,3]", "Sigma[4,4]"), n.iter = 20000,
  thin = 20, progress.bar = "none")

post.sample <- samp[[1]][, c(5:8, 1:4)]

df.rtwos <- rtwos.marg(df[, 2:5], post.sample, 4) # 4 repeated measures

L
.Vals.I <- matrix(0, 4, dim(df.rtwos)[2])
L
.Vals.J <- matrix(0, 4, dim(df.rtwos)[2])

```

```
L
.Vals.T <- matrix(0, 4, dim(df.rtwos)[2])

for (i in 1:dim(df.rtwos)[2]) {
  gofn <- df.rtwos[, i]
  obj.Gelman <- partition(gofn, pcan = 4, var.names = names(df[, 2:5]))
  L
  .Vals.I[, i] = obj.Gelman$IJ[, 1]
  L
  .Vals.J[, i] = obj.Gelman$IJ[, 2]
  L
  .Vals.T[, i] = obj.Gelman$IJ[, 3]
}
```


Bibliography

- Alexander, D. L., Tropsha, A., and Winkler, D. A. (2015). Beware of R²: Simple, Unambiguous Assessment of the Prediction Accuracy of QSAR and QSPR Models. *Journal of Chemical Information and Modeling*, **55**, 1316–1322. [4](#)
- Bland, J. M. and Altman, D. G. (1995). Calculating correlation coefficients with repeated observations: Part 1–Correlation within subjects. *BMJ (Clinical research ed.)*, **310**, 446. [18](#)
- Chevan, A. and Sutherland, M. (1991). Hierarchical partitioning. *American Statistician*, **45**, 90–96. [3](#)
- Fitzmaurice, G. M., Laird, N. M., and Ware, J. H. (2011). *Applied longitudinal analysis*. Wiley. [17](#)
- Gelman, A., Goodrich, B., Gabry, J., and Ali, I. (2017). R-squared for Bayesian regression models *. Technical report. [4](#), [5](#), [23](#)
- Grömping, U. (2006). Relative Importance for Linear Regression in R : The Package relaimpo. *Journal of Statistical Software*, **17**, 1–27. [1](#), [3](#), [8](#)
- Grömping, U. (2007). Estimators of relative importance in linear regression based on variance decomposition. *American Statistician*, **61**, 139–147. [1](#), [6](#)
- Grömping, U. (2015). Variable importance in regression models. *Wiley Interdisciplinary Reviews: Computational Statistics*, **7**, 137–152. [1](#), [1](#), [3](#), [4](#), [21](#)
- Holzinger, K. J. and Swineford, F. (1939). A study in factor analysis: The stability of a bi-factor solution. *Supplementary Educational Monographs*, **48**, 1–91. [13](#)
- Jackman, S. (2009). *Bayesian Analysis for the Social Sciences*. Wiley. [7](#)
- Kelley, K. (2017). Mbes (version 4.0.0 and higher) [computer software and manual]. R package version 4.0.0 and higher. [13](#)
- Kvalseth, T. O. (1985). Cautionary Note about R². *The American Statistician*, **39**, 279. [4](#)
- Link, W. A. and Eaton, M. J. (2012). On thinning of chains in MCMC. *Methods in Ecology and Evolution*, **3**, 112–115. [9](#)
- Nakagawa, S. and Schielzeth, H. (2013). A general and simple method for obtaining R² from generalized linear mixed-effects models. *Methods in Ecology and Evolution*, **4**, 133–142. [17](#), [18](#)
- Nimon, K., Lewis, M., Kane, R., and Haynes, R. M. (2008). An R package to compute commonality coefficients in the multiple regression case: An introduction to the package and a practical example. *Behavior Research Methods*, **40**, 457–466. [13](#), [21](#)
- Nimon, K. F. and Oswald, F. L. (2013). Understanding the Results of Multiple Linear Regression: Beyond Standardized Regression Coefficients. *Organizational Research Methods*, **00**, 1–25. [1](#)

- Schäfer, J. and Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, **4**, 1–30. [8](#)
- Stan Development Team (2016). rstanarm: Bayesian applied regression modeling via Stan. R package version 2.13.1. [9](#)
- Stan Development Team (2017). *Stan Modeling Language: User’s Guide and Reference Manual*. [9](#)
- Walsh, C. and Nally, R. M. (2015). Title Hierarchical Partitioning. Technical report. [i](#), [4](#)