Mixed-effects random forest model for quantifying relations in clustered data

Rutten, Thomas A.S.

Award date:
2021

Eindhoven University of Technology

Graduation Thesis

Department of Mathematics and Computer Science

Mixed-effects random forest model for quantifying relations in clustered data

Author:
Thomas Rutten
Student ID: 0954160

Supervisors:
Drs. Jan Willem Bikker (CQM)
Prof. dr. Edwin van den Heuvel (TU/e)
Ir. Michiel Paus (CQM)

February 20, 2021

Abstract

In many real life data problems people want to understand and quantify relationships in the data so that one can understand what is driving a certain response variable. In this thesis we investigate if the machine learning model random forest can be used for explaining and quantifying relationships in datasets when there is structured data with random effects. By doing an extensive simulation study we compare the performance of this mixed-effects random forest model in terms of marginal effect estimation to regular random forest and to linear regression and linear mixed model. The performances are assessed by computing bias and variance. The study shows that mixed-effects random forest falls significantly short of the linear mixed model in cases where the fitted model captures the true underlying model, but at the same time it also shows that it can be a clear improvement over regular random forest. Furthermore, when the true underlying model may be quite complex, the mixed-effects random forest can still give reasonable estimates for the effects and might thus in that way provide a good first step in a modeling process. Moreover we investigate how data settings influence the performances of the models, and we find that mixed-effects random forest performs quite poor when the amount of data is relatively small and it improves rapidly when more data is added. Another notable finding is that the hyperparameters of the random forest model may have a big influence on the model's performance.

Contents

Chapter 1

Introduction

Many problems in the field of statistics and data science deal with observational data. In this thesis, we focus on studying relations in this type of datasets. A typical goal in CQM's consultancy projects is explaining the effect of predictor variables in order to better understand the underlying domain problem. A central question often is what will happen if you change a predictor and the others remain constant. This is in contrast to making a predictive model, which is often the sole focus in a machine learning model, and which often is much easier. Although there do exist methods that try to explain the effects of variables for machine learning models, it is not straightforward to In many projects analyzing the data before actually making a model can be quite time consuming and sometimes tedious. This is because the first step of exploratory data analysis involves visualization, tables, summaries, checking missingness and often more advanced statistical checks, such as col-linearity. This is all needed because traditional statistical models usually have quite a few underlying assumptions such as normality, homogeneity and independence, and these need to be verified to some degree in order to make a sensible model and receive meaningful results. Contrary to this, machine learning methods are often more robust to those assumptions and can for example deal with missingness, outliers in the blink of an eye. The price one pays by using those models is that they are much more limited in their output: prediction is their main focus and getting detailed information about relations and uncertainty is not straightforward at all. Although there do exist methods that help identifying relations in machine learning models, estimating effects with accompanying confidence intervals and deriving statistical tests for significance is not really part of those methods. All this begs the question whether it is possible to combine best of both worlds.

In this thesis we are particularly interested in finding relations in the setting of clustered data. What these type of datasets typically entail is that observations occur at two or more levels, e.g. a group level and an individual level or a subject level with repeated measures. Especially when there are large differences between how variables behave across various groups or subjects, relatively simple models typically fail to get a grasp of the relationships in the data. So in those cases linear mixed models may be used for taking into account the structure of the data and the possible random effects that are at play.
Since the field of machine learning is quite broad, we will limit in this thesis ourselves

to one very popular machine learning technique: *random forest*. Therefore the research questions will also be formulated around random forest, rather than any machine learning model. The main research at stake is the following:

- Can random forest be used for constructing a reliable explorative model for effect estimation in the setting of clustered observational data?

Different data aspects that can often be found in real world datasets will be considered for answering this research question so that we can discuss under which data circumstances mixed-effects random forest might be a suitable model and under which data circumstances it behaves poorly (if at all) in terms of the modeling goal. These circumstances and the way the models will be assessed are reflected in the following research questions:

- What are existing models that combine random forest models with random effects estimation and what are their known strengths, weaknesses and implementation goals?
- Under which data circumstances can random forests be used as a suitable alternative for traditional statistical model for finding relations and why? And when should random forest not be used as an alternative and why? More specifically the following subquestions will guide us to a quantitative answer:
    1. How well does a mixed-effect random forest estimate fixed effects in terms of bias and variance? How does this compare to traditional statistical models, and is there an improvement compared to a regular random forest model?
    2. How well can a random forest model estimate random effects and how does this compare to traditional statistical models?
    3. What is the bias-variance trade-off for mixed-effects random forest for both fixed effects and random effects? How does this compare to linear mixed models?
- How does the selection of random forest hyperparameters influence the mixed-effects random forest model fixed and random effects estimations?

The different data aspects that will be considered are mostly about data size - both in terms of number of observations and predictors, data complexity and sizes of various effects. In this thesis data complexity is mainly about correlated predictors and whether the effects of predictors on the response are linear or non-linear. In the latter case traditional linear models are often less applicable.

1.1 Thesis outline

In this thesis we will first discuss traditional statistical models such as linear regression and linear mixed models. For linear mixed model we will also elaborate on an estimation method that can be used for solving mixed-effects random forest models.

Chapter 3 is all about random forest and the influence of its hyperparameters and in

Chapter 4 we will explain how the random forest model can be extended to a model that takes into account random structures.

In Chapter 5 we will discuss how the several models will be compared using a simulation study: Section 5.1 details the model parameters for both the traditional statistical models as well as for the random forest models. It also contains the structure of the data that is used for the simulation study and what the data parameters are. Section 5.2 outlines how the fixed effects and random effects are to be estimated by the models and how we will assess the estimation performance of the various models.

Chapters 6 and 7 contain the results, key finding and conclusions and ultimately a discussion of the research.

Chapter 2

Traditional Statistical Models

2.1 Linear Regression

One of the oldest and most used tools for finding relationships within data is linear regression. Although linear regression models go all the way back to the 18th century they are still very popular nowadays, and many more complicated models are generalizations or extensions of these linear models [15].
Assume we have an observational dataset with explanatory variables $X = (\boldsymbol{x}_1, ..., \boldsymbol{x}_p)$, where $\boldsymbol{x}_j = (x_{1j}, x_{2j}, ..., x_{nj})^T$ and the single dependent variable $\boldsymbol{y} = (y_1, ..., y_n)^T$. Then the linear regression model can be written as:

$$\boldsymbol{y} = \beta_0 + \sum_{j=1}^{p} \boldsymbol{x}_j \beta_j + \boldsymbol{\varepsilon}, \tag{2.1}$$

where $\boldsymbol{\varepsilon} = (\varepsilon_1, ..., \varepsilon_n)^T$ and the assumption is made that the error terms ε_i are all independent and identically distributed, following the normal distribution with mean zero and unknown variance σ^2, i.e. $\varepsilon_i \sim \mathcal{N}(0, \sigma^2) \quad \text{for} \quad i = 1, ..., n$. The model is always linear in terms of the unknown parameters $\boldsymbol{\beta} = (\beta_1, ..., \beta_p)$, also called regression coefficients. The regression variables themselves can be transformations or functions of other variables. To estimate the regression coefficients a variety of estimation methods exist, but the one typically used and often implemented in statistical software packages is *ordinary least squares* (OLS). The goal of this method is to pick the regression coefficients in such a way that the residual sum of squares (RSS) will be minimized. The RSS is defined as follows:

$$\text{RSS}(\boldsymbol{\beta}) = \left|\left| \boldsymbol{y} - \left(\beta_0 + \sum_{j=1}^{p} \boldsymbol{x}_j \beta_j \right) \right|\right|^2 = \sum_{i=1}^{n} \left(y_i - \left(\beta_0 + \sum_{j=1}^{p} \boldsymbol{x}_{ij} \beta_j \right) \right)^2,$$

which can also be written in matrix form as follows:

$$\text{RSS}(\boldsymbol{\beta}) = ||X\boldsymbol{\beta} - \boldsymbol{y}||^2 = (X\boldsymbol{\beta} - \boldsymbol{y})^T (X\boldsymbol{\beta} - \boldsymbol{y}).$$

Differentiating the above expression with respect to $\boldsymbol{\beta}$ yields

$$\frac{\partial \text{RSS}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = 2X^T (X\boldsymbol{\beta} - \boldsymbol{y}),$$

and setting this to $\mathbf{0}$ subsequently gives us the results for the estimates:

$$\hat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T \boldsymbol{y}, \tag{2.2}$$

where we assume that the matrix X has full rank, since otherwise $X^T X$ would not be invertible. The fitted response values $\hat{\boldsymbol{y}}$ are now:

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

Apart from the regression coefficients, there is another unknown parameter involved in this regression model, which is the variance of the error term $\boldsymbol{\varepsilon}$. The common estimator of σ^2 is based on the sum of squares and is given by:

$$\hat{\sigma}^2 = \frac{1}{n-1-p} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2, \tag{2.3}$$

where the denominator in this expression is deliberately chosen such that the estimator is unbiased.

2.2 Linear Mixed Model

A disadvantage of ordinary linear regression models is that they cannot take into account random effects, due to the underlying assumption that observations within the same group are independent. This is a shortcoming in settings where for example repeated measurements are executed on the same patient, or measurements on clusters of related units, e.g. students in the same school. Another way random effects can come into play is by having a random slope, i.e. a variable has a different effect for every subject. For these kind of settings mixed models can be used, which include both fixed effects and random effects. In this section we will discuss linear mixed models, which is an extension of linear regression described in the previous section [13]. Suppose there are K groups, where each group i has n_i observations (e.g. repeated measures) for $i = 1, ..., K$. Now the data for group i consists of three parts: X_i is the $n_i \times p$ design matrix related to the fixed effects $\boldsymbol{\beta} = (\beta_1, ..., \beta_p)$, Z_i is the $n_i \times m$ design matrix related to the random effects $\boldsymbol{u}_i = (u_{i1}, ..., u_{im})$, and finally the response variable $\boldsymbol{y}_i = (y_{i1}, ..., y_{in_i})$. Then, the linear mixed model is denoted as follows:

$$\boldsymbol{y}_i = X_i \boldsymbol{\beta} + Z_i \boldsymbol{u}_i + \boldsymbol{\varepsilon}_i, \quad \text{for } i = 1, 2, ..., K. \tag{2.4}$$

In this model $\boldsymbol{y}_i$ is independent of $\boldsymbol{y}_j$ for $i \neq j$. We now assume that the error term $\boldsymbol{\varepsilon}_i$ follows the multivariate normal distribution with means zero and $n_i \times n_i$ variance matrix R_i, i.e. $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, R_i)$. R_i only depends on group i in terms of its size, not in terms of structure. So in other words, in a balanced dataset where all groups have an equal number of observations it holds that $R_i = R$ for any i. Furthermore there is the assumption that also the random effects are multivariate normally distributed, with $m \times m$ variance-covariance matrix G, i.e. $\boldsymbol{u}_k \sim \mathcal{N}(\mathbf{0}, G)$. The structures of the R_i and G matrices need to be specified before estimating and this affects the number of parameters that are involved in the model.

The easy and often used choice for the residuals matrix R_i is simply σI_{n_i}, with I_{n_i} being the $n_i \times n_i$ identity matrix. The residuals and random effects are assumed to be independent, in other words that $\text{COV}(\boldsymbol{u}_i, \boldsymbol{\varepsilon}_i) = 0$ for all i.

The model in (2.4) can be rewritten in large matrix form as follows:

$$\boldsymbol{y} = X\boldsymbol{\beta} + Z\boldsymbol{u} + \boldsymbol{\varepsilon}, \tag{2.5}$$

where

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_K \end{bmatrix}, \quad X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_K \end{bmatrix}, \quad Z = \begin{bmatrix} Z_1 & 0 & \dots & 0 \\ 0 & Z_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Z_K \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \vdots \\ \boldsymbol{u}_K \end{bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \boldsymbol{\varepsilon}_1 \\ \boldsymbol{\varepsilon}_2 \\ \vdots \\ \boldsymbol{\varepsilon}_K \end{bmatrix}.$$

For sake of convenience we will now assume that the residual matrices R_is are indeed of the form $R_i = \sigma_R^2 I_{n_i}$ and that the matrix G that belongs to the random effects is specified by a vector of parameters called $\boldsymbol{\sigma}_b$ of unspecified length (but at most m^2), so that $G = G(\boldsymbol{\sigma}_b)$. It also follows that now $\boldsymbol{u} \sim \mathcal{N}(0, G^*(\boldsymbol{\sigma}_b))$, with G^* being the extended covariance matrix:

$$G^*(\boldsymbol{\sigma}_b) = \begin{bmatrix} G & 0 & \dots & 0 \\ 0 & G & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & G \end{bmatrix}.$$

Hence the set of parameters that needs to be estimated in this model is $P = (\boldsymbol{\beta}, \boldsymbol{\sigma}_b, \sigma_R)$. Note that $\boldsymbol{u}$ is also unknown, but that those estimates follow from P.

Various methods exist to estimate P, but many of those estimation techniques are based on maximum likelihood (ML) estimation. So therefore we will first provide the likelihood formulas that correspond to the model as introduced in (2.5), and then a method for estimating P will be discussed in further detail.

2.2.1 Maximum likelihood estimation

Let $L(P|\boldsymbol{y}) = f(\boldsymbol{y}|P)$ denote the likelihood function of (1.4) with $f_P(\boldsymbol{y})$ being the density function of the response depending on the parameters P. Obviously this function depends heavily on the assumptions of normality, since from (2.5) we know that the distribution of $\boldsymbol{y}_i|\boldsymbol{u}$ is a follows:

$$\boldsymbol{y}_i|\boldsymbol{u}_i \sim \mathcal{N}(X\boldsymbol{\beta} + Z_i\boldsymbol{u}_i, \sigma_R^2 I_N), \tag{2.6}$$

With this property we can now derive the likelihood $L(P|\boldsymbol{y}_i)$:

$$\begin{aligned}
L(P|\boldsymbol{y}) &= \prod_{i=1}^{m} f(\boldsymbol{y}_i|P) = \prod_{i=1}^{K} f(\boldsymbol{y}_i|\boldsymbol{\beta}, \boldsymbol{\sigma}_b, \sigma_R) \\
&= \prod_{i=1}^{K} \int f(\boldsymbol{y}_i|\boldsymbol{\beta}, \boldsymbol{\sigma}_b, \sigma_R, \boldsymbol{u}_i) f(u_i|\boldsymbol{\sigma}_b) \mathrm{d}u_i \\
&= \prod_{i=1}^{K} \left\{ \int \left[\frac{1}{(2\pi\sigma_R^2)^{n_i/2}} \exp\left(-\frac{1}{2\sigma_R^2} (\boldsymbol{y}_i - X_i\boldsymbol{\beta} - Z_i\boldsymbol{u}_i)^T (\boldsymbol{y}_i - X_i\boldsymbol{\beta} - Z_i\boldsymbol{u}_i) \right) \right. \right. \\
&\qquad \left. \left. \cdot \frac{1}{(2\pi)^{m/2}} \frac{1}{|G(\boldsymbol{\sigma}_b)|^{1/2}} \exp\left(-\frac{1}{2} \boldsymbol{u}_i^T G^{-1}(\boldsymbol{\sigma}_b) \boldsymbol{u}_i \right) \right] \mathrm{d}\boldsymbol{u}_i \right\} \\
&\approx \prod_{i=1}^{K} \left\{ \frac{1}{(2\pi\sigma_R^2)^{n_i/2}} \exp\left(-\frac{1}{2\sigma_R^2} (\boldsymbol{y}_i - X_i\boldsymbol{\beta} - Z_i\boldsymbol{u}_i^*)^T (\boldsymbol{y}_i - X_i\boldsymbol{\beta} - Z_i\boldsymbol{u}_i^*) \right) \right. \\
&\qquad \left. \cdot \frac{1}{|G(\boldsymbol{\sigma}_b)|^{1/2}} \exp\left(-\frac{1}{2} \boldsymbol{u}_i^{*T} G^{-1}(\boldsymbol{\sigma}_b) \boldsymbol{u}_i^* \right) \frac{1}{|Z_i^T Z_i + G^{-1}(\boldsymbol{\sigma}_b)|^{1/2}} \right\},
\end{aligned} \tag{2.7}$$

where u^* maximizes the integrand in the previous line. In the above computation we used in the last step Laplace's approximation method:

$$\int h(\boldsymbol{x}) \exp(M f(\boldsymbol{x})) \mathrm{d}\boldsymbol{x} = \left(\frac{2\pi}{M} \right)^{d/2} h(\boldsymbol{x}^*) \frac{\exp(M f(\boldsymbol{x}^*))}{|-H(f)(\boldsymbol{x}^*)|^{1/2}}, \quad \text{as } M \to \infty,$$

where $\boldsymbol{x}^*$ maximizes $f(\boldsymbol{x})$ and $H(f)(\boldsymbol{x})$ denotes the Hessian matrix of $f(\boldsymbol{x})$. It should be noted that this approximation method is not necessary, but it is commonly used for generalized linear mixed model parameter estimation [2]. For simple linear mixed model - like here - is it possible to express the maximum likelihood estimators for β and σ in terms of the random effects u and subsequently substituting these expressions in the log-likelihood and solving for u. The reason we elaborate here on this approximation method is that it will later we used for the mixed-effects random forest model.

Now transforming (2.7) into the log-likelihood $l(P|\boldsymbol{y}) = \log(L(P|\boldsymbol{y}))$ we get that:

$$\begin{aligned}
l(P|\boldsymbol{y}) = \sum_{i=1}^{K} & -\frac{n_i}{2} \log(2\pi\sigma_R^2) - \frac{1}{2} \log|G(\boldsymbol{\sigma}_b)| - \frac{1}{2} \log|Z_i^T Z_i + G^{-1}(\boldsymbol{\sigma}_b)| - \\
& \frac{1}{2\sigma_R^2} (\boldsymbol{y}_i - X_i\boldsymbol{\beta} - Z_i\boldsymbol{u}_i^*)^T (\boldsymbol{y}_i - X_i\boldsymbol{\beta} - Z_i\boldsymbol{u}_i^*) - \frac{1}{2} \boldsymbol{u}_i^{*T} G^{-1}(\boldsymbol{\sigma}_b) \boldsymbol{u}_i^*
\end{aligned}$$

Alternatively, we can write the log-likelihood in terms of the larger matrices and omit the constant parts:

$$l(P|\boldsymbol{y}) = -\frac{N}{2}\log(2\pi\sigma_R^2) - \frac{K}{2}\log|G(\boldsymbol{\sigma}_b)| - \frac{K}{2}\log|Z^TZ + G^{-1}(\boldsymbol{\sigma}_b)| - \frac{K}{2\sigma_R^2}(\boldsymbol{y} - X\boldsymbol{\beta} - Z\boldsymbol{u}^*)^T(\boldsymbol{y} - X\boldsymbol{\beta} - Z\boldsymbol{u}^*) - \frac{K}{2}\boldsymbol{u}^{*T}G^{-1}(\boldsymbol{\sigma}_b)\boldsymbol{u}^*, \tag{2.8}$$

where N is the total number of observations and K the number of subjects. This log-likelihood needs to be maximized in order to find the maximum likelihood estimators. However, we do not know what the random effects $\boldsymbol{u}_i$ $(1 \leq i \leq K)$ are yet. The estimates for random effects and the estimates of the other parameters depend on each other, so they cannot be estimated simultaneously at once. Luckily, there exist several methods that deal with issue, and one of them will be discussed in the next section.

2.3 Expectation Maximization Algorithm

The most common tool used in statistical software (for example in R and Python) for solving the log-likelihood functions in the previous section is the expectation maximization algorithm (EM). The idea behind this algorithm is that the maximum likelihood estimators that follow from maximizing the log-likelihood in (2.8) can be found by iteratively executing two steps:

- **Initialization**: The random effect estimates for $\boldsymbol{u}$, σ_R and the variance-covariance matrix G^* need to be initialized. A common choice for this initialization is to have the identity matrix for $\hat{G}^*(0)$, $\hat{\boldsymbol{u}}(0) = \boldsymbol{0}$ and $\hat{\sigma}_R(0) = 1$.

- **Expectation Step**: Conditional on the current estimates of $P = (\boldsymbol{\beta}, \boldsymbol{\sigma}_b, \sigma_R)$ find the random effects $\hat{\boldsymbol{u}}$ at step $k \geq 1$:

$$\hat{\boldsymbol{u}}_{(k)} = G^*_{(k-1)}Z^TV^{-1}_{(k-1)}(\boldsymbol{y} - X\hat{\boldsymbol{\beta}}_{(k-1)}),$$

  with $V_{(k-1)} = \hat{\sigma}_R^2 I_N + ZG^*Z^T$.

- **Maximization Step**: Obtain the new estimates for $\boldsymbol{\sigma}_b$ and σ_R.

$$\hat{\boldsymbol{\beta}}_{(k)} = (X^TV^{-1}_{(k-1)}X)^{-1}X^TV^{-1}_{(k-1)}\boldsymbol{y},$$

$$\hat{\sigma}^2_{R(k)} = \frac{1}{N}\sum_{i=1}^{n}\boldsymbol{\varepsilon}^T_{i(k)}\boldsymbol{\varepsilon}_{i(k)} + \hat{\sigma}^2_{R(k-1)}(n_i - \hat{\sigma}^2_{R(k-1)} \cdot \text{trace}(V^{-1}_{i(k-1)}))$$

$$\hat{G}^*_{(k)} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{u}_{i(k)}\boldsymbol{u}^T_{i(k)} + \hat{G}^*_{(k-1)} - \hat{G}^*_{(k-1)}Z_i^TV^{-1}_{i(k-1)}Z_i\hat{G}^*_{(k-1)}$$

- **Iterate**: Iterate the previous two steps until convergence. This is typically a criterion based on the likelihood of the model, i.e. stop iterating once the change is the likelihood is sufficiently small.

This algorithm is purely based on maximum likelihood estimates. There is however criticism on this method, since it has a tendency to underestimate the variance components due to ignoring a loss of degrees of freedom caused by estimating $\boldsymbol{\beta}$. The restricted maximum likelihood (REML) methods corrects for this issue and is often the preferred option for estimating the effects in linear mixed models. With this method the likelihood becomes a little bit more complicated, and therefore the estimators also change, but the principle of the EM algorithm remains unchanged. We leave the technicalities of the REML algorithm to the reader [9].

Chapter 3

Random Forest

Random forest is an ensemble method that makes use of regression (or decision) trees and can be used for either classification problems or regression. In this chapter we will discuss random forests and their strengths, but first we will explain what a regression tree is.

3.1 Regression Tree

Our dataset is denoted by $(X, \boldsymbol{y})$ with $X = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n)^T$, i.e. there are n observations in the dataset such that observation i corresponds to $(x_{i1}, x_{i2}, ..., x_{ip}, y_i)$. A binary regression tree is an algorithm that keeps making binary splits on the input variables $(\boldsymbol{x}_1, ..., \boldsymbol{x}_p)$ such that it can predict the response value $\boldsymbol{y} = (y_1, ..., y_n)$ [6]. Each split is made at an internal node (i.e. a node that is split) and the predictive value for all observations within a node is simply the average response value. The process of splitting continues until a stopping criterion has been met or node split cannot be further split. Such a node that will not be partitioned further is called a leaf node. Let $\hat{V}$ and V^* denote the set of internal nodes and leaf nodes respectively. The response at any node (either leaf or internal) is a constant c_m where m corresponds to the node. Considering that in any tree each observation ends up in one unique leaf node, the prediction of the random forest of observation i with its input $\boldsymbol{x}_i = (x_{i1}, x_{i2}, ..., x_{ip})$ is as follows:

$$f(\boldsymbol{x}_i) = \sum_{m \in V^*} c_m \mathbf{1}\{\boldsymbol{x}_i \in m\}. \tag{3.1}$$

For regression trees the optimization criterion is simply the residual sum of squares as we already have seen with linear regression:

$$\text{RSS} = \sum_{i=1}^{n} (y_i - f(x_i))^2.$$

Let I_m denote the set of observations that are in node m. Then if we want to minimize the sum of squares and take into account the expression in (3.1) it follows directly that the best estimate of c_m is simply the average:

$$\hat{c}_m = \frac{1}{|I_m|} \sum_{\boldsymbol{x}_i \in I_m} y_i,$$

here $|I_m|$ simply denotes the number of observations in this node.

Now remains the question how it is decided what the splitting variable and what the splitting criterion should be. If we make a split on a variable j (that is $\boldsymbol{x}_j = (x_{1j}, x_{2j}, ..., x_{nj})$) with split point threshold q the child nodes are:

$$m_1(j,q) = \{I_m | \boldsymbol{x}_j \leq q\}, \quad m_2(j,q) = \{I_m | \boldsymbol{x}_j > q\}.$$

The optimum values for j and q in terms of the largest decrease in mean squared error are as follows:

$$\min_{j,q} \left[\sum_{\boldsymbol{x}_i \in m_1(j,q)} \left(y_i - \frac{1}{|I_{m_1(j,q)}|} \sum_{\boldsymbol{x}_i \in I_{m_1(j,q)}} y_i \right)^2 + \sum_{\boldsymbol{x}_i \in m_2(j,q)} \left(y_i - \frac{1}{|I_{m_2(j,q)}|} \sum_{\boldsymbol{x}_i \in I_{m_2(j,q)}} y_i \right)^2 \right].$$

The specific value for q in this optimization problem is found by trying out all values that the variable $\boldsymbol{x}_j$ takes in this dataset and subsequently choosing the one that gives the lowest value. Since the sum of the mean squared errors of the child nodes is lower than the mean squared error of the parental node, there is necessarily a deacreas in mean squared error.

In principle it is possible to grown a tree until each leaf node consists only of one observation. These fully grown trees potentially overfit the data, so there is a certain need to impose a stopping criterion. Different criteria for this end exist, such a tree size, e.g. maximum depth or maximum number of nodes, minimum decrease in mean squared error, minimum node size, and more. Arguably the most popular one is maximum *node size*, that says a node should not be split further if its number of observations is less than or equal to maximum node size [12]. For regression trees node size is typically set at 5, but research has shown that tuning this hyperparameter (that is trying different values for this parameter) may lead to substantially better results [8].

3.2 Random Forest algorithm

A random forest model can be based on either regression trees or classification trees. In this thesis we are solely interested in regression trees. It is an extension on bootstrap aggregating methods (bagging). The general idea behind bagging is that by averaging many models with high variance and low bias, we can substantially decrease the variance, resulting in a more stable model.

A random forest consisting of B trees consists of the following steps [1]:

- For all trees $b = 1, 2, ..., B$:
    1. Draw randomly n observations with replacement from the data. We call this the bootstrap sample D_b

2. Then grow a regression tree T_b on this bootstrapped data until a certain minimum nodesize has been reached. The regression tree follows the same procedure as described in Section (3.1), except for one change: at each split not all p predictor variables are considered. Instead, at each split $m \leq p$ variables are randomly sampled without replacement. The the optimal split is then based on those m predictors only. For example: if we have 5 predictor variables and we pick m to be 3, we might at the first split randomly draw variables 2 and 5, and then the split will be based on one of these two variables. This process of drawing m variables is done for every split in each tree.

- Ensemble all B trees: $\{T_b\}_{1 \leq b \leq B}$, where $T_b(\cdot)$ denotes the resulting prediction function for the b-th grown tree.

A special property of random forests is that their predictor makes use of out-of-bag samples. This means that for the prediction of an observation $\boldsymbol{x}_i$ that was contained in the training dataset, we only make use of those regression trees for which $\boldsymbol{x}_i$ was not in its bootstrap sample. Let $\xi(\boldsymbol{x}_i)$ denote the set of regression trees that did not contain observation $\boldsymbol{x}_i$ as a sample in its training data. Then in formulas we have that the random forest regression predictor for $\boldsymbol{x}_i$ is:

$$f_{\mathrm{RF}}^{B}(\boldsymbol{x}_i) = \frac{1}{|\xi(\boldsymbol{x}_i)|} \sum_{b \in \xi(\boldsymbol{x}_i)} T_b(\boldsymbol{x}_i).$$

A good choice for hyperparameter m in regression settings is suggested as $p/3$, but this is actually a hyperparameter that needs to be tuned, since the behaviour of the random forest model can quite depend on its hyperparameters, as we will discuss in the next section [6].

3.3 Influence Hyperparameters

When using existing implementations for random forest in both *Python* and *R* and using the same set of hyperparameters for random forest we saw that that for several simulation scenarios we did not always get the same parameter estimates. Therefore we did a closer inspection of the differences between random forest implementations in *Python* using sklearn's *RandomForestRegressor* [10], in *R* using *randomForest* [7].

First of all, in order to be able to perform a fair comparison, we must ensure that the various hyperparameters that are involved in a random forest algorithm have the same values across the different implementations. Although the two implementations have different sets of hyperparameters that can be changed, it is possible to make sure that *Python* and *R* should be doing exactly the same based on the hyperparameter values. The following list of hyperparameters needs to be set, since by default they are not the same for a random forest of regression trees:

- Number of trees, called *ntree* in *R*, *n_estimators* in *Python*.
- Minimum number of bootstrapped observations required to split a node, called *nodesize* in *R*, *min_features_split* in *Python*.

- Number of predictor variables that are randomly sampled as candidates for a node split, called *mtry* in *R*, *max_features* in *Python*.

For all other hyperparameters, such as maximum number of terminal nodes, we use the default settings, which in all cases mean no further restrictions on growing trees. With respect to *randomForest*'s nodesize it should be noted that the above definition is not in alignment with the package's explanation, which reads "minimum size of terminal nodes", i.e. the minimum number of observations in any terminal node. By closer inspection of the grown trees, we found that minimum size of terminal nodes is often lower than nodesize and after examining the source code we saw that nodesize actually refers to the number of observations required to split an internal node. In other words, the nodesize hyperparameter is a much looser constraint on growing a trees than is implied.

Furthermore there is a significant difference between the two implementation regarding the strictness of the *mtry* parameter.

Chapter 4

Mixed Effects Random Forest

The model considered is mixed-effects random forest for clustered data [4]. Related literature can be found in [3] and [5]. The model is quite similar to linear mixed model, but now the fixed part, which was previously estimated by $X\boldsymbol{\beta}$ is now replaced by a random forest model. The fixed part is the same for all observations, and the random part is unique for every cluster within the data. Say there are K groups, and every group i has n_i observations for $i = 1, 2, ..., K$. Furthermore assume there is just one outcome variable y, then the MERF model is defined as follows:

$$\begin{aligned} &\boldsymbol{y}_i = \boldsymbol{f}(X_i) + Z_i\boldsymbol{u}_i + \boldsymbol{\varepsilon}_i, \\ &\boldsymbol{u}_i \sim \mathcal{N}(\boldsymbol{0}, G), \quad \boldsymbol{\varepsilon}_i \sim \mathcal{N}(\boldsymbol{0}, R_i), \quad i = 1, ..., K, \end{aligned}$$

where $\boldsymbol{y}_i = (y_{i1}, ..., y_{in_i})$ is the vector of outcomes for cluster i, X_i and Z_i are the known design matrices belonging to the fixed-effects and random-effects respectively, $\boldsymbol{u}_i$ is the unknown vector of random effects and $\boldsymbol{\varepsilon}_i$ is the vector of residuals. The fixed part $\boldsymbol{f}(X_i)$ is estimated by a random forest. In principle f could be estimated by any other machine learning tool, such as support vector machines or neural networks. Furthermore we assume that $\boldsymbol{u}_i$ and $\boldsymbol{\varepsilon}_i$ are independent and normally distributed with covariance matrices G and R respectively. It is also assumed that observations from different clusters are independent. For the remainder of this thesis we will we also make the assumption that R_i is a diagonal matrix of the form $R_i = \sigma^2 I_{n_i}$. So there is no difference between the residual structures and sizes of different clusters.

4.1 Algorithm

The algorithm used to find the random effects coefficients in the model is similar to the expectation-maximization algorithm that is used for estimating the variance components in linear mixed-effects models [9]. In words, the algorithm consists of the following steps:

- STEP 0: initialize all random effect coefficients at 0, $\sigma_R^2 = 1$ and G as the identity matrix: $G = I_m$. Set $k = 0$ as the iteration number.
- STEP 1: i) update $k = k + 1$, then subtract the random part (the current estimate) from the outcome variable: $\boldsymbol{y}^*_{i(k)} = \boldsymbol{y}_i - Z_i\hat{\boldsymbol{u}}_{i(k-1)}$;

ii) train the random forest, based on the updated outcome variable $\boldsymbol{y}^*_{i(k)}$ using the method described in previous chapter;
iii) obtain the estimate for each observation j using trees that didn't have this observation j in their training set and;
iv) update $\boldsymbol{u}_i$:

$$\hat{\boldsymbol{u}}_{i(k)} = \hat{G}_{(k-1)} Z_i^T V^{-1}_{i(k-1)} (\boldsymbol{y}_i - \hat{f}(X_i)_{(k)}), \text{ for } i = 1, , ..., n.,$$

where the V matrix is given by:

$$V_{(ik-1)} = Z_i \hat{G}_{(k-1)} Z_i^T + \hat{\sigma}^2_{R(k-1)} I_{n_i}.$$

- STEP 2: update the covariance matrix G and the estimate for σ^2_R based on the updated residual estimates.

$$\hat{\sigma}^2_{R(k)} = \frac{1}{N} \sum_{i=1}^{n} \hat{\boldsymbol{\varepsilon}}^T_{i(k)} \hat{\boldsymbol{\varepsilon}}_{i(k)} + \hat{\sigma}^2_{R(k-1)} (n_i - \hat{\sigma}^2_{R(k-1)} \cdot \text{trace}(V_{i(k-1)}))$$

$$\hat{G}_{(k)} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{u}^T_{i(k)} \boldsymbol{u}_{i(k)} + \hat{G}_{(k-1)} - \hat{G}_{(k-1)} Z_i^T V^{-1}_{i(k-1)} Z_i \hat{G}_{(k-1)},$$

where $\hat{\boldsymbol{\varepsilon}}_{i(k)}$ is simply the part not explained by the random forest and random effects estimates: $\hat{\boldsymbol{\varepsilon}}_{i(k)} = \boldsymbol{y}_i - \hat{f}(X_i)_{(k)} - Z_i \hat{\boldsymbol{u}}_{i(k)}$ for all $i = 1, 2, ..., K$.
It should be noted that these steps are very similar to the steps of the EM-algorithm we explained in Chapter 2 regarding the linear mixed model.

- STEP 3: keep repeating steps 1 and 2 until convergence is satisfied according to the generalized likelihood criterion:

$$\begin{aligned} \text{GLL}(f, \boldsymbol{u}|\boldsymbol{y}) = &\sum_{i=1}^{n} (\boldsymbol{y}_i - f(X_i) - Z_i \boldsymbol{u}_i)^T R_i^{-1} (\boldsymbol{y}_i - f(X_i) - Z_i \boldsymbol{u}_i) + \boldsymbol{u}_i^T D^{-1} \boldsymbol{u}_i \\ &+ \log |G| + \log |R_i|) \end{aligned}$$

Let GLL_k define the above generalized likelihood criterion after iteration k, then we say that the algorithm has converged if:

$$\frac{|\text{GLL}_k - \text{GLL}_{k-1}|}{\text{GLL}_{k-1}} < \delta,$$

for some small $\delta > 0$.

We use a relative convergence criterion here instead of an absolute one, since the actual value of the generalized likelihood criterion may vary greatly between different problems and datasets, so it does not make much sense to make the convergence criterion independent of the real value of the GLL.

In recent literature the MERF models have been used in predictive analysis, but have not been evaluated in terms of parameter estimation. Research papers reporting the use of MERF in real world modeling problems indicate that MERF can outperform regular random forest models and other machine learning models in many settings, even with a simple random structure [11] [14]. So clearly the MERF shows great potential in terms of predictive performance and goodness of fit (criteria such as R^2). However, these papers using MERF do not check if for example the random effect estimates are reasonable, or if the fitted model can actually determine which predictor variables influence the response and in what way.

4.2 Initialization sensitivity

The MERF algorithm makes use of a standard initialization. It is commonly known that initialization values can sometimes have a significant influence on the performance of algorithms, especially if they are relatively far away from the real values. Since there is no literature available on initialization sensitivity of MERF model, we conducted a small research on this issue using a data setting that is similar to the one of the mixed-effects random forest paper [4]. Using initialization values for the random intercepts on group level that are close to the real ones instead of 0, we did not find a better performance after a sufficient amount of iteration steps, i.e. until convergence. This holds true even if the random effects are much larger than the fixed effects.

Chapter 5

Model Evaluation

in this Chapter we will explain how we will perform a simulation study to evaluate estimation of marginal fixed effects and random effects for the model we described in the previous Chapters. First, in Section (5.1) we explain how the data is simulated and which models we will fit using this data. Then Section (5.2) will outline how the performances of the various fitted models will be assessed and compared.

5.1 Simulation Set Up

5.1.1 Data

The different models as discussed in previous section are to be evaluated and compared on the basis of an extensive simulation study. The general frame of the simulated data is as follows:

- Observations are distributed over different groups and each observation has one single response variable. The number of groups and how many observations are within each group are to be varied. We do however impose balanced groups: all groups are of the same size.
- Predictor variables exist on two levels: group level and individual level. Correlation between variables on the same level is also considered and those variables are multivariate normally distributed with mean 0, unit variance and some correlation.
- Each group has a random intercept, which is i.i.d. normally distributed with mean 0 and variance σ_{INT}.
- The residuals are also i.i.d. normally distributed with mean 0 and variance σ_R.
- Different response models $f(X)$ are considered: one polynomial and one that also contains non-polynomial terms. The specific formulas will be displayed later.

In formulas the simulated data can be summarized in this manner (where $f()$ will be explained later on):

$$y_{ij} = \boldsymbol{f}(\boldsymbol{x}_{ij}) + u_i + \varepsilon_{ij},$$
$$u_i \sim \mathcal{N}(0, \sigma_{\text{INT}}), \quad \boldsymbol{\varepsilon}_{ij} \sim \mathcal{N}(0, \sigma_{\text{R}}), \text{ for } \quad i = 1, ..., K, \text{ and } j = 1, ..., n.$$

Furthermore we can make a distinction between within group variables and between group variables: there are p predictors in this two-level model: $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_p$, and 40% of these predictor variables vary only between groups, we call this set X_b, and the remaining 60% vary within groups and we call this set X_w.

$$\boldsymbol{x}_{ij} = (\boldsymbol{x}_{w,ij}, \boldsymbol{x}_{b,i}), \quad \boldsymbol{x}_{w,ij} \sim \mathcal{N}(\boldsymbol{0}, M(\rho)_{3p/5}), \quad x_{b,i} \sim \mathcal{N}(\boldsymbol{0}, M(\rho)_{2p/5}),$$

where $M(\rho)_r$ denotes the following $r \times r$ matrix:

$$M(\rho)_r = \begin{bmatrix} \rho & 1 & \dots & 1 \\ 1 & \rho & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & \rho \end{bmatrix}$$

Table 5.1 gives an overview of the various settings for the data simulation that are considered, along with their respective labels and simulated levels. An explanation of those factors will be given below.

Factor	Label	Levels
Number of groups	K	{25,100}
Group size	n	{3,15}
Number of Predictors	p	{5,25}
Correlation Fixed Effects	ρ	{0,0.8}
Random Effect Size	σ_{INT}	{0.7,3}
Residual Size	σ_{R}	{1,2}
Response complexity	$f(X)$	$\{f_n, f_p\}$

Table 5.1: Factors for simulation study with their corresponding levels

We will execute a full-factored simulation where every possible combination is considered, so that means we will have a total of $2^7 = 128$ distinct scenarios. The first two factors mentioned in the table, number of groups and group size, determine the number of observations that are in the dataset. Since both have two levels, there are four flavours regarding the number of observations: 75, 300, 375 and 1500.
Random Effect Size and Residual Size together determine the intraclass correlation coefficient (ICC):

$$\text{ICC} = \frac{\sigma^2_{\text{INT}}}{\sigma^2_{\text{INT}} + \sigma^2_{\text{R}}}$$

Table 5.2 gives a quick overview of the four different ICC levels that are in our simulation.

σ_{INT}	σ_{R}	ICC
0.7	1	0.329
0.7	2	0.109
3	1	0.9
3	2	0.692

Table 5.2: Intraclass Correlation Coefficient

Regarding the response complexity we have two models, one which is purely polynomial f_p and one that also contains non-polynomial terms f_n:

$$\begin{aligned} f_p(X) &= \boldsymbol{x}_1 + \boldsymbol{x}_2 - 2\boldsymbol{x}_2^2 - 0.5\boldsymbol{x}_3 + 1.5\boldsymbol{x}_4 - 2\boldsymbol{x}_5 + \boldsymbol{x}_5^2 \\ f_n(X) &= \boldsymbol{x}_1 - 2\boldsymbol{x}_1^3 + 0.5\exp(\boldsymbol{x}_2) + 2\log(|\boldsymbol{x}_3\boldsymbol{x}_5|) - 4 \cdot \mathbf{1}\{|\boldsymbol{x}_4 > 0.5|\}, \end{aligned} \tag{5.1}$$

where $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are between group variables, and the other predictors are within group variables. Figure 5.1 shows the behaviour of $f_n(X)$.

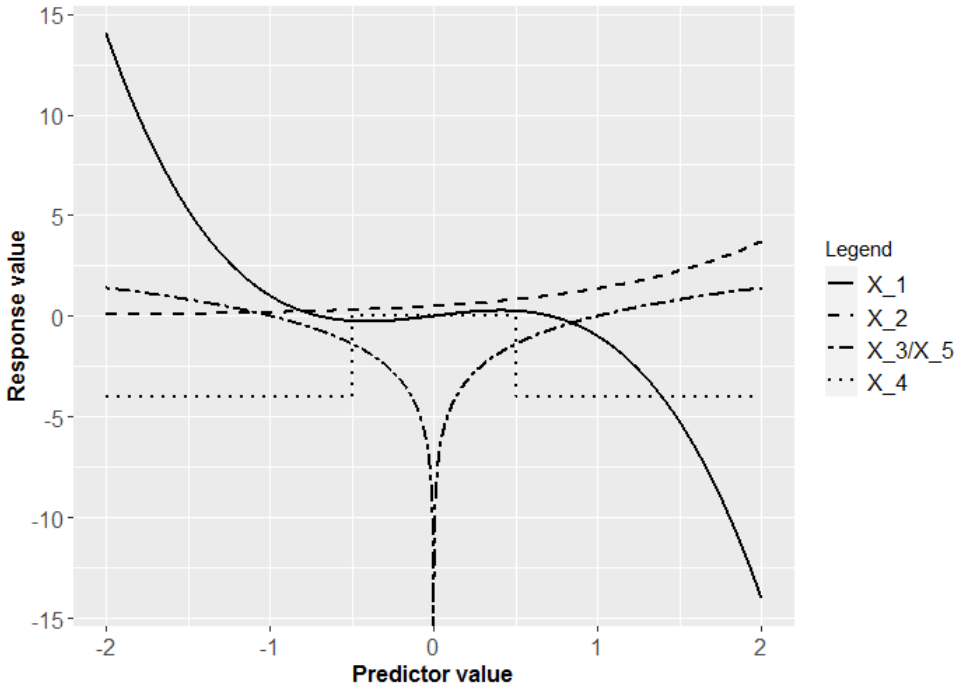

Figure 5.1: The influence of various predictor variables on the response for the non-polynomial response model f_n.

5.1.2 Models

Using the simulated datasets we will compare four models: linear regression, linear mixed model, random forest, and mixed-effects random forest. Each of those models was evaluated using statistical software *R*. The modeling choices for those models are as follows:

- For linear regression (**LM**), as discussed in Section 2.1, *R*'s *lm* function is used and for the linear mixed model (**LME**) we use the *lmer* function with the restricted maximum likelihood (REML) method. As for the parameters to be estimated we will use a second order model without any interactions and furthermore we assume that we know the random structure (there is only a random itnercept), so that the fitted model for observation j of group i is as follows:

$$y_{ij} = \sum_{k=1}^{p} \beta_{k,1} x_{k,ij} + \beta_{k,2} x_{k,ij}^2 + u_i + \varepsilon_{ij}, \tag{5.2}$$

  where p is the number of predictor variables. For the linear regression model the u_i is obviously not taken into account.
  Furthermore this fitted model is independent of the number of variables in the dataset. So if there are 25 variables, we estimate 50 fixed-effects coefficients. Hence, there is no variable selection executed before a final fit has been made.

- For random forest models, both the regular one and the mixed-effects, we use the function *randomForest*. Here we take 500 regression trees, but the models differ in terms of other hyperparameters. **RF1** and **MERF1** use nodesize 5 and m (*mtry* in *R*) $= p/3$. **RF2** and **MERF2** use nodesize 2 and $m = 0.8p$.

All in all we will fit six model variants on every dataset. How these fits and performance will be compared is outlined in the next Section.

5.2 Evaluation Criteria

As mentioned in the research questions, the quality of both the fixed effect estimates as well as the random effect estimates will be assessed in terms of bias and variance and their trade-off. For the random effects retrieving the estimates is straightforward since the MERF models and the linear mixed model give direct estimates of σ_{INT} and σ_{R}. However, since random forests do not estimate regression coefficients as opposed to the linear models, comparing fixed effect estimates for all models is not straightforward.

5.2.1 Estimating fixed effects

In order to make a fair comparison between the random forest models and the linear models we will use *partial dependence functions* [16]. Those functions evaluate what the marginal effect is of a variable on the outcome of a model. It is typically used for machine learning models, but it can be used for any model.

Let $\boldsymbol{x}_S$ denote the variable for which we like to evaluate the partial dependence function and $\hat{f}$ the model. Then for a given point a the partial dependence function $\hat{f}_{\boldsymbol{x}_S}$ is computed as follows:

$$\hat{f}_{\boldsymbol{x}_S}(a) = \frac{1}{N} \sum_{i=1}^{N} \hat{f}(x_{iS} = a, X_{iC}) \tag{5.3}$$

where X_C denotes the set of all other predictor variables and N is the total number of observations. The partial dependence function evaluated at a point $\boldsymbol{x}_S = a$ replaces all values of $\boldsymbol{x}_S$ with a, but keeps all other variables at their original values, and then computes what the model would predict for any of the observations and computes the average of them. Now, if we want to know what the marginal effect is of predictor $\boldsymbol{x}_S$ on the response, we can look at an interval $[u, t]$ and compute how much the partial dependence function differs between the endpoints u and t. In formula form the partial dependence estimate of model $\hat{f}$ for the fixed effect of variable $\boldsymbol{x}_S$ is denoted by:

$$FE_{\boldsymbol{x}_S}(u,t) = \frac{1}{t-u}\left(\frac{1}{N}\sum_{i=1}^{N}[\hat{f}(x_{iS}=t, X_{iC}) - \hat{f}(x_{iS}=u, X_{iC})]\right), \quad t > u \in \mathbb{R}. \tag{5.4}$$

In our simulation study the fixed effects will be evaluated on two intervals: on $[-1, 1]$, which is minus and plus the standard deviation of the predictor variables, and on $[0, 1]$. Since the linear models only have first and second order terms in their model formulas (5.2), the interval $[-1, 1]$ corresponds to the linear regression coefficients $\beta_{i,1}$ for $i = 1, ..., p$.
For all six model fits (LM, LME, RF1, RF2, MERF1, MERF2) the fixed effects will be evaluated according to (5.4). That way we can make model comparisons fairly. The two different types of fixed effect estimates that will be made for predictor variable $\boldsymbol{x}_j$ $(j = 1, 2, ..., p)$ are as follows:

$$\alpha_{j,1} = FE_{\boldsymbol{x}_j}(-1, 1), \quad \alpha_{j,2} = FE_{\boldsymbol{x}_j}(0, 1) - FE_{\boldsymbol{x}_j}(-1, 1).$$

When the simulated response is dependent on only first and second order terms $\alpha_{j,1}$ and $\alpha_{j,2}$ correspond to the coefficients of the first and second order term respectively. Of course, if higher order terms or non-polynomial terms are added to the simulated model then this property does not hold any more. In order to clarify this we give two short examples based on the model formulas f_n and f_p (5.1).

Example 1: we want to determine the theoretical values of $\alpha_{2,1}$ and $\alpha_{2,2}$ for the polynomial model f_p. Note that we only have to look at the way $\boldsymbol{x}_2$ is incorporated in this formula, since it appears independent of the other variables.

$$\begin{aligned}
&FE_{\boldsymbol{x}_2}(0,1) = 1 \cdot 1 - 2 \cdot 1^2 - (1 \cdot 0 - 2 \cdot 0^2) = -1 \\
&FE_{\boldsymbol{x}_2}(-1,1) = 1/2 \cdot (1 \cdot 1 - 2 \cdot 1^2 - (1 \cdot -1 - 2 \cdot (-1)^2)) = 1 \\
&\implies \alpha_{2,1} = 1, \quad \alpha_{2,2} = -1 - 1 = -2,
\end{aligned}$$

which are indeed the coefficients as seen in the formula.

Example 2: we want to determine the theoretical values of $\alpha_{1,1}$ and $\alpha_{1,2}$ for the non-polynomial model f_n. Again, we only have to look how much $\boldsymbol{x}_1$ changes over the relevant intervals.

$$\begin{aligned}
&FE_{\boldsymbol{x}_2}(0,1) = 1 \cdot 1 - 2 \cdot 1^3 - (1 \cdot 0 - 2 \cdot 0^3) = -1 \\
&FE_{\boldsymbol{x}_2}(-1,1) = 1/2 \cdot (1 \cdot 1 - 2 \cdot 1^3 - (1 \cdot -1 - 2 \cdot (-1)^3)) = -1 \\
&\implies \alpha_{2,1} = -1, \quad \alpha_{2,2} = -1 + 1 = 0.
\end{aligned}$$

Note that $\alpha_{2,1}$ now does not align with the coefficient of the linear of $\boldsymbol{x}_1$, which is 1 instead of -1.
Table 5.3 summarizes the theoretical coefficients for the marginal fixed effects. For the non-polynomial model the effects of the variables that appear in the logarithm cannot be computed, since the logarithm does not have a real value at $x = 0$. We can however still compare the estimates of the various model fits, but the bias is not defined in this case.

Variable (j)	f_p		f_n	
	$\alpha_{j,1}$	$\alpha_{j,2}$	$\alpha_{j,1}$	$\alpha_{j,2}$
1	1	0	-1	-1
2	1	-2	$1/4(\exp(1) - \exp(-1))$	$1/4(\exp(1) + \exp(-1))$
3	-0.5	0	0	*
4	1.5	0	0	-4
5	-2	1	0	*

Table 5.3: Theoretical values of the marginal effects for the linear and non-linear response formula. * means that this coefficient does not have a real value, since $\log x$ goes to $-\infty$ as x goes to 0 (albeit very slowly).

5.2.2 Bias, variance, mean squared error

Every simulation scenario will be run 100 times, meaning that all six model fits are evaluated 100 times on different datasets that are constructed using the same data parameters. There are $2 \times p + 2$ estimates for to be estimated for the mixed-effects models and $2p$ for simple linear regression and random forest. For those estimators the bias, variance and mean squared error will be computed.

If we want to estimate the parameter ϑ by $\hat{\vartheta}$, the bias is $\mathbb{E}[\hat{\vartheta}] - \vartheta$. In case of a simulation, where we have the simulation estimators $\hat{\vartheta}_i$ for $i = 1, 2, ..., N$, the estimated bias of $\hat{\vartheta}$ is:

$$b_\vartheta(\hat{\vartheta}) = \frac{1}{N} \sum_{i=1}^{N} \hat{\vartheta}_i - \vartheta.$$

The estimated variance of that estimator is as follows:

$$\mathrm{Var}_\vartheta(\hat{\vartheta}) = \frac{1}{N-1} \sum_{i=1}^{N} \left(\hat{\vartheta}_i - \frac{1}{N} \sum_{j=1}^{N} \hat{\vartheta}_j \right)^2.$$

And the mean squared error is by the bias-variance decomposition given by:

$$\mathrm{MSE}_\vartheta(\hat{\vartheta}) = \mathrm{Var}_\vartheta(\hat{\vartheta}) + b_\vartheta(\hat{\vartheta})^2.$$

For the random effects estimates we will instead use a 'scaled' bias:

$$b_\vartheta(\hat{\vartheta}) = \frac{1}{N} \sum_{i=1}^{N} \frac{\hat{\vartheta}_i}{\vartheta}.$$

The variance is still computed in the same manner, but the bias-variance decomposition of the MSE does not hold any more with this definition.

5.2.3 Goodness of fit

It should be noted that the way the linear regression model and linear mixed model have been chosen (equation (5.1)), the linear models are able the capture the true response model in case of the polynomial model f_p, but will certainly be less predictive for the non-polynomial model f_n. Therefore we will also conduct an investigation whether the chosen linear mixed model is still somewhat a reasonable choice or not. This will be done by computing the R^2, which is the coefficient of determination and it the proportion of the variance in the response variable that can be explained by the model. Ideally you want this coefficient to be very close to 1. Furthermore we will check whether adding higher order terms to the fitted model can actually improve the R^2. The new fitted model is then:

$$y_{ij} = \sum_{k=1}^{p} \beta_{k,1} x_{k,ij} + \beta_{k,2} x_{k,ij}^2 + \beta_{k,3} x_{k,ij}^3 + u_i + \varepsilon_{ij}, \tag{5.5}$$

Chapter 6

Results

6.1 High level visualization

As a first impression of the results we make box plots of the estimates, where any facet within the plot represents one single scenario. The boxplots themselves summarize the 100 estimates of a model, as can be seen in figure 6.1. In the following Subsections we will for the fixed effects, random effects and prediction accuracy display grids of boxplots, that contain for a specific response formula all scenarios (i.e. 64 boxplots). Beneath these figures the most important and notable findings will be enumerated. Regarding the fixed effects, the scenarios for the two different response models, the polynomial and non-polynomial model, will be discussed separately, since those models do not have the same fixed effect input parameters.

6.1.1 Fixed Effects

When it comes to the fixed effects there are two types of predictors variables: the ones that vary between groups only and the ones that vary within groups. For each of those two types one variable will be highlighted and discussed more extensively, since for the other variables of the same type the overall findings are quite similar. These plots contain only one MERF and RF model, and more specifically RF2 and MERF2, as well as the linear mixed model and linear regression model. This is mainly done for sake of readability.

Polynomial model

As can be seen in figure 6.1 the random forest and mixed-effects random forest models often perform quite poorly regarding estimating the fixed effect of a predictor variable that varies between groups. Figure 6.2 is accompanied within a summary of the most notable findings for a within group variable. It is clear that the within group variable estimates ($\boldsymbol{x}_3$, x_4, x_5) are more promising for the MERF model. The estimates for the other predictors can be found in the appendix, in figures A.1 ($\boldsymbol{x}_2$), A.2 ($\boldsymbol{x}_3$) and A.3 ($\boldsymbol{x}_4$).

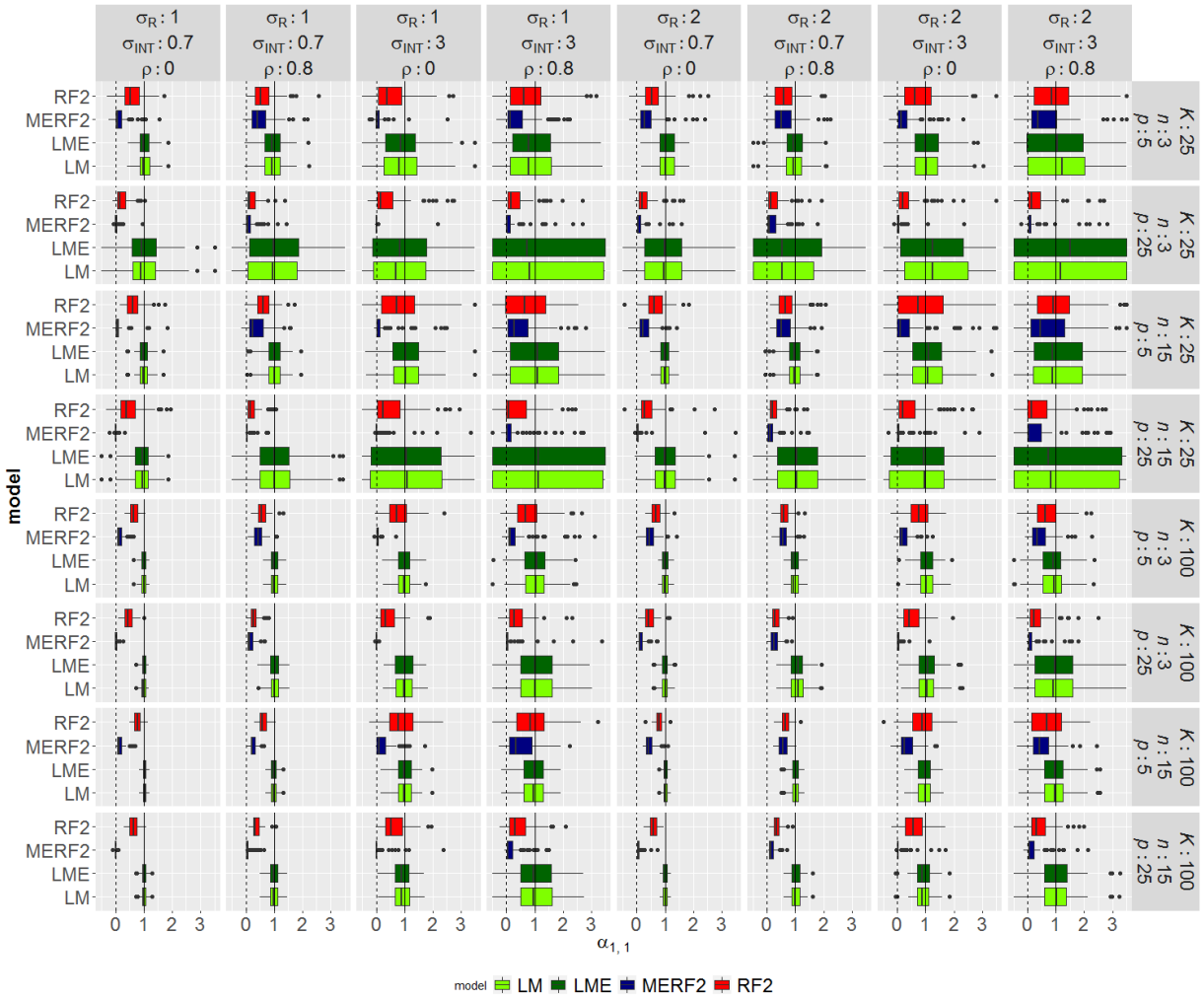

Figure 6.1: **Fixed effects of x_1, which is a between group variable, evaluated for the polynomial model**.

- When it comes to the MERF model, the estimates are in almost all cases heavily biased towards zero, meaning that the MERF model cannot adequately identify the effects of a between group variable. This 'malfunction' for MERF is more notable in the scenarios where there are many predictors ($p = 25$), but based on the plots the estimates are only somewhat better when there are few predictors ($p = 5$).

- The pairwise comparison between MERF and RF show that in none of these scenarios MERF is able to better pick up on a between group variable fixed effect than its regular counterpart.

- Although we can see that as the number of observations increases the variance of estimates decrease, the biases seem not to decrease for the RF and MERF models.

- For the MERF and RF models the data parameters as displayed in the columns (ρ, σ_{R} and σ_{INT}) seem to have rather little influence on the behaviour of estimates. For the linear models however a higher correlation and higher random effect have a negative influence on the variance.

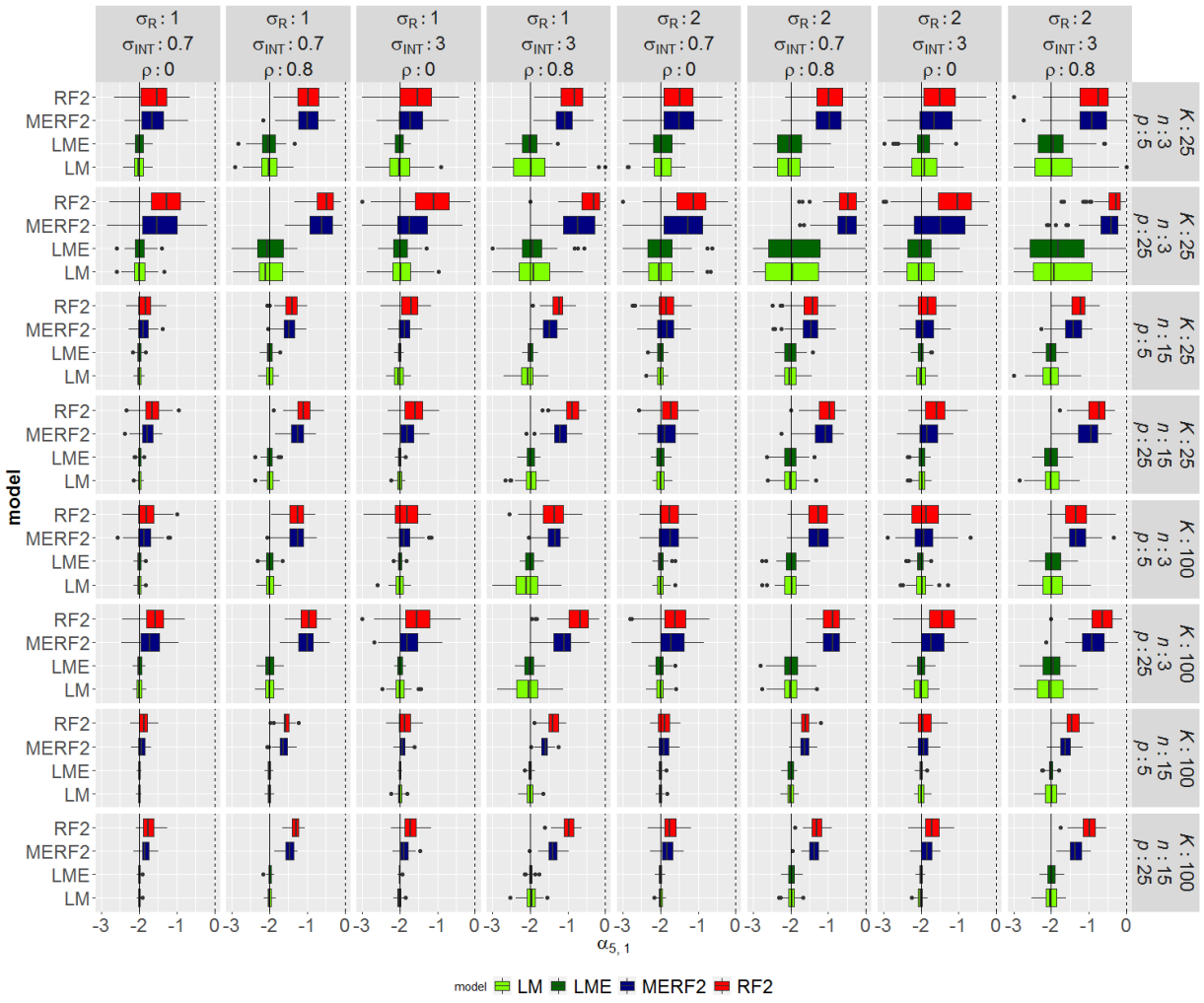

Figure 6.2: Fixed effects of x_5, which is a within group variable, evaluated for the polynomial model.

- Compared to between group variables estimates, the MERF model performs much better. However, in the vast majority of simulation scenarios the MERF and RF model are both clearly outperformed by the linear models in terms of bias and variance.
- By first sight it looks like the σ_{R} and σ_{INT} have very little influence on the MERF and RF estimates. On the other hand, $\rho = 0.8$ clearly has a negative affect on the MERF and RF estimates (seen by doing pairwise column comparisons).
- For all models both variance and bias decrease as the number of groups K and number of observation per group n increase, as one would expect.
- At least in terms of bias, the MERF model seems to outperform the RF model in virtually all scenarios.

Non-polynomial model

For the non-polynomial response, the linear models no longer have the right model formula, so fixed effect estimates will no longer be unbiased. Figures A.5 and 6.4 show the fixed effect estimates for a between group variable and within group variable respectively. The As for the polynomial response, we can conclude that the MERF model cannot adequately estimate the fixed effect of a between group variable.

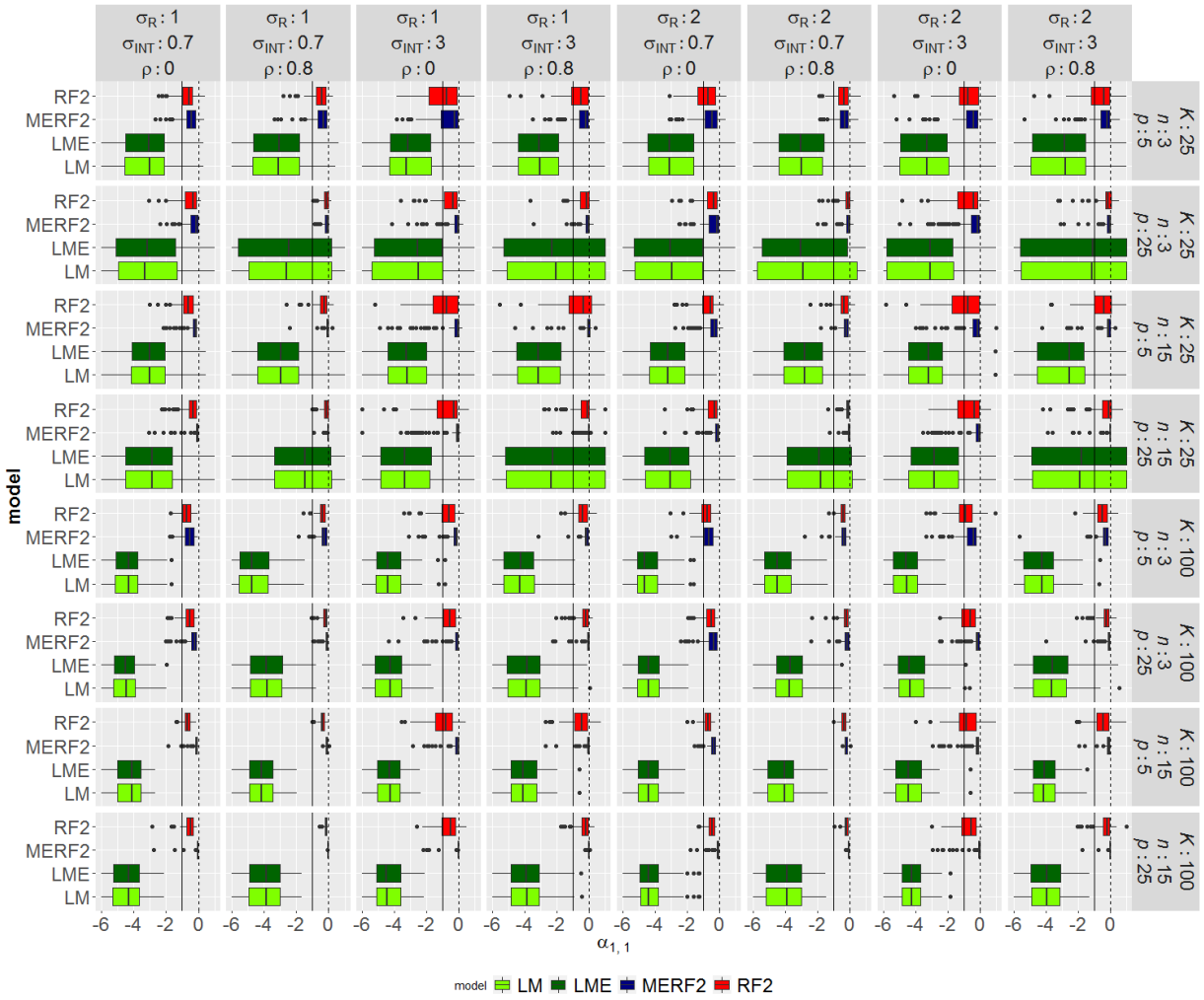

Figure 6.3: Fixed effects of x_1, which is a between group variable, evaluated for the non-polynomial model.

- MERF estimates are in most cases very close to 0, with relatively low variance. The random forest model seems to perform best in terms of bias. Also quite notably, the linear models' estimates have a rather large variance.

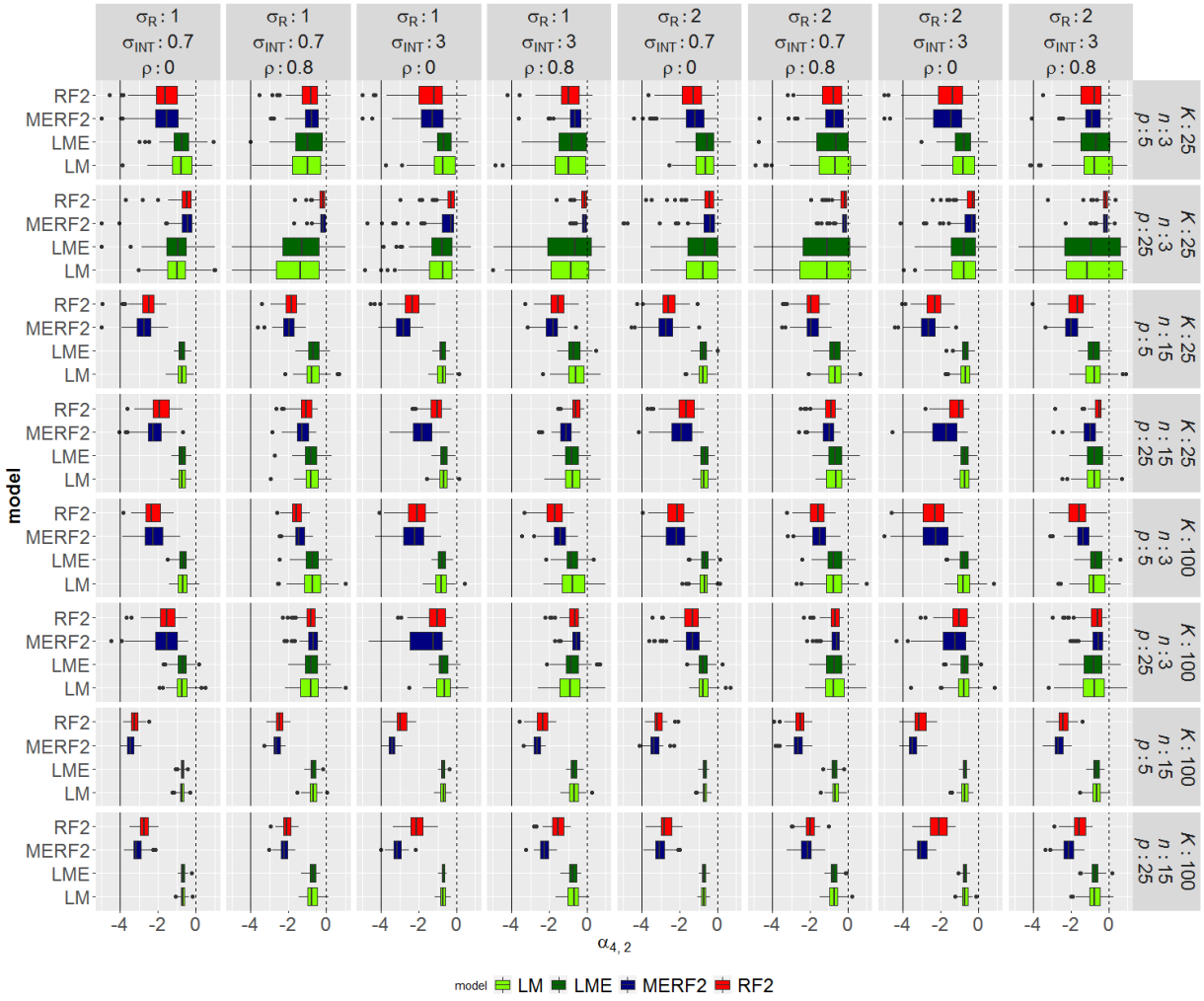

Figure 6.4: Fixed effects of x_4, which is a within group variable, evaluated for the non-polynomial model.

- We can see again that especially the MERF and RF model are sensitive for the correlation between predictors: it substantially increases the bias. Variance on the other hand does not increase at all.

- The MERF model outperforms the RF model in most cases in terms of bias. The difference in variance seems quite small between those two models.

6.1.2 Random Effects

Figure 6.5 displays the scaled bias of σ_{INT} for the polynomial model and A.7 for the non-polynomial response.

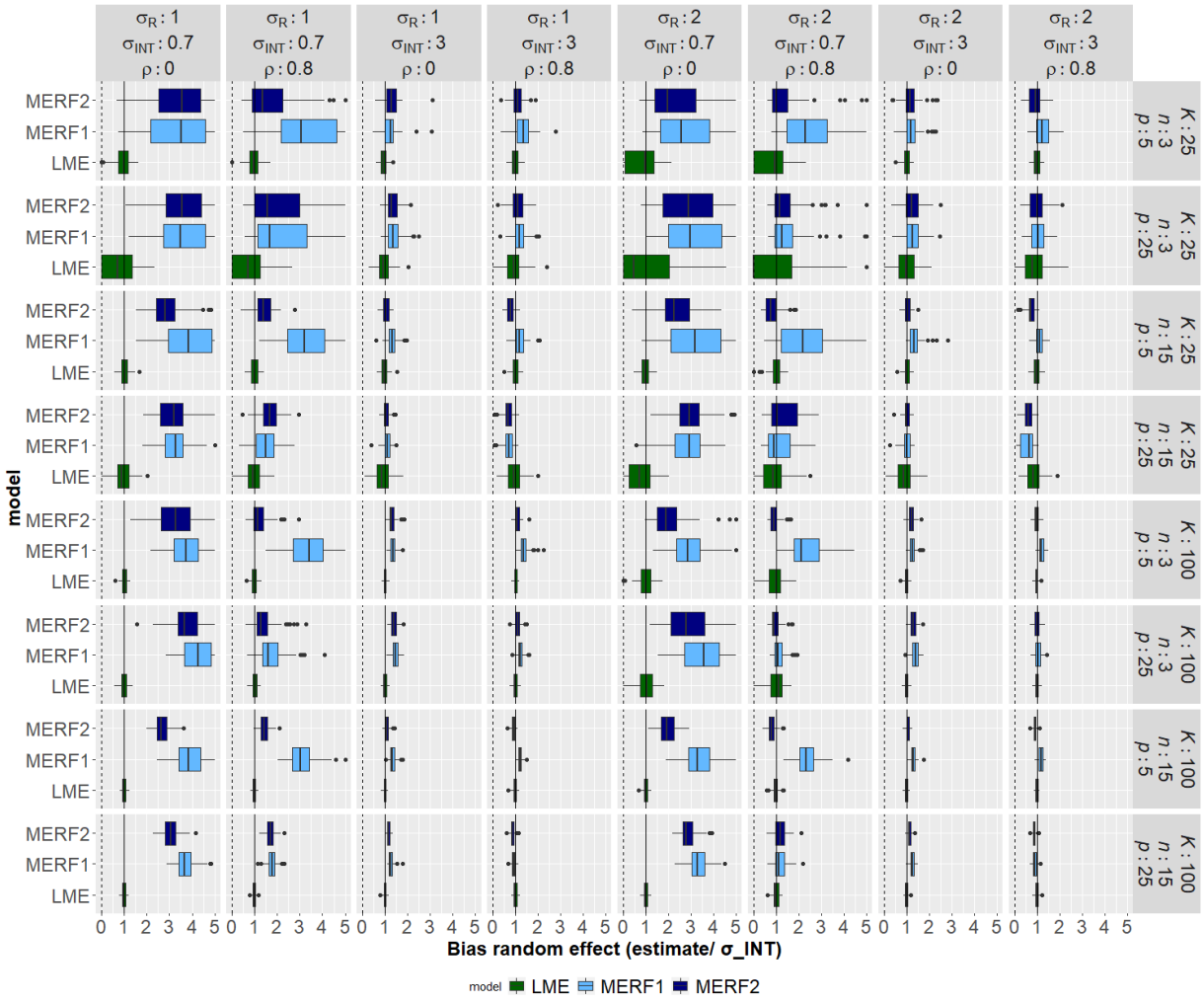

Figure 6.5: Random effect σ_{INT}, for the polynomial response model.

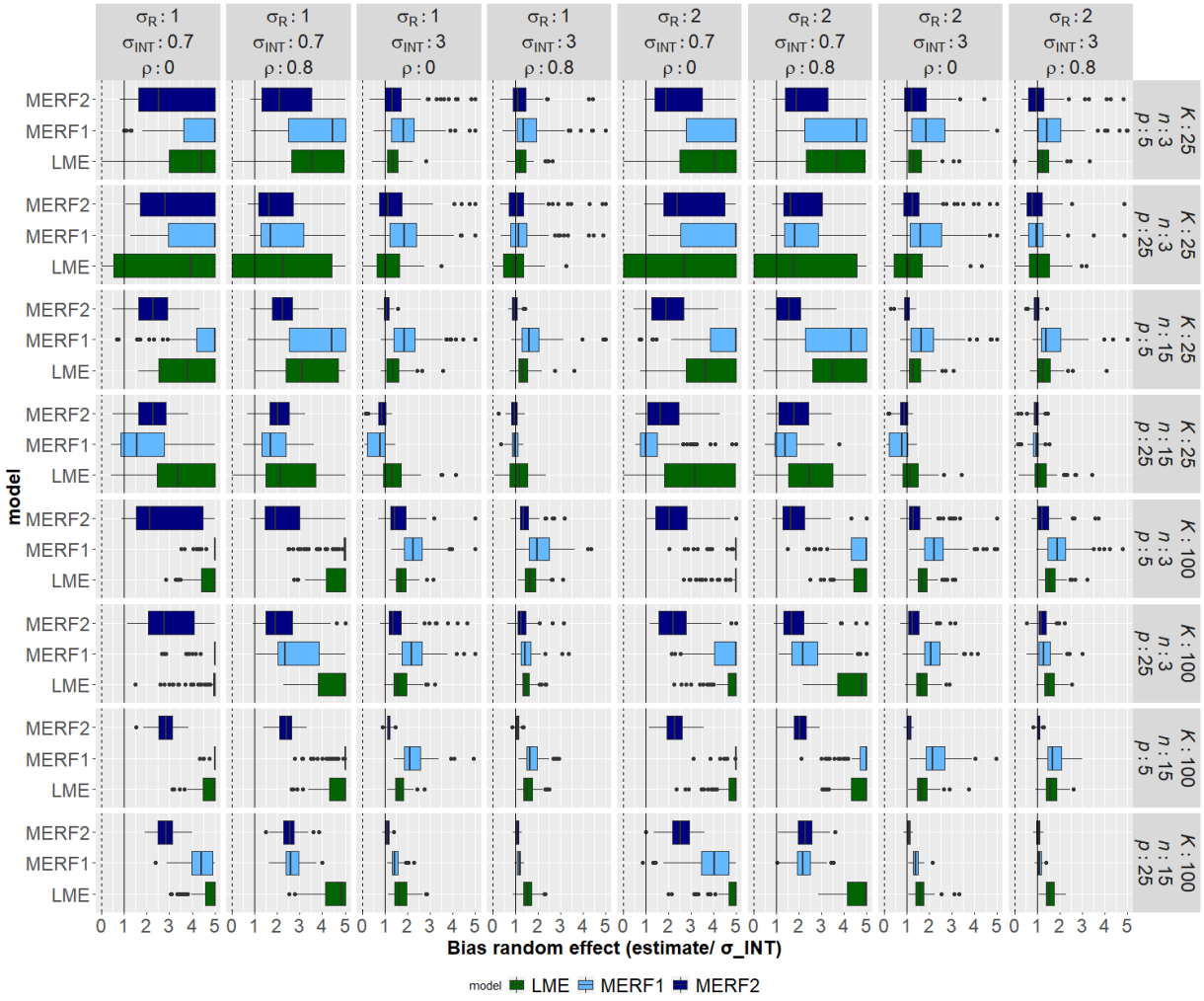

Figure 6.6: Random effect σ_{INT}, evaluated for the non-polynomial response model.

- We can see again that especially the MERF and RF model are sensitive for the correlation between predictors: it substantially increases the bias. Variance on the other hand does not increase at all.
- The MERF model outperforms the RF model in most cases in terms of bias. The difference in variance seems quite small between those two models.

6.2 Factor fixed effect plots

In this section we will based on the results of the previous section, make for the fixed effects box plots that summarize the results. For each factor level and simulation formula (polynomial or non-polynomial) there are 32 scenarios that have been simulated (and thus 3200 simulation runs), and the box plots display the averages of those 3200 simulation runs. The box plots show the squared bias placed on top of the variance, so that there is a clear visualization of the composition of the mean squared error, i.e. what part of the estimator's mean squared error can be attributed to the variance of the estimator and what part to its bias.
The fixed effect plots for the first fixed effect (referred to as $\alpha_{j,1}$) are displayed in this section, for the other fixed effect the plots can be found in the appendix. It should be noted that for the non-polynomial response model the displayed bias for $\alpha_{3,2}$) and $\alpha_{5,2}$) is actually not the bias at all, since these two coefficients are not defined. Instead, these show the squared estimates.

6.2.1 Polynomial response

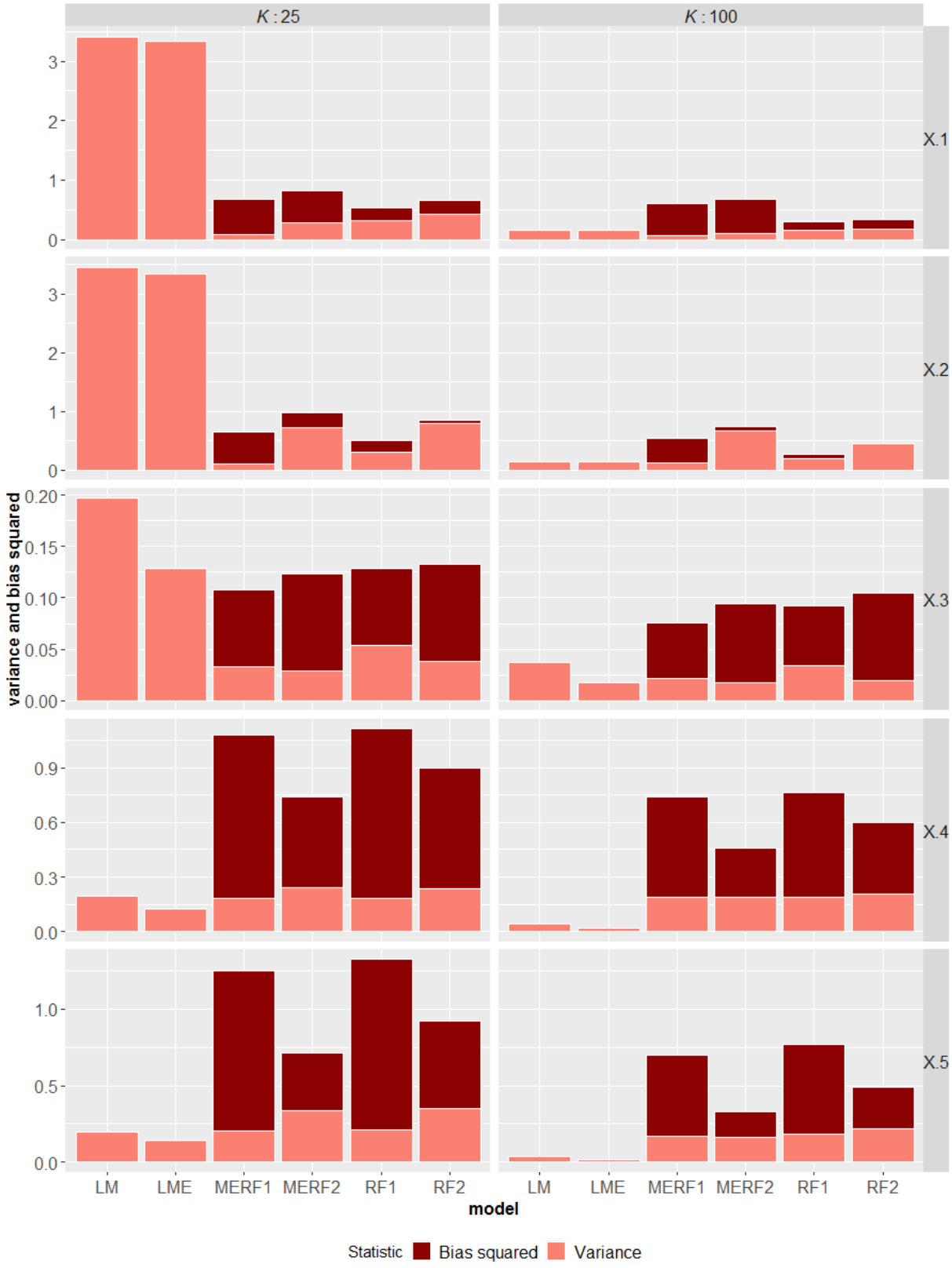

Figure 6.7: Bar plot of the fixed effect estimates ($\alpha_{j,1}$) for the number of groups factor.

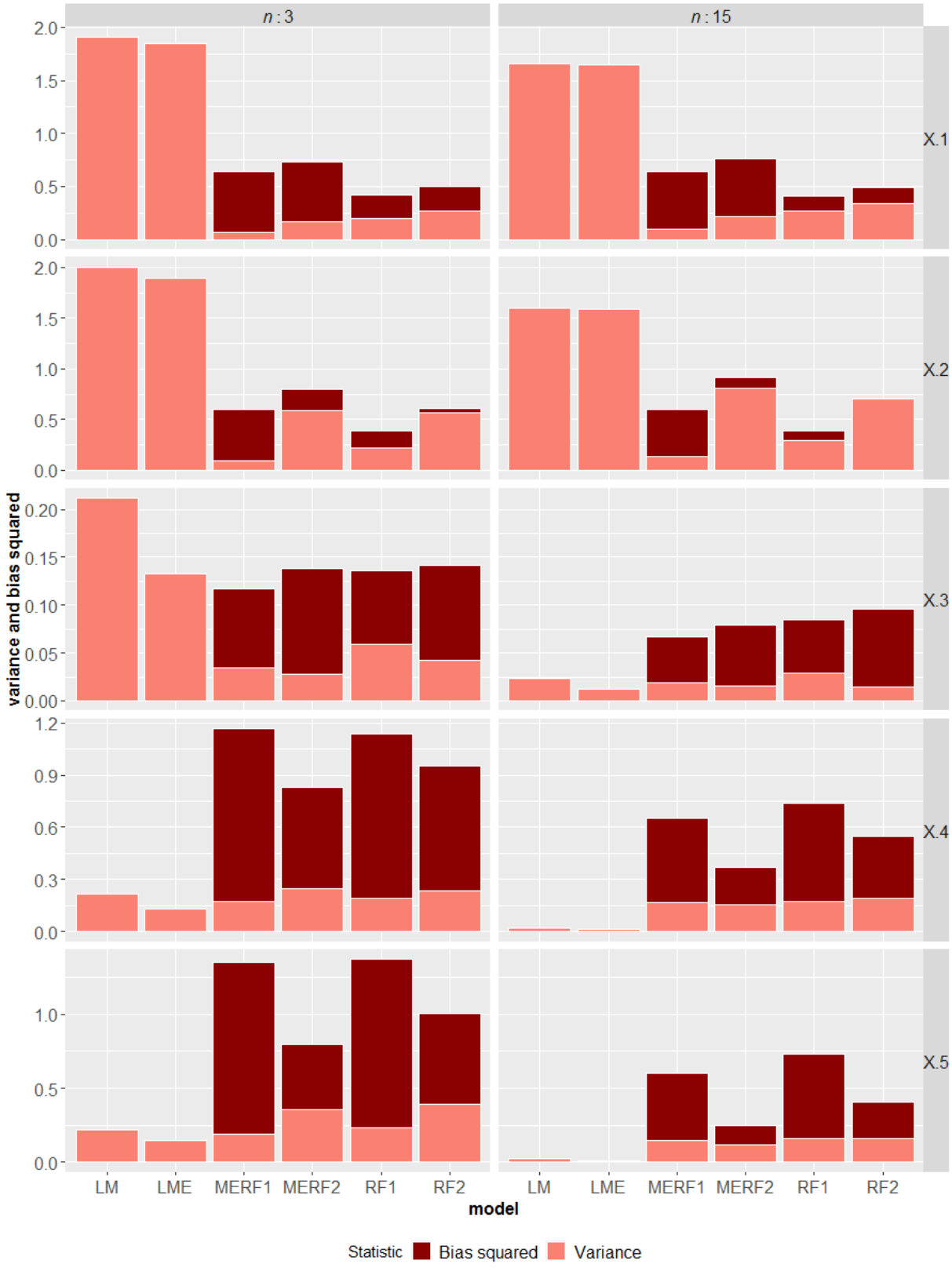

Figure 6.8: Bar plot of the fixed effect estimates ($\alpha_{j,1}$) for the group size factor.

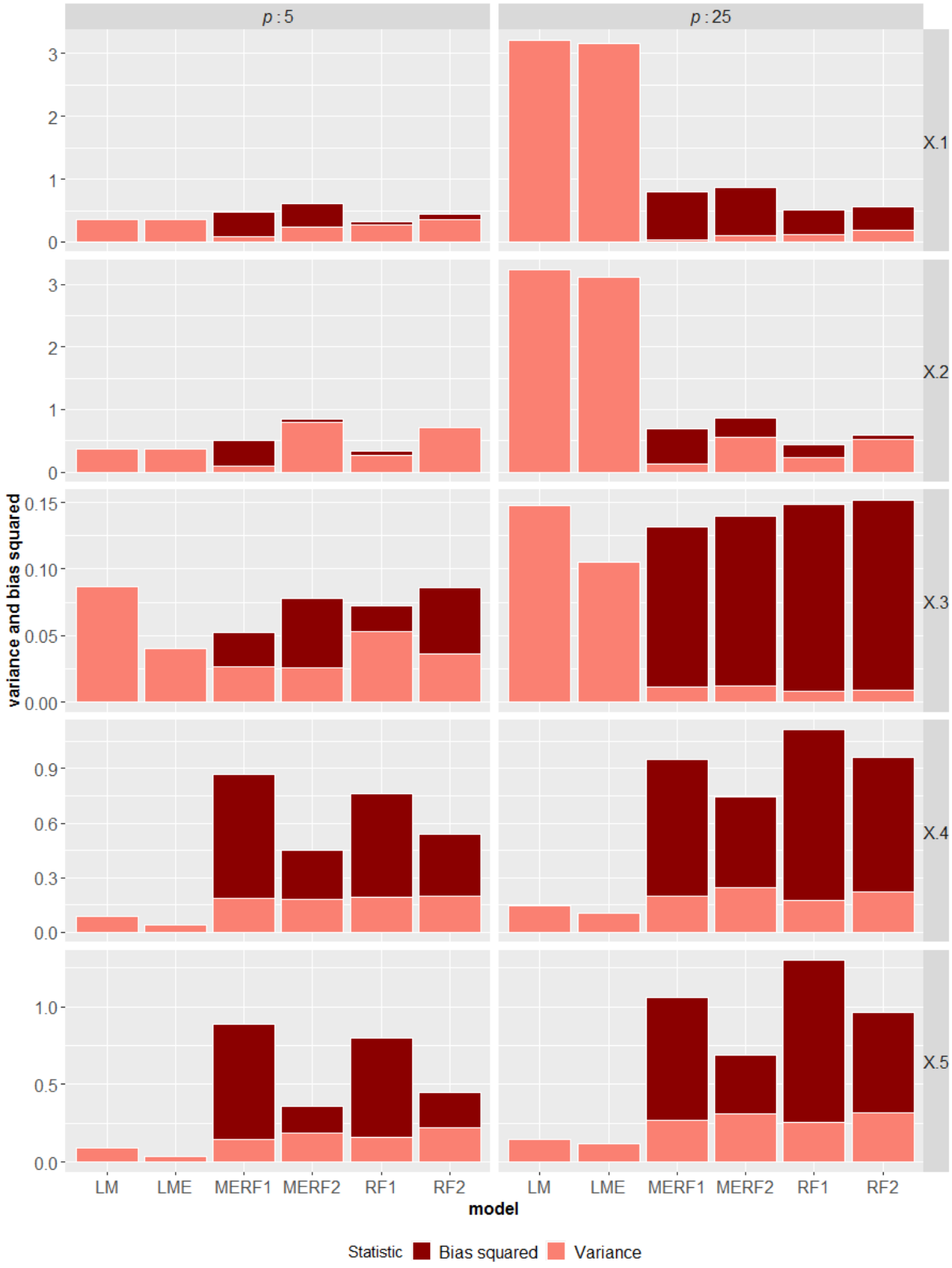

Figure 6.9: Bar plot of the fixed effect estimates ($\alpha_{j,1}$) for the number of predictors factor.

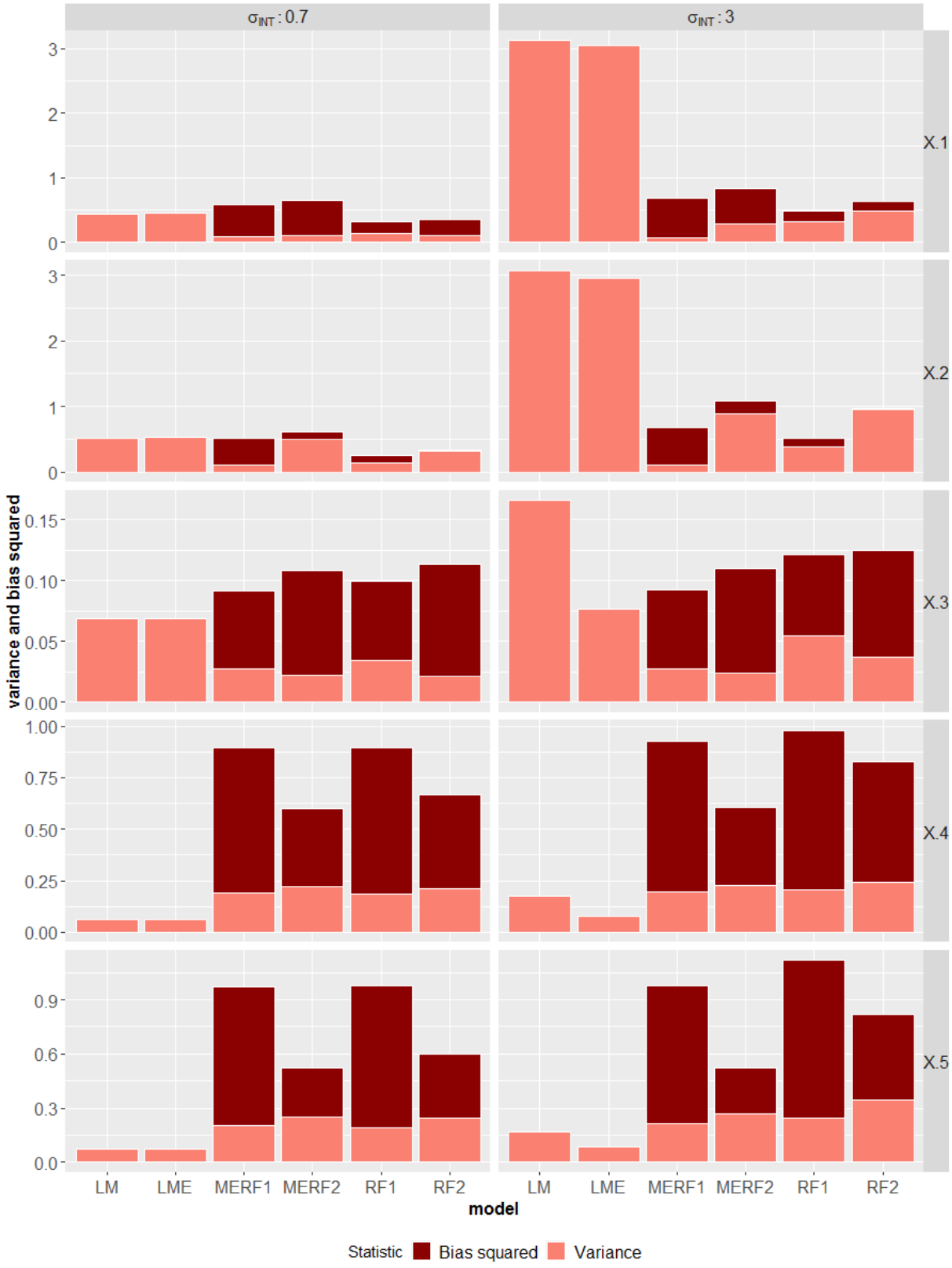

Figure 6.10: Bar plot of the fixed effect ($\alpha_{j,1}$) estimates for the random intercept size factor.

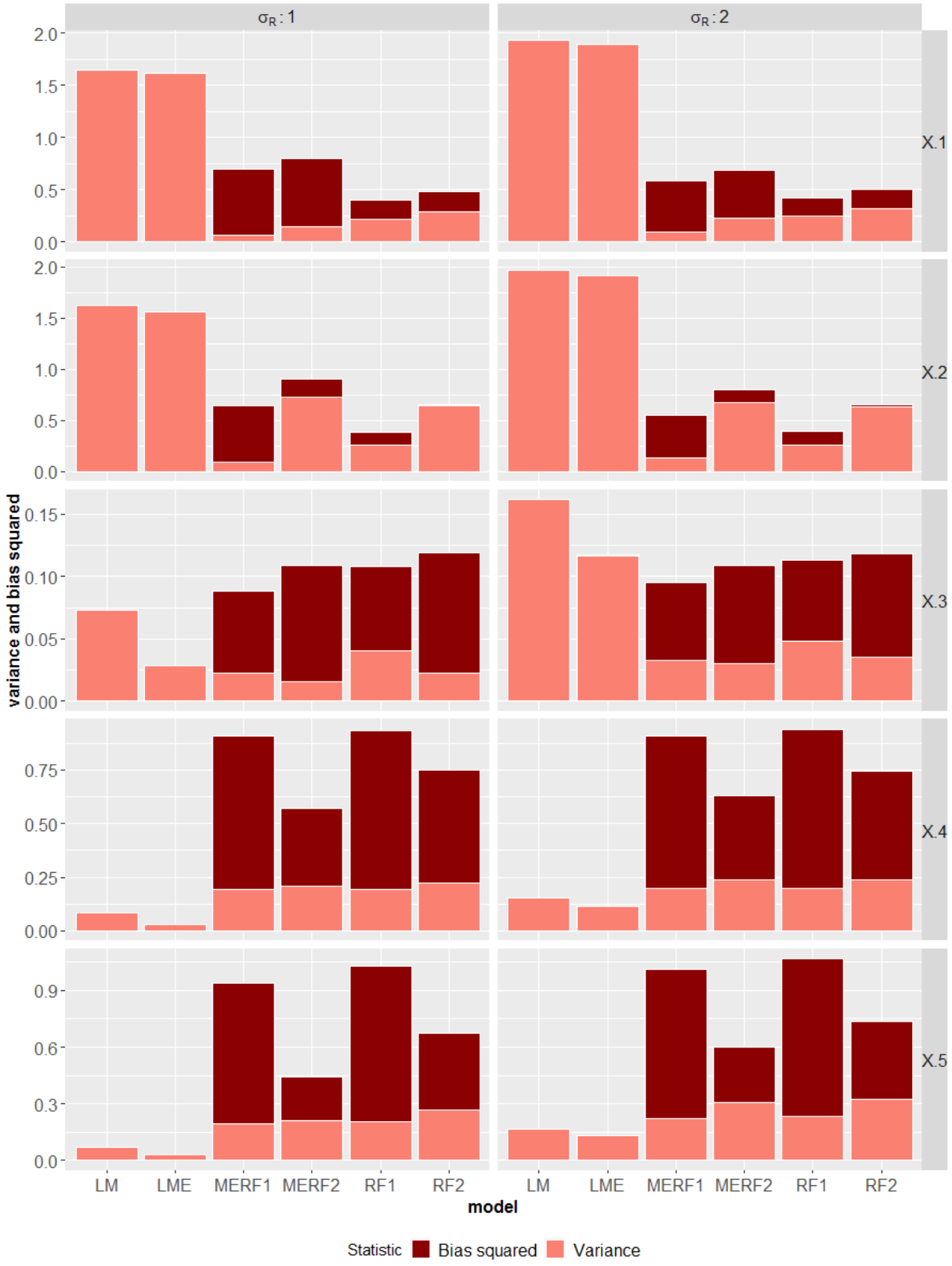

Figure 6.11: Bar plot of the fixed effect estimates ($\alpha_{j,1}$) for the residual size factor.

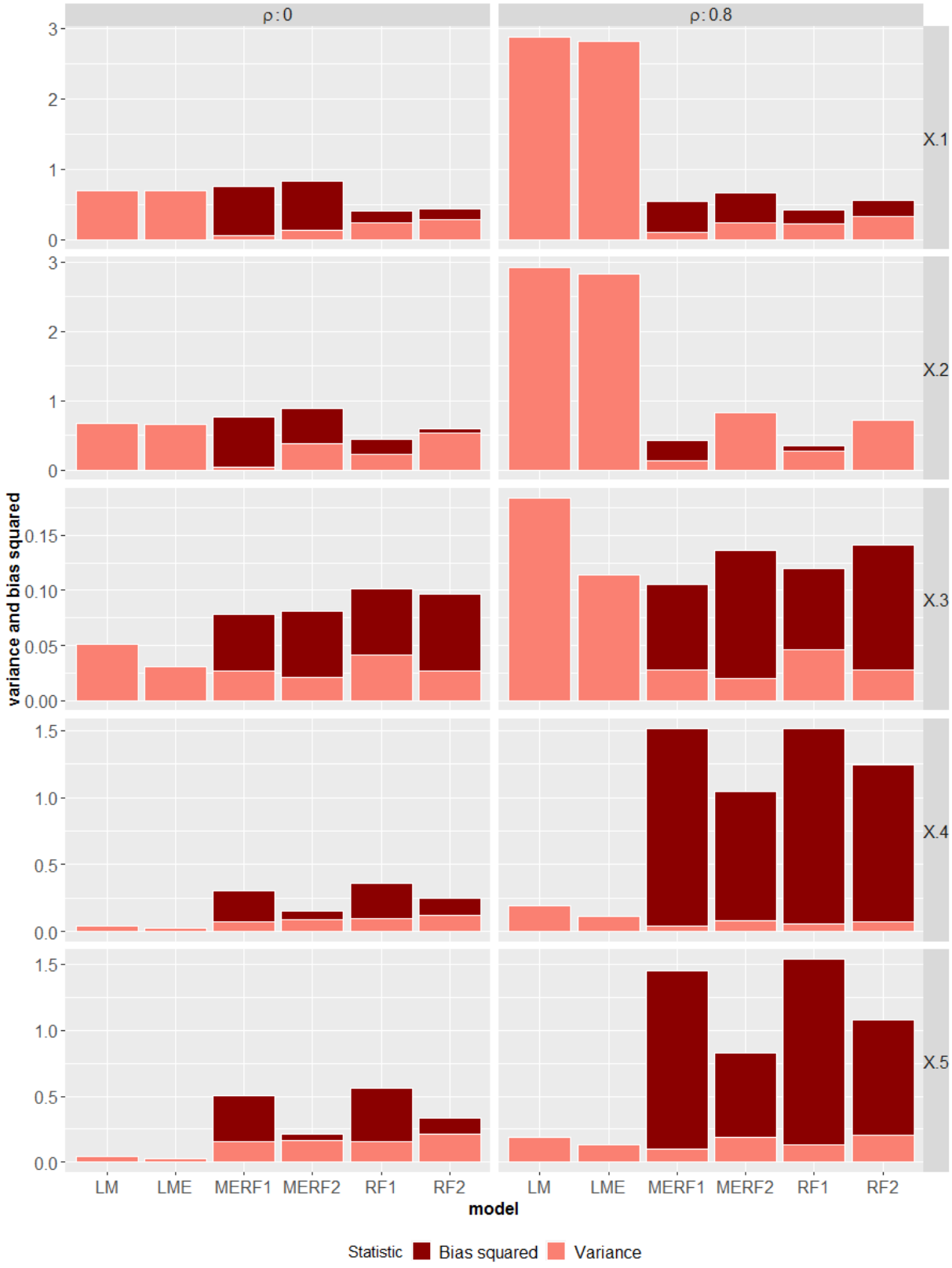

Figure 6.12: Bar plot of the fixed effect estimates ($\alpha_{j,1}$) for correlation factor.

6.2.2 Non-polynomial response

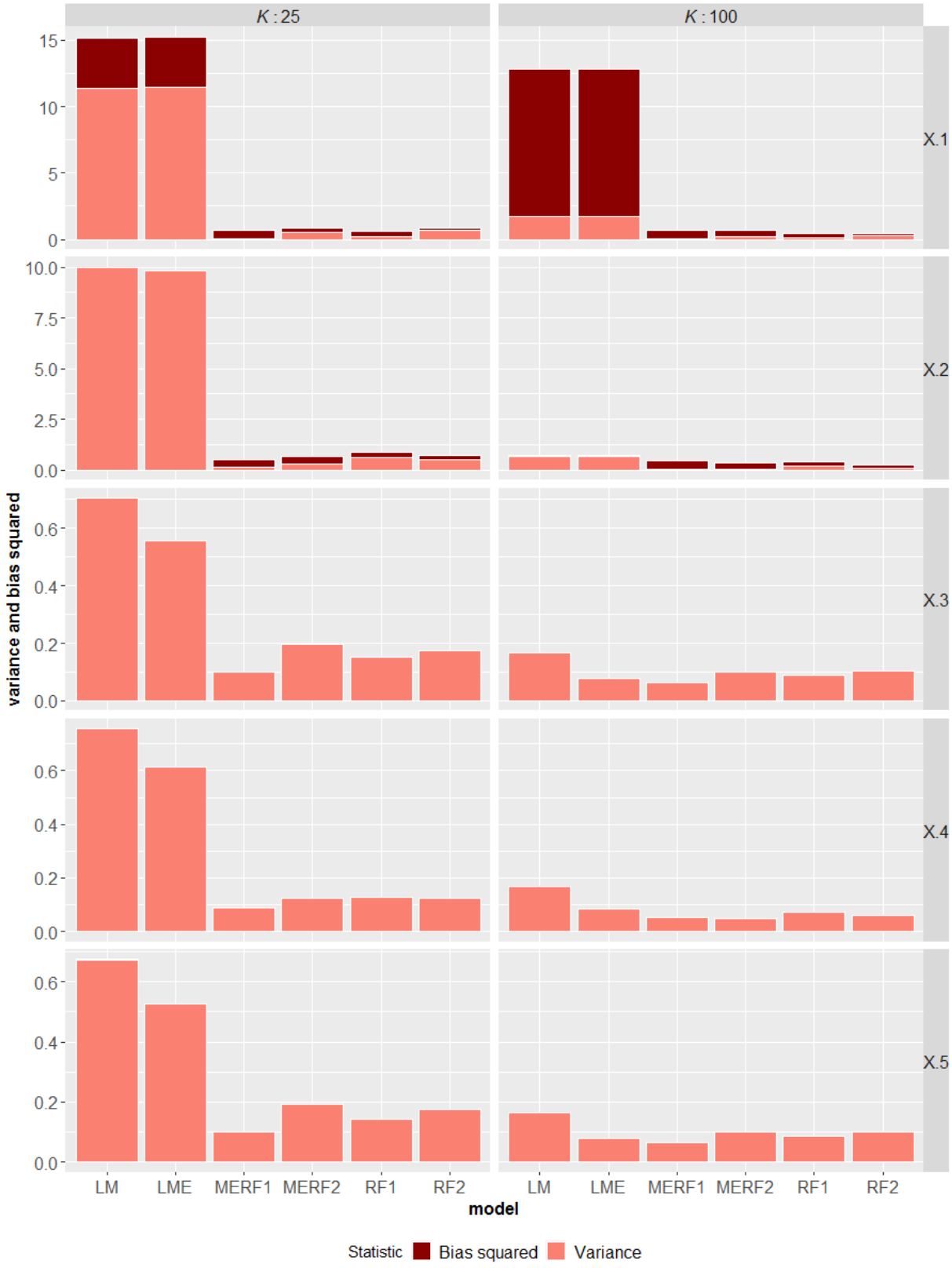

Figure 6.13: Bar plot of the fixed effect estimates ($\alpha_{j,1}$) for the number of groups factor.

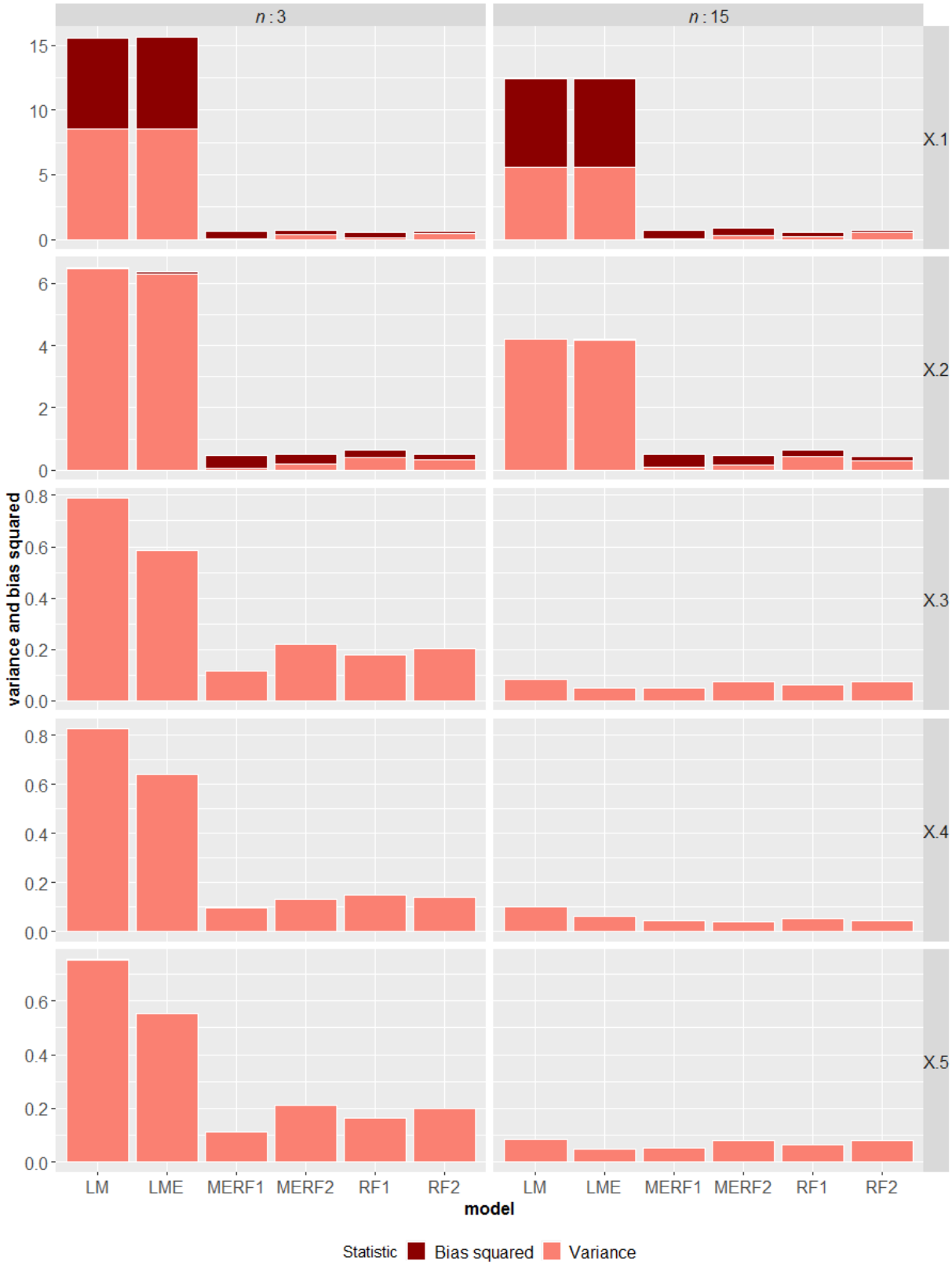

Figure 6.14: Bar plot of the fixed effect estimates ($\alpha_{j,1}$) for the group size factor.

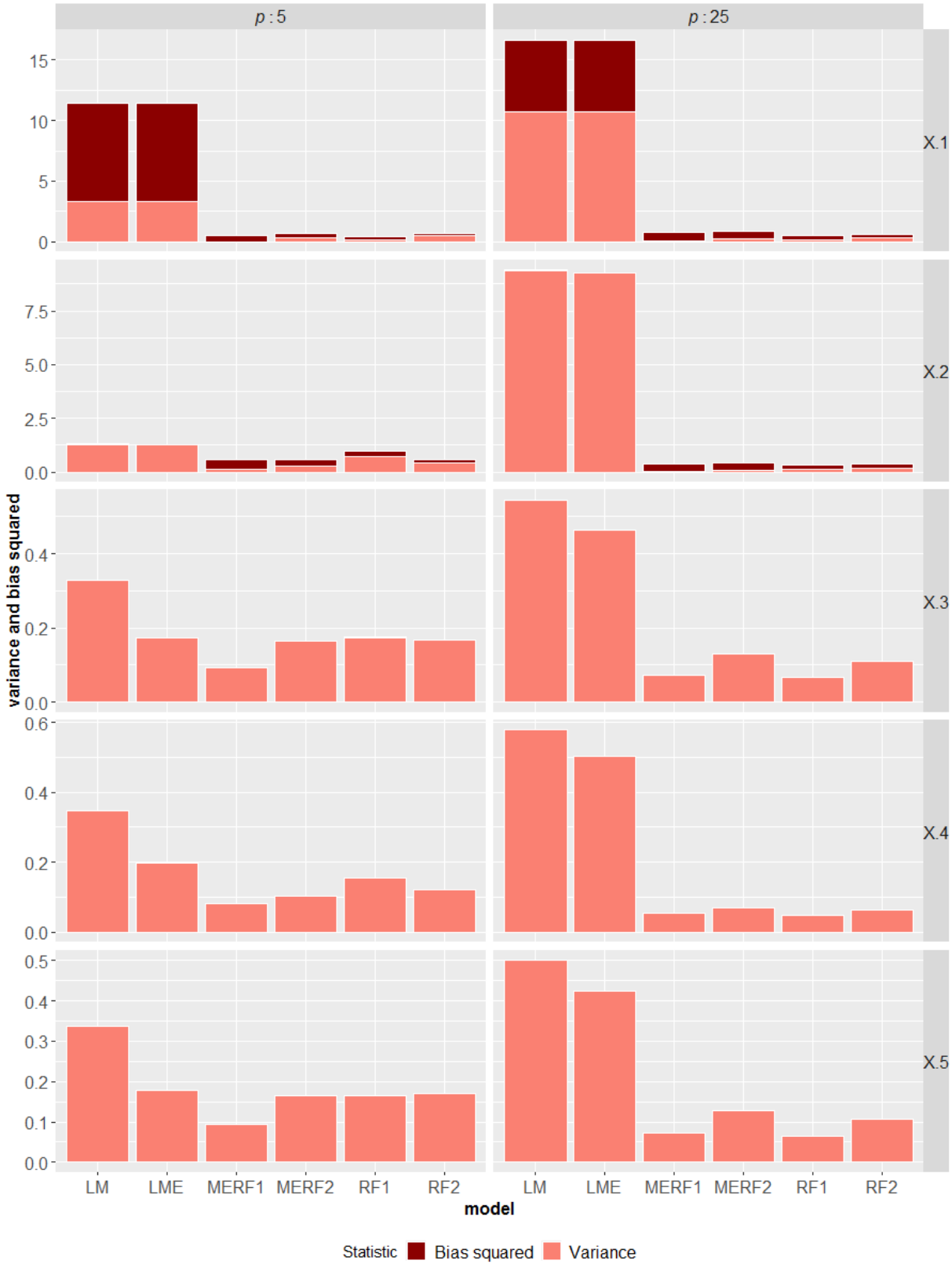

Figure 6.15: Bar plot of the fixed effect estimates ($\alpha_{j,1}$) for the number of predictors factor.

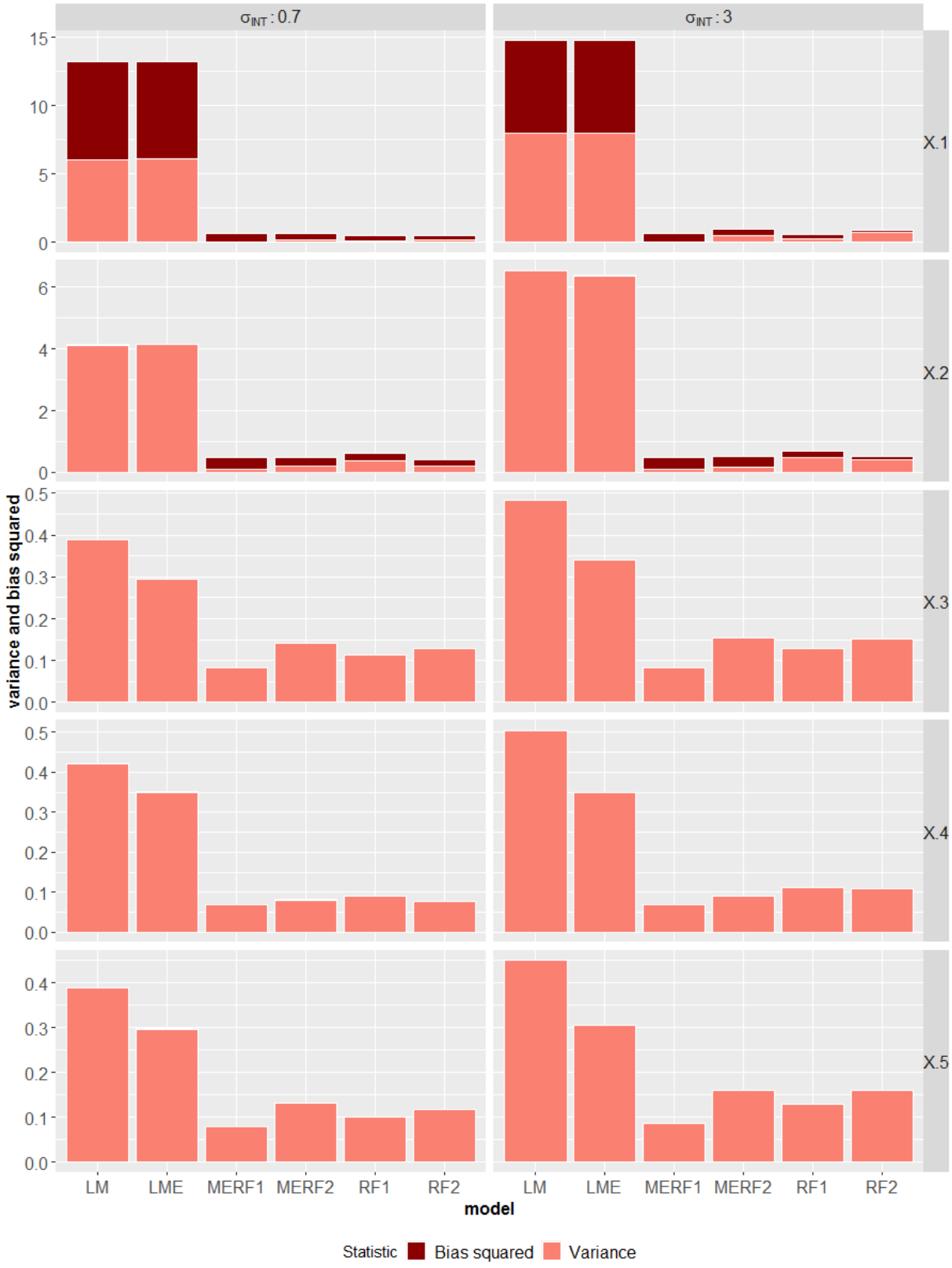

Figure 6.16: Bar plot of the fixed effect estimates ($\alpha_{j,1}$) for the random intercept size factor.

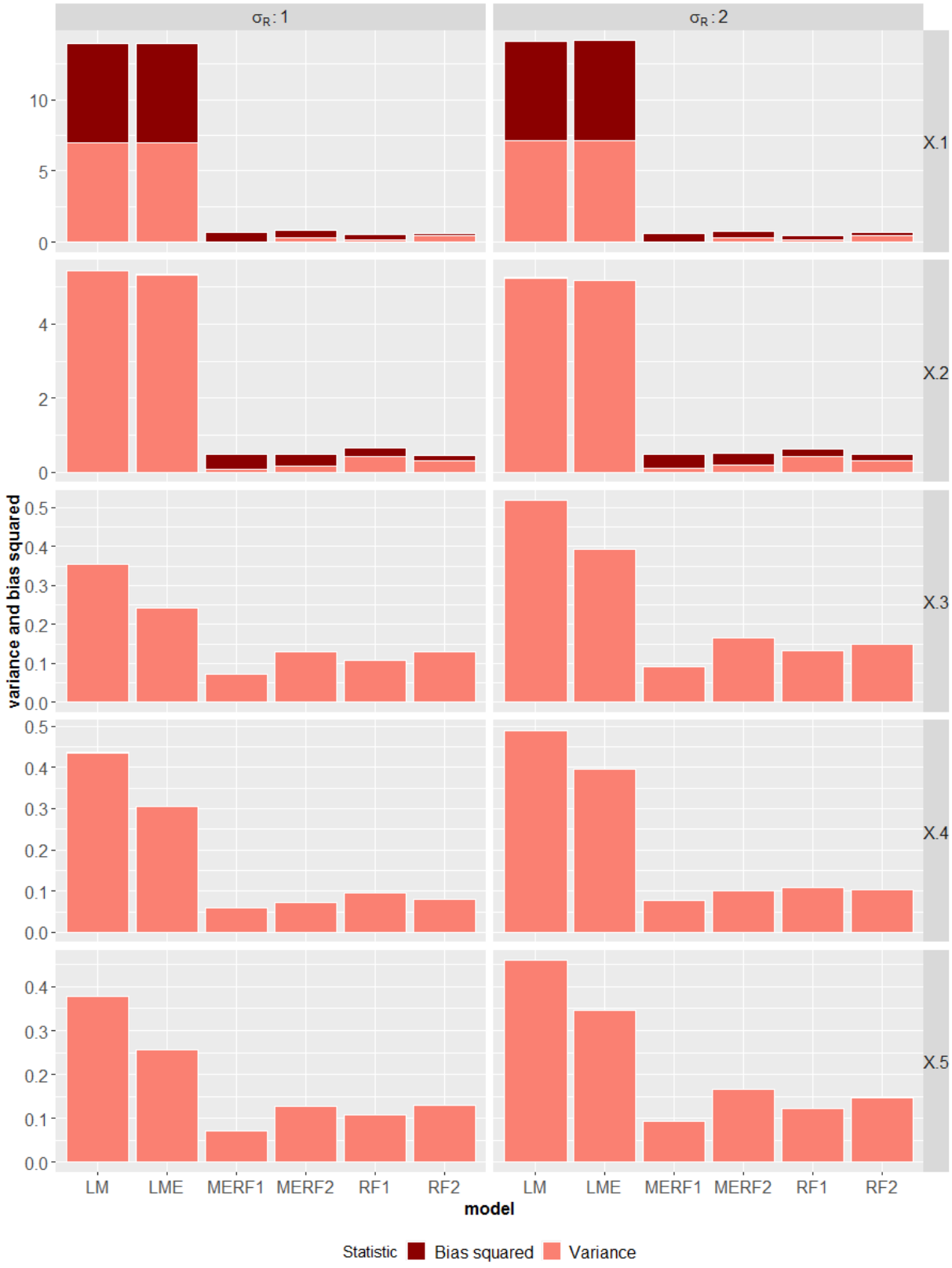

Figure 6.17: Bar plot of the fixed effect estimates ($\alpha_{j,1}$) for the residual size factor.

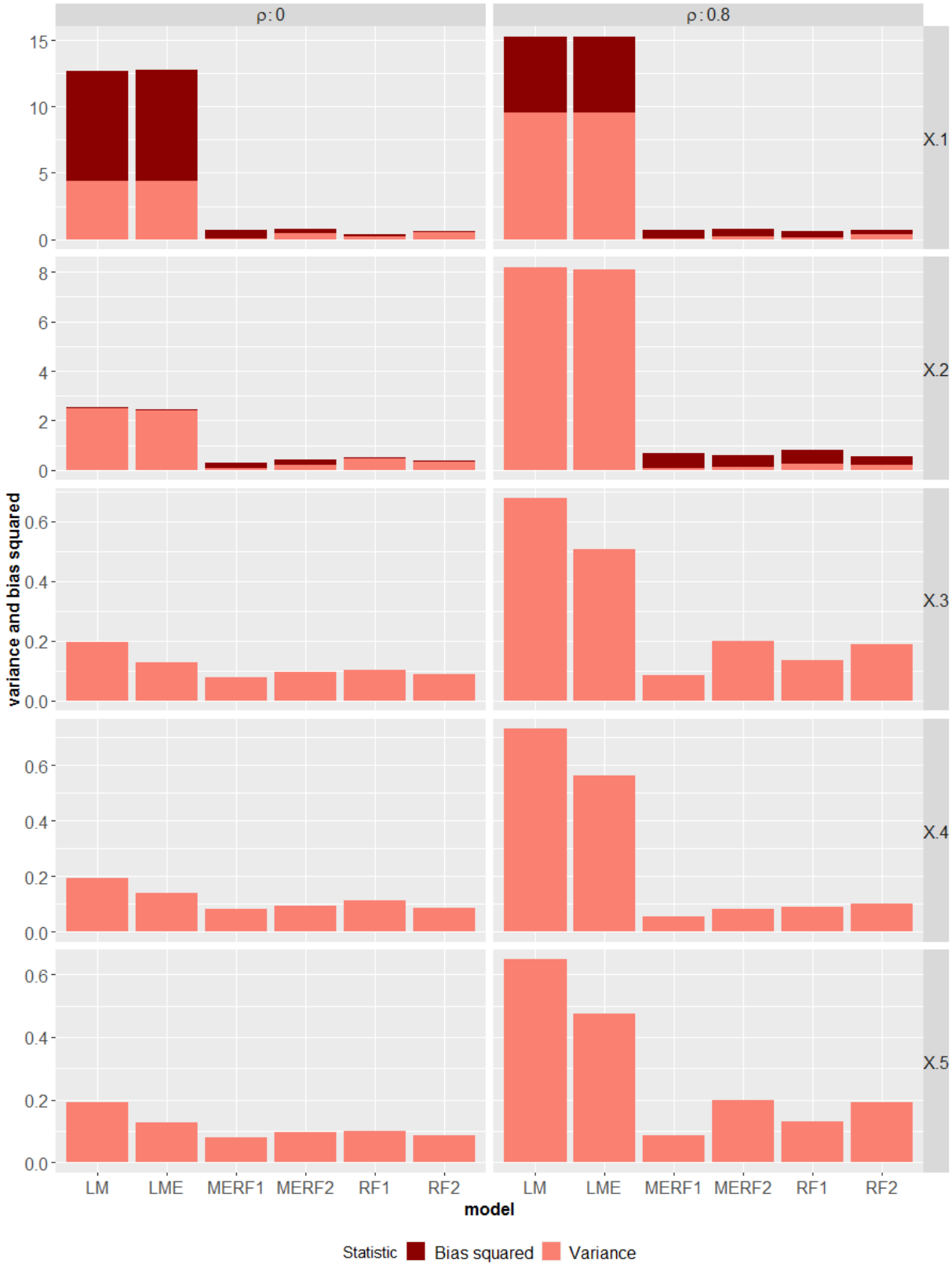

Figure 6.18: Bar plot of the fixed effect estimates ($\alpha_{j,1}$) for correlation factor.

6.3 Random effect plots

Somewhat similar as for the fixed effect plots, we make plots of average estimates of the random effect parameters for all factors, but only in terms of the bias. These plots contain the scaled bias of random effects, that is the estimate divided by the true parameter value. The plots also contain 95% Wald confidence intervals.

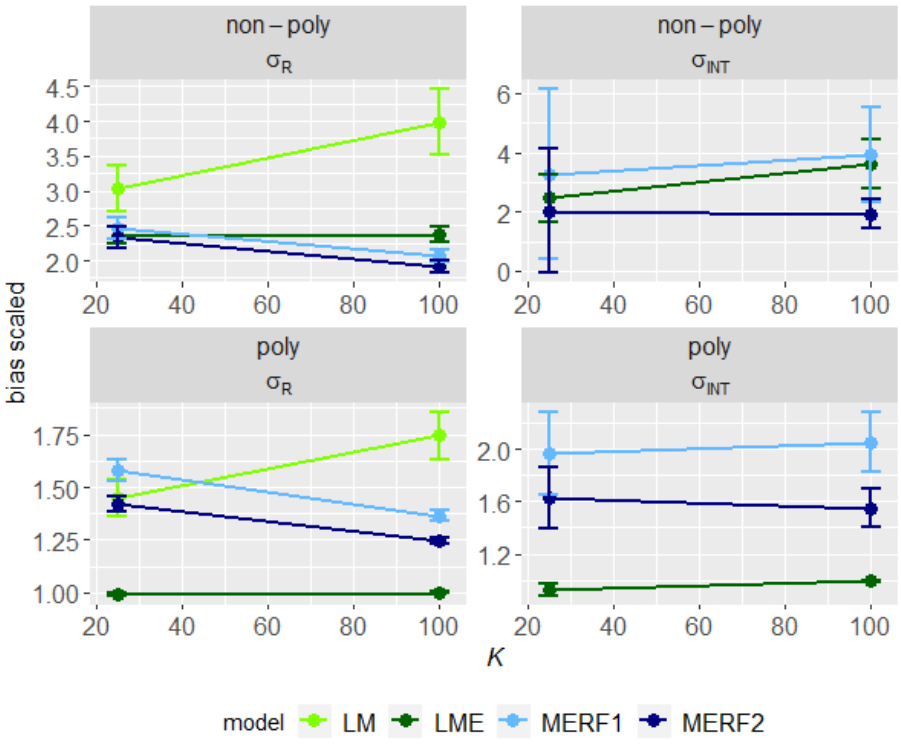

Figure 6.19: Random effect plots for the number of groups factor

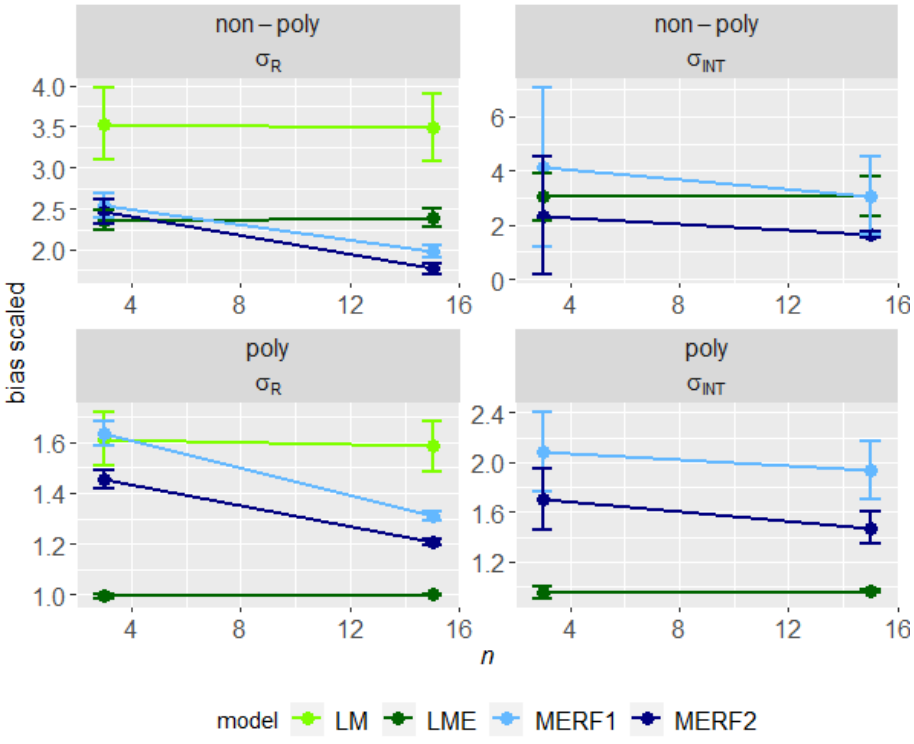

Figure 6.20: Random effect plots for group size factor

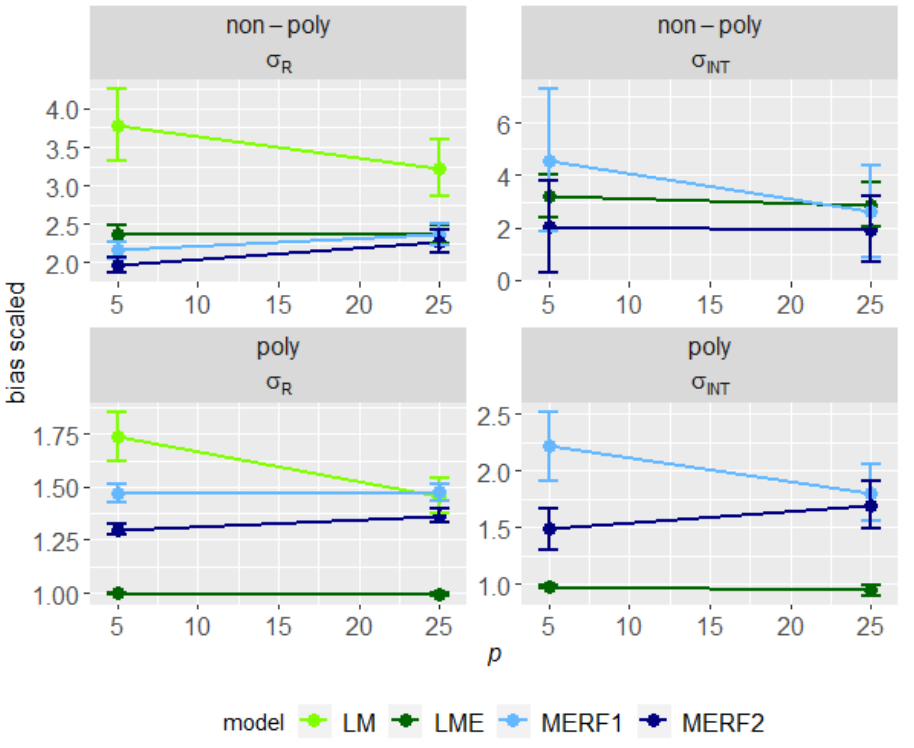

Figure 6.21: Random effect plots for number of predictors factor

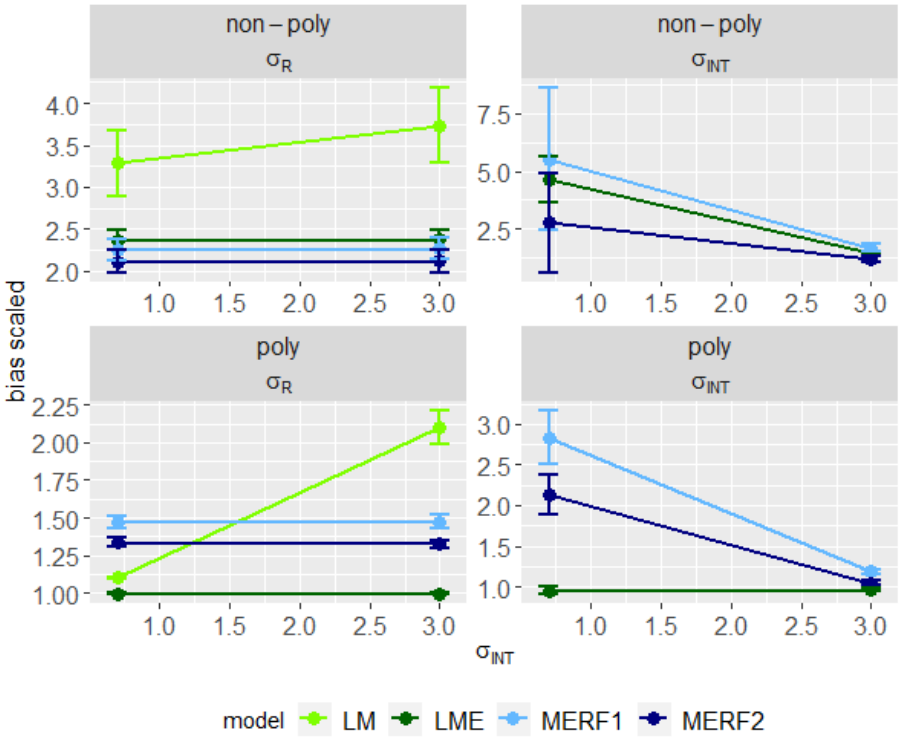

Figure 6.22: Random effect plots for random intercept size factor

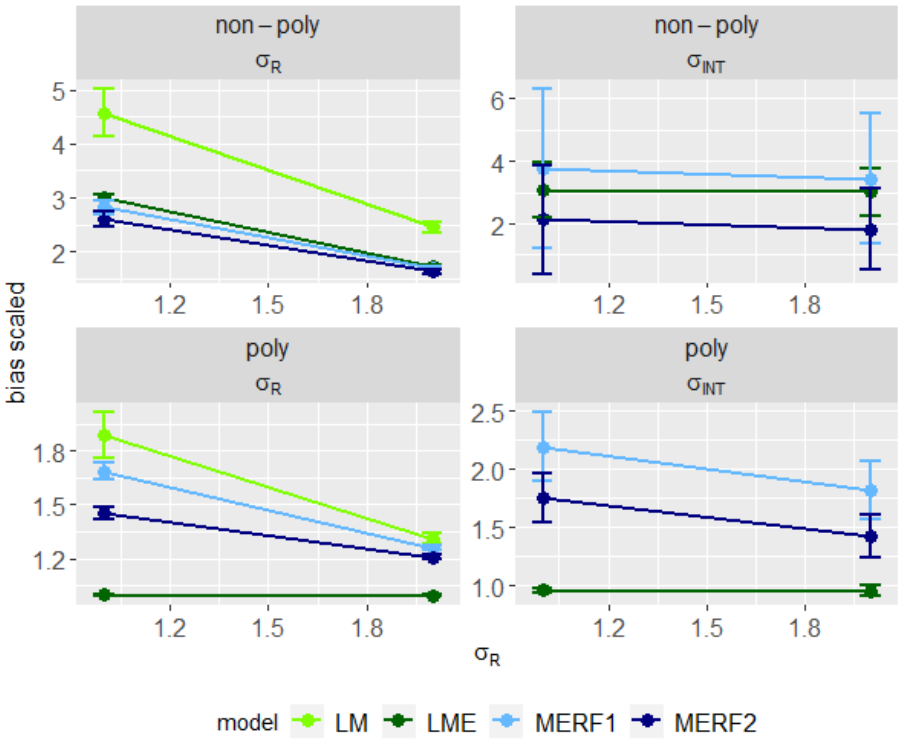

Figure 6.23: Random effect plots for residual size factor

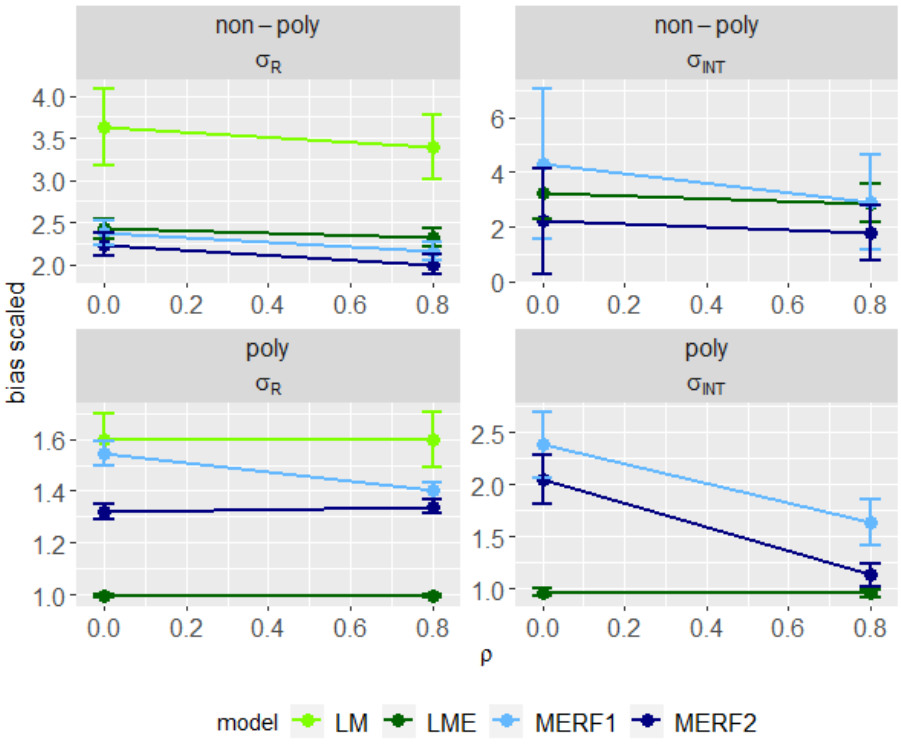

Figure 6.24: Random effect plots for correlation factor.

6.4 Goodness of fit LME

For the non-polynomial response formula we found that the R^2 for the linear mixed model (LME) varies between 0.68 and 0.99 among the 64 scenarios with a mean of 0.87. For the scenarios where there are 5 variables ($p = 5$) the average R^2 is 0.78 whereas for the scenarios where $p = 25$ the average is 0.96.
When the LME is fitted with adding third order terms (see 5.5), we can only simulate the case where the number of groups (K) equals 100, otherwise the model is not properly defined since there would be more coefficients to be estimated than that are observations. The average R^2 over these 32 scenarios is 0.88, which is the same for those 32 scenarios when fitted with the smaller model. The 32 R^2's vary between 0.74 and 0.99.

Chapter 7

Conclusion and Discussion

7.1 Summary

We have seen that several machine learning models exist that try to incorporate random effects for clustered data, and we took a closer look at mixed effects random forest. The applications and intentions of such models were focused solely on prediction. With the simulation study we tried to investigate whether such models can also be used for explaining relations in the data, and compare that to the performance of more 'conventional' statistical methods such as linear regression and linear mixed models. First we give a summary of the key findings of studying the results, which is followed by conclusions and a discussion.

7.1.1 Fixed effect estimation

For the polynomial response model f_p we can see that MERF model is inferior to the linear model and linear mixed model in terms of fixed effect estimation. The bias is often large, especially for between group variables, and for these effects MERF clearly not an improvement on regular random forest models. For within group variables MERF performs much better, but in terms of bias still significantly worse than the linear models. However, for estimating these within group variable effects MERF seems to be a significant improvement on random forest. This does not only hold for scenarios where the random effects are large, but to a lesser extent also for scenarios where the random intercepts on group level are smaller in size. f_p is a scenario where the linear models are typically designed for, so it is no surprise that these models outperform random forest, which prime goal is not to quantify fixed effects. All in all, it seems that when there is a sufficient amount of data (1500 observations in our simulation design) and variables are not correlated, MERF can estimate within group variable effects reasonably well. In all other scenarios MERF is not a suitable alternative to traditional statistical models.

For the non-polynomial response model f_n the comparison is different for the obvious reason that the linear models' formulas do not contain the actual input effects. However, what of course does not change is the MERF's inability for picking up on between group variable effects. And in fact, for this type of fixed estimation a linear model is still better,

even if the model formula has not been correctly specified.
On the contrary, for within group variables, especially the one with a step function ($\boldsymbol{x}_4$), RF and MERF have a much lower bias than linear models. This is because of the nature of regression trees, which basically consists of a lot of cutoff function. Residual size and random effect size have very little influence on fixed effect estimation for RF and MERF, but that is not the case for predictor correlations ρ. An increasing correlation yields a clear increasing bias for fixed effects estimates for MERF. The variance however is not affected by this. So if the data is not too much correlated, and it is hard to identify a reasonable model input for linear regression, MERF can be a good alternative, since it is able to identify fixed effects as long as there is sufficient amount of data.

7.1.2 Random effect estimation

Again for the polynomial response, the linear mixed model yields unbiased estimates for σ_{INT}. When the random effect is relatively large MERF is also able to estimate the random effect quite well, with an average bias of 1.05, i.e. it overestimates the bias by about 5%. On the other hand, if the random effect size is small and the number of observations is low, MERF tends to overestimate its size by a big margin, around 3 times as large on average. Another notable results is that introducing correlation between the variables decreases the bias of σ_{INT}. This holds particularly for the polynomial response formula (Figure 6.26).

Furthermore we see that increasing the size the random intercept has no influence on the bias of the residual size σ_{INT} for the MERF models (figure 6.22). However, the linear regression model sees here a sharp increase in its bias.
Making the residual size bigger (i.e. increasing σ_{R}) yields a lower bias of σ_{R} for all models, and lower bias of σ_{INT} for the MERF models, but not for the linear mixed model.

7.1.3 Influence of hyperparameters

When we compare the performance of the two mixed-effects random forest fits MERF1 and MERF2 we see that they generally behave quite similar, in the sense that they show the same trends when simulation settings are changed. However, it is also quite clear that there are in some cases large difference between the estimated bias and variances of both models: when it comes to the polynomial response formula, MERF1 seems to perform more or less the same as MERF2 for the fixed effects of the between-group variables in terms of MSE, but a bit worse for the within-group variables. This holds for both $\alpha_{j,1}$ and $\alpha_{j,2}$. Although MERF2 has consistently a higher variance for its estimates than MERF1, the increase in bias reduction seems often large enough to overcome this and eventually result in a lower MSE. For the non-polynomial response model we see a similar pattern when it comes to the fixed effects, although it here MERF1 produces slightly lower mean-squared errors for the between-group variables.

When it comes to the random-effects we see that MERF2 gives consistently lower average

bias across all simulation factors. One remarkable results is displayed in figure 6.21, where we see that the bias of random effects estimates of MERF2 increases when the number of variables increases, but those of MERF1 decreases. For this behaviour of MERF1 there seems a rather obvious explanation: in combination with the standard setting of the *mtry* parameter this model produces quite bad results since it makes its cut-offs based on only one predictor variable for $p = 5$. However when $p = 25$ the cut-offs are much more likely to be based on a predictor that can reduce the mean squared error by a bigger margin. For MERF2's behaviour a possible explanation is overfitting: in case $p = 25$ it becomes more likely that a tree will overfit its bootstrapped training data since every split considers 20 prediction variables, and therefore the out-of-bag estimates have a larger mean squared error, which will eventually lead to an overestimation of the random effects.
We can conclude that hyperparameter settings can have significant influence on the performance of mixed-effects random forest, which we already knew was the case for regular random forest model.

7.2 Conclusion

Based on our simulation study random forest enhanced with random effect estimation is generally not a reliable explorative model when it comes to effect estimation. The bias of MERF estimators is in many of the simulated cases simply too high. If a linear mixed model's fitted formula is reasonable close to the true model, the results of the MERF models are vastly inferior to the linear model. So in cases where a response variable can be reasonably well approached by a relatively simple model formula (i.e. a polynomial model), the MERF model is not a suitable alternative. However, when the true model is overly complex, the MERF model has potential to give a sense of the right direction, but for real parameter estimation, especially for marginal fixed effects, the random forest model seems not adequate.

7.3 Discussion

Although the results of the mixed-effects random forest models were rather disappointing, especially considering the great potential they had shown in previous studies when it comes to making predictions, it still gives some new knowledge about the strengths and limitations of such machine learning models.
At the same time we also have to acknowledge the limitations of the study. For example, instead of a random forest, any machine learning model could be used to make a mixed-effects machine learning model. However, since previous papers that have been dealing with mixed-effects machine learning models have shown that MERF is one of the most powerful in prediction settings, it does not seem likely that simply replacing the random forest in MERF with another model would necessarily improve parameter estimations significantly. Possible alternative model are neural networks and support vector machines.
Furthermore the MERF model as discussed in this thesis is simply one of the possible manners to incorporate machine learning with random effect estimation. So there might be

ways to improve this mixed-effects random forest. A possible disadvantage of the MERF algorithm is the way it performs the bootstrapping step, which is also mentioned by the authors of the paper [4]. After removing the random effect estimates from the observations, the data structure is ignored in the re-sampling process. This means that from a certain group any number of observations can be drawn. Of course, if the random structure is correctly defined and its estimates are unbiased, the observations are all independent after removing the random effect estimates. Nonetheless, if this estimation has not been done correctly, there still remains some correlation between the observations within one group, and in that case it would make more sense to re-sample entire groups instead of individual observations.

Another limitation is naturally the simulation study itself. Although we covered seven different simulation settings resulting in a total of 128 simulation scenarios, there are still many other data properties that can be found in real world data problems, such as outliers, missing values, and more advanced random structures (we considered only a random intercept). Exploring more data settings would give a more complete overview of strengths and weaknesses of mixed-effects random forest.

Bibliography

[1] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[2] N. E. Breslow and D. G. Clayton. Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association*, 88(421):9, Mar. 1993.

[3] A. Hajjem, F. Bellavance, and D. Larocque. Mixed effects regression trees for clustered data. *Statistics & Probability Letters*, 81(4):451–459, Apr. 2011.

[4] A. Hajjem, F. Bellavance, and D. Larocque. Mixed-effects random forest for clustered data. *Journal of Statistical Computation and Simulation*, 84(6):1313–1328, Nov. 2012.

[5] A. Hajjem, D. Larocque, and F. Bellavance. Generalized mixed effects regression trees. *Statistics & Probability Letters*, 126:114–118, July 2017.

[6] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer New York, 2009.

[7] A. Liaw and M. Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.

[8] Y. Lin and Y. Jeon. Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, 101(474):578–590, June 2006.

[9] M. J. Lindstrom and D. M. Bates. Newton-raphson and EM algorithms for linear mixed-effects models for repeated-measures data. *Journal of the American Statistical Association*, 83(404):1014, Dec. 1988.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[11] P. Ploton, N. Barbier, P. Couteron, C. Antin, N. Ayyappan, N. Balachandran, N. Barathan, J.-F. Bastin, G. Chuyong, G. Dauby, V. Droissart, J.-P. Gastellu-Etchegorry, N. Kamdem, D. Kenfack, M. Libalah, G. Mofack, S. Momo, S. Pargal, P. Petronelli, C. Proisy, M. Réjou-Méchain, B. Sonké, N. Texier, D. Thomas, P. Verley, D. Z. Dongmo, U. Berger, and R. Pélissier. Toward a general tropical forest biomass prediction model from very high resolution optical satellite images. *Remote Sensing of Environment*, 200:140–153, Oct. 2017.

[12] P. Probst, M. N. Wright, and A.-L. Boulesteix. Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(3), Jan. 2019.

[13] G. Verbeke and G. Molenberghs. *Linear Mixed Models in Practice*. Springer New York, 1997.

[14] S. Yan, H. Hosseinmardi, H.-T. Kao, S. Narayanan, K. Lerman, and E. Ferrara. Estimating individualized daily self-reported affect with wearable sensors. In *2019 IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE, June 2019.

[15] X. Yan and X. G. Su. *Linear Regression Analysis*. WORLD SCIENTIFIC, June 2009.

[16] Q. Zhao and T. Hastie. Causal interpretations of black-box models. *Journal of Business & Economic Statistics*, pages 1–10, July 2019.

Appendix A

Figures

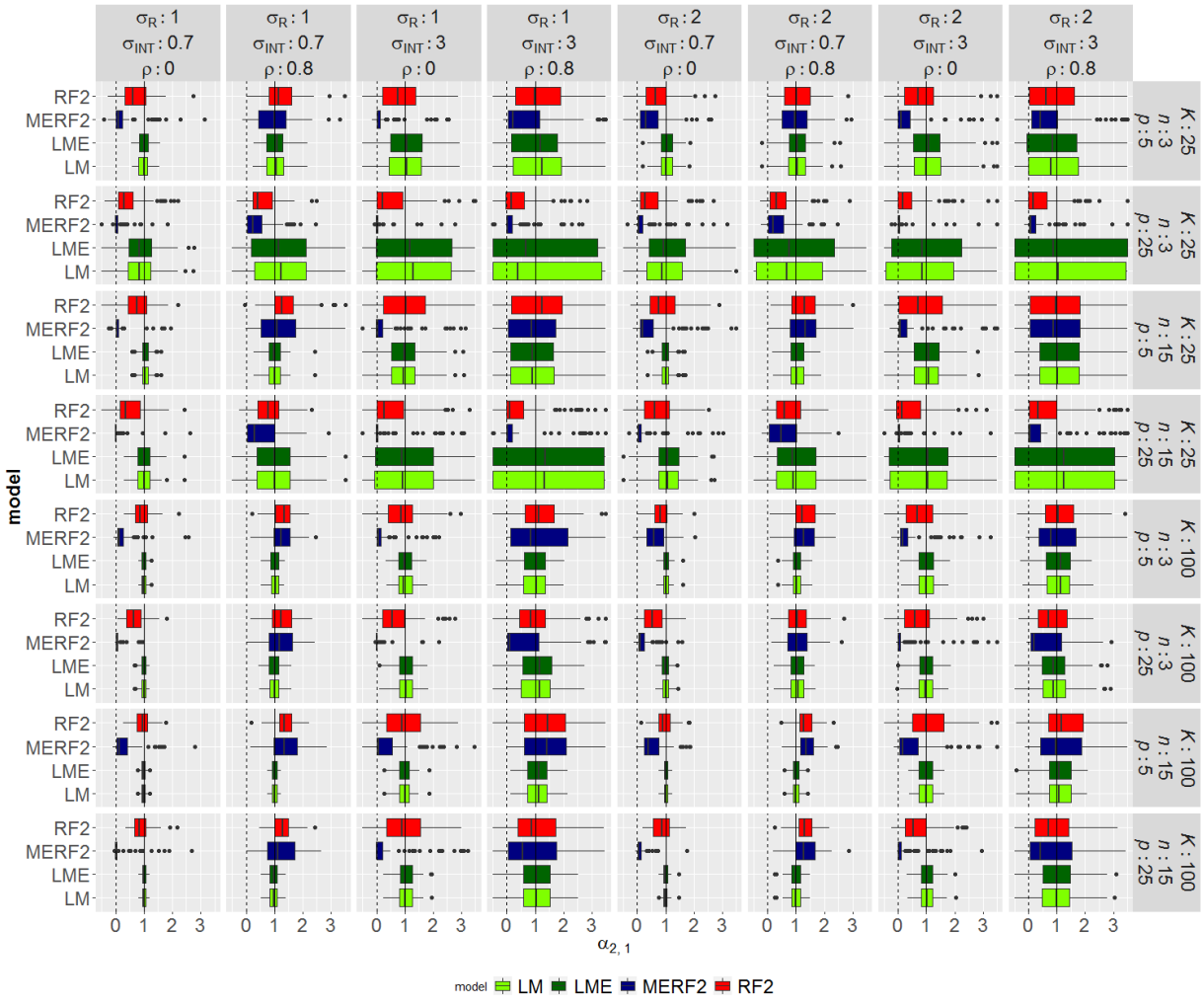

Figure A.1: Fixed effects of x_2, which is a between group variable, evaluated for the polynomial model.

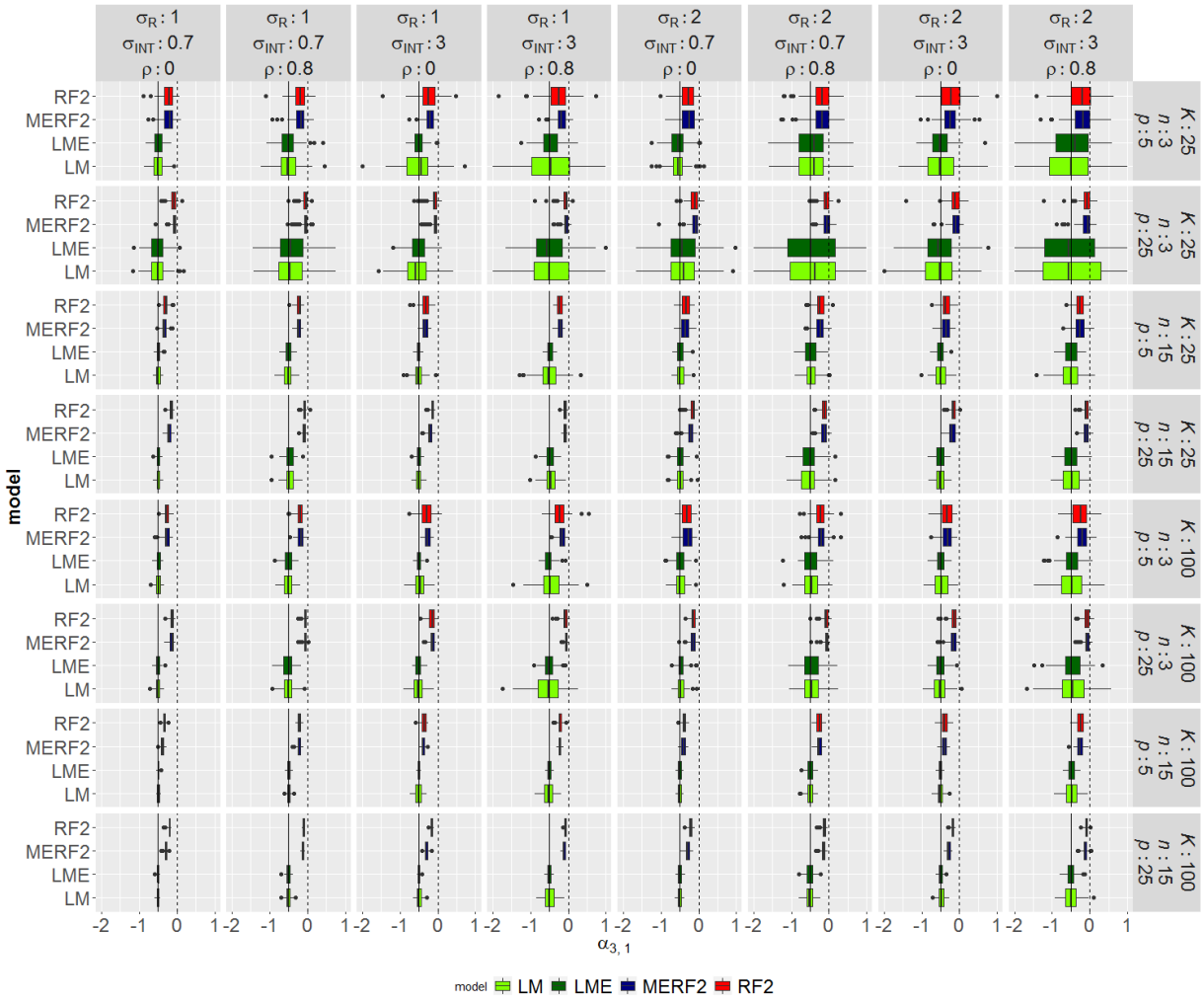

Figure A.2: Fixed effects of x_3, which is a within group variable, evaluated for the polynomial model.

A.0.1 Polynomial response

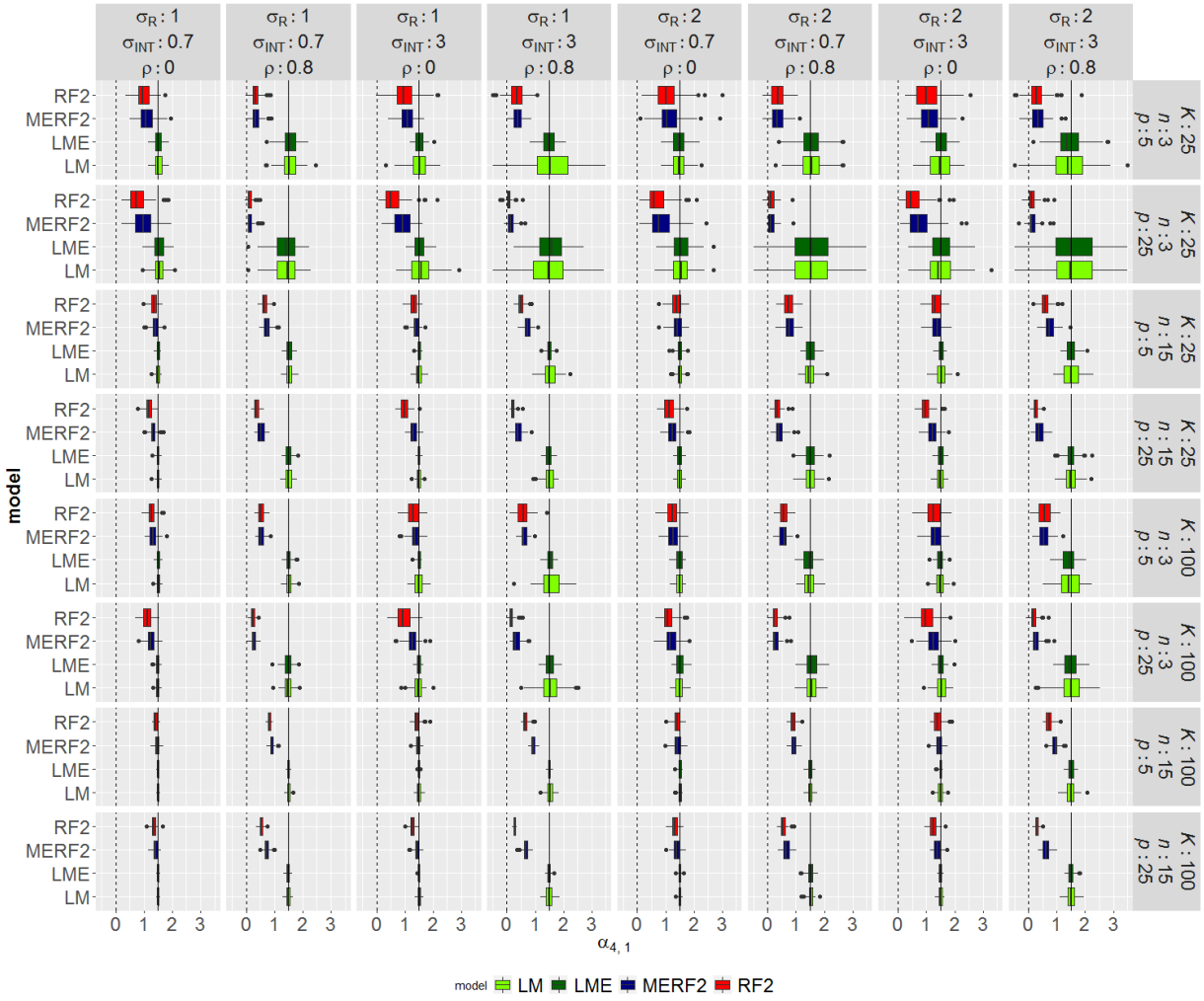

Figure A.3: Fixed effects of x_4, which is a within group variable, evaluated for the polynomial model.

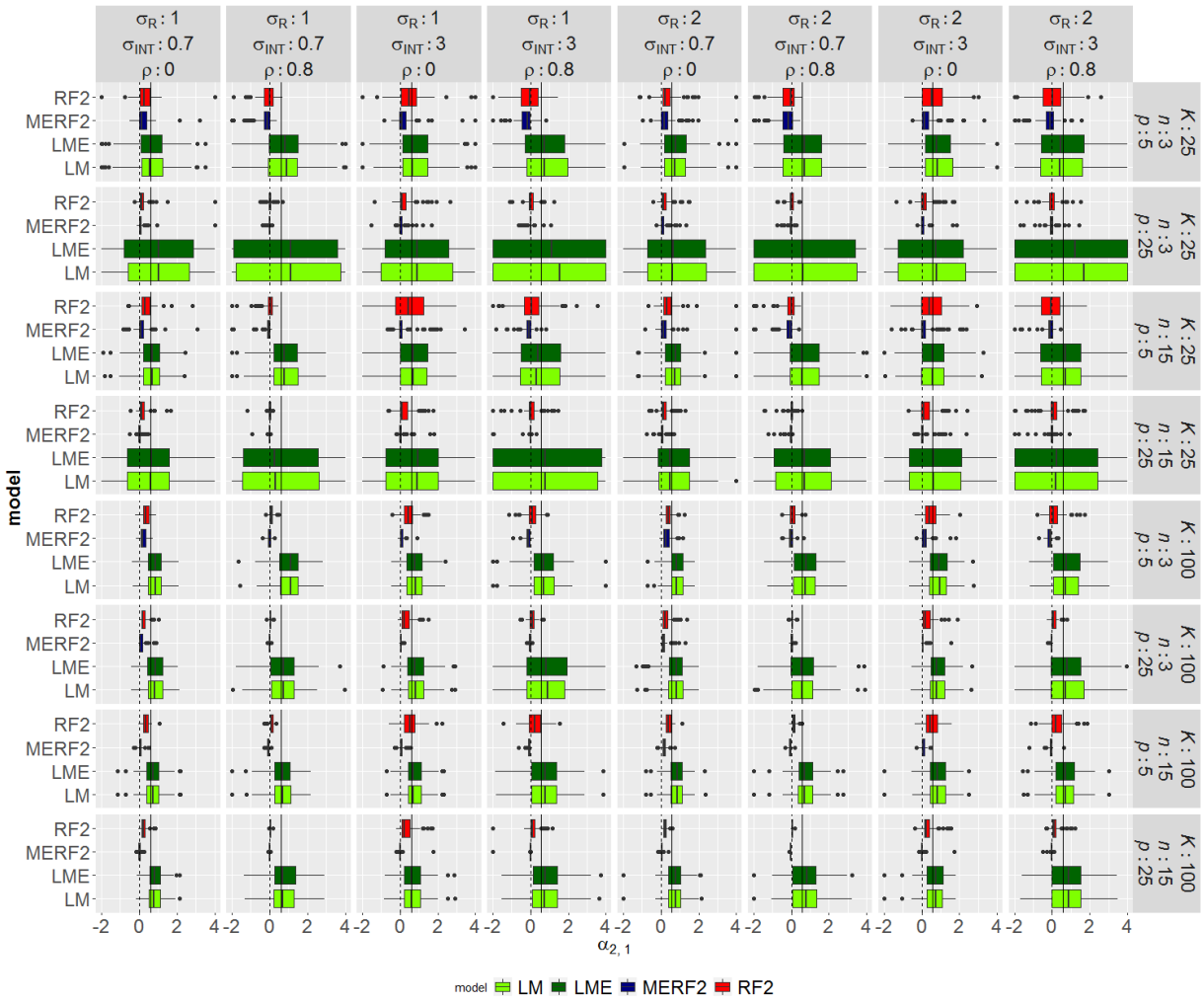

Figure A.4: Fixed effects of x_2, which is a between group variable, evaluated for the non-polynomial model.

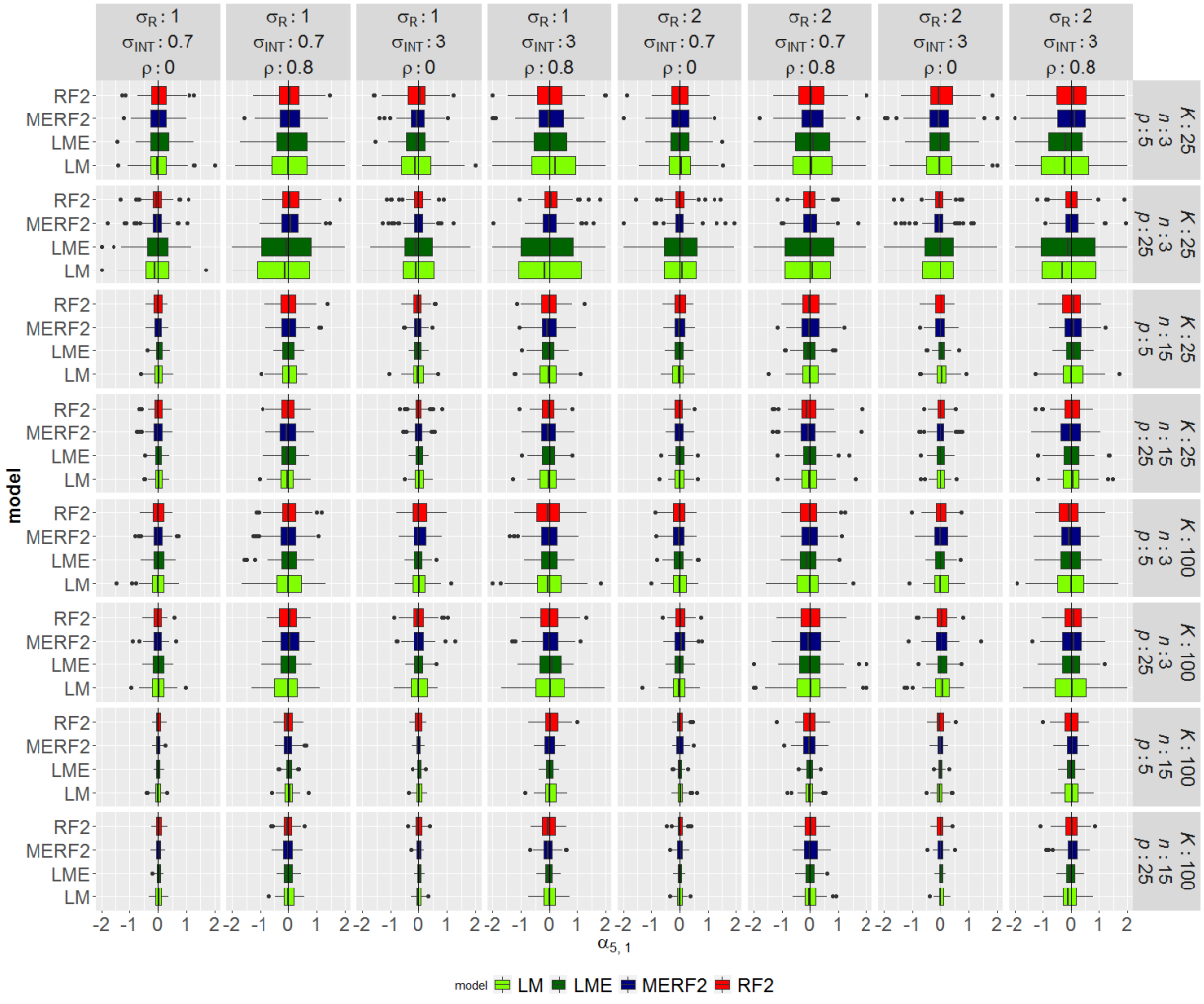

Figure A.5: Fixed effects of x_3, which is a within group variable, evaluated for the non-polynomial model. This plot is very similar to the plot of x_5 ($\alpha_{5,1}$), and therefore plot is omitted.

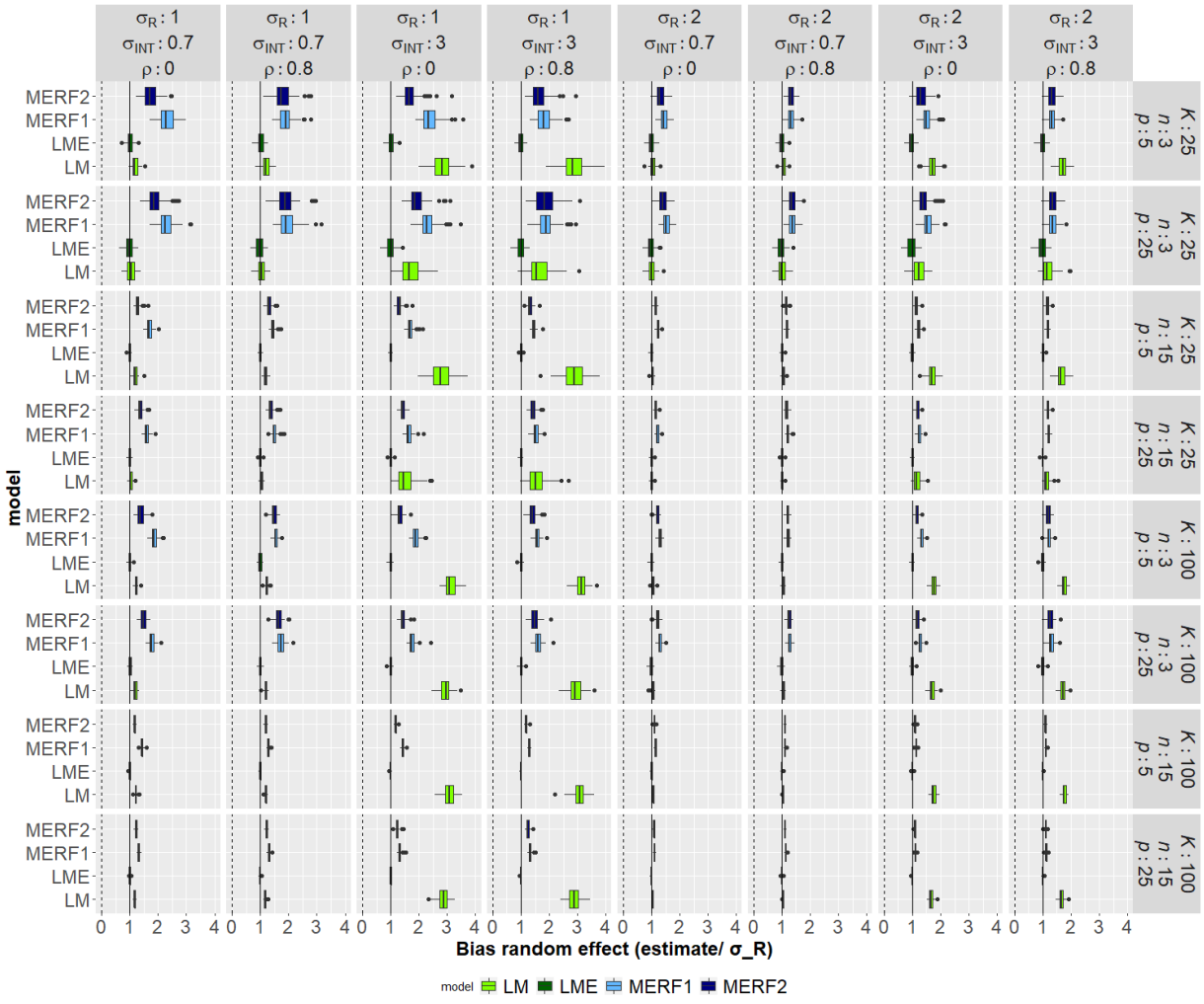

Figure A.6: σ_R, evaluated for the polynomial response model.

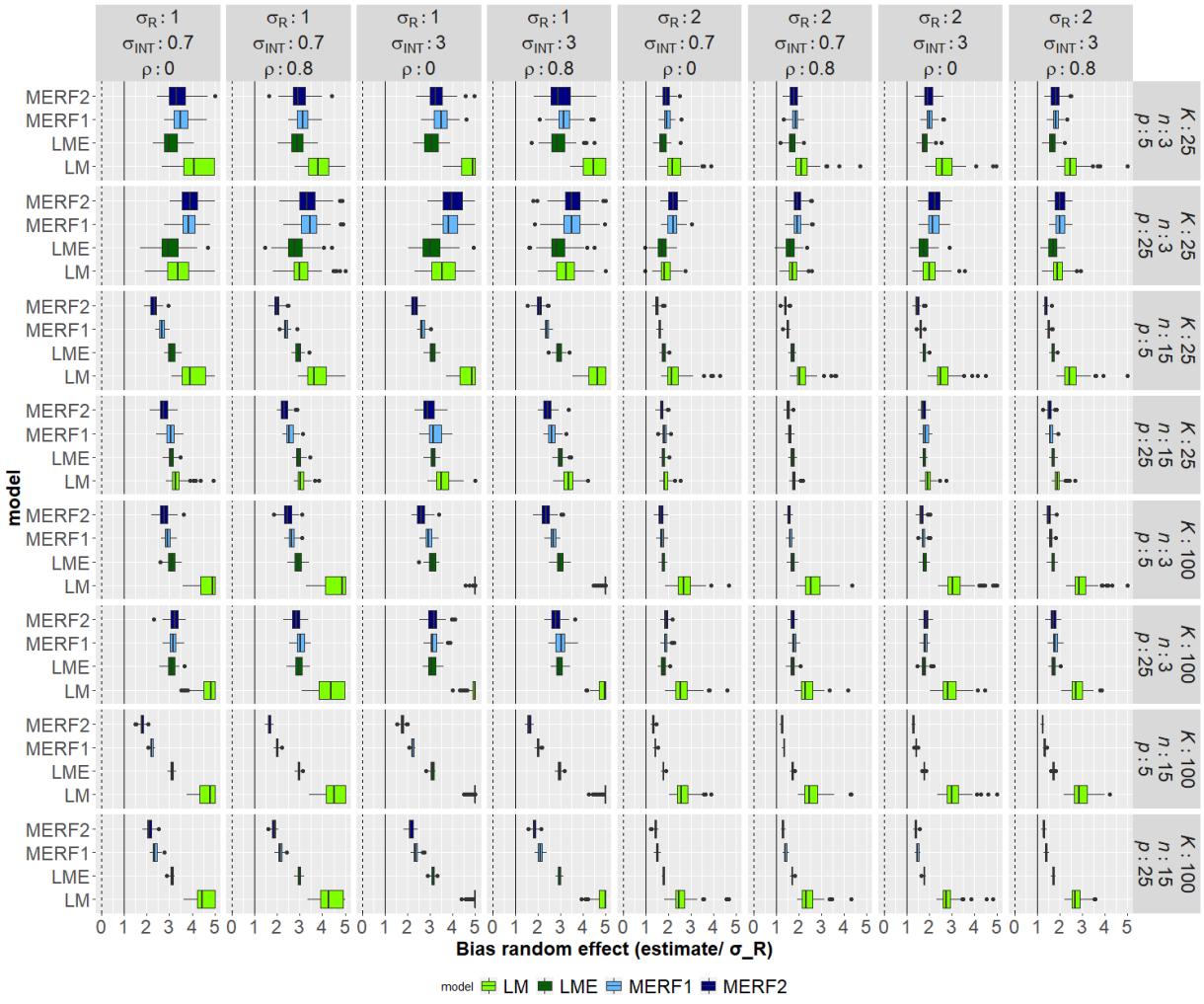

Figure A.7: σ_R, evaluated for the non-polynomial response model.

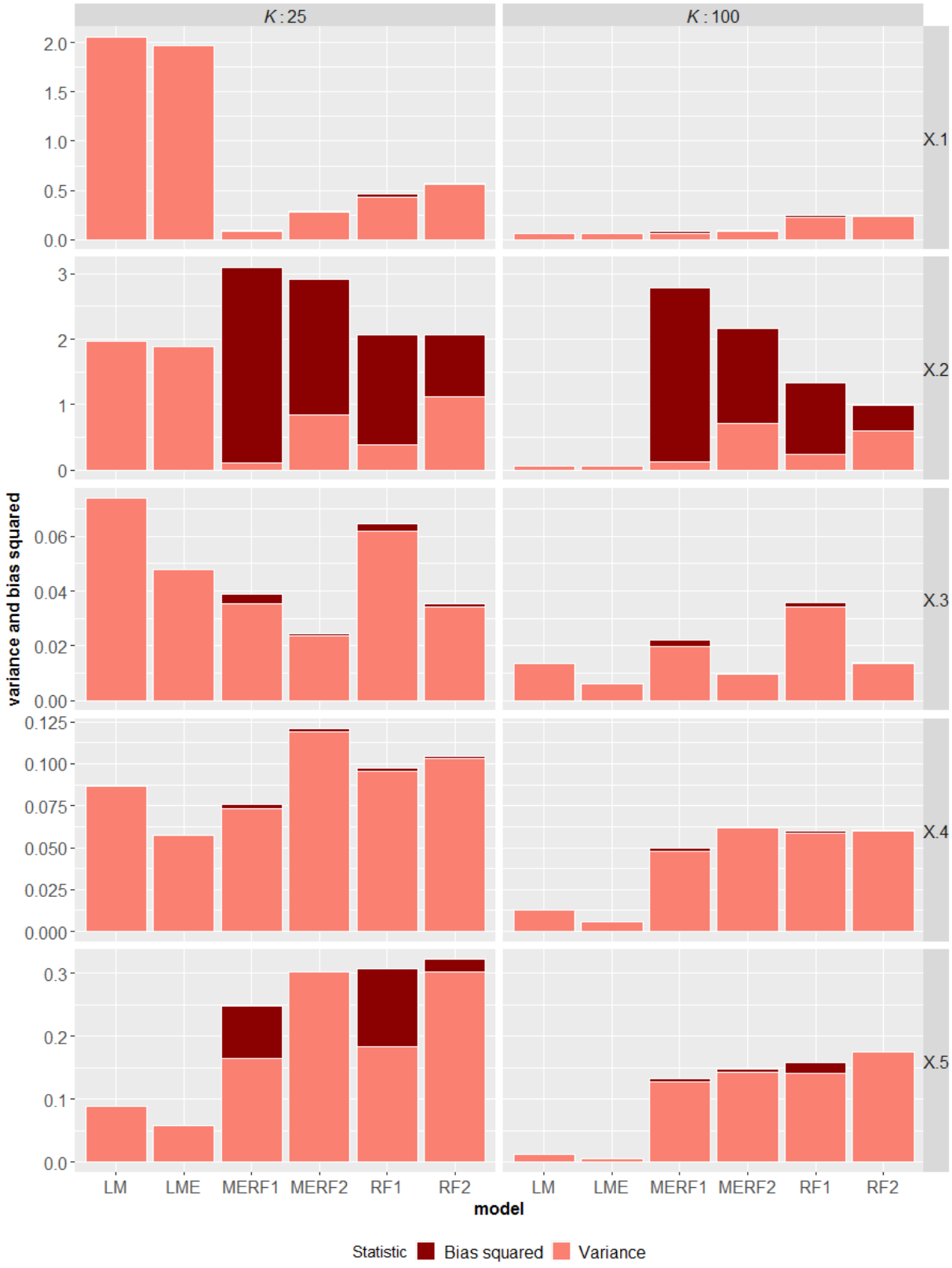

Figure A.8: Bar plot of the fixed effect estimates ($\alpha_{j,2}$) for the number of groups factor.

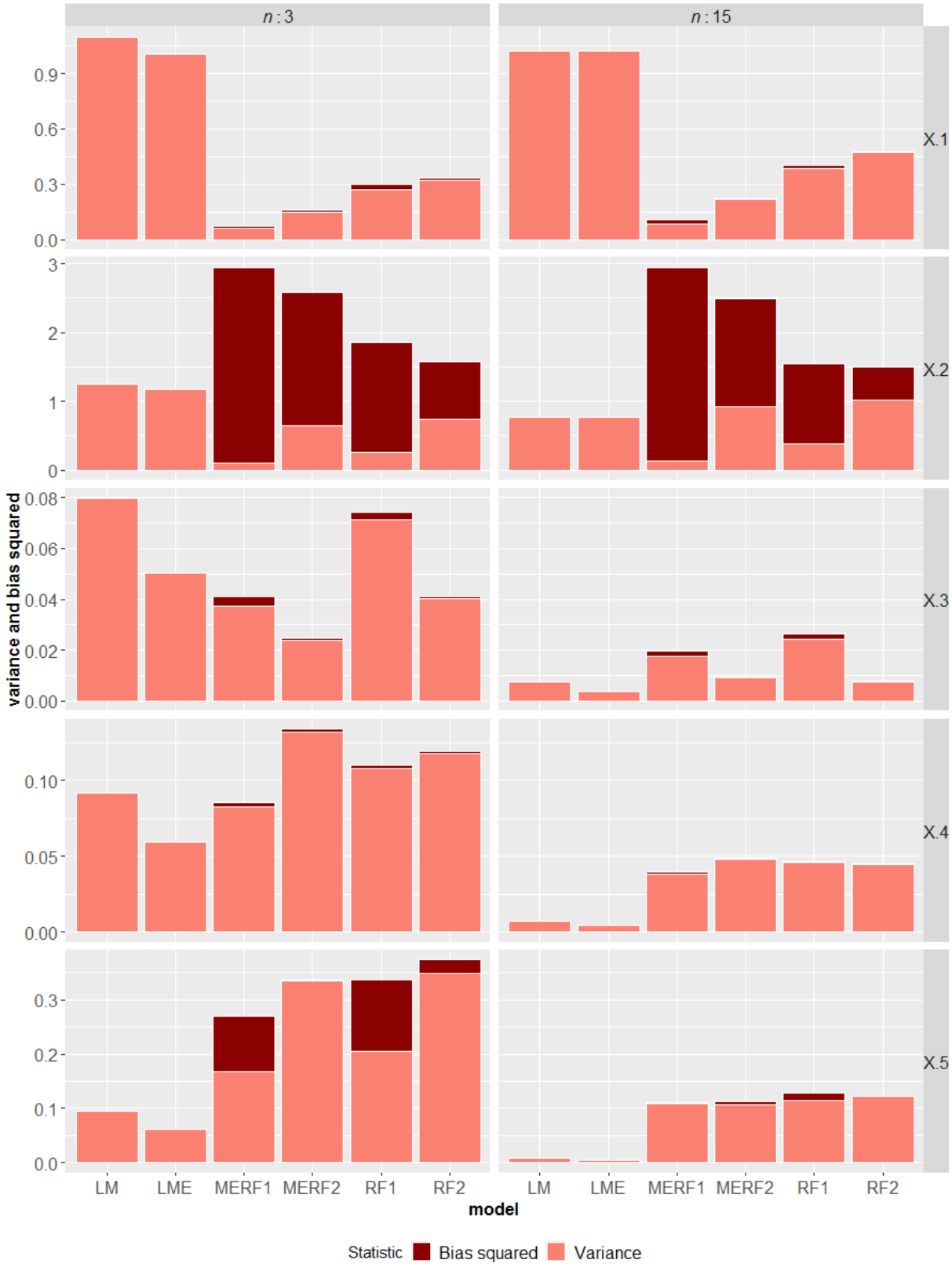

Figure A.9: Bar plot of the fixed effect estimates ($\alpha_{j,2}$) for the group size factor.

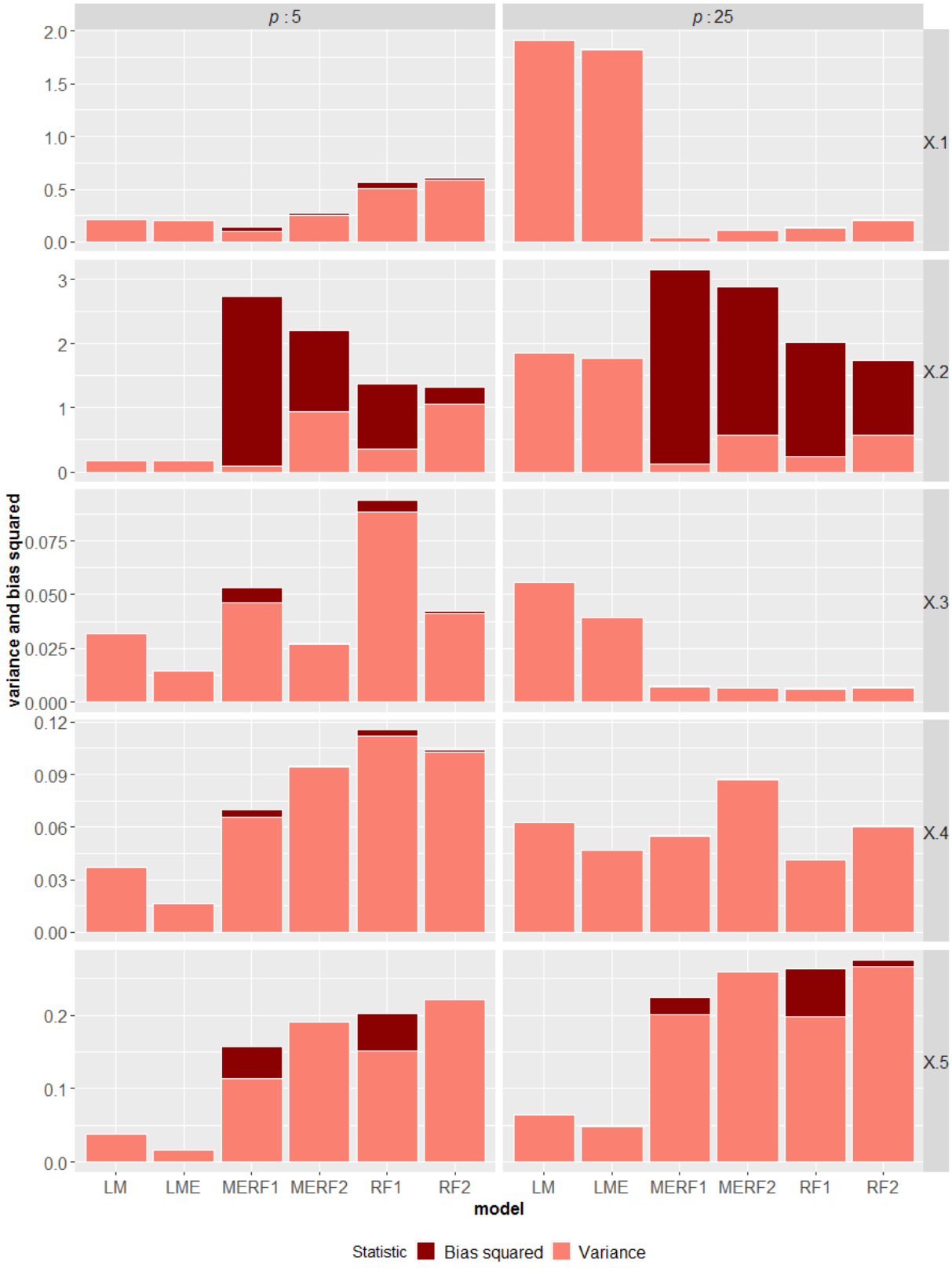

Figure A.10: Bar plot of the fixed effect estimates ($\alpha_{j,2}$) for the number of predictors factor.

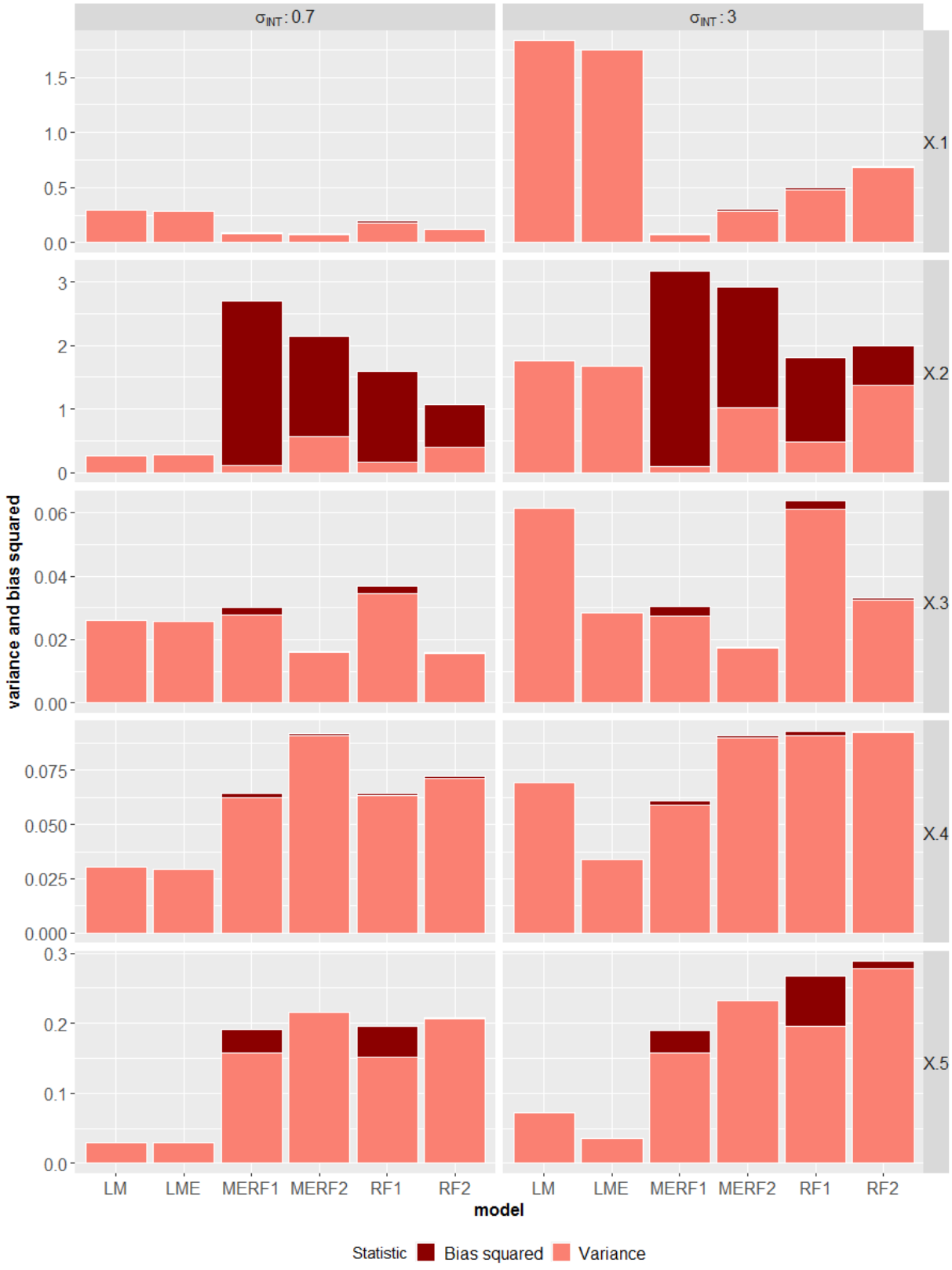

Figure A.11: Bar plot of the fixed effect estimates ($\alpha_{j,2}$) for the random intercept size factor.

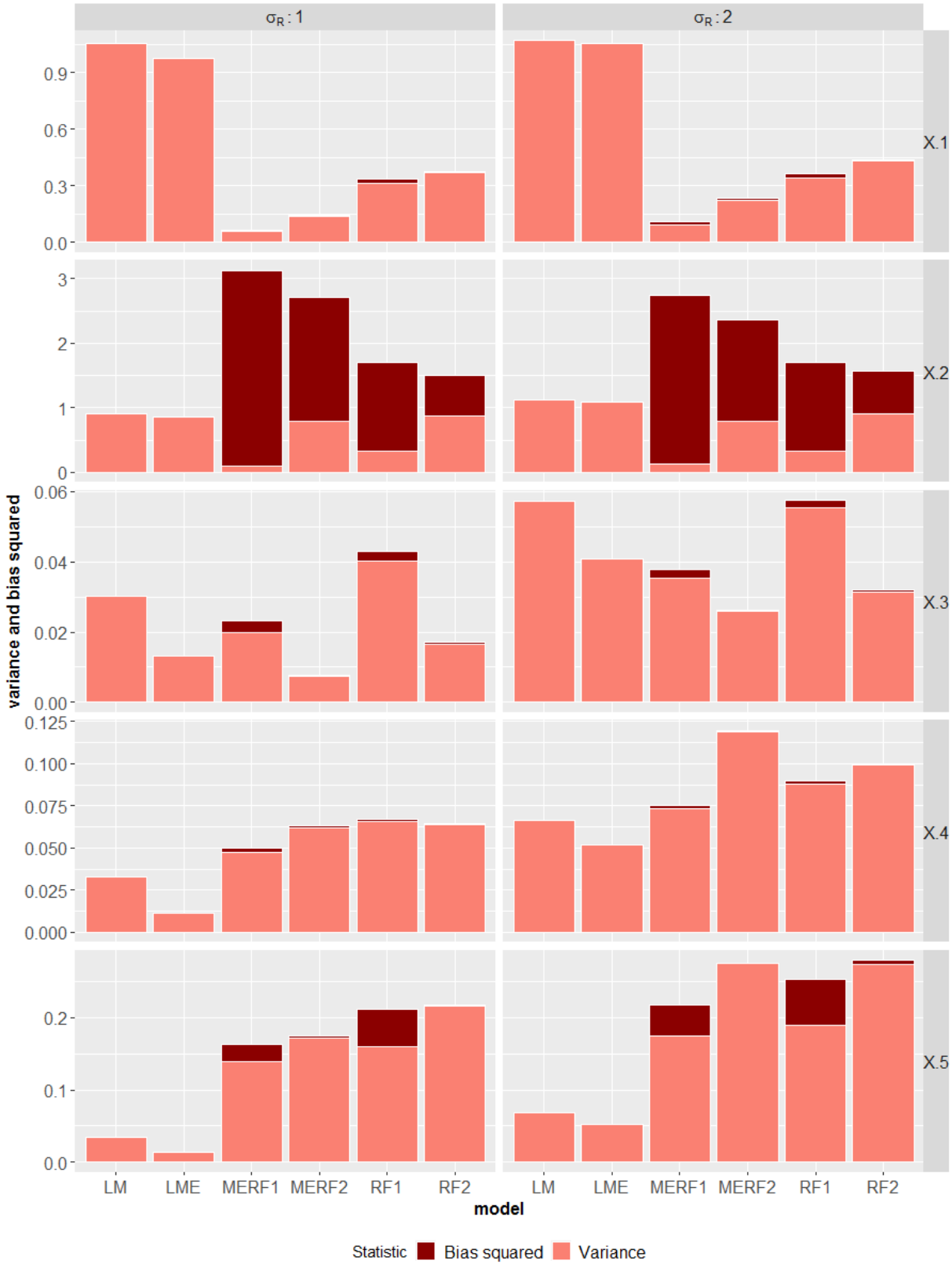

Figure A.12: Bar plot of the fixed effect estimates ($\alpha_{j,2}$) for the residual size factor.

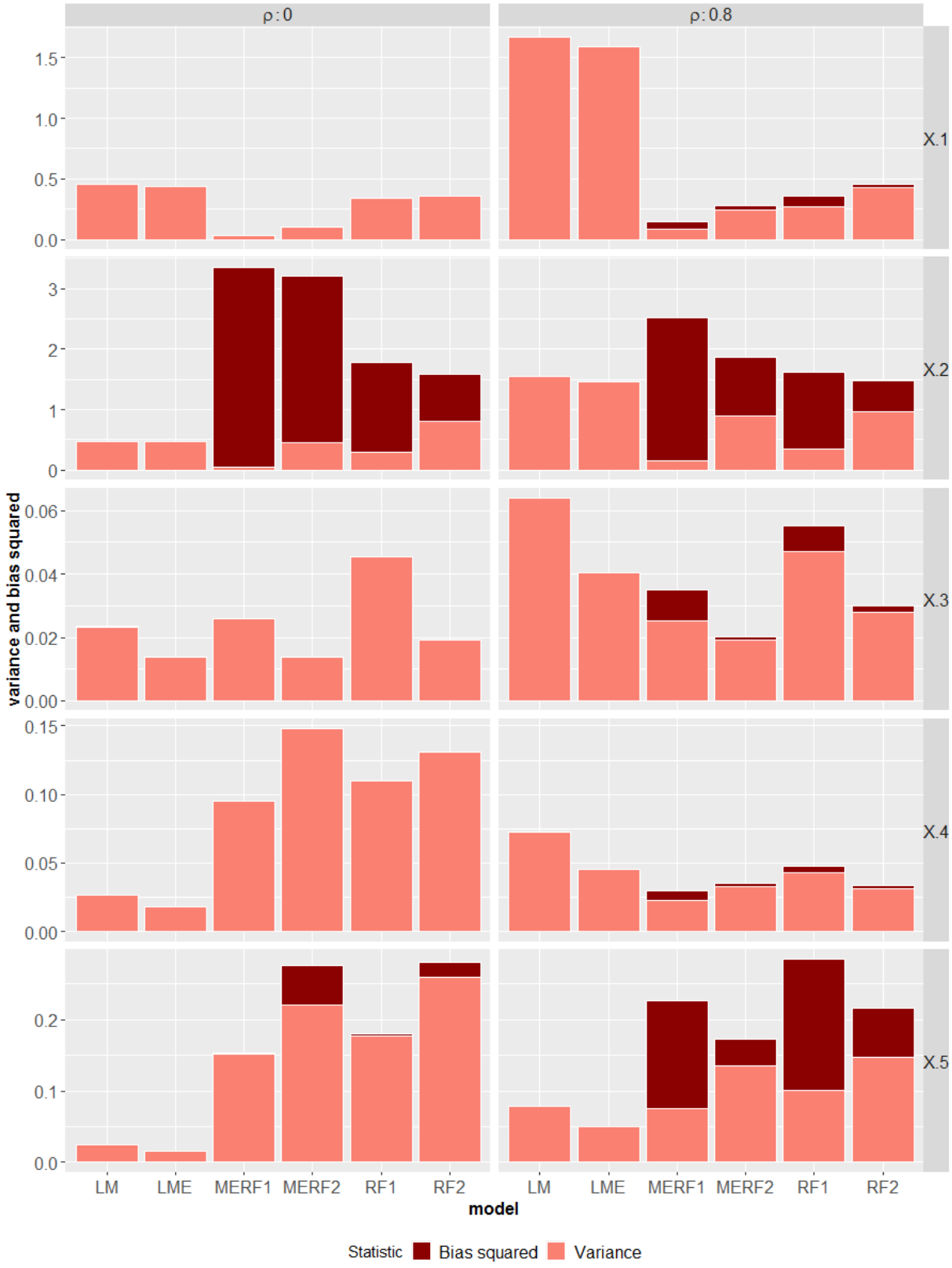

Figure A.13: Bar plot of the fixed effect estimates ($\alpha_{j,2}$) for correlation factor.

A.0.2 Non-polynomial response

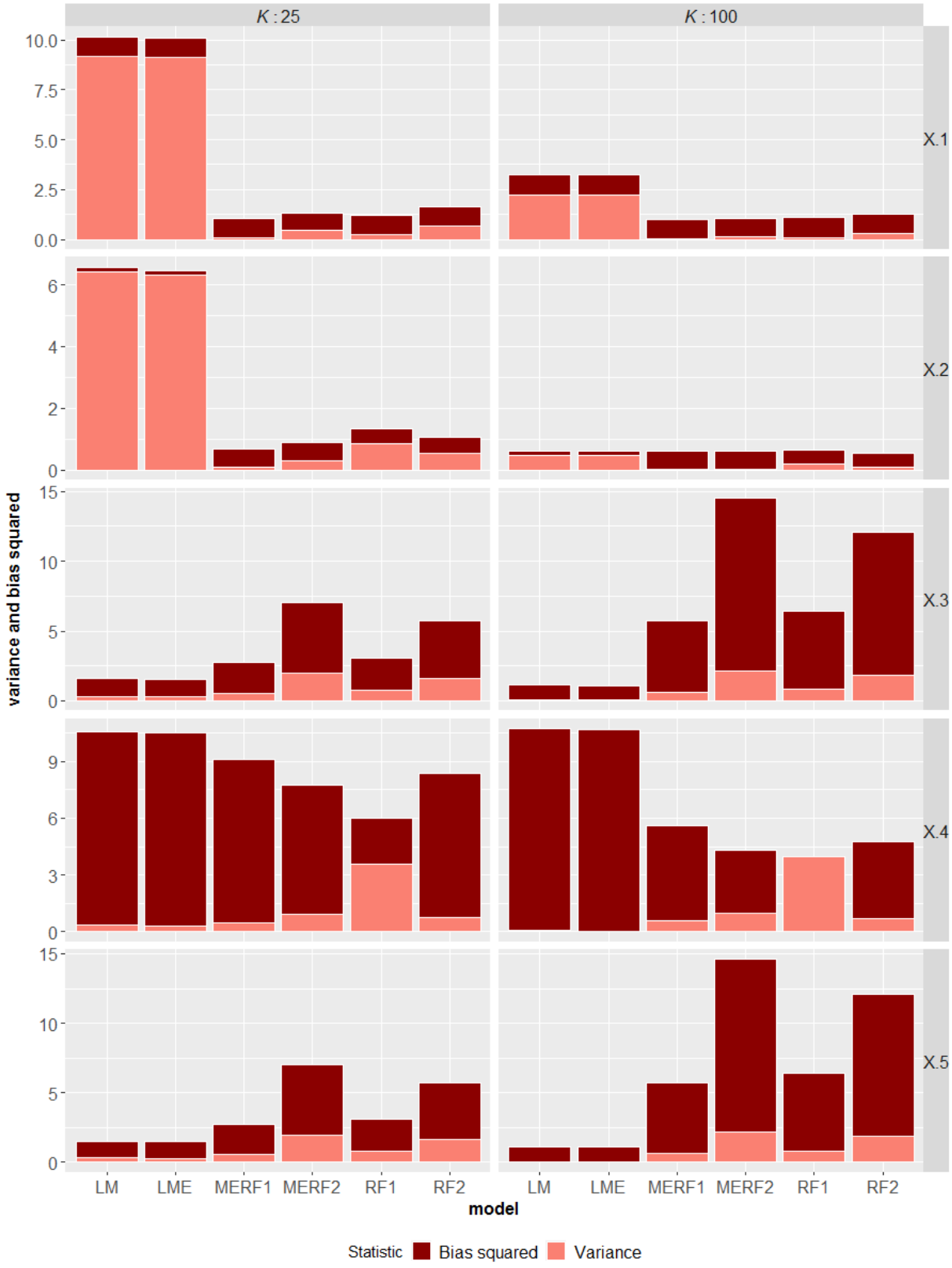

Figure A.14: Bar plot of the fixed effect estimates ($\alpha_{j,2}$) for the number of groups factor.

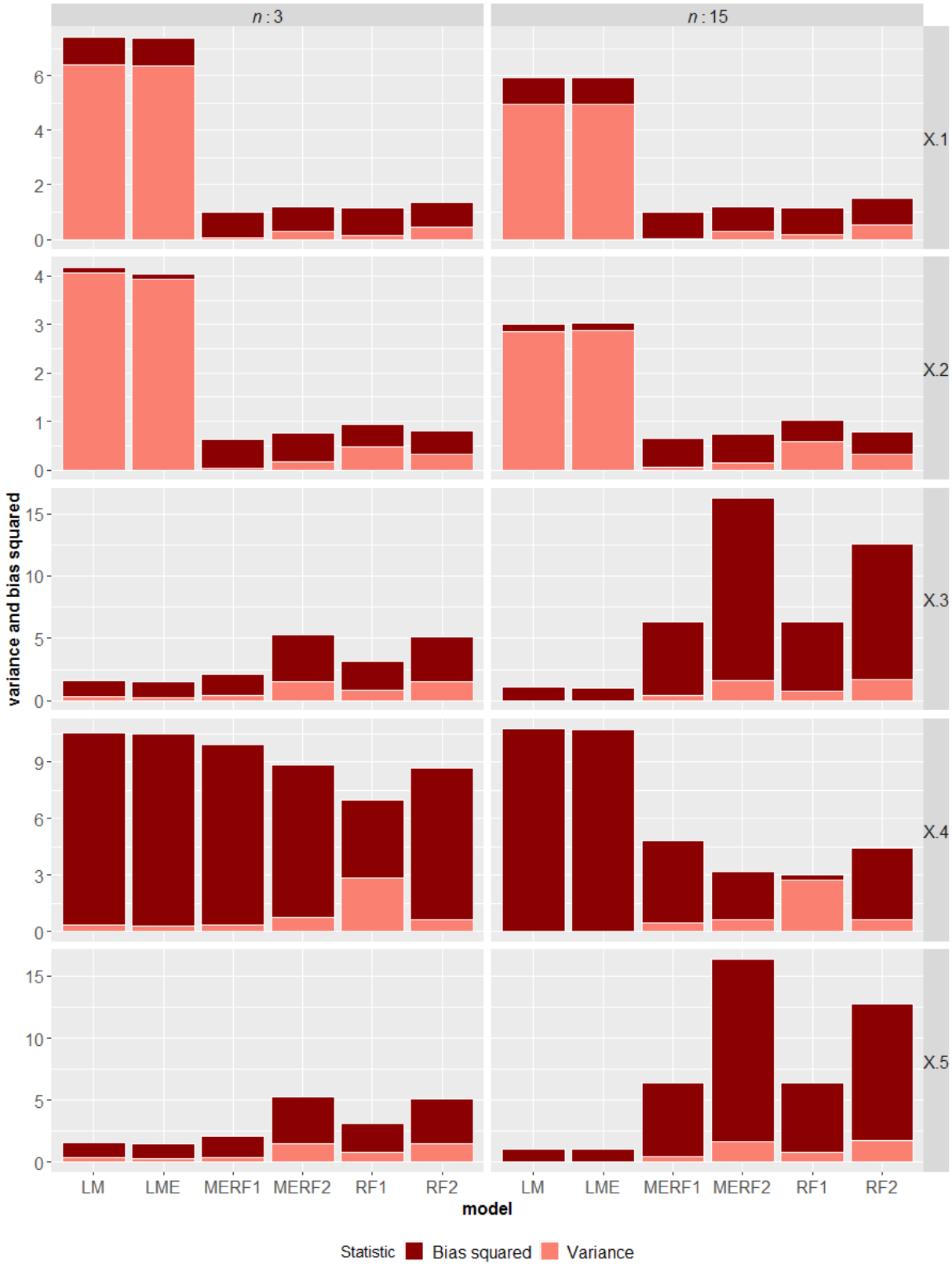

Figure A.15: Bar plot of the fixed effect estimates ($\alpha_{j,2}$) for the group size factor.

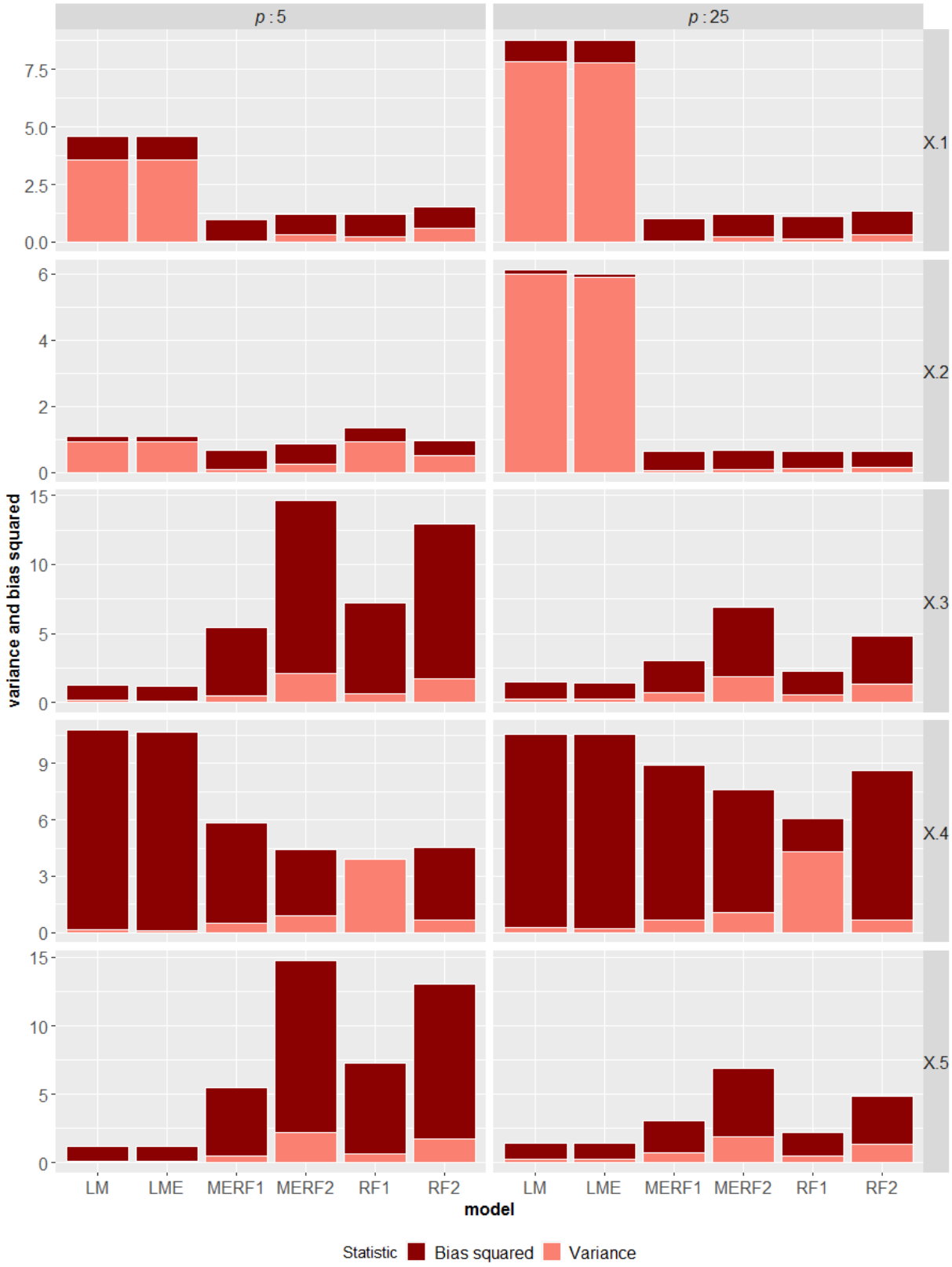

Figure A.16: Bar plot of the fixed effect estimates ($\alpha_{j,2}$) for the number of predictors factor.

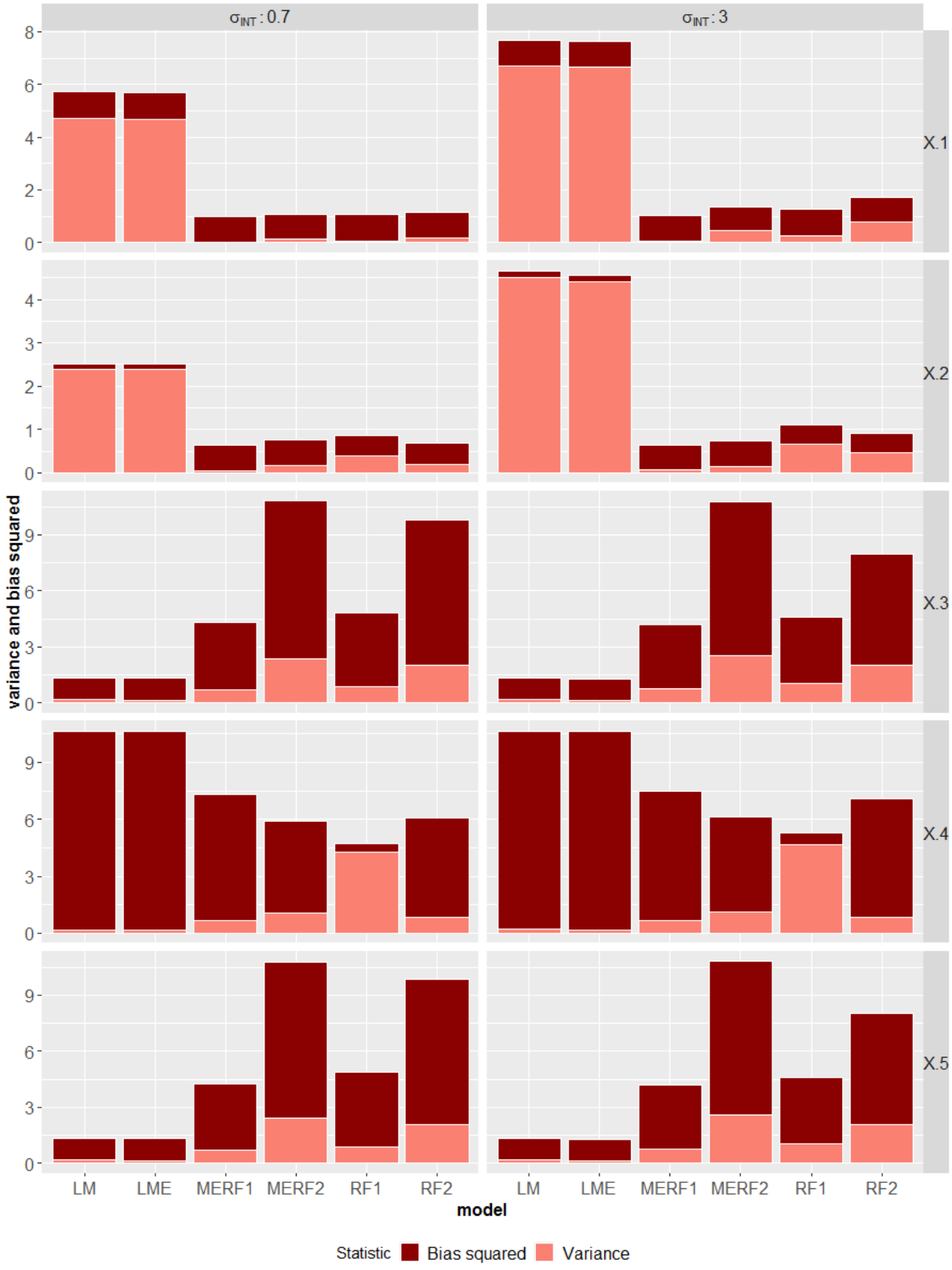

Figure A.17: Bar plot of the fixed effect estimates ($\alpha_{j,2}$) for the random intercept size factor.

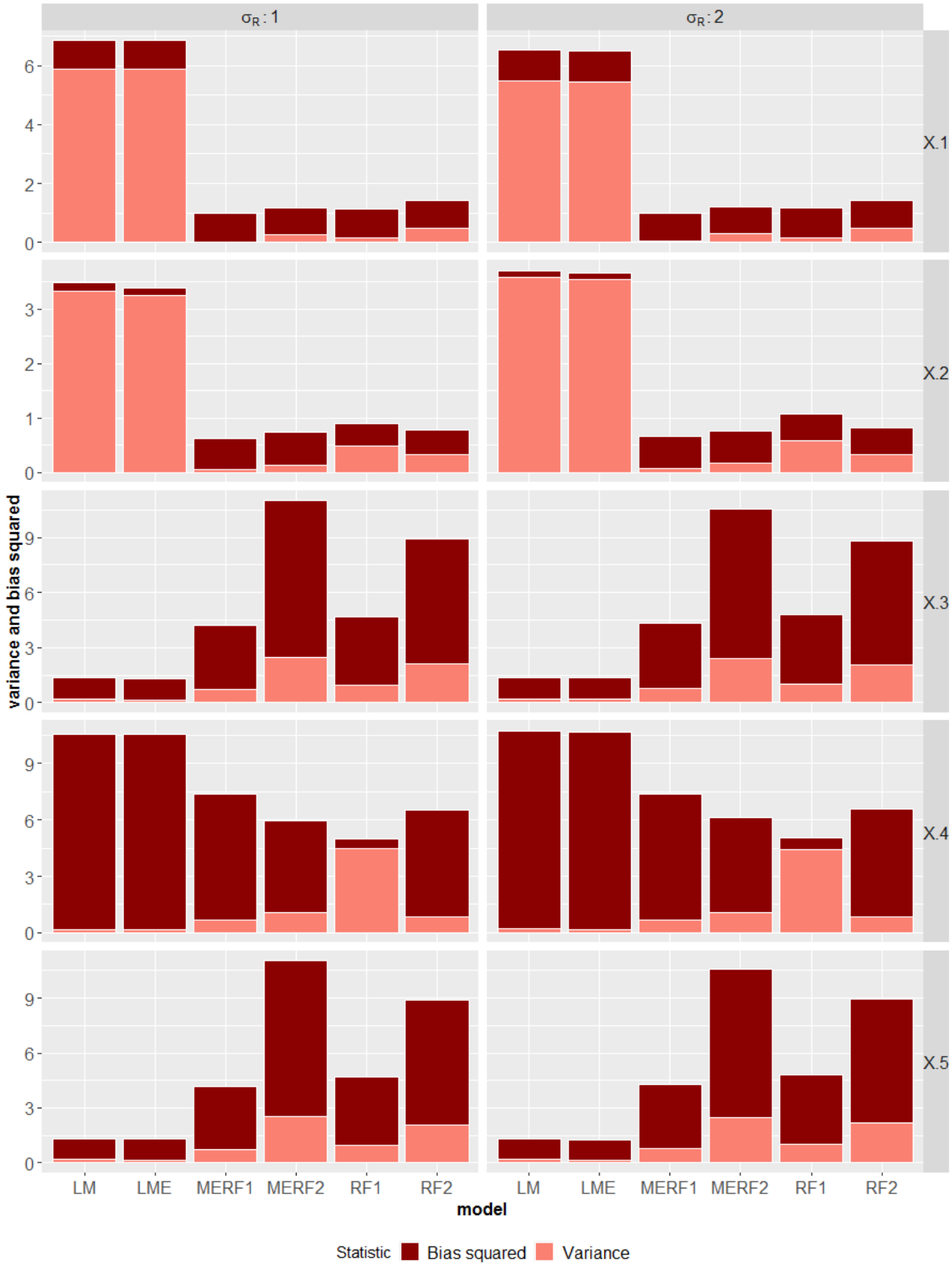

Figure A.18: Bar plot of the fixed effect estimates ($\alpha_{j,2}$) for the residual size factor.

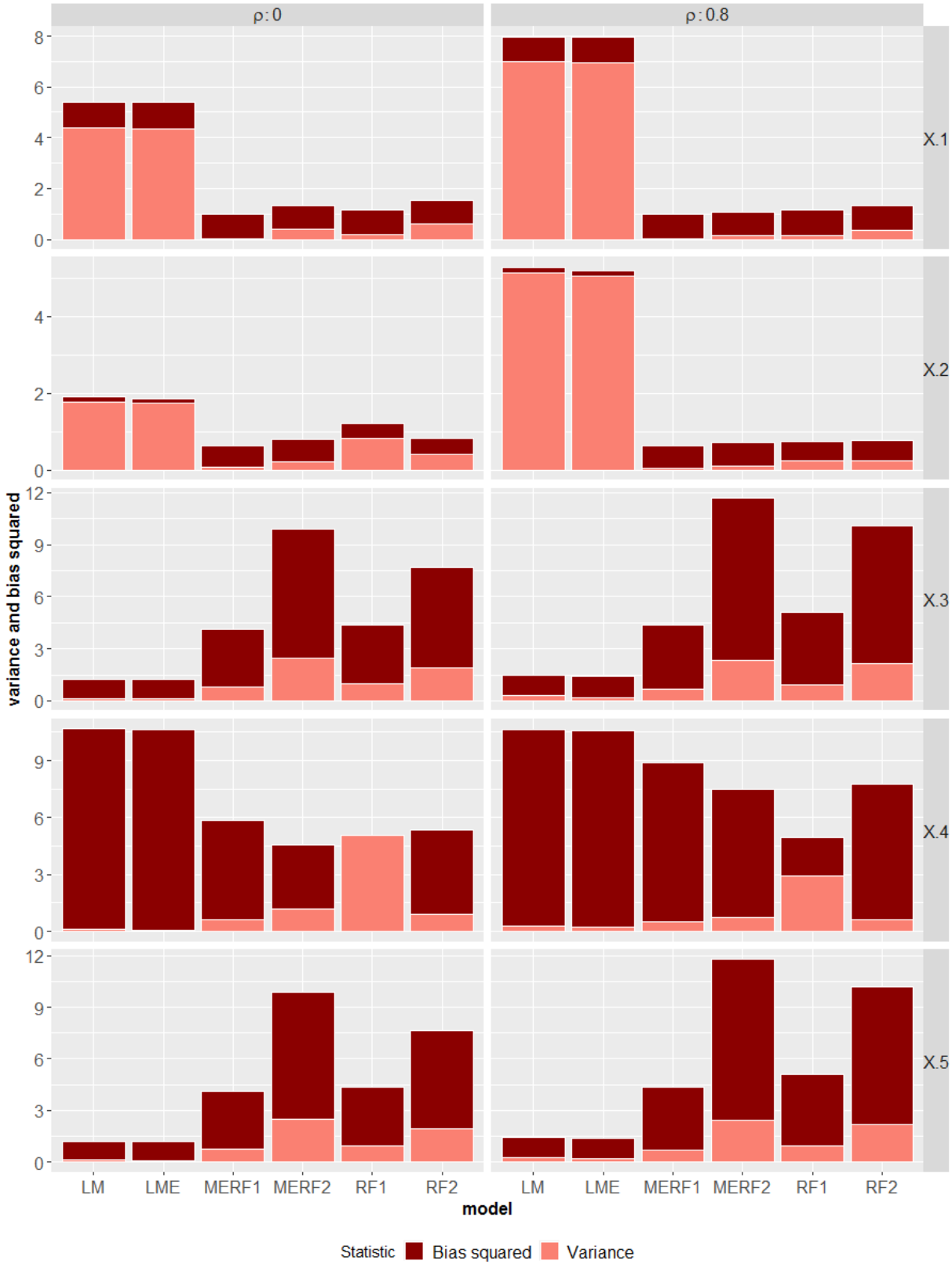

Figure A.19: Bar plot of the fixed effect estimates ($\alpha_{j,2}$) for correlation factor.