A new variable importance measure for random forests with missing data

Alexander Hapfelmeier
Institut für Medizinische Statistik und Epidemiologie,
Technische Universität München,
Ismaninger Str. 22, 81675 Munich, Germany,
Alexander.Hapfelmeier@tum.de

Torsten Hothorn
Institut für Statistik,
Ludwig-Maximilians-Universität,
Ludwigstraße 33, 80539 München, Germany

Kurt Ulm
Institut für Medizinische Statistik und Epidemiologie,
Technische Universität München,
Ismaninger Str. 22, 81675 Munich, Germany

Carolin Strobl
Department of Psychology,
University of Zurich,
Binzmühlestrasse 14, 8050 Zurich, Switzerland

March 13, 2013

Abstract

Random forests are widely used in many research fields for prediction and interpretation purposes. Their popularity is rooted in several appealing characteristics, such as their ability to deal with high dimensional data, complex interactions and correlations between variables. Another important feature is that random forests provide variable importance measures that can be used to identify the most important predictor variables. Though there are alternatives like complete case analysis and imputation, existing methods for the computation of such measures cannot be applied straightforward when the data contains missing values. This paper presents a solution to this pitfall by introducing a new variable importance measure that is applicable to any kind of data – whether it does or does not contain missing values. An extensive simulation study shows that the new measure meets sensible requirements and shows good variable ranking properties. An application to two real data sets also indicates that the new approach may provide a more sensible variable ranking than the widespread complete case analysis. It takes the occurrence of missing values into account which makes results also differ from those obtained under multiple imputation.

Keywords: variable importance measures, permutation importance, random forests, missing values, missing data

The original publication (Hapfelmeier et al., 2012) is available at `www.springerlink.com`.

1 Introduction

Recursive partitioning methods – in particular classification and regression trees, bagging and random forests – are popular approaches in statistical data analysis. They are applied in many research fields such as social, econometric and clinical science. Among others, there are approaches like the famous CART algorithm introduced by Breiman et al. (1984), the C4.5 algorithm by Quinlan (1993) and conditional inference trees by Hothorn et al. (2006). A detailed listing of further application areas and methodological issues, along with discussions about the historical development and state-of-art, can be found in Strobl et al. (2009). The popularity of trees is rooted in their easy applicability and interpretability. Advantages over common approaches like logistic and linear regression are their ability to implicitly deal with missing values, collinearity and high dimensional data sets. Moreover, recursive partitioning is able to detect even complex interaction effects, as pointed out by, e.g., Lunetta et al. (2004), Cutler et al. (2007) and Tang et al. (2009).

Likewise, random forests (cf. Breiman, 2001) – besides achieving an improved prediction accuracy – can be used to identify relevant variables. Therefore variable importance measures, which are in main focus of this

paper, provide an assessment of the predictive power of a variable in a forest. In addition they are often used for variable selection, too. The work of Tang et al. (2009), Yang and Gu (2009), Rodenburg et al. (2008), Sandri and Zuccolotto (2006) and Díaz-Uriarte and Alvarez de Andrés (2006) shows that different approaches that are meant to improve prediction accuracy and select the most relevant variables have been developed. Yang and Gu (2009) show that a forest may benefit from previous variable selection by means of an example from the field of genome-wide association studies. Altmann et al. (2010), on the other hand, discuss that variable selection may equally harm and improve analysis. A summary of several selection methods and related publications can be found in Archer and Kimes (2008).

Despite the widespread usage of random forest variable importance measures, to our knowledge there is no straightforward suggestion on how to compute such measures in the presence of missing values. Therefore, in the following we will propose a new approach for dealing with this problem: We will show how our method can deal with missing values in an intuitive and straightforward way, yet retaining the widely appreciated qualities of existing random forest variable importance measures. The properties of the new method are investigated in an extensive simulation study. The results show that a list of sensible demands we had pre-specified are completely fulfilled. An application to two real data sets shows the practicability of the new method in real life situations and a potential superiority to complete case analysis. Results also differ from those obtained under multiple imputation as the new method reflects the data situation at hand, taking the occurrence of missing values into account.

2 Missing Data

In an early work Rubin (1976) specifies the issue of correct statistical inference from data containing missing values. A key instrument is the declaration of the process that causes the missingnes. Based on this considerations many model strategies for inference and elaborate definitions of the subject have been developed. An extensive summary can be found in Schafer and Graham (2002). In general, three types of missingness are distinguished:

- Missing completely at random (MCAR): $P(R|X_{\text{comp}}) = P(R)$
- Missing at random (MAR): $P(R|X_{\text{comp}}) = P(R|X_{\text{obs}})$
- Missing not at random (MNAR): $P(R|X_{\text{comp}}) = P(R|X_{\text{obs}}, X_{\text{mis}})$

The status of missingnes (yes = 1/no = 0) is indicated by R and its probability $P(R)$. The letter R, that was adopted from the original notation, may emerge from the fact that Rubin (1987) originally was dealing with 'R'esponse rates in surveys. The complete variable set X_{comp} consists of the observed values X_{obs} and the missing ones X_{mis}. $X_{\text{comp}} = \{X_{\text{obs}}, X_{\text{mis}}\}$. Therefore MCAR indicates that the probability of a missing value is independent of the observed and unobserved data. By contrast for MAR this probability is dependent on the observed data. Finally, in MNAR it depends on the missing values themselves – for example because those subjects with high response values systematically drop out of a study.

The findings of Little and Rubin (2002) showed that usual sample estimates – for example in linear regression – stay unaffected by the MCAR scheme. However, Strobl et al. (2007) outlined that in classification and regression trees even MCAR may induce a systematic bias, that may be carried forward to random forests based on biased split selections. Therefore, in our following simulation study, one MCAR, four MAR and one MNAR process to generate missing values are investigated to shed light on the sensitivity of the proposed method to these schemes.

3 Methods

3.1 Recursive Partitioning

The main idea of recursive partitioning can be best described by the example of the CART algorithm. The construction of trees is based on sequential splits of the data into binary subsets. These are conducted in single variables due to different criteria depending on the response type. A popular choice for binary responses is to use the Gini Index (cf. Breiman et al., 1984, for details). The growth of the tree is stopped when a certain criterion, such as a limiting number of observations in the final subsets, is reached. After a tree has been grown to its maximal size it can be "pruned" back: For pruning the performance of the tree is evaluated via cross-validation at different growth stages. A popular choice of the final size is given by the smallest tree that

still produces an error within a range of one standard error to the best performing tree (cf. Breiman et al., 1984; Hastie et al., 2009, for a more detailed description of the entire approach).

Unfortunately, the CART algorithm – and consequently all random forest algorithms based on the same construction principles – favour splits in continuous variables and variables with many categories. Likewise, predictors with many missing values may be preferred if the Gini Index is employed (Strobl et al., 2007). To overcome this problem, several unbiased tree algorithms have been suggested (see, e.g., White and Liu, 1994; Kim and Loh, 2001; Dobra and Gehrke, 2001; Hothorn et al., 2006; Strobl et al., 2007). From these unbiased approaches, our following work will be based on the algorithm suggested by Hothorn et al. (2006), who introduced conditional inference trees. Basically, these trees are grown by node splitting which is performed in two steps. 1) The association of predictors to the response is assessed in a permutation test framework. Predictors that reach statistical significance are found to be eligible for splitting. 2) Among those, the node split is conducted in the predictor that showed the strongest relation to the response. This approach guarantees unbiased variable selection and shows several other good statistical properties.

3.2 Ensemble Methods

Breiman (1996) enhanced the tree methodology by means of "bagging" (bootstrap aggregation). He was able to show that the performance benefits from using ensembles of trees instead of single trees as the variance of predictions is reduced. In bagging, several trees are fit to bootstrap or subsamples of the data. Averaged values or majority votes of the response values predicted by each single tree are used as predictions. As an advancement of bagging, random forests (Breiman, 2001; Breiman and Cutler, 2008) extend this approach: In random forests, splits are performed only in a random selection of variables, which enables a more diverse set of variables to contribute to the joint prediction. There is no general advice on how many trees should be used in a forest. Breiman (2001) proves that random forests don't overfit with a rising number of trees (while the results of Lin and Jeon, 2006, indicate that they do overfit when trees are grown too large). Further research of Biau et al. (2008) lead to theorems about the consistency of Random Forest approaches and other averaging rules. Likewise, Genuer (2010) was able to show the superiority of a variant of Random Forests to single trees and therefore proved the attendant question of variance reduction in this special case.

From the recursive partitioning approach of Hothorn et al. (2006), random forests can be constructed following the same rationale as Breiman's original approach. The advantage of this method is again that it guarantees unbiased variable selection and variable importance measures when combined with subsampling (as opposed to bootstrap sampling) (Strobl et al., 2007). This framework for constructing random forests is used in the following. Note, however, that the new importance measure can generally be applied in all other random forest algorithms without any restriction.

3.3 Surrogate Splits

There are several possibilities to handle missing values. One of them is to stop the throughput of an observation at the node at which the information for the primary split rule is missing (the prediction is then based on the conditional distribution of the responses that are element of this node). Another approach makes the missing values simply follow the majority of all observations with observed values (Breiman et al., 1984). However, the by far most popular way of handling missing observations is to use surrogate decisions based on additional variables (Breiman et al., 1984; Hothorn et al., 2006). These splits try to mimic the primary split by archiving the same partitioning of the observed values. When several surrogate splits are computed they can be ranked according to their ability of resembling the original split. When an observation contains further missing values in surrogate variables they are processed in order of similarity to the primary split until a decision for a missing value is found. The number of possible surrogate splits is usually determined by the user.

3.4 Variable importance measures

The work of Sandri and Zuccolotto (2006), Altmann et al. (2010), Wang et al. (2010) and Zhou et al. (2010) shows that the development of new importance measures is still an ongoing process. In the following we present the most common and popular ones.

3.4.1 Gini importance

The Gini importance, that is available in many random forest implementations, accumulates the Gini gain over all splits and trees to evaluate the discriminatory power of a variable (Hastie et al., 2009). One advantage of this measure is that – in principle – it is applicable to missing data. However, all classification tree and random

forest algorithms based on the Gini splitting criterion are prone to biased variable selection (Strobl et al., 2007; Hothorn et al., 2006). Recent results also indicate that it has undesirable variable ranking properties, especially when dealing with unbalanced category frequencies (Nicodemus, 2011). Therefore the Gini importance is not considered here.

3.4.2 Permutation importance

The most popular and most advanced variable importance measure for random forests is the permutation accuracy importance. One of its advantages is its broad applicability and unbiasedness (when used in combination with subsampling as shown by Strobl et al., 2007). The permutation importance is assessed by comparing the prediction accuracy of a tree before and after random permutation of the predictor variable of interest. If it is of relevance the accuracy should decrease as the original association to the response is destroyed by permutation. The average of differences over all trees provides the final importance score. Large values of the permutation importance indicate a strong association between the predictor variable and the response. Values around zero (or even small negative values, cf. Strobl et al., 2009) indicate that a predictor is of no value for predicting the response.

In the computation of the permutation importance, the assessment of the prediction accuracy – in terms of correct classification or mean squared error – is usually based on observations that were not part of the sample used for constructing the respective tree (the so called "out of bag" (OOB) observations; cf. Breiman (2001)). This way the OOB accuracy provides a more realistic estimate of the prediction accuracy that can be expected for new data (cf., e.g., Boulesteix et al., 2008; Strobl et al., 2009).

In summary, the computation of the permutation importance consists of the following steps:

1. Compute the OOB accuracy of a tree.
2. Permute the predictor variable of interest in the OOB observations.
3. Recompute the OOB accuracy of the tree.
4. Compute the difference between the original and recomputed OOB accuracy.
5. Repeat step 1 to 4 for each tree and use the average difference over all trees as the overall importance score.

Besides the original version of the permutation importance, a conditional version (that more closely resembles the behavior of partial correlation or regression coefficients) was introduced by Strobl et al. (2008). The discussions in Nicodemus et al. (2010) and Altmann et al. (2010) show that both kinds of measures, conditional and unconditional, can be of specific value depending on the research question. However, the variety of recent publications, like that of Yu et al. (2011), indicate that the original permutation importance is still extremely popular in many research fields. Therefore, we will concentrate on the construction of an unconditional importance measure in the following, but will address an extension for conditional variable importance in further research.

The main problem of the approach for computing the permutation importance as described above is that there is no straightforward way to compute this measure in the presence of missing values. In particular, it is not clear how conclusions about the importance of variables can be drawn from the permutation approach when surrogate splits are used for the computation of the OOB accuracy but are not part of the permutation scheme.

4 New proposal

A new approach is suggested here in order to provide a straightforward and intuitive way of dealing with missing values in the computation of a random forest variable importance measure. The construction of the new measure closely sticks to the existing methodology and the reader will find that it deviates from the original permutation importance only in one – yet one substantial – step.

The main idea of the new proposal is the following: Instead of permuting the values of a variable (that may be missing), the observations are randomly allocated to one of the child nodes if the split of their parent node is conducted in the variable of interest. This procedure detaches any decision from the raw values of the variable, and therefore circumvents any problems associated with the occurrence of missing values and the application of surrogate splits for the computation of the OOB accuracy.

The rest of the computation procedure, however, is not affected by this "trick": In the first step of the computation one proceeds as normal by recording the prediction accuracy of a tree (including all surrogate splits, which can be considered as an implicit imputation of the missing values). In a second step, the prediction accuracy is again recomputed by randomly assigning observations that were originally split in the variable of interest to the corresponding child nodes.

Formally introducing a binary random variable D, that indicates the decision for one of the child nodes, the probability of being send to the left ($D = 0$) or to the right ($D = 1$) child node, respectively, is given by $P_k(D = 0)$ and $P_k(D = 1) = 1 - P_k(D = 0)$ for a node k. The random allocation of the observations – just like the random permutation of the values of a predictor variable X_j itself in the original permutation importance – mimics the null hypothesis that the assignment of observations to nodes does not depend on this particular predictor variable any more: Under the null hypothesis, the probability to end up in a specific child node of node k is $P_k(D|X_j) = P_k(D)$. Therefore it also does not matter whether a value of X_j is missing or not, as it is not used for the decision of how to further process an observation.

For the practical computation of the prediction accuracy, the probability $P_k(D = 0)$ is replaced by its empirical estimator, the relative frequency:

$$\hat{p}_k(D = 0) = n_{k,\text{left}}/n_k$$

where $n_{k,\text{left}}$ and n_k are the number of observations that were originally send to the left child node and were present in the parent node k, respectively. In contrast to the original permutation importance measure presented in section 3.4.2, the computation of the new measure consists of the following steps, highlighting the essential difference in step 2 and 3:

1. Compute the OOB accuracy of a tree.
2. **Randomly assign each observation with $\hat{p}_k(D = 0)$ to the child nodes of a node k that uses X_j for its primary split.**
3. Recompute the OOB accuracy of the tree **following step 2**.
4. Compute the difference between the original and recomputed OOB accuracy.
5. Repeat step 1 to 4 for each tree and use the average difference over all trees as the overall importance score.

5 Simulation studies and analysis settings

5.1 Simulation

5.1.1 Rationale

An extensive simulation study was designed to shed light on the characteristics of the proposed importance measure. Some properties are supposed to be close to those of the original permutation importance measure like the fact that correlated variables obtain higher importances than uncorrelated ones in an unconditional assessment. Others are still to be investigated like the effect of different correlation structures (that determine the quality of surrogate splits), schemes of missingness, the amount of missing values and so forth.

In order to assess the performance of the newly suggested approach, before running the simulation experiments we formulated a list of requirements that should be met by a sensible variable importance measure designed for dealing with missing values:

- (R1) When there are no missing values, the measure should provide the same variable importance ranking as the original permutation importance measure.
- (R2) The importance of a variable is not supposed to artificially increase, but to decrease with an increasing amount of missing values (because the variable loses information, cf. Strobl et al., 2007).
- (R3) Like the original permutation importance (that is a marginal importance in the sense of Strobl et al., 2008), the importances of a variable is supposed to increase with an increasing correlation to other influential predictor variables.

- (R4) The importance ranking of variables not containing missing values is supposed to stay unaffected by the amount of missing values in other variables. This is only required within groups of equally correlated variables as differences in correlation directly affect variable importances and therefore may well change the ranking.
- (R5) The importance of influential variables is supposed to be higher than the importance of non-influential variables. This should hold for equally correlated variables with equal amounts of missing values – considering that both facts influence the importance of a variable. For example a non-influential variable which does not contain missing values and is correlated with an influential variable can achieve a higher importance than an influential variable containing missing values (cf. Strobl et al., 2008). In any way, the lowest importance should always be assigned to non-influential variables that are uncorrelated to any other influential variables.

In order to investigate these requirements and further characteristics of the newly suggested approach, an extensive simulation study was set up, as described in the following sections.

5.1.2 Settings

There are several factors that need to be varied in the simulation setting, in particular the amount of missing values, correlation strength, the number of correlated variables (termed block size in the following), variable influence and different missing value mechanisms. A detailed explanation of the setup is given in the following.

- *Influence of predictor variables*

  The proposed importance measure is supposed to be applicable in both classification and regression problems. Thus, the simulated data contained a categorical (binary) and a continuous response. In both cases the coefficients β, that were used to produce 20 variables in the data generating model described below, are:

  $$\beta = (4, 4, 3, 4, 3, 4, 3, 4, 3, 0, 0, 2, 2, 0, 0, 0, 0, 0, 0, 0)^\top$$

  The idea of this setup was that repeatedly choosing the same values for β enables a direct comparison of importances belonging to variables which are, by construction, equally influential. The different β values are crossed with the other experimental factors to allow an evaluation of differences and provide reference values for each setting. In addition, the non-influential variables with $\beta = 0$ help to investigate possible undesired effects and again serve as a baseline.

- *Data generating models*

  A continuous response was modeled by means of a linear model:

  $$y = x^\top \beta + \epsilon \text{ with } \epsilon \sim N(0, .5).$$

  The binary response was modeled by means of a logistic model:

  $$P(Y = 1|X = x) = \frac{e^{x^\top \beta}}{1 + e^{x^\top \beta}}.$$

  The variable space X itself contains 100 observations drawn from a multivariate normal distribution with mean vector $\vec{\mu} = 0$ and covariance matrix Σ:

- *Correlation*

$$\Sigma = \begin{pmatrix}
1 & r & r & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
r & 1 & r & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
r & r & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 1 & r & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & r & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & r & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & r & 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & r & r & r & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & r & 1 & r & r & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & r & r & 1 & r & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & r & r & r & 1 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

As the variances of each variable are chosen to be 1, the covariance equals the correlation in this special case. The strength of correlation was varied by setting r to 0, .3, .6 and .9. The structure of the 20×20 dimensional covariance matrix Σ reveals that there are four blocks of different size, each of them consisting of 3, 2, 2 and 4 variables. Thus it was possible to investigate the effect that the strength and extent of the correlation had on the importance measure.

- *Missing values*

  In analogy to the simulation setting of Rieger et al. (2010), who investigate the performance of random forests on missing data, several MCAR and MAR schemes of creating missing values were implemented. In addition, a further MNAR setting was investigated, too. In each scheme, a given fraction m of observed values is replaced by missing values. As the amount of missing values is of major concern in our simulation experiments, m takes the values 0%, 10%, 20% and 30%.

  In a MAR setting, the probability for missing values in a variable depends on the values of another variable. In the MNAR scheme this probability is determined by a variables own values. Accordingly, each variable containing missing values has to be linked to at least one other variable or itself. Table 1 lists the corresponding relations.

Table 1: List of the variables containing missing values and variables determining the probability of missing values.

contains missing values (MCAR, MAR & MNAR)	determines missing values (MAR)	(MNAR)
X_2	X_3	X_2
X_4	X_5	X_4
X_8	X_9	X_8
X_{10}	X_{11}	X_{10}
X_{12}	X_{13}	X_{12}
X_{14}	X_{15}	X_{14}

The schemes for producing missing values are:

- MCAR: Values are randomly replaced by missing values.
- MAR(rank): The probability of a value to be replaced by a missing value rises with the rank the same observation has in the determining variable.
- MAR(median): The probability of a value to be replaced by a missing value is 9 times higher for observations whose value in the determining variable is located above the corresponding median.
- MAR(upper): Those observations with the highest values of the determining variable are replaced by missing values.
- MAR(margins): Those observations with the highest and lowest values of the determining variable are replaced by missing values.
- MNAR(upper): The highest values of a variable are set missing.

A schematic illustration of β summarizes all factors varied in the simulation design below. Correlated blocks of variables are enumerated by roman figures and separated by '|'. Bold figures indicate variables that contain missing values:

$$\beta = (\underbrace{4, \mathbf{4}, 3}_{\text{I}} | \underbrace{\mathbf{4}, 3}_{\text{II}} | \underbrace{4, 3}_{\text{III}} | \underbrace{\mathbf{4}, 3, \mathbf{0}, 0}_{\text{IV}} | \underbrace{\mathbf{2}}_{\text{V}} | \underbrace{2}_{\text{VI}} | \underbrace{\mathbf{0}}_{\text{VII}} | \underbrace{0}_{\text{VIII}} | \underbrace{0}_{\text{IX}} | \underbrace{0}_{\text{X}} | \underbrace{0}_{\text{XI}} | \underbrace{0}_{\text{XII}} | \underbrace{0}_{\text{XIII}})^\top$$

In summary, there are 2 response types, 6 missing value schemes, 4 fractions of missing values and 4 correlation strengths, summing up to as much as 192 different simulation settings. Variable importances were recorded by repeating each setting 1000 times. Corresponding R-Code is given in appendix B.

5.2 Real Data

In addition to the extensive simulation study, two well known data sets were used to show the applicability of the new approach in real life situations. Both were chosen to provide a varying number of missing values in

several variables. The total number of variables equals 8 and 9 to allow for an easy and clear comparison of importance measures.

In addition, the widely used complete case analysis – where observations that contain missing values are entirely omitted from the data – and multiple imputation by chained equations (MICE; cf. Van Buuren et al., 2006; White et al., 2011) – which permits the imputation of data with missing values in several variables – were used to enable the application of the original permutation importance measure despite the occurrence of missing values. Finally the ranking of variable importances within each approach were compared and discussed.

The considered data sets are the following:

- The **Pima Indians Diabetes Data Set** was obtained from the open source UCI Machine Learning Repository (Frank and Asuncion, 2010). It contains information about the diabetes disease of 768 pima indian women, which are at least 21 years old. In addition to age, the number of pregnancies, plasma glucose concentration, diastolic blood pressure, triceps skin fold thickness, 2-Hour serum insulin, BMI and diabetes pedigree function were recorded. This makes 8 independent variables used for the classification problem to determine whether a women showed signs of diabetes according to the WHO definition.

  At a first glance this data set does not seem to contain any missing data. However, the missing values are actually "hidden" behind by many zero values that are biologically implausible or impossible. Pearson (2006) calls this situation "disguised missing data" and gives a profound discussion about its occurrence in the Pima Indians Diabetes Data Set. According to his description, there are 5 variables that contain missing data. The total numbers of missing values in these variables are 5, 35, 227, 374 and 11, which equals fractions of 0.7%, 4.6%, 29.6%, 48.7% and 1.4%. Overall 376 (49.0%) of all observations contain at least one missing value. Therefore the complete case analysis can only employ 392 of the 768 available observations.

- The **Mammal Sleep Data** comprises features of 62 species ranging from mice over opposums and elephants to man. It can be obtained from the `R` package `VIM` and was originally used by Allison and Cicchetti (1976) to examine relations between sleep, ecological influences and constitutional characteristics. The observed sleep features include information about duration and depth of sleep phases as well as occurrence of dreams. Constitution is given by measures like body weight and brain weight. The safety of sleep is assessed by scaling for overall danger, danger for being hunted, sleep exposure as well as gestation time etc. One of the main findings in the original paper was a negative correlation between slow-wave sleep and body size. In alignment with these investigations the data containing 9 independent variables was used in a regression analysis for the prediction of body weight. There are 20 (32.3%) observations which are not completely observed for all variables which leaves 42 observations for the complete case analysis. It is interesting to note that Allison and Cicchetti (1976) had originaly chosen a complete case analysis as they found the incomplete data to be "... not suitable for the multivariate analyses ...". There are five variables containing 4 (3 times), 12 and 14 (6.5% (3 times), 19.4% and 22.6%) missing values.

5.3 Implementation

All computations were performed with the `R` system for statistical computing (R Development Core Team, 2010). An implementation of unbiased random forests based on a conditional inference framework is provided by the function `cforest()` which is part of the package `party` (Hothorn et al., 2008). The original permutation importance measure and the new approach were computed by the function `varimp()` which is also included in this package. The settings for the simulation studies were chosen to result in a computation of $ntree = 50$ trees and $maxsurrogate = 3$ surrogate splits in each node. The number of randomly selected variables serving as candidates for splits was set to be $mtry = 8$. Sticking to the default setting $mincriterion = 0$ there were no restrictions concerning the significance of a split. Trees were grown until terminal nodes contained less than $minsplit = 20$ observations while not allowing for splits that lead to less than $minbucket = 7$ observations in a child node. As the number of complete observations becomes extremely low in the additional complete case analysis of the simulation study these parameters were set to $minsplit = 2$ and $minbucket = 1$ in this case. The examination of the two real data sets was based on random forests that consisted of $ntree = 5000$ trees in order to produce stable variable importance rankings. The number of variables chosen for splits was set to $mtry = 3$ considering that the data contained only 8 and 9 variables. The number of surrogate splits and observations that needed to be included in terminal nodes and parent nodes was the same as in the simulation studies. The function `mice()` of the package `mice` (van Buuren and Groothuis-Oudshoorn, 2010, version 2.11) was used to produce five imputed datasets. Normal, logistic and polytomous regression models were applied to impute continuous, binary and polytomous variables, respectively; therefore the corresponding setting of `mice()` was chosen to be defaultMethod = c("norm", "logreg", "polyreg").

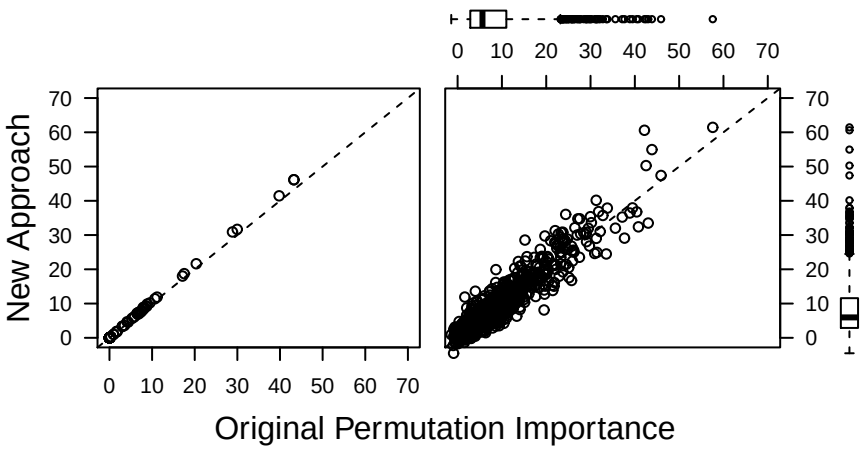


Figure 1: Comparison of the new approach and the original permutation importance. Left: Median importance measures across 1000 simulation runs for all variables and correlations when there is no missing data (m = 0%). Right: Distribution of values observed in 1000 simulation runs for the example of variable 5 (r = 0.6).

Genuer et al. (2008) have conducted elaborate studies to investigate the effect of the number of observations, *ntree* and *mtry* on the computation of importance measures. They found that the stability of estimation improved with a rising amount of observations and trees (*mtry*). However, the rankings of importance measures – which are in main focus of this work – remained almost untouched. This instance is also supported by the fact that simulation studies are repeated 1000 times; aiming at an averaged assessment of rankings. It is a common choice to make *mtry* equal the square root of the number of predictors (cf. Díaz-Uriarte and Alvarez de Andrés, 2006; Chen et al., 2011). Again, Genuer et al. (2008) found this value and even higher values for *mtry* to be convenient for the identification of relevant variables by means of importance measures. Therefore, all of the parameter settings of the simulation studies are in accordance with these considerations.

6 Results

6.1 Simulation

The following investigations are based on the regression analysis in the MAR(rank) scheme. Due to the study design each requirement can be explored by the presentation of results for specific sets of variables. However, it has to be pointed out that non-influential variables are only partly presented as they all gave the same results and did not show any unexpected effects. Thus, variables 16 to 20 are omitted from presentation. A discussion about the reproducibility of findings in the investigated classification problem and further missing data generating processes is given at the end of this section.

Requirement (R1) is satisfied for all of the investigated variables and correlation strength (cf. Figure 1). The newly suggested approach and the original permutation importance measure even approximately equal each other when there are no missing values (m = 0%). Deviations of single assessments are due to the computation inherent variability of importance measures. Therefore, results are also presented as median importance across 1000 simulation runs to stress the average equality.

Requirement (R2) is met as the importance of variables decreases the more missing values they contain (cf. Figure 2). This holds for all variables and correlation strength (cf. Figure 9 in appendix A).

Requirement (R3) holds as correlations with influential variables induce higher importances (cf. Figure 3). This is true for all variables and fractions of missing values (cf. Figure 10 in appendix A; A comparison of Blocks I and II shows that block size is another factor that affects variable importance. However, non-influential variables – given in Block IV – do not contribute to this effect.).

The effects of correlation and missing values appear to be interacting (cf. Block I in Figure 4): Although all variable importances rise with a rising strength of correlation, the importance of variable 2 drops in relation to the variables of the same block when the amount of missing values increases. An investigation of selection frequencies – i.e. the number of times a variable is chosen for splits in a Forest (displayed as horizontal lines) – reveals that it is replaced by other variables in the tree building process. This effect follows a simple rule: the more similar the information of variables becomes due to an increased correlation and the more information a variable is lacking because of missing values the more often it will be replaced by others.

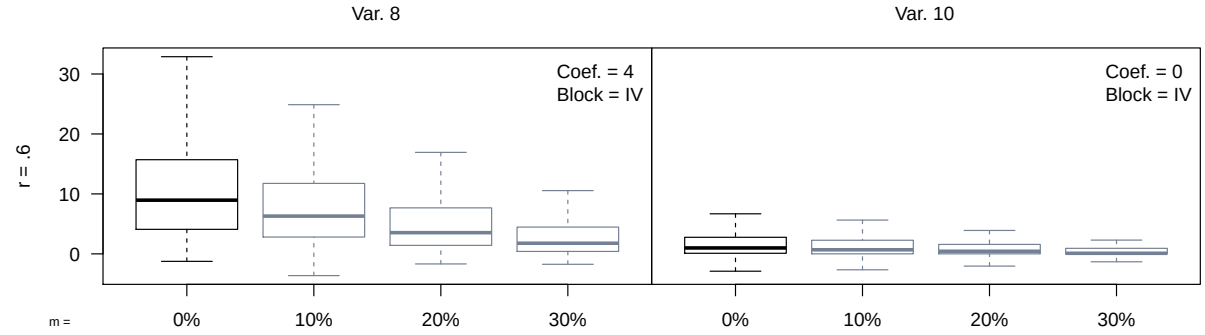


Figure 2: Variable importance of variables 8 and 10 with correlation r = .6 and m = 0%, 10%, 20%, 30% missing values. Boxplots of variables with missing values are colored grey. Outliers are omitted from illustration for clarity.

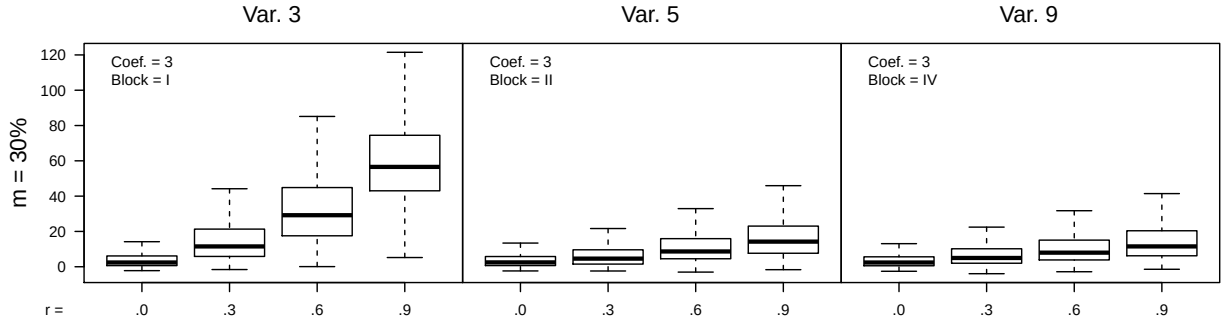


Figure 3: Variable importances of variables 3, 5 and 9 with correlations r = 0, .3, .6, .9. Outliers are omitted from illustration for clarity.

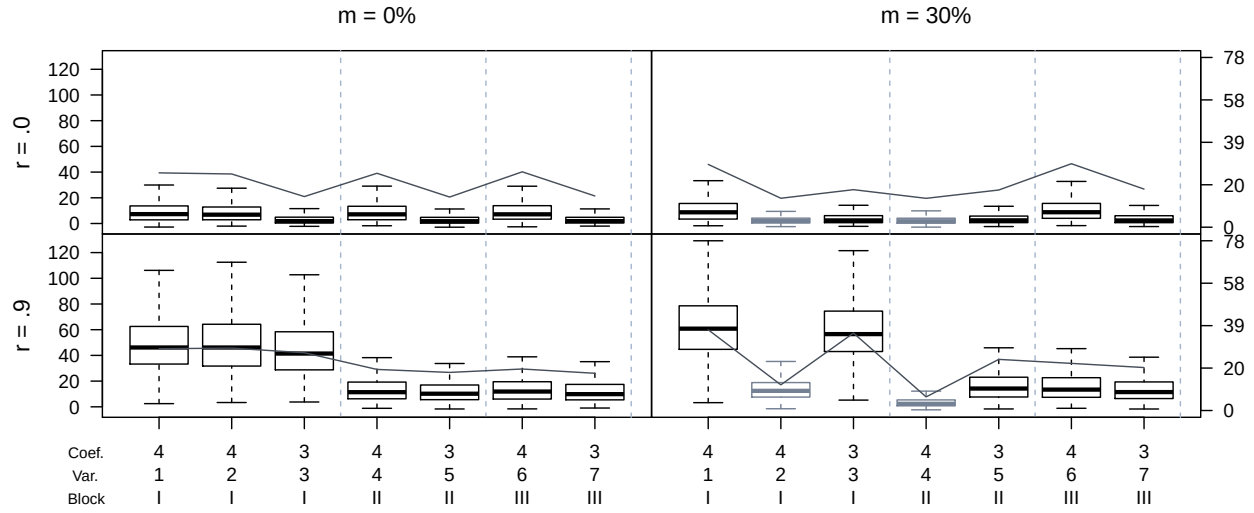


Figure 4: Variable importances (left axis) of variables 1-7 (Block I, II, III) and correlations r = 0, .9 for fractions of missing values m = 0%, 30%. Boxplots of variables that contain missing values are colored grey. Horizontal lines indicate selection frequencies (right axis). Vertical dashed lines indicate correspondance to the same block. Outliers are omitted from illustration for clarity.

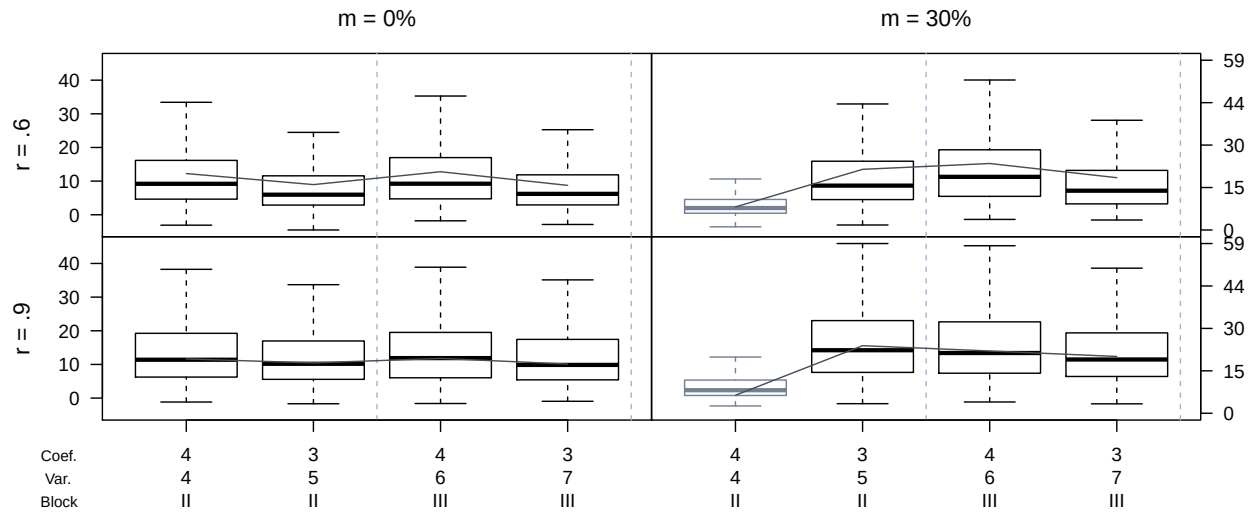


Figure 5: Variable importances (left axis) of variables 4-7 (Blocks II, III) and correlations r = .6, .9 for fraction of missing values m = 0%, 10%, 20%, 30%. Boxplots of variables that contain missing values are colored grey. Horizontal lines indicate selection frequencies (right axis). Vertical dashed lines indicate correspondance to the same block. Outliers are omitted from illustration for clarity.

Requirement (R4) is satisfied as the ranking of fully observed variables from the same block stays unaffected by the amount of missing values in other variables (cf. Figure 4). Note that between blocks the variable rankings may change (cf. variables 5 and 7). The importance of variable 5 increases as it is able to replace variable 4 that contains missing values. It rises above variable 7 with the same (and for strong correlations and many missing values even above variable 6 with a higher) influence on the response. Another question emerging from the fact that variables may replace others in a tree is if this also holds for isolated blocks that are not correlated with any variables that contain missing values. Figure 5 shows that this is almost not the case as selection frequencies and variable importances stay on a certain level (cf. Block II and III). This finding even partly extends (R4) which demands stable rankings for fully observed variables only within blocks, not across blocks.

Requirement (R5) is met as the importance of influential variables is higher than for non-influential variables (cf. Figure 6). This holds for variables with and without missing values – but not necessarily for comparisons between the two cases. Importances of influential variables may drop below those of non-influential ones if the former contain missing values and the latter are part of a correlated block with influential variables. An example is given by Block IV: Variable 8 shows a higher importance than variable 10 (both containing missing values) and variable 9 shows a higher importance than variable 11 (both without missing values). However, the importance of the influential variable 8 drops below that of the non-influential variable 11, as the former contains missing values and the latter is correlated to variable 9. The importance of variable 11 even rises above that of influential variables contained in other blocks (e.g. variable 13). However, the lowest importance should always be assigned to non-influential variables that are uncorrelated to any other influential variables. This claim is approved by the examples of variable 14 and 15.

In conclusion, factors like the occurrence of missing values, the number and influence of correlated variables as well as the correlation strength, can positively affect the importance of variables. However, these are properties to be expected from a marginal variable importance measure when dealing with variables that lose information due to missing values, yet are correlated to other variables that can "take over for them".

Results for the entire simulation setting show the same properties as for the specific cases investigated above (cf. Figure 11 in appendix A for a broad overview). An additional examination of all missing data generating processes (MCAR, MAR(rank), MAR(median), MAR(upper), MAR(margins) and MNAR(upper)) demonstrates that all findings of the previous analyses can be retraced in each scheme (cf. Figure 12 in appendix A). Therefore, the proposed approach appears to be applicable to a wide range of MCAR, MAR and MNAR settings. In addition, results for the classification problem show the same properties and are not displayed or discussed to omit redundancy.

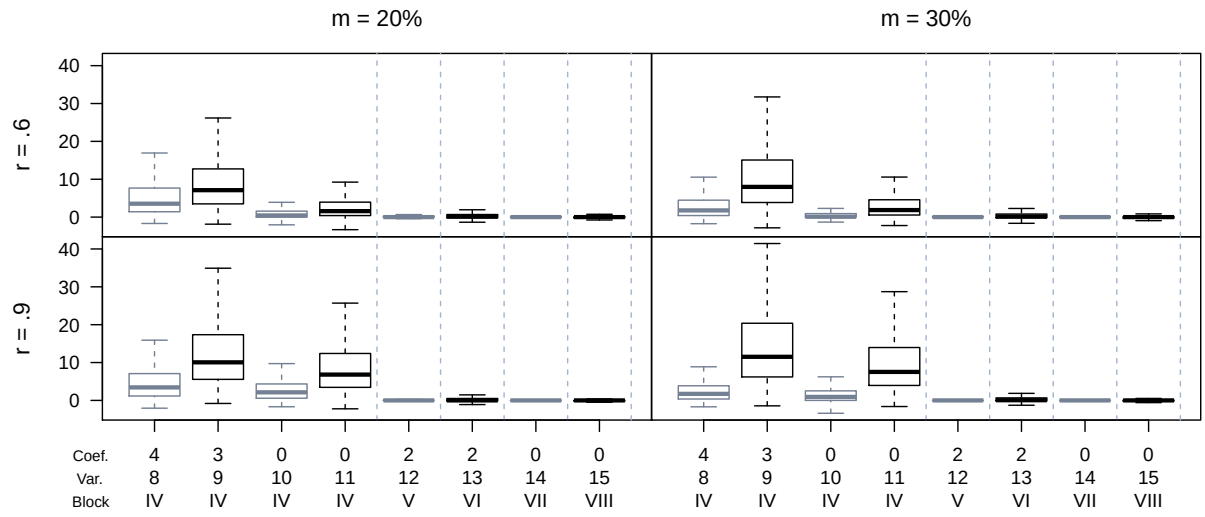


Figure 6: Variable importances of Block IV-VIII and correlations r = .6, .9 for fraction of missing values m = 20%, 30%. Boxplots of variables that contain missing values are colored grey. Vertical dashed lines indicate correspondance to the same block. Outliers are omitted from illustration for clarity.

6.2 Real Data Application

Figure 7 shows importance measures computed by the new approach for the two real data sets. As an alternative, the original permutation importance measure was computed for multiple imputed data and in a complete case analysis.

In the Pima Indians Diabetes Data the ranking of predictor variables shows obvious differences between methods. These become evident by the example of the variables BMI, number of pregnancies, diabetes pedigree function, age and 2-Hour serum insulin. The strongest and weakest variables, however, plasma glucose concentration, diastolic blood pressure and triceps skin fold thickness are ranked equally. Similar findings can be observed for the Mammal Sleep Data. The variables slow wave sleep ('NonD' = 'nondreaming'), dreaming sleep, maximum life span and the overall danger index are ranked differently. Even if the ranking of some variables is the same between methods, like for the sleep exposure index ('Exp'), there may still be substantial differences in the magnitude of importances assigned to them. The results also show that differences do not directly (or solely) depend on whether or not a variable contains missing values. In addition, there is some diversity in the ranking of variables between each of the methods.

A plausible reason for these differences is that complete case analysis can induce a bias when observations are not MCAR. This is well-known – yet complete case analysis is still frequently applied in practice. It can modify the entire importance ranking just because information is omitted when observations are excluded from the analysis. By contrast, multiple imputation tries to restore the information that would be provided by the complete dataset and therefore affects the computation of importance measures, too. A much more detailed investigation of this issue has been published by Hapfelmeier et al. (2012).

The random allocation of observations, instead of a random permutation of a variables values, is the key element in the computation of the new importance measure. Despite the comprehensible character of this approach one might fear an increased computational complexity. For this reason we examined the computational costs by means of CPU time used in the application studies. For a fair comparison the original permutation importance measure and the new approach have been applied to data of equal size which is given by the imputed data and the original datasets. Computations have been performed on a Pentium(R) Dual-Core CPU E5200 processor with 2.50 GHz, 3 GB RAM and a Microsoft windows XP 32 bit system. The results given by Table 2 indicate that the new approach was computed even faster than the original one (both importance measures are implemented by the `R`-function `varimp()` which is part of the package `party`).

To illustrate one possible scenario that can lead to a change in the variable ranking when complete case analysis or multiple imputation are applied to data that is not MCAR, another small simulation was conducted. Given

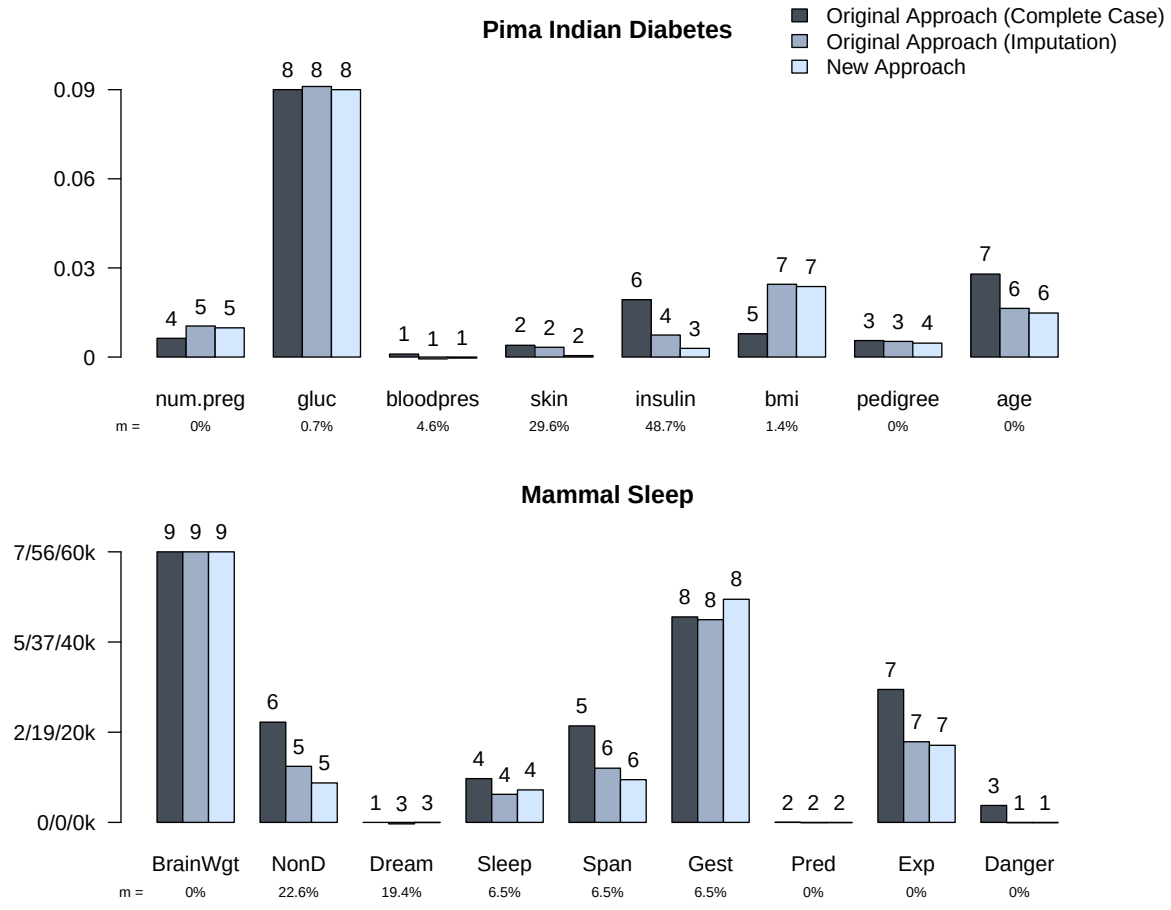
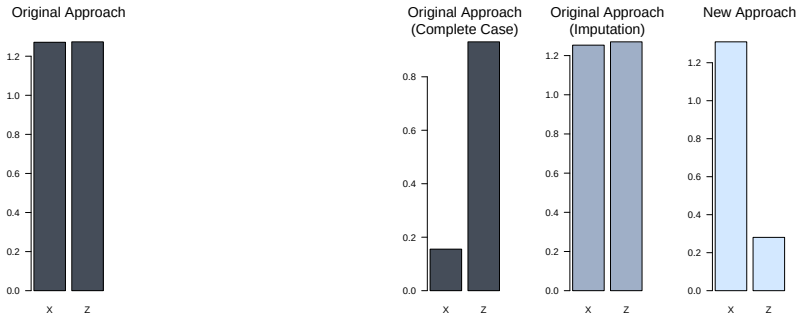


Figure 7: Variable Importances for the Pima Indians and the Mammal Sleep Data. Bars correspond to the original permutation importance measure – computed via complete case analysis and multiple imputation – and the new approach. Figures above bars indicate ranks within methods. The importance of variables decreases from the highest to the lowest ranks. The fraction of missing values per variable is given by m.

Table 2: Time elapsed for the computation of the new importance measure and the original permutation importance measure in the data application studies.

Data	new approach	orig. approach
Pima Indians	134.3 ± 0.9	144.5 ± 0.6
Mammal sleep	1.58 ± 0.02	6.95 ± 0.05

Mean ± SD in sec.



(a) No missing values

(b) 30% missing values in Z

Figure 8: Variable importances of X and Z computed by the original permutation importance measure and the new approach. Case (a) bases on the entire data without any missing values. In case (b) 30% of variable Z are set missing. A complete case analysis and multiple imputation is used to compute the original permutation importance measure.

a pair of binary variables (X, Z) the response Y follows the distribution:

$$Y \sim \begin{cases} N(2,1) & \text{if } (x,z) = (1,0) \\ N(0,1) & \text{if } (x,z) = (0,0) \text{ or } (x,z) = (1,1) \\ N(-2,1) & \text{if } (x,z) = (0,1) \end{cases}$$

The relative frequencies of class 0 and class 1 in X and Z are 80% and 20%, respectively. They are not correlated. Missing values are induced into Z dependent on the highest values of Y which resembles the MAR(upper) scheme. To produce stable results the simulation bases on 5,000 observations and random forests growing 5,000 trees. According to our expectation Figure 8a displays the same importance for both variables when there are no missing values in the data. In Figure 8b a fraction of 30% of Z is set missing. The application of multiple imputation makes the importances stay on an even level. The new approach assigns a reduced importance to Z while X remains of high relevance. This finding also meets our expectations for data that contains variables of reduced information. At this point it has to be emphasized that neither of the results obtained for the new approach and the original one used with multiple imputation are more accurate or correct. They simply differ in the research goal they are best suited for: the new approach takes the occurrence of missing values into account and therefore reflects the actual information of the data. By contrast, multiple imputation seems to be a convenient means to assess the importance of variables that could be observed if the data was complete. In a complete case analysis however, X suffers the loss of its explanatory power although it does not contain any missing values at all. It is not even correlated to Z. The explanation of this effect is quite simple: the highest values of Y which cause the missing values in Z are most frequently related to $x = 1$. Deleting these observations in a complete case analysis makes X mainly consist of class 0. As a consequence it loses its discriminatory power. This example demonstrates how a complete case analysis can distort the ranking of variable importances when the missingnes scheme is not MCAR.

7 Discussion and conclusion

In summary, our simulation results have shown that all requirements that were previously formulated were fulfilled by the newly suggested importance measure for different types of MCAR, MAR and MNAR missing data. Most importantly: In the absence of missing values, both the original permutation importance measure and the newly suggested approach produce similar results. The importance of variables containing missing

values does not artificially increase but decreases with the number of missing values and the respective decrease of information. Moreover, in the presence of correlation the measure shows all properties that are to be expected from a marginal variable importance measure.

A particularly interesting effect is that, with regard to the variable selection frequencies, variables with increasing numbers of missing values are increasingly replaced by fully observed variables that are correlated with them: the complete variables "take over" for those with missing values within a group of correlated ones. In this sense, the effects of correlation and missing values are interacting. This is an intuitive property, since both affect the amount of information a variable retains. What is important to note here is that, besides effects of the correlation on the permutation importance, that were already pointed out by Strobl et al. (2008), in the presence of missing values the correlation is also linked to the quality of surrogate variables.

The exact role that surrogate variables play for the variable importance is still ambiguous: On one hand they help to reconstitute missing information, but on the other hand they also compete for the selection in the tree. However, the selection frequencies displayed in our results indicate that the latter effect is of more relevance.

Besides the findings for the simulation analysis the new approach also appeared well suited to deal with missing values in real data: There were some profound differences between the variable ranking suggested by the new approach and a complete case analysis. As the latter is known to produce biased results in many situations (cf., e.g., Janssen et al., 2009, 2010) this strongly indicates that the omission of observations with missing values has induced artifacts because the values were not missing at random. It also has to be pointed out that the rationale of our approach is not to undo the influence missing values have on the information carried by a variable – as it is for multiple imputation – but to reflect the remaining information that the variable has with the respective values missing. Results of corresponding simulation studies support this claim.

The advantage of the new approach proposed in this work is that it incorporates the full information provided by the data set. Moreover, it utilizes one of the most appreciated properties of recursive partitioning methods, namely their ability to deal with missing values by means of surrogate variables. Accordingly, the resulting importance rankings depend not only on the amount of missing values but also on the quality and availability of surrogate variables.

References

Allison, T. and D. V. Cicchetti (1976). Sleep in mammals: ecological and constitutional correlates. *Science 194*(4266), 732–734.

Altmann, A., L. Tolosi, O. Sander, and T. Lengauer (2010). Permutation importance: a corrected feature importance measure. *Bioinformatics 26*(10), 1340–1347.

Archer, K. and R. Kimes (2008). Empirical characterization of random forest variable importance measures. *Computational Statistics & Data Analysis 52*(4), 2249–2260.

Biau, G., L. Devroye, and G. Lugosi (2008). Consistency of Random Forests and Other Averaging Classifiers. *Journal of Machine Learning Research 9*, 2015–2033.

Boulesteix, A.-L., C. Strobl, T. Augustin, and M. Daumer (2008). Evaluating microarray-based classifiers: An overview. *Cancer Informatics 6*, 77–97.

Breiman, L. (1996). Bagging predictors. *Machine Learning 24*(2), 123–140.

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

Breiman, L. and A. Cutler (2008). *Random forests*. `http://www.stat.berkeley.edu/users/breiman/RandomForests/cc_home.htm`. (accessed 03.02.2011).

Breiman, L., J. Friedman, C. J. Stone, and R. A. Olshen (1984). *Classification and Regression Trees*. Chapman & Hall/CRC.

Chen, X., M. Wang, and H. Zhang (2011). The use of classification trees for bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1*(1), 55–63.

Cutler, D. R., T. C. Edwards, K. H. Beard, A. Cutler, K. T. Hess, J. Gibson, and J. J. Lawler (2007). Random forests for classification in ecology. *Ecology 88*(11), 2783–2792.

Díaz-Uriarte, R. and S. Alvarez de Andrés (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinformatics 7*(1), 3.

Dobra, A. and J. Gehrke (2001). Bias correction in classification tree construction. In C. E. Brodley and A. P. Danyluk (Eds.), *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA*, pp. 90–97. Morgan Kaufmann.

Frank, A. and A. Asuncion (2010). UCI machine learning repository.

Genuer, R. (2010). Risk bounds for purely uniformly random forests. Rapport de recherche RR-7318, INRIA.

Genuer, R., J.-M. Poggi, and C. Tuleau (2008). Random Forests: some methodological insights. Rapport de recherche RR-6729, INRIA.

Hapfelmeier, A., T. Hothorn, and K. Ulm (2012). Random forest variable importance with missing data.

Hapfelmeier, A., T. Hothorn, K. Ulm, and C. Strobl (2012). A new variable importance measure for random forests with missing data. *Statistics and Computing*, 1–14.

Hastie, T., R. Tibshirani, and J. H. Friedman (2009). *The Elements of Statistical learning* (Corrected ed.). Springer.

Hothorn, T., K. Hornik, C. Strobl, and A. Zeileis (2008). *party: A laboratory for recursive part(y)itioning.* R package version 0.9-9993.

Hothorn, T., K. Hornik, and A. Zeileis (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics 15*(3), 651–674.

Janssen, K. J., A. R. Donders, F. E. Harrell, Y. Vergouwe, Q. Chen, D. E. Grobbee, and K. G. Moons (2010). Missing covariate data in medical research: to impute is better than to ignore. *Journal of clinical epidemiology 63*(7), 721–727.

Janssen, K. J., Y. Vergouwe, A. R. Donders, F. E. Harrell, Q. Chen, D. E. Grobbee, and K. G. Moons (2009). Dealing with missing predictor values when applying clinical prediction models. *Clinical chemistry 55*(5), 994–1001.

Kim, H. and W. Loh (2001). Classification trees with unbiased multiway splits. *Journal of the American Statistical Association 96*, 589–604.

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

Little, R. J. A. and D. B. Rubin (2002). *Statistical Analysis with Missing Data, Second Edition* (2 ed.). Wiley-Interscience.

Lunetta, K., B. L. Hayward, J. Segal, and P. Van Eerdewegh (2004). Screening large-scale association study data: exploiting interactions using random forests. *BMC Genetics 5*(1).

Nicodemus, K. (2011). Letter to the editor: On the stability and ranking of predictors from random forest variable importance measures. *Briefings in Bioinformatics*.

Nicodemus, K., J. Malley, C. Strobl, and A. Ziegler (2010). The behaviour of random forest permutation-based variable importance measures under predictor correlation. *BMC Bioinformatics 11*(1), 110.

Pearson, R. K. (2006). The problem of disguised missing data. *SIGKDD Explor. Newsl. 8*(1), 83–92.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)* (1 ed.). Morgan Kaufmann.

R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Rieger, A., T. Hothorn, and C. Strobl (2010). Random forests with missing values in the covariates.

Rodenburg, W., A. G. Heidema, J. M. A. Boer, I. M. J. Bovee-Oudenhoven, E. J. M. Feskens, E. C. M. Mariman, and J. Keijer (2008). A framework to identify physiological responses in microarray-based gene expression studies: selection and interpretation of biologically relevant genes. *Physiological Genomics 33*(1), 78–90.

Rubin, D. B. (1976). Inference and missing data. *Biometrika 63*(3), 581–592.

Rubin, D. B. (1987). *Multiple Imputation for Nonresponse in Surveys.* J. Wiley & Sons, New York.

Sandri, M. and P. Zuccolotto (2006). Variable selection using random forests. In S. Zani, A. Cerioli, M. Riani, and M. Vichi (Eds.), *Data Analysis, Classification and the Forward Search*, Studies in Classification, Data Analysis, and Knowledge Organization, pp. 263–270. Springer Berlin Heidelberg. 10.1007/3-540-35978-8_30.

Schafer, J. L. and J. W. Graham (2002). Missing data: our view of the state of the art. *Psychol Methods 7*(2), 147–177.

Strobl, C., A.-L. Boulesteix, and T. Augustin (2007). Unbiased split selection for classification trees based on the gini index. *Computational Statistics & Data Analysis 52*(1), 483–501.

Strobl, C., A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis (2008). Conditional variable importance for random forests. *BMC Bioinformatics 9*(1), 307.

Strobl, C., A.-L. Boulesteix, A. Zeileis, and T. Hothorn (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics 8*(1), 25.

Strobl, C., J. Malley, and G. Tutz (2009). An introduction to recursive partitioning: Rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological Methods 14*(4), 323–348.

Tang, R., J. Sinnwell, J. Li, D. Rider, M. de Andrade, and J. Biernacka (2009). Identification of genes and haplotypes that predict rheumatoid arthritis using random forests. *BMC Proceedings 3*(Suppl 7), S68.

Van Buuren, S., J. P. L. Brand, C. G. M. Groothuis-Oudshoorn, and D. B. Rubin (2006). Fully conditional specification in multivariate imputation. *Journal of Statistical Computation and Simulation 76*(12), 1049–1064.

van Buuren, S. and K. Groothuis-Oudshoorn (2010). Mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software in press*, 01–68.

Wang, M., X. Chen, and H. Zhang (2010). Maximal conditional chi-square importance in random forests. *Bioinformatics 26*(6), 831–837.

White, A. and W. Liu (1994). Bias in information based measures in decision tree induction. *Machine Learning 15*(3), 321–329.

White, I. R., P. Royston, and A. M. Wood (2011). Multiple imputation using chained equations: Issues and guidance for practice. *Statistics in Medicine 30*(4), 377–399.

Yang, W. W. W. and C. C. Gu (2009). Selection of important variables by statistical learning in genome-wide association analysis. *BMC proceedings 3*(7).

Yu, X., J. Hyyppä, M. Vastaranta, M. Holopainen, and R. Viitala (2011). Predicting individual tree attributes from airborne laser point clouds based on the random forests technique. *ISPRS Journal of Photogrammetry and Remote Sensing 66*(1), 28 – 37.

Zhou, Q., W. Hong, L. Luo, and F. Yang (2010). Gene selection using random forest and proximity differences criterion on dna microarray data. *JCIT 5*(6), 161–170.

A Supplementary Material

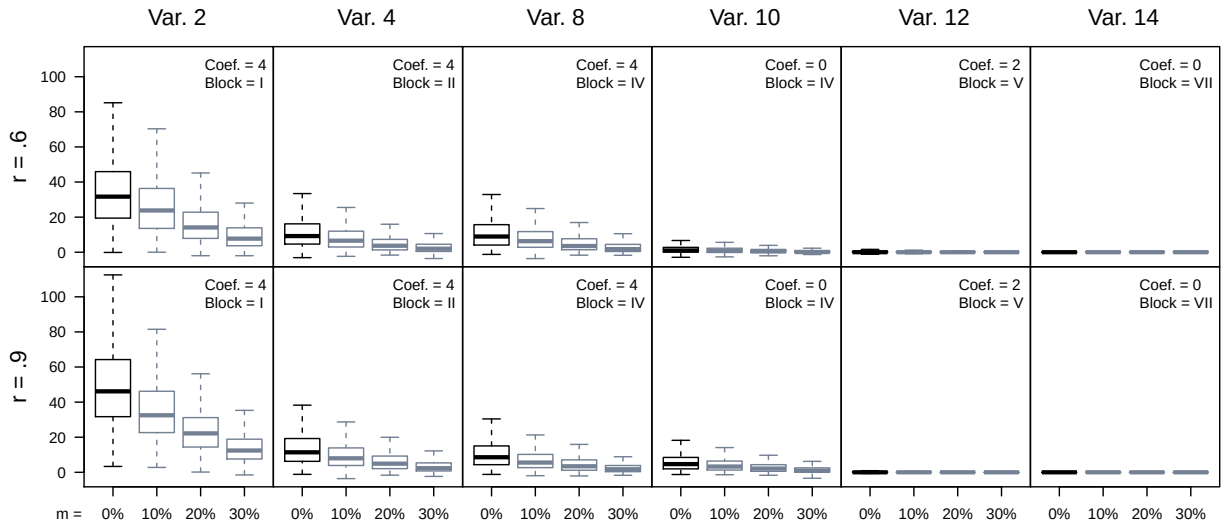


Figure 9: Variable importance of variables 2, 4, 8, 10, 12, 14 – that contain missing values when m > 0% – for correlations r = .6, .9 and fractions of missing values m = 0%, 10%, 20%, 30%. Boxplots of variables that contain missing values are colored grey. Outliers are omitted from illustration for clarity.

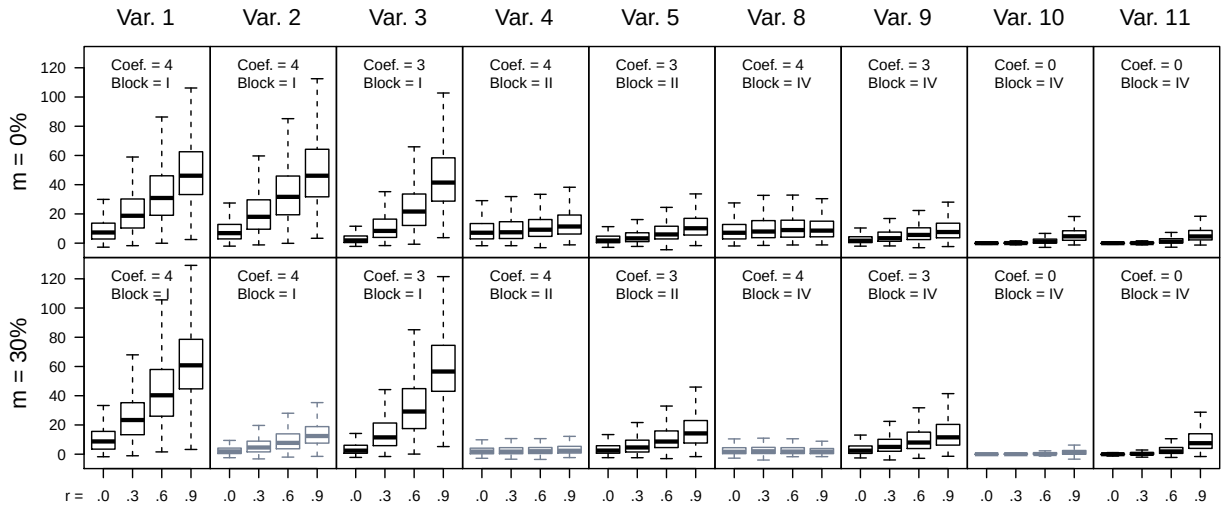


Figure 10: Variable importances of variables 1-5, 8-11 (Blocks I, II, IV) and correlations r = 0, .3, .6, .9 for fractions of missing values m = 0%, 30%. Boxplots of variables that contain missing values are colored grey. Outliers are omitted from illustration for clarity.

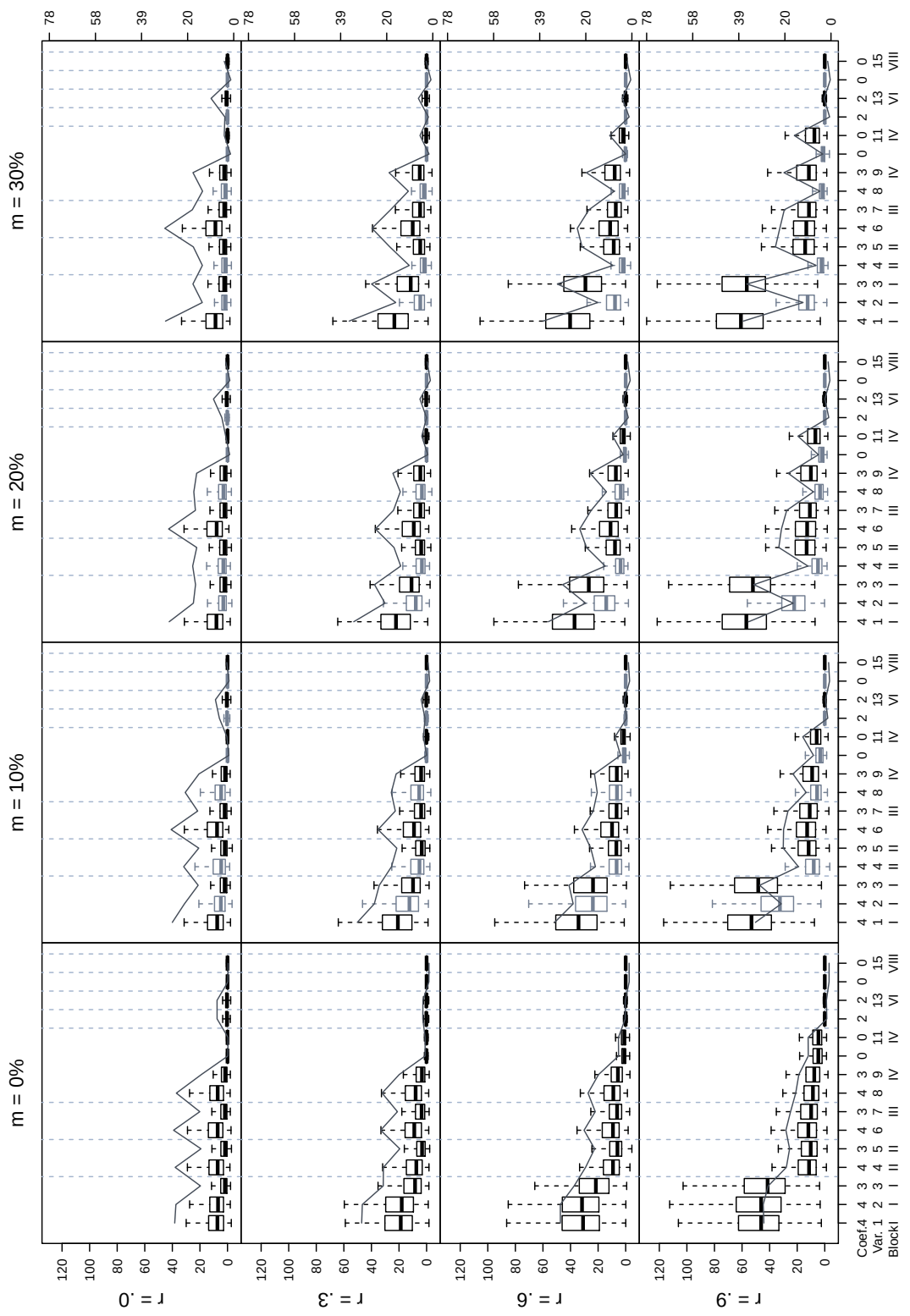


Figure 11: Variable importances (left axis) of Block I-VIII and correlations r = 0, .3, .6, .9 for fraction of missing values m = 0%, 10%, 20%, 30% in the MAR(rank) setting and regression problem. Boxplots of variables that contain missing values are colored grey. Horizontal lines indicate selection frequencies (right axis). Vertical dashed lines indicate correspondance to the same block. Outliers are omitted from illustration for clarity.

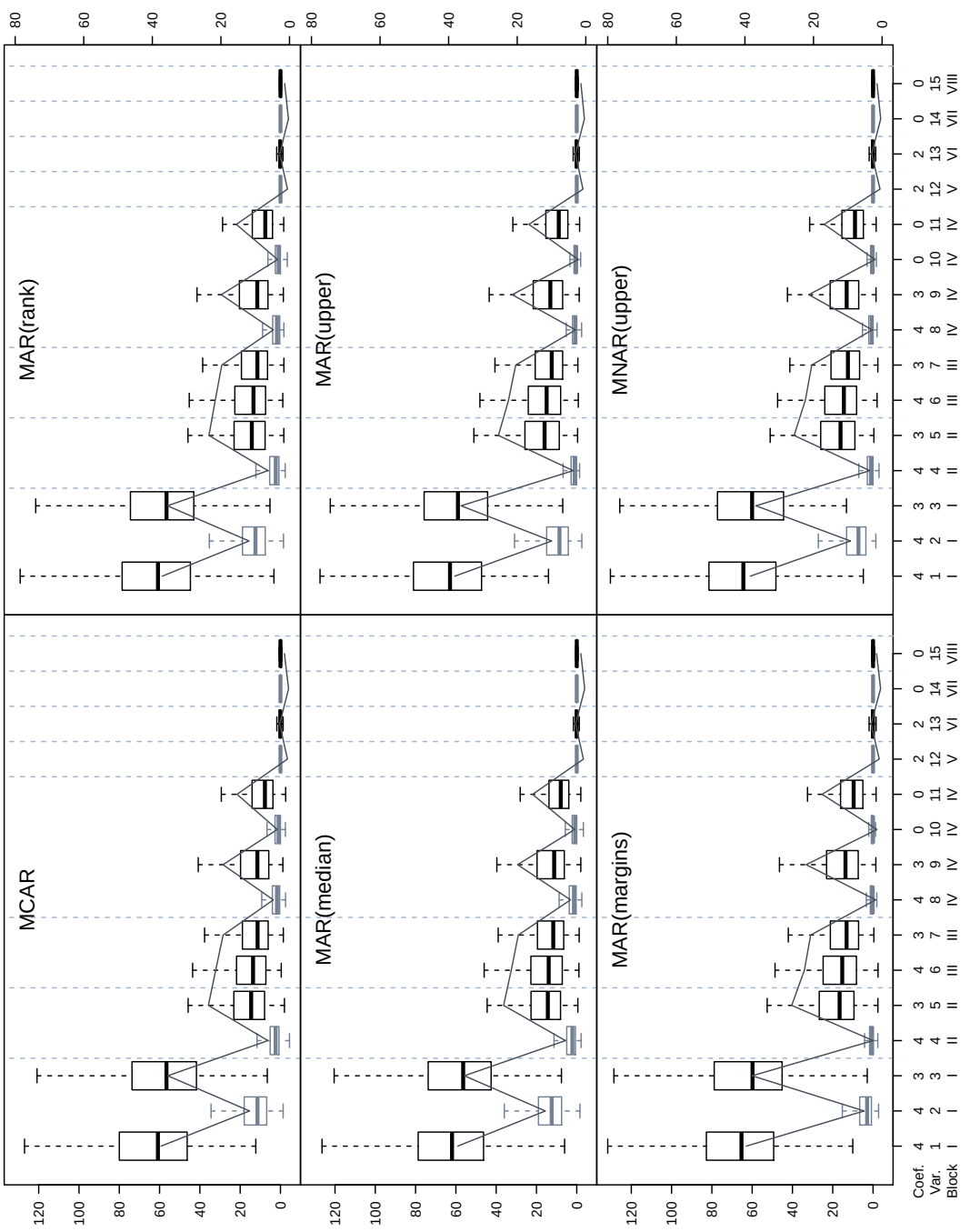


Figure 12: Variable importances (left axis) of variables 1-15 for a correlation of r = .6 and a fraction of missing values m = 20% in the regression analysis. Boxplots of variables that contain missing values are colored grey. Horizontal lines indicate selection frequencies (right axis). Vertical dashed lines indicate correspondance to the same block. Outliers are omitted from illustration for clarity.

B R-Code

```
# load required packages
library("party"); attach(asNamespace("party")) # version 1.0-0
library(mvtnorm) # version 0.9-9992
library(mice) # version 2.11

# Function to count selection frequencies of variables in Random Forests
count <- function(forest, inames = NULL) {
  # forest: object of class "RandomForest" created by the function cforest()
  # inames: names of variables to be assessed (defaults to NULL, using all
  # variables in a forest)
  if (is.null(inames) && extends(class(forest), "RandomForest"))
  inames <- names(forest@data@get("input"))
  resultvec <- rep(0, length(inames))

  myfunc <- function(x, inames, resultvec) {
    names(x) <- c("nodeID", "weights", "criterion", "terminal",
        "psplit", "ssplits", "prediction", "left", "right")
    names(x$criterion) <- c("statistic", "criterion", "maxcriterion")
    if (!x$terminal) {
     resultvec[x$psplit[[1]]] <- resultvec[x$psplit[[1]]] + 1
     resultvec <- myfunc(x$left,  inames = inames, resultvec = resultvec)
     resultvec <- myfunc(x$right, inames = inames, resultvec = resultvec)
     }
    return(resultvec)
    }
  for (k in 1:length(forest@ensemble)) {
      resultvec <- myfunc(forest@ensemble[[k]], inames, resultvec)
    }
names(resultvec) <- inames
return(resultvec)
}
environment(count) <- environment(varimp)


# create a list of covariance matrices for each correlation strength
sig <- lapply(1:4, function(x) {r <- c(0,.3,.6,.9)[x]; y <- diag(20);
              y[1:3, 1:3] <- r; y[4:5, 4:5] <- r; y[6:7, 6:7] <- r;
              y[8:11, 8:11] <- r; diag(y) <- 1; return(y)})

# create lists that contain arrays used to collect the results of 20 variables
# for 4 fractions of missing values, 4 correlation strength and
# 6 missing data generating processes in 1000 simulation runs
# The common importance measure (‘old’), the new approach (‘new’) and
# selection frequencies (‘count’) are recorded for the regression (‘reg’) and
# classification (‘clas’) problem.
reg.old <- clas.old <- reg.new <- clas.new <- reg.count <- clas.count <-
lapply(1:6, function(y) lapply(1:4, function(x) array(dim = c(1000, 20, 4))))

set.seed(1234) # set a random seed for reproducibility of results

# 1000 simulation runs start here
for (i in 1:1000) {
  for (r in 1:4) { # there are 4 correlation strength
    dat <- as.data.frame(rmvnorm(100, sigma = sig[[r]])) # create the data
    x.beta <- 4 * dat$V1 + 4 * dat$V2 + 3 * dat$V3 + 4 * dat$V4 + 3 * dat$V5 +
              4 * dat$V6 + 3 * dat$V7 + 4 * dat$V8 + 3 * dat$V9 + 2 * dat$V12 +
              2 * dat$V13
    dat$y.reg  <- x.beta + rnorm(100, 0, .5)
    dat$y.clas <- rbinom(100, 1, exp(x.beta) / (1 + exp(x.beta)))

    for (m in 1:4) { # there are 4 fractions of missing values
```

```
      # the data is replicated for each of 6 missing data generating process
      dat.mis <- lapply(1:6, function(x) dat)
      if (m != 1) {
      for (k in c("V2", "V4", "V8", "V10", "V12", "V14")) {
      ind <- switch(k, "V2"  =  "V3",  "V4" =  "V5",  "V8" =  "V9",
                       "V10" = "V11", "V12" = "V13", "V14" = "V15")
      # induce missing values MCAR
      is.na(dat.mis[[1]][,k])[sample(1:100, (m - 1) * .1 * 100)] <- TRUE
      # induce missing values MAR(rank)
      is.na(dat.mis[[2]][,k])[sample(1:100, (m - 1) * .1 * 100,
                              prob = rank(dat.mis[[2]][,ind]) / 5050)] <- TRUE
      # induce missing values MAR(median)
      is.na(dat.mis[[3]][,k])[sample(1:100, (m - 1) * .1 * 100,
                              prob = ifelse(dat.mis[[3]][,ind] >=
                              median(dat.mis[[3]][,ind]), .9, .1))] <- TRUE
      # induce missing values MAR(upper)
      is.na(dat.mis[[4]][,k])[dat.mis[[4]][,ind] >=
                    sort(dat.mis[[4]][,ind], T)[(m - 1) * .1 * 100]] <- TRUE
      # induce missing values MAR(margins)
      is.na(dat.mis[[5]][,k])[dat.mis[[5]][,ind] >=
                    sort(dat.mis[[5]][,ind], T)[(m - 1) * .1 * 100 / 2] |
                              dat.mis[[5]][,ind] <=
                    sort(dat.mis[[5]][,ind])[(m - 1) * .1 * 100 / 2]] <- TRUE
      # induce missing values MNAR(upper)
      is.na(dat.mis[[6]][,k])[dat.mis[[6]][,k] >=
                    sort(dat.mis[[6]][,k], T)[(m - 1) * .1 * 100]] <- TRUE
      }}
      for (j in 1:6) { # compute results for 6 missing data generating processes
      # create random forests and compute importances and selection frequencies
      forest.reg  <- cforest(as.formula(paste("y.reg",  paste("V", 1:20, sep = "",
                             collapse = " + "), sep = " ~ ")), data = dat.mis[[j]],
                             controls = cforest_unbiased(mtry = 8, ntree = 50,
                             maxsurrogate = 3))
      forest.clas <- cforest(as.formula(paste("y.clas", paste("V", 1:20, sep = "",
                             collapse = " + "), sep = " ~ ")), data = dat.mis[[j]],
                             controls = cforest_unbiased(mtry = 8, ntree = 50,
                             maxsurrogate = 3))

      reg.new[[j]][[r]][i, , m]    <- varimp(forest.reg)
      clas.new[[j]][[r]][i, , m]   <- varimp(forest.clas)
      reg.count[[j]][[r]][i, , m]  <- count(forest.reg)
      clas.count[[j]][[r]][i, , m] <- count(forest.clas)
      if (m == 1) {
      reg.old[[j]][[r]][i, , m]  <- varimp(forest.reg,  pre1.0_0 = TRUE)
      clas.old[[j]][[r]][i, , m] <- varimp(forest.clas, pre1.0_0 = TRUE)
      }
      }
      }
    }
  }
```