Detection of missing data mechanisms with graphical models and their imputation A simulation study

Master Thesis

Department of Statistics

Ludwig-Maximilians-Universität München

Eleftheria Papavasiliou

Supervisor Prof. Dr. Christian Heumann

August 2023

Abstract

Knowing the missing mechanisms, called MCAR, MAR and MNAR, in a given dataset is central to handling missing data, minimizing biases that can arise due to missingness and also adapt the imputation method. To detect these mechanisms a graphical representation of these will be first theoretically introduced and tested later on in a simulation study. It will be shown that graphical models provide a good tool for comprehending, encoding, and communicating assumptions about the missingness process. To improve the traditional graphical models which assume the data to be fully observed an alternative algorithm, named mvpc-algorithm will be introduced, based on the theory of Mohan et. al. (19)(29) and tested. The results proved also in practice that the algorithm works better than the traditional pc-algorithm. Nevertheless also the traditional approach yielded satisfying results and has the big advantage that it is easy to use. Based on the detection of the missingness mechanisms, the imputation should be adapted for MNAR variables and validated afterwards. Whereas for MCAR/MAR variables multiple imputation with Amelia, which is an expectation-maximization with bootstrapping algorithm that works faster and easier but at the same time provides equally good results as various Markov chain Monte Carlo approaches, was used. For the MNAR variables a weighted knn imputation was carried out. The results in the simulation study showed that, although MNAR imputations are biased when using traditional multiple imputation methods which include model assumptions that are violated through this missing mechanism, the imputations are quite similar to those of the MCAR/MAR variables when using Amelia. On the other side the imputations suffered from changing the method to the weighted knn imputation. Even if this algorithm should provide in theory better results it can not be recommended to use as an alternative to Amelia based on the results of this simulation study.

Contents

List of Tables

List of Figures

1 Introduction

There are many possibilities to analyse a dataset. Starting with an innumerable number of exploratory data analysis tools as plotting kernel density estimations and box plots or calculating standard deviations and quantiles and moving on with an even larger amount of modelling techniques as fitting a linear regression or training a neural model. All these procedures have something in common: They take the available data and try to make conclusions from that for a larger population. Thereby it is often forgotten that not only available data is containing important information about the population but also the data which is not available, hence missing. Ignoring this fact might lead to false interpretations.

Is it enough to analyse the available data from an questionnaire to detect if the problem of obesity is increasing in our community? If we assume that every missing answer to the question about your weight is due to overlooking this question the answer would be *yes*, assuming that the available amount of data is still large enough to calculate reliable averages. If we assume on the other hand that younger persons feel in general more uncomfortable to answer questions about their weight the answer would be *no* since we are averaging then only over the weights of older persons and missing these of the younger ones. If we assume that people with obesity feel bad about their weight and therefore reject to answer this question about their weight we should answer our question again with *no*.

This is only one of many examples why it is important to analyse not only the available data but also the missing data. But how should we analyse something if this something is missing? Answering this satisfactorily is difficult. There are different approaches which cover different parts of the question. Splitting this general question into smaller questions is a first step to dive into the topic of missing values.

As you may have noticed in the provided example there are three different types of missingness, namely *missing completely at random* (*MCAR*), *missing at random* (*MAR*) and *missing not at random* (*MNAR*). Understanding at the beginning of a data analysis which type of missingness is present in the different variables is already a challenge and will be the first part of this work. This information is not only relevant for detecting patterns in the population which can be used to improve e.g the general well-being in a community, but also to adjust imputation methods. This will be the second part of this work. Mohan et. al. propose to use graphical models to detect the different missing data mechanisms (19) (29). Further they also provide a new graphical model algorithm which considers missing values since traditional graphical models work only with fully observed observations. Whereas they concentrated themselves on proofing that their algorithm works in theory, this work is primarily interested in testing the algorithm in practice which is why a simulation study stands at the core of this work following the necessary theory.

After having classified the missing value types with graphical models the information is used to adapt imputation. Imputation in general is one of the most relevant topics when working with missing data. It is not only useful and even necessary if a lot of data is missing and nearly no statistical model can be fitted or at least cannot be fitted satisfiable due to too few observations but it is also important to recover relations between variables which might got lost when using only the list wise deleted data. Most imputation models assume the missing data mechanism to be MCAR or MAR but not MNAR. For the MNAR mechanism imputation is at least in theory very difficult due to this. Using just the traditional imputation methods will yield biased results in theory which is why it seems important to find an unbiased imputation method also for MNAR. There exist already some approaches but most of them assume that we have further knowledge about the missingness process which is often unrealistic. In this work we will use the knn algorithm which is an non parametric procedure without restrictive model assumptions to impute the MNAR variables to examine if the imputation will be indeed better than using traditional imputation methods as Amelia. In the simulation study we will compare the results between the complete case analysis, the knn imputation and the multiple imputation method done with Amelia.

Before we start with the presentation of the simulation study, the work will introduce you the topic of missing data in more detail and the theoretical background of multiple imputation with Amelia and graphical models with and without incorporating the ideas of Mohan et. al.. The simulation study was done in *R*, the related code can be found on `https://github.com/Eleftheria1/Classifying-missing-values-with-graphical-models`.

2 Missing Data

Respondents declining to answer certain questions in a questionnaire, subjects dropping out of panels, technical problems with measuring instruments and much more are scenarios which will be well-known to scientists handling data especially in social or medical science. Datasets received in one of these scenarios are subject to nonresponse, which means that not all information of interest has been observed, and are called missing datasets. Many statistical methods e.g. linear regression models, however, assume the absence of missing data and software packages often simply remove the observations with missing entries automatically if the researcher ignores the problem of having missing data. This is called listwise deletion or complete case analysis and is one way to handle missing data. But obviously this can yield several problems as for example removing nearly all observations of the dataset if the missing quote is very high and not having enough data points to fit the model anymore. Other ad-hoc solutions to deal with this problem are the pairwise deletion method, also called available case analysis or mean imputation where every missing data point is replaced by the mean. Using the latter one resolves the problem of loosing many observations during the analysis but generates heavy biases such as underestimating the variance and disturbing the relations between variables which is why it should not be used either. A widely used method which performs under certain assumptions far better than those mentioned is multiple imputation. The basic idea is to generate m plausible values for each missing value to generate m completed datasets. These m datasets are then analysed as being completely observed datasets and the m results are combined according to Rubin's rules 2.2. Using this or other methods to deal with the problem of missing data means that assumptions are to be made about the process leading to nonresponse as for example what kind of missingness is present. Detecting this is very difficult and will be one of the main topics of this work. Mohan et al. proposed to estimate missingness graphs 3.1 to support the assumption a researcher may have in advance, we will see later if this procedure works also in practice(19). (30) (12) (17) (10)

2.1 Types of missing values

As mentioned before we have to make assumptions about the kind of missingness we have at hand to be able to choose the correct method for imputation and receive unbiased results. For that we will introduce the three types of missingness defined by Rubin. To do that we have to start with some notation. Let Y denote the *(n* $\times$ *p)* matrix containing the observations on p variables for all n units in the sample. The missigness indicator R is defined as an *(n* $\times$ *p)* $0-1$ matrix. The elements of Y and R are denoted by y_{ij} and r_{ij}, where $i = 1, ..., n$ and $j = 1, ..., p$. If y_{ij} is observed, then $r_{ij} = 0$, else $r_{ij} = 1$. Y_{obs} is the observed data and Y_{mis} the missing data. Hence the hypothetically complete data is $Y = (Y_{obs}, Y_{mis})$. We are mainly interested in the distribution of R which may depend on Y. This relation is modelled by the missing data model of which ψ is the parameter. This yields the general expression $P(R = 1|Y_{obs}, Y_{mis}, \psi)$. The data are said to be MCAR if $P(R = 1|Y_{obs}, Y_{mis}, \psi) = P(R = 1|\psi)$. That means the probability of being missing depends only on some parameters ψ. Furthermore we call the data MAR if $P(R = 1|Y_{obs}, Y_{mis}, \psi) = P(R = 1|Y_{obs}, \psi)$, so the probability of being missing may depend on observed information. Finally the data are MNAR if $P(R = 1|Y_{obs}, Y_{mis}, \psi)$. In this case the missing probability also depends on unobserved information. Multiple imputation can handle MAR and MCAR but not MNAR which is why it is important to be able to classify this three types of missing data. The ψ parameters on the other side are generally unknown and it would simplify the analysis if we could ignore them. We are allowed to do this if it is possible to determine the parameters θ for the full data Y which are of scientific interest without knowing ψ. The joint density function $f(Y_{obs}, R|\theta, \psi)$ is proportional to the likelihood of θ and ψ, i.e., $l(\theta, \psi|Y_{obs}, R) \propto f(Y_{obs}, R|\theta, \psi)$. Little and Rubin proved that we are able to determine θ and hence ignore the missing data mechanism for likelihood inference if the missing data are missing at random and if the parameters θ and ψ are distinct. For us the more important condition is the MAR requirement. Note that *ignorable* in that sense does not mean that we can ignore the fact that we have missing data at hand. We have to condition on those factors that influence the missing data rate for valid inference but the distribution of the data Y is the same in the response and nonresponse groups. It follows that we can model the posterior distribution $P(Y|Y_{obs}, R = 0)$ from the observed data and use this model to create imputations. On the other hand we should include R itself in the imputation model if the missingness is not ignorable since then $P(Y|Y_{obs}, R = 0) \neq P(Y|Y_{obs}, R = 1)$ holds. Since there is no information to estimate any regression weight for R because the corresponding data is missing, one needs assumptions external to the data to be able to specify $P(Y|Y_{obs}, R = 1)$. This assumptions must be made with some prior knowledge of the scientist. If no such knowledge exist it is nearly impossible to get unbiased imputations. (30) (12) (17) (10)

2.2 Multiple Imputation

In this chapter we will assume that our data is MCAR or MAR. In the next chapter 3.1 we will then see how someone can conclude this from a missingness graph.

The main goal of multiple imputation is to find an estimate $\hat{Q}$ for a quantity of scientific interest Q that is unbiased and confidence valid. Q can be expressed as a known function of the population data as we do for example when calculating regression coefficients and can only be received if the entire population is observed which means that no missing values are allowed to occur. This in turn is a very unrealistic scenario as we have discussed before and justifies the need of an unbiased and confidence valid estimate $\hat{Q}$. While the definition of unbiasedness should be known ($E(\hat{Q}|Y) = Q$), the definition of confidence validity has to be explained more in detail.

We call an estimate *confidence valid* if the average of the estimated variance-covariance matrix of $\hat{Q}$, denoted as U over all possible samples is equal or larger than the variance of $\hat{Q}$ which is caused by the sampling process. This yields the following formula:

$$E(U|Y) \geq V(\hat{Q}|Y)$$

Based on that a procedure is said to be confidence valid if it holds that a statistical test with significance level α should reject the null hypothesis in at most $\alpha\%$ of the cases when in fact the null hypothesis is true.

When drawing imputations for Y_{mis}, denoted as $\dot{Y}_{mis}$ one uses $P(Y_{mis}|Y_{obs})$. After that $P(Q|Y_{obs}, \dot{Y}_{mis})$ can be used to calculate the quantity of interest Q from the imputed data (Y_{obs}, Y_{mis}). Then we repeat this two steps with new imputed data. These are substeps for deriving the actual posterior distribution $P(Q|Y_{obs})$ of Q

$$P(Q|Y_{obs}) = \int P(Q|Y_{obs}, Y_{mis}) \cdot P(Y_{mis}|Y_{obs}) dY_{mis}$$

which we are interested in. We can see that it is equal to the average over the repeated draws of Q. Using this it can be shown that the posterior mean of $P(Q|Y_{obs})$ is equal to

$$E(Q|Y_{obs}) = E(E(Q|Y_{obs}, Y_{mis})|Y_{obs})$$

which is the average of the posterior means of Q over the repeatedly imputed data. From this follows the suggestion for combining the results of repeated imputations:

$$\bar{Q} = \frac{1}{m} \sum_{l=1}^{m} \hat{Q}_l$$

where $\hat{Q}_l$ is the estimate of the l^{th} repeated imputation and m the number of imputations. The posterior variance of $P(Q|Y_{obs})$ is the sum of the within-variance and the between-variance:

$$V(Q|Y_{obs}) = E(V(Q|Y_{obs}, Y_{mis})|Y_{obs}) + V(E(Q|Y_{obs}, Y_{mis})|Y_{obs})$$

The within-variance is the average of the repeated complete-data posterior variances of Q. The between-variance is the variance between the complete-data posterior means of Q.

If we assume an infinitely large number of imputations, then the posterior variance of Q is $T_\infty = \bar{U}_\infty + B_\infty$ where $\bar{U}_\infty$ is the estimated within and B_∞ the estimated between variance. Taking now the fact into account that usually we do not have infinity many imputations we have to adjust the calculation of T and its single components. The complete data variances are calculated now as

$$\bar{U} = \frac{1}{m}\sum_{l=1}^{m} \bar{U}_l$$

where $\bar{U}_l$ is the variance-covariance matrix of $\hat{Q}_l$ On the other hand the standard unbiased estimate of the variance between the m complete data estimates is given by

$$B = \frac{1}{1-m}\sum_{l=1}^{m}(\hat{Q}_l - \bar{Q}) \cdot (\hat{Q}_l - \bar{Q})^T$$

Note that when we want to compute now the total variance T we have to consider the fact that $\bar{Q}$ itself is estimated using finite m and thus only approximates $\bar{Q}_\infty$. Rubin shows that the contribution to the variance of this factor is systematic and equal to $\frac{B_\infty}{m}$. Since B approximates B_∞, this yields

$$T = \bar{U} + B + \frac{B}{m}$$

Vividly this means that $\bar{U}$ stands for the variance caused by the fact that we are taking a sample rather than observing the entire population, B for the extra variance caused by the fact that there are missing values in the sample and $\frac{B}{m}$ for the extra simulation variance caused by the fact that $\bar{Q}$ itself is estimated for finite m. Note that the last term ensures that confidence intervals are not too short but it also shows that the traditional choice of $m = 5$ may be to low and hence the effect of simulation error on the total variance to high, which is why it is recommended to use a higher m.

We call the procedure to combine the repeated imputation results *Rubin's Rules.*

After having created a theoretical foundation of how the estimate $\hat{Q}$ should be correctly calculated, we can have a closer look at the main steps in multiple imputation. For that we use the scheme in figure 1 which can be found also in (30). As you can see combining the results of the imputation with *Rubin's Rules* belongs to the last part of the scheme, called also pooling. But let's start from the beginning. In the first step we start with the observed data and think about what multiple imputation method may be the best for our application. The *mice* package for example generates multivariate imputations by Chained Equations. It is a Markov chain Monte Carlo method, where the state space is the collection of all imputed values. This method is based on Fully Conditional Specification, where each incomplete variable is imputed by a separate model. When choosing this method one has to decide for each variable which predictor variables with potential interactions should be included in the imputation model. That means that it is advantageous to know which variables are independent and which not. Of course one could use always all variables for the imputation but that may disturb the model. Note also that in case of an independent variable one cannot do proper imputation. Another technique to multiple impute data is done in the *Amelia II* package. Here a novel bootstrapping approach, the EMB (expectation-maximization with bootstrapping) algorithm is used. The advantage of Amelia is that " it combines the comparative speed and ease-of-use of the algorithm with the power of multiple imputation, to let you focus on your substantive research questions rather than spending time developing complex application-specific models for nonresponse in each new dataset " (12). This is also the reason why we will use this algorithm in the simulation study. We will dive deeper in the theory of this algorithm soon. Nevertheless this procedure has also a significant disadvantage which should be mentioned. The algorithm assumes that the complete data, thus observed and unobserved data, follow a multivariate normal distribution. Of course one can use different transformations to force the given data to fit better in a normal distribution but we will see later that this does not improve the algorithm a lot. Though it has already been shown that the model works also quite well for mixed or categorical data, which is why we will use it despite this assumption violation.

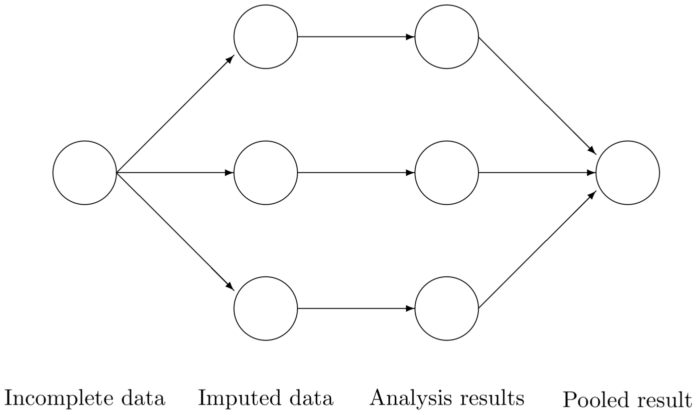


Figure 1: Scheme of main steps in multiple imputation

After having decided which imputation method to use one creates several complete versions of the data by replacing the missing values by plausible data values. These plausible values are drawn from a distribution specifically modelled for each missing entry by using the chosen imputation method. In the figure you can see three imputed datasets. In our simulation study we will use five. The three imputed datasets are identical for the observed data entries, but differ in the imputed values. The magnitude of these difference reflects our uncertainty about what value to impute.

Next we estimate the parameters of interest e.g. regression coefficients from each imputed dataset exactly the same as we would do it when having complete data. Of course the results of the different imputations will differ since we have different entries for the missing data points. These differences are caused only because of the uncertainty about what value to impute. This yields us the problem of deciding which value to keep of these m imputations. This is done in the last step where we pool our results as discussed before. (30) (12) (17)

2.2.1 Multiple Imputation with Amelia II

Amelia II implements a new expectation-maximization with bootstrapping algorithm that works faster and is easier to use, than various Markov chain Monte Carlo approaches, but gives at the same time essentially the same answers. The algorithm uses the EM (expectation-maximization) algorithm on multiple bootstrapped samples of the original incomplete data to draw values of the complete-data parameters. After that the algorithm draws imputed values from each set of boostrapped parameters, replacing the missing values with these draws. We will have a deeper look at the EM algorithm in chapter 2.2.1. The package also improves imputation models by allowing expert knowledge to incorporate Bayesian priors on individual cell values. We will not make use of this since we assume in our simulation that we do not have any further knowledge about the data but it is still worth mentioning it. Another advantage of this package is that compared to other packages it virtually never crashes.

As always there is also a dark side, at least in theory. The imputation model assumes namely that the complete $(n\ x\ k)$ data D which is composed of the observed data D_{obs} and the unobserved missing data D_{mis} follows a multivariate normal distribution with mean μ and covariance matrix Σ.

$$D \sim N_k(\mu, \Sigma)$$

Even if this is a strong restriction in theory in practice there is evidence that the model works also for mixed and categorical data. Moreover one can transform the data in many different ways such that it makes the normality assumption more plausible, some of this transformations as *square root transformation* are also incorporated in the package and will be used later on in the simulation study.

Another disadvantage is that as in nearly every multiple imputation method the algorithm

assumes the data to be *MAR* or *MCAR* but not *MNAR*. This is also the reason why we will discuss *knn-imputation* in the next section 2.2.2 to account also for *MNAR* missingness.
The algorithm itself wants to take draws from the following posterior

$$p(\theta|D_{obs}) \propto p(D_{obs}|\theta) = \int p(D|\theta) dD_{mis} \tag{1}$$

by using the classic EM algorithm and additionally bootstrap the data for each draw to simulate estimation uncertainty before applying the EM algorithm to find the mode of the posterior for the bootstrapped data.
The posterior itself is obtained by taking first into account that our observed data is not only D_{obs} but also the missigness matrix M which gives us information about which entries are missing and which not. Therefore the likelihood of the observed data is $p(D_{obs}, M|\theta)$ and can be written, if we assume that the data at hand is *MAR*, as

$$p(D_{obs}, M|\theta) = p(M|D_{obs}) \cdot p(D_{obs}|\theta)$$

where $\theta = (\mu, \Sigma)$ are the complete-data parameters we are interested in. Since $p(M|D_{obs})$ does not depend on the complete data parameters, which we are looking for we can rewrite the likelihood as

$$L(\theta|D_{obs}) \propto p(D_{obs}|\theta).$$

When using the law of iterated expectation this yields

$$p(D_{obs}|\theta) = \int p(D|\theta) dD_{mis}.$$

Using this likelihood and a flat prior on θ we obtain the posterior in equation 1 and the EM algorithm can be applied on bootstrapped data to find the mode of it. (12)

Addendum: Expectation Maximization algorithm

The expectation maximization algorithm in general enables parameter estimation in probabilistic models with incomplete data. It alternates between guessing a probability distribution over completions of missing data given the current model (E-step) and then reestimating the model parameters using these completions (M-step). Chuong B Do and Serafim Batzoglou have generated a helpful figure in their paper *What is the expectation maximization algorithm?* (5), which we will use to understand how the algorithm works.

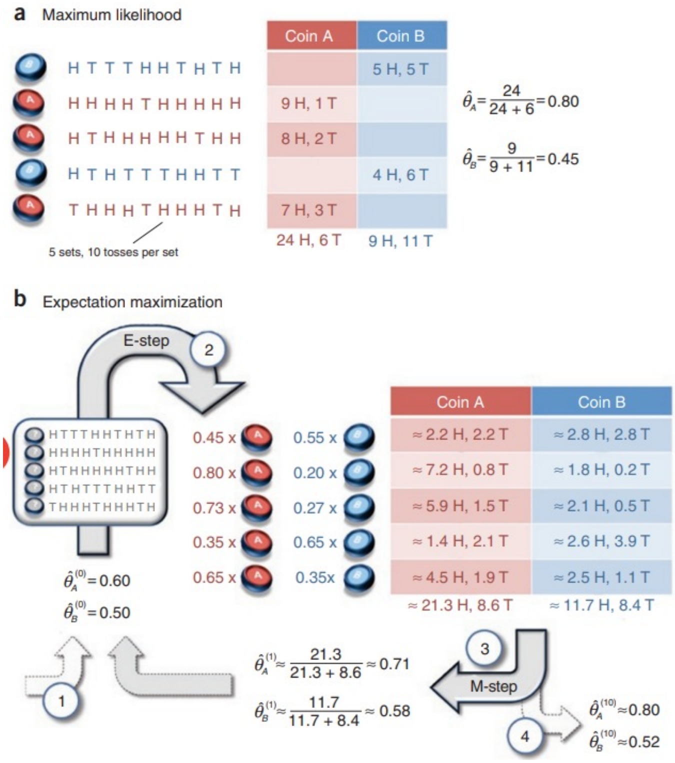


Figure 2: Scheme of EM algorithm (5)

As you can see we start with a coin flipping experiment in which a pair of coins A and B of unknown biases are given. θ_A and θ_B represent the probability of A landing on head and B landing on head, respectively. Consequentially $1 - \theta_A$ describes the probability of A landing on the tail. The goal is to estimate these probabilities θ by randomly choosing one of the two coins and performing ten independent coin tosses with the selected coin and afterwards repeating this procedure five times. Note that the probability of choosing one of the two coins should be equal. Parameter estimation in this setting is known as the complete data case because we know the values of all variables and can be done e.g. with maximum likelihood estimation, which yields the estimated parameter

$$\hat{\theta}_A = \frac{Number\ of\ heads\ using\ coin\ A}{Total\ number\ of\ flips\ using\ coin\ A}$$

if we consider that our likelihood is $p(\mathbf{x}, \mathbf{z}, \theta)$, where $\mathbf{x} = (x_1, ..., x_5)$ and respectively $x_i \in \{0, 1..., 10\}$ is the number of heads observed during the ith set of tosses and $\mathbf{z} = (z_1, ..., z_5)$ and analogously $z_i \in \{A, B\}$ is the identity of the coin used during the ith set of tosses. Since in this scenario all observations are known to us the objective function, namely the likelihood has a single global maximum which is easily computable as we can see.

Now we assume that we do not know the identities z of the coins used for each set of tosses but we can observe recorded head counts x. The resulting incomplete data likelihood $p(x, \theta)$

has now multiple local maxima and no closed form solution. Hence computing proportions of heads for each coin is no longer possible, since the coin used for each set of tosses is unknown. The EM algorithm solves this problem by splitting the optimization problem in a sequence of simpler optimization problems, whose objective functions have unique global maxima that can often be computed in closed form.

According to this we can reduce parameter estimation for this case with missing data to maximum likelihood estimation with complete data if we are able to guess correctly which coin was used in each of the five sets. To do that we use a iterative procedure that looks as follows. We start with some initial parameters $\theta^{(0)} = (\theta_A^{(0)}, \theta_B^{(0)})$, $\theta^{(t)} = (\theta_A^{(t)}, \theta_B^{(t)})$ determine for each of the five sets whether coin A or coin B was more plausible to have generated the observed flips by using the current parameter estimates. Then, assume these guessed coin assignments to be correct and apply the regular maximum likelihood estimation procedure to get $\theta^{(t+1)}$ like visualised in step three of the figure 2. These steps are then repeated until the algorithm converges, i.e. the estimates do not change anymore and the local maximum is reached (step four 2). This procedure can be improved by not choosing just the single best assignment but by computing probabilities for each possible assignment of the missing data, using the current parameters to include uncertainty (see step two of the figure 2). This yields a weighted training set consisting of all possible assignments of the data and their probabilities. With that training set a modified version of maximum likelihood estimation can be applied and used to obtain new parameter estimates $\theta^{(t+1)}$. (5)

After applying the EM algorithm on the different bootstrapped datasets as visualised also in figure 3, we have draws of the posterior of the complete-data parameters and can create imputations by drawing values of D_{mis} from its distribution conditional on D_{obs} and the draws of θ. (12)

Next we have to analyse each imputed dataset and combine the results according to *Rubin's Rules* 2.2.

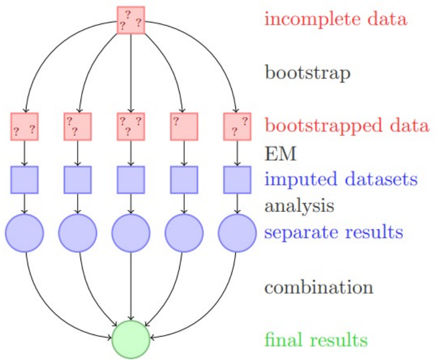


Figure 3: Scheme of Honaker's, King's and Blackwell' approach to multiple imputation with the EMB algorithm in their paper *Amelia II: A Program for Missing Data* (12)

2.2.2 Weighted knn Imputation

We have discussed already that most imputation methods assure asymptotically unbiased results only in the MCAR and MAR scenario and are biased for MNAR data. A widely discussed method for improving imputation for MNAR variables is to include auxiliary variables that have an relationship with the missing variable in the imputation model (20). Similarly to that there exist also the idea of *imputation stacking* (2) which uses assumptions about the not-at-random missingness to calculate weights as a function of the imputed data and these assumptions. Afterwards one tries to correct the multiple imputation by a weighted analysis. These two approaches have something in common, namely the assumption that we have knowledge about the missingness process. If we will look at the missingness graphs 3.4 we will see that we have used them to find out which missingness type is present. This means that we assume we have no further knowledge about the missingness process and this yields not being able to use these methods as they require further information about the data. Another procedure which could be used is the not-at-random fully conditional specification (NARFCS) procedure (28). The underlying idea is to perform a sensitivity analysis to departures from MAR. This is performed by defining a single MNAR model with one of more unidentified parameters, known as sensitivity parameters, allowing the distribution on specific variables to vary between missing and observed data points. This would indeed yield good results if this sensitivity parameters are specified correctly which is not straightforward since also here expert knowledge is required and/ or a tipping point analysis which may or may not include the true parameters. Thus we need another procedure to handle MNAR data.

Another idea is to look at the abilities of neural networks, since deep learning seems to be the answer for nearly everything in the last years. Yoon, Jordon and van der Schaar (33) have proposed to use Generative Adversarial Nets to impute missing data and have showed that you can obtain indeed good imputations with this procedure. However Lalande and Doya have published in their paper *Numerical Data Imputation: Choose kNN over Deep Learning* (18) some results that speak for using knn instead of neural networks, since in their simulation study the results where quit similar, sometimes even better, and knn is far less complicated than neural networks and easier to interpret. Also Petrazzini et al. (24) showed that knn imputation gives expecially in MNAR scenarios good imputations. Thus it seems like a good idea to try after the principle of Occam's razor knn imputation for MNAR data before over complicating things with neural networks.

Before we start to speak about knn imputation it is important to note that most of the literature evaluate the goodness of the imputation based on the results of the prediction, e.g. by comparing the RMSEs. We will focus on the imputations themselves by looking at the empirical densities of the missing variables and the parameter estimates instead of looking at the overall goodness of prediction. That should in theory of course not change the evaluation of the different imputation methods but nevertheless it should be kept in mind.

A big advantage of the knn algorithm in general is that it does not have any model assumptions, as the MAR assumption or the normality assumption in Amelia, which can be not fullfilled as it is a non-parametric method and hence also called *lazy model.* Therefore there is also no necessity for creating a predictive model for each attribute with missing data as done in *mice.* Thus, the k-nearest neighbour can be easily adapted to work with any attribute as class, by just modifying which attributes will be considered in the distance metric. Hence it can predict both discrete attributes (the most frequent value among the k nearest neighbours) and continuous attributes (the weighted mean among the k nearest neighbours).

For data imputation tasks, the knn algorithm selects the k nearest neighbours of a given incomplete observation, and uses available data from the selected neighbours to estimate missing values. If we have a high proportion of missing values in the data at hand the available data becomes very sparse and imputation gets bad. This is why we will change the approach slightly and use for every variable which we do not want to impute in the current state the Amelia imputations which we made before to have more data at hand to calculate the distances and also the value of the missing entry. This makes it also possible to do some kind of multiple imputation with knn since we do knn m times with the m different imputed predictor variables from Amelia and then pool the results.

The knn imputes missing values using a weighted average of the selected neighbours. We choose the Euclidean distance, which looks as follows in the two dimensional space

$$d(x_i, x_j) = \Big(\sum_{s=1}^{p} (x_{is} - x_{js})^2 \Big)^{\frac{1}{2}}$$

where x_j denotes the jth nearest neighbour of x_i, and use the resulting distances as input for the epanechnikov kernel

$$\frac{3}{4}(1 - d^2) \cdot \mathbb{1}(|d| \leq 1)$$

to calculate the weights and furthermore also the values of the missing entry.

One can easily imagine that the resulting value for the missing variable may be biased if we include a lot of variables which are not correlated with the missing variable or its reason for missingness if this variables influence highly the distance. Moreover it makes the algorithm slower without improving it. Therefore we will use only that variables for computing the nearest neighbours which are connected with the variable itself or with the missingness indicator variable in the graph to improve the imputation (18) (24) (1) (9). After having mentioned the graphical models a few times we will now have a look at the theory behind this models.

3 Graphical models

3.1 Graph theory

A graph $G = (V, E)$ is a pair of finite sets $V = V(G)$ and $E = E(G)$. V is the nodes set which corresponds here to the variables of our dataset and E are the edges between these nodes. Each pair of nodes can have either no or one edge between them, and this in turn can be directed either in one or in two directions or undirected. If there is an edge of any form between two nodes u and v, they are said to be adjacent, denoted by $u \sim v$. Later on we will focus on directed graphs which are defined as graphs containing only directed edges, drawn as arrows. Further we assume that the graph contains no self-loops, that is $(u, u) \notin E$ for all $u \in V$. A graph is called complete if there is an edge between every pair of nodes. Besides graphs also subgraphs which are derived by using only a subset of nodes $A \subset V$ can be complete. If this subset is then not contained in a larger complete subset we call it a *clique*. Therefore a *clique* is a maximal complete subset.

Moreover we define a path of length n between u and v as a sequence of nodes $u = u_0, ..., u_n = v$ such that $u_{i-1} \sim u_i$ for $i = 1, ..., n$. A graph is said to be connected if there is a path between every pair of nodes, respectively. A *n-cycle* is a path of the form $u_1, u_2, ..., u_n, u_1$.

If additionally the nodes of the cycle are distinct, and if $u_j \sim u_k$ is true only if $|j - k| = 1$ or $n - 1$, then we call the cycle a *chordless* cycle. This is important because we will need the definition of triangulated graphs later, which are defined as graphs that have no *chordless* cycles of length greater or equal to four.

An acyclic graph on the other had is a graph with no cycles. For three subsets A, B, and S of V, we say S separates A and B if all paths from A to B intersect S. (6) (11)

We can use all these definitions to build graphical models, where we focus on models under which some conditional independence relations of the form $X \perp\!\!\!\perp Y|$ (some other variables) hold for all densities in the model. In this work we will assume according to common practice multivariate normal distributions for our densities which will yield the use of *graphical gaussian models.*

3.2 Conditional independence

Since we have to use conditional independence relations when working with graphical models, it makes sense to refresh shortly our knowledge about independence in the mathematical sense.

Two random variables $X_A = (X_v)_{v \in A}$ and $X_B = (X_v)_{v \in B}$ with A and B being subsets of V are said to be independent if their joint density factorises into the product of their marginal densities:

$$f_{X_A, X_B}(x_A, x_B) = f_{X_A}(x_A) \cdot f_{X_B}(x_B).$$

As you can see, we already foreshadow the connection to graphs, by indexing with a node $v \in V$. If we want to reformulate the definition of independence such that we do not need to make use of the density of X_A we can look at the conditional density of X_B given $X_a = x_a$:

$$f_{X_B|X_A}(x_B|x_A) = f_{X_B}(x_B).$$

When turning to the concept of conditional independence we include beside the two variables X_A and X_B a third variable X_C. X_A and X_B are called conditionally independent given X_C, denoted by $A \perp\!\!\!\perp B|C$, if for each value $x_C \in X_C$, X_A and X_B are independent in the conditional distribution given $X_C = x_C$. This yields:

$$f(x_A, x_B|x_C) = f(x_A|x_C) \cdot f(x_B|x_C).$$

An alternative characterisation of this relation can be done if we take the *factorization criterion* into account. Then we can write the joint density as a product of two functions $g(\cdot)$ and $h(\cdot)$, where $g(\cdot)$ only depends on x_A and x_C but not on x_B and $h(\cdot)$ does not depend on x_A:

$$f(x_A, x_B, x_C) = g(x_A, x_C) \cdot h(x_B, x_C).$$

(6) (11)

3.3 Directed Graphs

3.3.1 Directed Acyclic Graphs

Acyclic graphs are as already mentioned before graphs with no cycles and directed graphs are graphs whose edges are arrows pointing to one direction. Directed acyclic graphs are the combination of these two. In particular that means that our graph has no cycles with the arrows pointing in the same direction all the way around. The word *directed* will be of high importance in this work, since we want to analyse the reason why a variable has missing values and which other variables could be the cause of that. The direction of the edges namely indicates influence of a variable on another or in some cases they represent even causal directions. Although it would be great when looking at the interpretation of these models if the latter would hold, it must not necessarily be true, which is very important to keep in mind.

One can show that the absence of any directed cycles is equivalent to the existence of an ordering of the nodes $v_0, ..., v_n$ such that $v_i \to v_j$ only if $i < j$. That means that arrows point only from lower-numbered nodes to higher-numbered ones. Note that a DAG with n nodes and no edges has $n!$ possible orderings, whereas a complete DAG has only one, which means that the ordering is not always unique. If we have a given ordering, which has to be proposed by the researcher with some prior knowledge of the data such that v_{i-1} is prior to v_i for all $i = 1, ..., n$, we can factorize the joint density of $v_1, ..., v_n$ as

$$f(v_1) \cdot f(v_2|v_1) \cdot ... \cdot f(v_n|v_{n-1}v_{n-2}...v_1). \tag{2}$$

Later we will see that in most cases the variable ordering will not be known in advance and must be inferred from the data, which can be done only up to Markov equivalence. Under the term *Model Selection* we understand exactly this procedure but we do not have to worry about that right now.

An arrow is drawn from v_i to v_j where $i < j$, unless $f(v_j|v_{j-1}v_{j-2}...v_1)$ does not depend on v_i, when building the DAG. Remembering the definition of conditional independence and taking into account that now we have additionally also a ordering this means that

$$v_i \perp\!\!\!\perp v_j|\{v_1...v_j\}\backslash\{v_i, v_j\}.$$

We can interpret this as conditional independence of v_i and v_j given all *prior* variables. In contrast to that when handling undirected graphs we say that the conditional independence relation holds given *all* remaining variables, as introduced in section 3.2 since we have no ordering. But the key message that a missing edge between two nodes is equivalent to a conditional independence relation between these two variables stays the same.

If we recall also the definition of a *path* and take into account that now we have edges with directions in our graph we have to adjust slightly the definition of it by changing the undirected sequence of nodes with a directed one, yielding the requirement $u_{i-1} \to u_i$ for $i = 1, ..., n$. With that extension we can introduce also parents and children of a node. The parents $pa(v)$ of a node v are those nodes u for which $u \to v$. Similarly the children $ch(u)$ of a node u are those nodes v for which $u \to v$. Parents in the broader sense are called ancestors $an(w)$ and are defined as directed paths from v to w. We can apply this definitions also on sets of nodes instead of single nodes. For a set $S \subseteq V$ this yields e.g. $pa(S) = \left(\bigcup_{v \in S} pa(v)\right) \backslash S$.

We can use this new terms to rewrite the joint density from equation (2) as

$$\prod_{v \in V} f(v|pa(v)) \tag{3}$$

and hence the pairwise conditional independence becomes $v_i \perp\!\!\!\perp v_j \mid an(\{v_i, v_j\})$ and is independent of any specific nodes ordering.

Until now we focused only on pairwise conditional independences, as a next step we want to investigate if it is possible to extend this and deduce even stronger dependencies from a graph. Therefore we have to introduce a new graph theoretic property, namely the *d-seperation* which represents general conditional independences in DAGs.

The main idea is to associate *dependencies* with *connectedness* in the graph. For that we can define three rules which tell us if our path is *d-connected* or not. Before starting with the first one we have to introduce *colliders*. A path is called a *collider* if it has converging arrows. Starting now with the first rule we say that A and B are *d-connected* if there is an unblocked path between them. Unblocked means that the path can be traced without traversing a pair of arrows that collide "head-to-head". This could be visualised as follows:

$$u \to r \to s \to t \leftarrow w \leftarrow y \to v$$

For the examples we consider for simplicity separation between two single variables, u and v but the extension to sets of variables can be done easily when using that two sets are separated if and only if each element in one set is separated from every element in the other. We could assume e.g that $u \in A$ and $v \in B$.

Going back to the given example we can see that t is our collider in this case which means that the whole path is not unblocked and hence also not d-connected. If a path is not *d-connected* we call it *d-seperated*. On the other hand we conclude from the graph that e.g the subgraph $u \to r \to s \to t$ is indeed *unblocked* and also *d-connected*.

Including now a third set S of variables which we will condition on, and therefore its values are assumed to be known, leads us to the second rule: u and v are *d-connected*, conditioned on a set S of nodes, if there is a collider-free path between u and v that traverses no element of S. Looking again at an example will make this definition more tangible. Now we assume that $r, y \in S$. The main structure of the graph looks the same but the interpretation is different.

$$u \to \mathbf{r} \to s \to t \leftarrow w \leftarrow \mathbf{y} \to v$$

While we had concluded in the first example that the subgraph $u \to r \to s \to t$ is *d-connected*, now it is *d-seperated* because the r node is known. We say that S has blocked the path $u \to r \to s \to t$ and hence also the path from u to v which was of interest in the first place.

The last rule handles *colliders* which are part of the known set S, that we use as condition. It states that a collider no longer blocks any path that traces it, if it is a member of S or has a *descendant* in S. A *descendant* is the same as an *ancestor* but for children. If we take again our path from the first example and extend it with assuming that now our collider $t \in S$, the path from u to v will be *d-connected* by S.

$$u \to r \to s \to \mathbf{t} \leftarrow w \leftarrow y \to v$$

Combining these three rules into one definition one can derive if the different paths or subpaths of an DAG are *d-seperated* or not.

This *d-seperation* definition can be seen as an equivalent of the global Markov property for DAGs.

If DAGs induce the same sets of conditional independence relations which will be the case if we do not know the exact ordering of the variables they are called Markov equivalent and are distributionally equivalent but differ in interpretation. You can see an example of Markov equivalent graphs in Figure 4. Although they will be distributionally equivalent you can obviously see that the interpretation is different for each graph. While in the first graph variable Y influences X and Z, in the second graph Y still influences Z but not X. X on the other side influences now Y. Similarly you can interpret also the third graph.

More precisely two graphs are Markov equivalent if and only if they have the same skeleton, which is the undirected graph, and same unshielded colliders. An unshielded collider occurs when two directed edges from non adjacent nodes meet.

Since knowing the ordering of the variables may not be a realistic scenario for our simulation later we have to think about how to represent these equivalence classes.

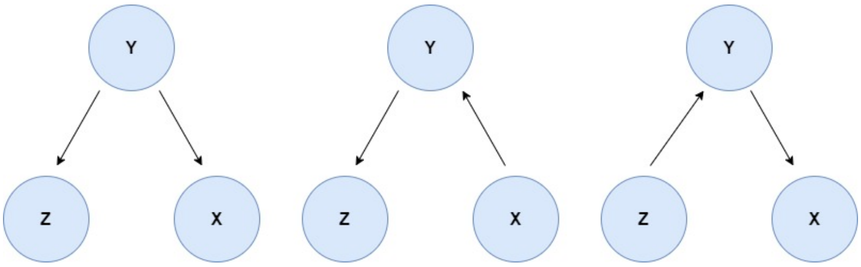


Figure 4: Example of Markov equivalent graphs

One option to do this is to build completed partial directed acyclic graphs CPDAGs. These graphs are constructed by orienting all edges whose directions are fixed in the equivalence class and letting edges be undirected if there are two members of the equivalence class that have arrows in opposite directions. The individual DAGs then in the equivalence classes can be obtained by assigning any orientation to the undirected edges, provided this does not introduce any cycles or unshielded colliders.

Independently of the method we use to represent an equivalence class the first step will be to construct the skeleton of the graph and build on that we can add directions to edges using the idea of e.g. CPDAGs. This is why the section 3.5.1 will introduce undirected gaussian graphical models even if they themselves are not of interest for this work.

(21) (6) (11)

3.3.2 Directed Gaussian graphical models

In this work we will focus on directed gaussian graphical models (DGGMs), which assume that the data at hand follows a multivariate normal distribution. We will see later on in the simulation study that the application of these models works also for data which is not multivariate normal distributed but can be approximated with a normal distribution if the sample sizes are big enough.

Even if there are also possibilities for modelling graphical models for discrete or other types of continuous data, it is justifiable to stick with the gaussian graphical models here. The main reason for that is the topic of model selection which will be of high importance for us since we want to keep only a few edges which indicate indeed a high dependence condition between our indicator variable for the missing variable and its reason for missingness. There exist different ways for doing this model selection as e.g. step wise approaches which are based on the AIC/BIC criterion. A problem with step wise selection procedures is that they tend to become time consuming for problems with many variables and usually only a small part of the relevant search space is covered during a search. Others can be found in (6). This kind of approaches can be used also with mixed data but they tend further to prefer complicated models with a lot of edges, which is not adequate for our purpose.

The *pc-algorithm*, which will be presented in the next section, on the other hand prefers simple models with few edges and reflects often better the structure of the true model. Therefore it is used by a lot of people in practice even if the data is presumably not normal distributed. The normal assumption is only necessary when using the *pc-algorithm* if we have continuous data at hand as we will see later. When handling discrete data as e.g binary data one can assume also the data to be binomial distributed. The problem is that we have to stick with one distribution assumption for all variables and cannot mix the distributions for the single variables of the data.

This is why we will assume for all variables in the simulation study the normal distribution. Another option would be to model two separate models one for continuous data with normal distribution assumption and one for discrete data with binomial assumption to reduce model violations but then we would not be able to see relations between a continuous and a discrete variable any more which is also not desirable. Hence seemingly the best solution is to argue with the central limit theorem and assume that all variables are normal distributed if the dataset is big enough.

Speaking of gaussian models we have to recap first the definition of the multivariate normal distribution. A random vector $\mathbf{X} = (X_1, ..., X_p)$ follows a multivariate normal distribution $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ in p dimensions with mean $\boldsymbol{\mu} \in \mathbb{R}^p$ and positive definite covariance matrix $\boldsymbol{\Sigma} =$

$\Sigma_{i,j} i, j \in (1, ..., p)$ if the density is given by

$$f_{\mathbf{X}}(\mathbf{x}) = det(2\pi\Sigma)^{-\frac{1}{2}} \; exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}))$$

The inverse of $\boldsymbol{\Sigma}$, which exists if we assume positive definiteness, is the precision matrix and we denote it by $\mathbf{K}$. This will be the key quantity in Gaussian graphical models and has the following form:

$$\mathbf{K} = \begin{pmatrix} k_{11} & k_{12} & \cdots & k_{1d} \\ k_{21} & k_{22} & \cdots & k_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ k_{d1} & k_{d2} & \cdots & k_{dd} \end{pmatrix}$$

Using the precision matrix in the density of the normal distribution instead of the covariance and defining h as $K\mu$ we can rewrite the density as

$$f_{\mathbf{X}}(\mathbf{x}) = (2\pi)^{-\frac{d}{2}} \; det(\mathbf{K})^{\frac{1}{2}} \; exp(-\frac{1}{2}\boldsymbol{\mu}^T\mathbf{K}\boldsymbol{\mu} + \mathbf{h}^T\mathbf{x} - \frac{1}{2}\mathbf{x}^T\mathbf{K}\mathbf{x})$$

Choosing $a = -\frac{d}{2}log(2\pi) + \frac{1}{2}log \; det(\mathbf{K}) - \frac{1}{2}\boldsymbol{\mu}^T\mathbf{K}\boldsymbol{\mu}$ we get

$$\begin{aligned} f_{\mathbf{X}}(\mathbf{x}) &= exp(a + \mathbf{h}^T\mathbf{x} - \frac{1}{2}\mathbf{x}^T\mathbf{K}\mathbf{x}) \\ &= exp(a + \sum_u h_u \, x_u - \frac{1}{2}\sum_u\sum_v k_{uv} \, x_u \, x_v) \end{aligned}$$

If the sets of the nodes A and B are separated by a set C in the graph we have $k_{uv} = 0$ for $u \in A, v \in B$. Then we can use the factorisation criterion from chapter 3.2 to show that by collecting appropriate terms we can write $f(x) = g(x_A, x_C) \; h(x_B, x_C)$. This is the proof that for DGGMs the global Markov property $A \perp\!\!\!\perp B|C$ holds. Moreover you can see from the formula that we have at most pairwise interactions between variables, which means that we do not have higher order interactions in DGGMs and therefore the model is completely determined by the edges of the graph.

Further it can be shown that the conditional distribution of (X_1, X_2) given the remaining variables is a bivariate normal distribution with inverse covariance

$$\mathbf{K}_{bivariate} = \frac{1}{k_{11}k_{22} - k_{12}^2} \begin{pmatrix} k_{22} & -k_{21} \\ -k_{12} & k_{11} \end{pmatrix}$$

Since we want to detect conditional (in-)dependencies between pairs of variables, more precisely between a indicator variable of our missing variable and its reason for missingness, it seems a good idea to use the precision matrix of the bivariate distribution to calculate partial correlations between them whereby this correlation is also invariant under change of scale.

The partial correlation between x_u and x_v given all other variables is then composed of

$$\rho_{uv|V\setminus(u,v)} = -k_{uv} \setminus \sqrt{k_{uu}k_{vv}} \tag{4}$$

We can see from the formula that k_{uv} has to be zero if and only if x_u and x_v are conditionally independent given all other variables hence we would not draw an edge between this two variables. Graphical Gaussian models are then defined by setting some elements of the precision matrix, and therefore partial correlation, to zero.

Recall that a probability function factorises w.r.t. a DAG if it can be expressed as a product of conditional densities of single variables given their parents as stated in equation 3. Since we focus on DGGMs here our univariate conditional models which we need to construct models of this type are linear regression models with gaussian errors. In particular this means that the conditional distributions in equation 2 are linear regressions.

As we have seen in the previous chapter 3.3.1 DAGs and hence also DGGMs that are Markov equivalent cannot be distinguished on the basis of sample distributions. This means that we can select only equivalence classes of DGGMs with e.g. CPDAGs when we use model selection algorithms, and not individual ones. A rare exception is the case where we know the exact ordering of the variables like stated in equation 2.

We will introduce the *pc-algorithm* in section 3.5.1 which estimates the CPDAG of the true causal structure of the data. (6) (11)

3.4 Graphical models for missing data

We have seen in section 2 that there exist three different types of missing values, namely MCAR, MAR and MNAR. Most imputation methods as e.g multiple imputation 2.2 provide asymptotically unbiased results only if the data is MAR or MCAR but not in the MNAR scenario. According to this almost all available software packages implicitly assume that data fall under this two categories. Failing these assumptions, there is no guarantee that estimates produced by software will be less biased than those produced by simply using complete case analysis. Consequently, it is important for the user to decide if the type of missingness present in the data is compatible with the requirements of MCAR or MAR. Without further knowledge it is nearly impossible to detect which type of missingness we have at hand. To solve this problem one can use graphical models to encode assumptions about the reasons for missingness. We call this specific kind of graphical model missingness graph or short *m-graph.*

To build an m-graph we have to introduce next to our variables X of the given dataset also auxiliary variables R_{v_i} which represent the missingness mechanism of the variable V_i at hand. If $R_{v_i} = 1$ the value of the variable is concealed, i.e. we will see an *NA* entry in the corresponding variable, on the other hand if $R_{v_i} = 0$ the true value of the variable is revealed, i.e. the entry in the corresponding variable is the observed value v_i. When looking at the graph we then have to slightly adjust the definition of our nodes set V. We can partition it into five categories $V = V_o \cup V_m \cup U \cup V^\star \cup R$ where V_o is the set of variables that are observed in all records and V_m is the set of variables that are missing in at least one record. Variable X is termed as fully observed if $X \in V_o$ and partially observed if $X \in V_m$. R_{v_i} and $V_i^\star$ are two variables associated with every partially observed variable, where $V_i^\star$ is a proxy variable that is actually observed, and R_{v_i} represents the status of the causal mechanism responsible for the missingness of $V_i^\star$ as explained above. This yields formally:

$$v_i^\star = f(r_{v_i}, v_i) = \begin{cases} v_i & if \;\; r_{v_i} = 0 \\ NA & if \;\; r_{v_i} = 1 \end{cases}$$

$V^\star$ is the set of all proxy variables and R is the set of all causal mechanisms that are responsible for missingness. Note that we will not explicitly show the proxy variables in the graph for the sake of clarity. U is the set of latent variables. Usually it is assumed that no variable in $V_o \cup V_m \cup U$ is a child of an R variable. As in the traditional graphical models two nodes X and Y can be connected by a directed edge $X \rightarrow Y$, indicating that X is a cause of Y , or by a bi-directed edge $X \leftrightarrow Y$ denoting the existence of a U variable that is a parent of both X and Y.

When working with m-graphs we have to adjust also the definition of the three types of missingness. Thereby we will replace the traditional MAR type of missingness by v-MAR. The main distinction rests on the fact that MAR introduced by Rubin is defined in terms of conditional independencies between events whereas that when speaking of v-MAR conditional independencies are defined between variables. In the following we may use the term MAR when speaking of v-MAR. Now we can translate the original definitions of MCAR, MAR and MNAR in graphical terms.

Data are MCAR if $V_o \cup V_m \cup U \perp\!\!\!\perp R$ holds in the m-graph. This means R is entirely independent of both the observed and the partially observed variables. This condition can be easily identified in an m-graph by the absence of edges between the R variables and variables in $V_o \cup V_m$. On the other hand data are v-MAR if $V_m \cup U \perp\!\!\!\perp R | V_o$ holds in the m-graph which means that missingness occurs at random conditional on the fully observed variables V_o. In graphical terms, v-MAR holds if (i) no edges exist between an R variable and any partially observed variable and (ii) no bi-directed edge exists between an R variable and a fully observed variable. Finally data that are not v-MAR or MCAR fall under the MNAR category. (19) (4) (27)

You can see in in the figure below 5 that it is very easy to detect the missingness type with this simple rules from the m-graph. Here we assume that our variable X is the variable of interest for which we want to check which missingness type is present. In the first graph we can see that the R_X variable has no edges at all which means that the missingness type falls into MCAR. The second graph shows an dependence structure between the R_X variable and an fully observed variable Y_o but no edge between R_X and an partially observed variable, namely X and no bi-directed edges, which means that here we can assume the missingness to be MAR. The last graph shows an edge between the R_X variable and an partially observed variable since Y_m has now also missing entries, hence this example tells us that the data of X is MNAR.

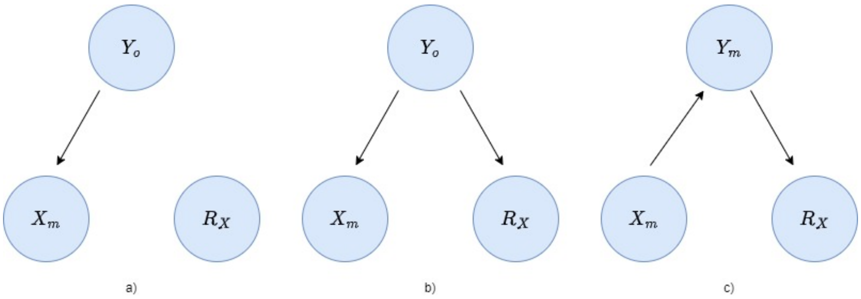


Figure 5: Example for detecting a) MCAR, b) v-MAR and c) MNAR in a m-graph

While the classification after having obtained the m-graph is indeed quite easy, the construction of the m-graph itself is not straightforward. Of course we could apply the widely used traditional pc-algorithm 3.5.1 to discover the causal relationships between missingness indicator variables and the other variables at hand but we have to keep in mind that this algorithm is designed for complete data and will at least theoretically not lead to unbiased results. We will check later if this is true also in practical use in the simulation study 4.

Ruibo Tu and others have presented in their paper *Causal Discovery in the Presence of Missing Data* (29) an adapted pc-algorithm for missing values mvpc which solves the problem of model violations. This algorithm will be introduced in section 3.5.2. It is important to note that the aim of this algorithm presented in the paper was in the first place to model a better and unbiased graph of the original variable set in contrast to the traditional pc-algorithm but without including missingness indicator variables and detecting the type of missingness. But before we have a deeper look at the adapted version of the pc-algorithm we have to first understand how the traditional one works.

3.5 Modelselection for DGGMs

3.5.1 Pc-algorithm

The *pc-algorithm* is an example for constraint based learning. Constraint based learning means that one can derive constraints which every distribution generated from a given causal structure must obey. Such constraints can be e.g. conditional independence statements which can be derived with the help of the partial correlation matrix and some independence test as we will see later. Constraint based learning checks for such constraints given data and thus ideally can reverse-engineer the causal structure of the data generating mechanism. Hence the goal of the *pc-algorithm* is to estimate the true causal structure of the data at hand which can be represented as a CPDAG.

The algorithm can be divided in two main parts where the first step is to estimate the skeleton, thus the undirected graph. In the second step one has to orient the unshielded triples into unshielded colliders if possible to receive in the end the CPDAG. But before we dive deeper into the algorithm we have to introduce shortly undirected gaussian graphical models UGGM to be able to understand how the skeleton is build. (6) (11) (14) (15)

Addendum: Undirected gaussian graphical models

The setup for UGGMs is the same as for DGGMs, presented in section 3.3.2. This means that of course the underlying density and its parameters as well as the precision matrix K and the correlation matrix do not change. In particular this also means that the global Markov property holds and that the entry k_{uv} of the precision matrix will again be zero if there is no edge between the nodes u and v. We call the resulting graph also *dependence graph* since it holds for all u, v, that if u and v are not adjacent, then $u \perp\!\!\!\perp v | V \setminus \{u, v\}$. The only thing which is different in UGGMs is the derivation of the joint likelihood. In the undirected case we do not use the formula from equation 2 since this one was derived by using the ordering from the variables which we do not have now anymore but instead we use the cliques $C_1, ..., C_k$ from chapter 3.1 of the graph to calculate the joint density. We say that the joint density $f(x_V)$ factorises according to the graph if for some functions $g_i()$ that depend only on x through x_{C_i} the joint density can be written as follows:

$$f(x_V) = \prod_{i=1}^{k} g_i(x_{C_i})$$

The set of this cliques is called the *generating class* for the model.

Models with closed-form maximum likelihood estimates are called decomposable. A graph is decomposable if and only if it is triangulated. Since in our case we look only at graphs which are acyclic they are also triangulated and hence decomposable. Therefore it seems appropriate to search for the maximum likelihood estimates in the next step. Hence we first have to build the likelihood and the log likelihood, respectively. To do that we denote the matrix of sums of squares and products as $W = \sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^T$ and the empirical covariance matrix as $S = \frac{W}{n}$. The log-likelihood based on the sample is then

$$l(K, \mu) = \frac{n}{2}\, log\, det(K) - \frac{n}{2}\, tr(KS) - \frac{n}{2}\,(\bar{x} - \mu)^T(\bar{x} - \mu)$$

where $tr(KS) = \sum_u \sum_v s_{uv}\, k_{uv}$. We are mainly interested in K therefore we look at the profile likelihood of K which is

$$l(K, \hat{\mu}) = \frac{n}{2}\, log\, det(K) - \frac{n}{2}\, tr(KS) \tag{5}$$

since for fixed K the log likelihood is maximised for $\hat{\mu} = \bar{x}$ and hence the last term becomes zero. The only elements s_{uv} of S that contribute to the likelihood are those for which the corresponding elements k_{uv} of K are not equal to zero.

If the UGGM has generating class $\mathcal{C} = \{C_1, ..., C_k\}$ it can be shown that the submatrices $S_{C_jC_j}$, for $j = 1, ..., k$ together with the sample mean $\bar{x}$ jointly form a set of minimal sufficient statistics. A submatrix of M_{AB} with entries m_{uv} for $u \in A$ and $v \in B$ is obtained by taking a matrix M with entries m_{uv} for $u \in V$ and $v \in V$ and two subsets $A \subset V, B \subset V$. The maximum likelihood estimate is then derived by finding the unique solution to the system of equations

$$\hat{\mu} = \bar{x},\ \hat{\Sigma}_{C_jC_j} = S_{C_jC_j}, j = 1, ..., k$$

which fulfills the restrictions on the concentration matrix. It follows that the MLE of the covariance between any pair of variables which are neighbours in the graph is equal to the corresponding empirical quantity i.e under the saturated model with no conditional independence restrictions $\hat{\Sigma} = S$ and hence also $\hat{K} = S^{-1}$, provided S is not singular. Neighbours of e.g. u are all nodes which are adjacent to u.

An iterative algorithm called iterative proportional scaling algorithm for computing maximum likelihood estimates for this graphical gaussian models was proposed by Speed and Kiiveri. We will introduce just the idea of this algorithm, for further theory see (26).

As a starting point for the precision matrix K we choose the identity matrix. Let $C \in \mathcal{C}$ be an element in the generator class and $B = N\backslash\mathcal{C}$. The submatrix K_{CC} of K is modified by an increment E so as to satisfy the constraints by the likelihood equations, namely $\hat{\Sigma}_{CC} = S_{CC}$ A cycle in the algorithm repeats this for each generator.

The required increment matrix E can be found as follows: We require that

$$\begin{pmatrix} S_{CC} & * \\ * & * \end{pmatrix} = \begin{pmatrix} K_{CC} + E & K_{CB} \\ K_{BC} & K_{BB} \end{pmatrix}$$

where $*$ denotes unspecified. Standard results on the inverse of partitioned matrices gives

$$S_{CC} = (K_{CC} + E - K_{CB}K_{BB}^{-1}K_{BC})^{-1}$$

Since we want to get the increment E we have to rewrite the equation as

$$\begin{aligned} E &= S_{CC}^{-1} - (K_{CC} - K_{CB}K_{BB}^{-1}K_{BC}) \\ &= S_{CC}^{-1} - \Sigma_{CC}^{-1} \end{aligned}$$

so K_{CC} is updated in each step as

$$K_{CC} \leftarrow S_{CC}^{-1} + K_{CB}K_{BB}^{-1}K_{BC}$$

until convergence.

The maximised value of the profile likelihood in equation 5 is $\frac{n}{2}\, log\ det(\hat{K}) - \frac{nd}{2}$ because $tr(\hat{K}S) = tr(\hat{K}\hat{\Sigma}) = d$ since $\hat{\Sigma}$ and S differ only on those entries for which $k_{uv} = 0$ as we have seen before. Thus the deviance of the model is

$$D = 2(\hat{l}_{sat} - \hat{l}) = n\, log(\frac{det(S^{-1})}{det(\hat{K})}) = -n\, log\ det(S\hat{K})$$

We could use the deviance to build the likelihood test and test conditional independence hypothesis. We will not introduce this procedure here since the *pc-algorithm* uses another approach to test for conditional independences, namely based on the asymptotic normality of Fisher's z transformation of the partial correlation.

The sample partial correlation $\rho_{u,v|\mathbf{k}}$ can be calculated via regression, inversion of parts of the covariance matrix like in equation 4 or recursively by using the following identity: for some $h \in \mathbf{k}$

$$\rho_{u,v|\mathbf{k}} = \frac{\rho_{u,v|\mathbf{k}\backslash h} - \rho_{u,h|\mathbf{k}\backslash h}\ \rho_{v,h|\mathbf{k}\backslash h}}{\sqrt{(1 - \rho^2_{u,h|\mathbf{k}\backslash h})(1 - \rho^2_{v,h|\mathbf{k}\backslash h})}}$$

where $\mathbf{k}$ is a subset of the neighbours of u excluding v of the complete undirected graph. To be able to test whether a partial correlation is zero or not, one can apply Fisher's z-transformation

$$Z(u,v|\mathbf{k}) = 0.5\, log\left(\frac{1+\hat{\rho}_{u,v|\mathbf{k}}}{1-\hat{\rho}_{u,v|\mathbf{k}}}\right)$$

Using classical decision theory we can reject the null-hypothesis $H_0 : \rho_{u,v|\mathbf{k}} = 0$ against the two sided alternative $H_1 : \rho_{u,v|\mathbf{k}} \neq 0$ if $\sqrt{n - |\mathbf{k}| - 3}\, Z(u,v|\mathbf{k}) > \Phi^{-1}(1 - \frac{\alpha}{2})$ with significance level α and cdf of the standard normal distribution $\Phi(\cdot)$. Based on that we use the *if*-statement $\sqrt{n - |\mathbf{k}| - 3}\ Z(u,v|\mathbf{k}) \leq \Phi^{-1}(1 - \frac{\alpha}{2})$ to decide whether two variables are conditionally independent. We remove step by step the edges from the complete graph where we were able to reject the null hypothesis of conditional independence wrt. the given significance level. In order to be able to infer from conditional independence to non existence of a edge between two nodes in a graph we have to make a further assumption, namely the *Causal Faithfulness Assumption.* This assumption states, that the conditional independence relations correspond to d-separations and vice versa. In general the probability distribution may have additional conditional independence relations that are not entailed by d-separation applied to a graph, but with this assumption we neglect this possibility. We will use a conservative α value of 0.01 to minimise the type 1 error. The tests we are doing are of successively increasing order. That means that we first test marginal independences and then further relations of the form $u \perp\!\!\!\perp v|S$ for $|S| = 1, 2, ...$ and so on. The pc-alogorithm takes advantage of the fact that at any time, when an edge between u and v is tested, it is sufficient to consider sets S which are subsets of the neighbours of u or v. This is very important to consider since we have to avoid performing a huge number of independence tests. The skeleton becomes more and more sparse, while edges are removed and simultaneously the cardinality of S increases, but such sets are very few. This process yields a list of identified conditional independences, denoted as triples (u, v, S) for which $u \perp\!\!\!\perp v|S$ holds. The S sets are called sepsets, since they correspond to sets which d-separate variables u and v in the unknown true graph. These sets are not of primary interest for the skeleton but will be needed when we want to extend the skeleton to the CPDAG in the next step. The present procedure results in the end in the skeleton of the causal structure and therefore closes also the first step of the pc-algorithm. (14) (15) (6) (11)

We have seen in the addendum to *undirected gaussian graphical models* 3.5.1 how the undirected graph and hence also the skeleton is build, therefore we can move on with the second part of the pc-algorithm, namely the extension of the skeleton to a CPDAG. We can distinguish in this step two substeps. First we orient the unshielded triples into unshielded colliders, where it is possible. An unshielded triple are three nodes a, b, c with $a - b, b - c$ where a and c are not connected. If node b is not in $sepset(a, c)$, the unshielded triples $a - b - c$ is oriented into an unshielded collider $a \rightarrow b \leftarrow c$. Otherwise b is marked as a non-collider on $a - b - c$. Next , the partially directed graph we just received is checked using three further rules to see if any other edges can be oriented while avoiding new unshielded colliders or cycles. You can see that in this step of the pc-algorithm we do not have to do further calculations but only consider some rules. Kalisch and Bühlmann (14) have combined the two substeps into one algorithm which can be seen below 1.

Algorithm 1 Extending the skeleton to a CPDAG

```
Input: Skeleton, seperation sets $S$

Output: CPDAG

for all pairs of nonadjacent variables $u, v$ with common neighbour $k$ do
  if $k \notin S(u, v)$ then
    Replace $u - k - v$ in Skeleton by $u \rightarrow k \leftarrow v$
  end if
end for
```

In the resulting PDAG, try to orient as many undirected edges as possible by repeated application of the following three rules:

R1 Orient $v - k$ into $v \rightarrow k$ whenever there is an arrow $u \rightarrow v$ such that u and k are nonadjacent.

R2 Orient $u - v$ into $u \rightarrow v$ whenever there is a chain $u \rightarrow k \rightarrow v$.

R3 Orient $u - v$ into $u \rightarrow v$ whenever there are two chains $u - k \rightarrow v$ and $u - l \rightarrow v$ such that k and l are nonadjacent.

R4 Orient $u - v$ into $u \rightarrow v$ whenever there are two chains $u - k \rightarrow l$ and $k \rightarrow l \rightarrow v$ such that k and l are nonadjacent.

By completing this step, the pc-algorithm terminates and the final $CPDAG$ is found. (14) (15) (6) (11)

3.5.2 Missing value pc-algorithm

We have discussed in section 3.4 that we have to adapt the traditional pc-algorithm when having missing data at hand since this algorithm is valid only for complete data analysis. An alternative algorithm is the so called *missing-value pc-algorithm*, short *mvpc*, which takes into account that some observations are missing. In the first place the aim of this algorithm is not to produce a m-graph but to produce a more realistic version of the graph based on the pc-algorithm when comparing it to the true underlying dependence graph. But we can use this algorithm also for our purpose of get a better version of the m-graph. Before we apply the algorithm we have to make some assumptions which are restrictive but necessary. (29)

Assumption 1 There is no confounder or selection bias relative to the set of observed variables. Further we assume causal sufficiency, which means that the latent variable set U is an empty set

Assumption 2 No missingness indicator can be the cause of any observed variable.

This means , if variables of interest X and Y are not d-separated by a variable set $Z \subseteq V \backslash (X, Y)$, they are not d-separated by Z together with their missingness indicators R_X, R_Y. Hence if they are conditionally independent given Z together with their missingness indicators, they are conditionally independent given only Z. Of course we cannot directly verify whether they are conditionally independent given Z and their missingness variables because we do not have the records for the considered variables when their missingness indicators $R = 1$. Therefore we need further assumptions:

Assumption 3 Any conditional independence relation in the observed data also holds in the unobserved data. This also means that there is no accidental conditional independence relation caused by missingness.

Assumption 4 No causal interactions between missingness indicators.

Assumption 5 Missingness in a variable that is caused directly by itself is called self masking missingness. In the m-graph this is depicted by $X \rightarrow R_X$, for $X \in V_m$. We assume that there are no such edges in the m-graph. This assumption is very restrictive in our case since self masking missingness is one type of MNAR and cannot be seen from the graph but cannot be circumvented due to the reason mentioned in assumption 2.

Now we can apply the mvpc-algorithm which looks as follows (29):

Algorithm 2 Missing value pc-algorithm

1: *Skeleton search with deletion-based PC:*

i) Graph initialization:

Build a complete undirected graph on the node set V as in traditional pc-algorithm.

ii) Causal skeleton discovery:

Remove edges in the graph with the same procedure as the pc-algorithm with the test-wise deleted data. Test-wise deletion means that we only delete records with missing values for variables involved in the current CI test when performing the pc-algorithm. This is far more data-efficient than the naive approach of list-wise deletion, i.e. deleting all records that have any missing value and then applying the pc-algorithm to the remaining data.

2: *Detecting direct causes of missingness indicators:*

For each variable $V_i \in V$ containing missing values and for each j that $j \neq i$, test the CI relation of R_i and V_j. If they are independent given a subset of $V \backslash (V_i, V_j), V_j$ is not a direct cause of R_i.

3: *Detecting potential extraneous edges:*

For each $i \neq j$, if V_i and V_j are adjacent and have at least one common adjacent variable or missingness indicator, the edge between V_i and V_j is potentially extraneous.

4: *Recovering the true causal skeleton:*

Perform correction method for removing the extraneous edges in the graph. 3

5: *Determining the orientation:*

Orient edges in the graph with the same orientation procedure as the traditional pc-algorithm.

The first thing we have to discuss is why it is not sufficient to just use the test-wise deletion pc-algorithm TD-PC, which would shorten the above algorithm to the steps one and five. At least in the case of MCAR TD-PC gives asymptotically correct results since $(V_m, V_o) \perp\!\!\!\perp R$ is satisfied. Let's say $R_y \perp\!\!\!\perp (X, Y, Z)$ holds which is represented in the graph by the absence of an edge between R_y and the other variables. Consequentially, we have $X \perp\!\!\!\perp Y|Z \leftrightarrow X \perp\!\!\!\perp Y|(Z, R_y)$. Using now assumption 2 3.5.2 we can conclude that $X \perp\!\!\!\perp Y|Z \leftrightarrow X \perp\!\!\!\perp Y^{\star}|(Z, R_y = 0)$. When applying the CI test to the test-wise deleted data of concerned variables X, Y, Z, we test whether $X \perp\!\!\!\perp Y^{\star}|(Z, R_y = 0)$ holds. Therefore, CI results imply d-separation relations of concerned variables in m-graphs when data are MCAR, which guarantees the asymptotic correctness of TD-PC.

Having MAR or MNAR data at hand the TD-PC-algorithm does not longer provide asymptotic correct results. The causal skeleton given by TD-PC has no missing edges though, but may contain extraneous edges. Extraneous edges are produced if dependence relations in test-wise deleted data imply the wrong corresponding relations in the m-graph because $X \not\perp\!\!\!\perp Y|(Z, R_x = 0, R_y = 0, R_z = 0) \nrightarrow X \not\perp\!\!\!\perp Y|Z$. Fortunately, such extraneous edges appear only under special circumstances which makes it possible to remove them afterwards. You can see an example in the following graph 6. W is the direct common effect of X and Y and the missingness indicator R_y is a descendant of W. Thus, the extraneous edge occurs between X and Y in the causal skeleton produced by TD-PC. The reason for that is that under the assumptions 3.5.2 for at least one variable in $X \cup Y \cup Z$ its missingness indicator is either the direct common effect or a descendant of the direct common effect of X and Y if we suppose that X and Y are not adjacent in the true causal graph and that for any variable set $Z \subseteq V \backslash (X, Y)$ such that $X \perp\!\!\!\perp Y|Z$, it is always the case that $X \not\perp\!\!\!\perp Y|(Z, R_x = 0, R_y = 0, R_z = 0)$. A detailed proof for that proposition can be found in (29). With this information we can improve the TD-PC-algorithm by using the mvpc-algorithm which considers this cases in step three. Note that step two is equivalent to performing TD-PC on $V \cup R$, which we will do in the simulation study since we want to depict the R variables themselves in the graph.

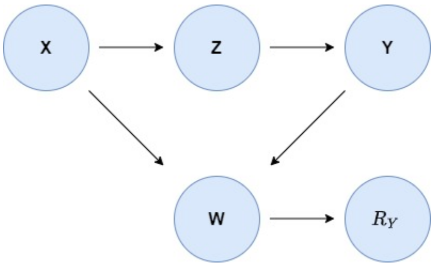


Figure 6: Example for potentially extraneous edge produced by TD-PC between X and Y in MAR scenario

The most difficult part of the algorithm is step four where we have to perform a correction for these extraneous edges. Ruibo Tu and others propose the permutation-based correction (29). This method does not cover all possible missingness cases but is nevertheless a substantial improvement in contrast to the TD-PC-algorithm. The conditions for the validity of PermC are

(i) $(R_x, R_y, R_z, R_w) \perp\!\!\!\perp (X, Y, Z)|W$, where the variable set W is the set of direct causes of missingness indicators R_x, R_y, R_z. If variables in W also have missing values, the direct causes of their missingness indicators R_w are also included in W.

(ii) In the m-graph, the missingness indicators of W follow the condition that $X \perp\!\!\!\perp Y|Z \leftrightarrow X \perp\!\!\!\perp Y|(Z, R_w)$.

For the proof you may have a look at (29).

Algorithm 3 Permutation-based correction

```
Input: data of concerned variables, such as $X, Y, Z$ in figure 6, and the direct causes of their corresponding missingness indicators, such as the direct cause $W$ of $R_y$
Output:: The CI relations among concerned variables, such as the CI relations among $X, Y, Z$
1: Delete records containing any missing value. We denote the deleted dataset by $D_d$, and denote the original dataset by $D_o$.
2: Regress $X, Y, Z$ on $W$ with $D_d$.
3: Shuffle data of $W$ in $D_o$, denoted by $W^S$, and delete records containing any missing value in $D_o$ (included $W^S$).
4: Generate virtual data of $\hat{X}, \hat{Y}, \hat{Z}$ with $W^S$ and the residuals.
5: Test the CI relations among $\hat{X}, \hat{Y}, \hat{Z}$ in the generated virtual data.
6: return
    The CI relations among $X, Y, Z$.
```

To gain a better understanding of the algorithm we will go through the steps two, three and four in more detail using the graph 6 (29). We have already discussed that in this example samples from the joint distribution $P(X, Y, Z)$ are not available in the observed dataset. In this case, we test the CI relations in the test-wise deleted data from $P(X, Y^\star, Z | R_y = 0)$, which leads to producing the extraneous edge between X and Y. The above algorithm introduces a random variable W which is the direct cause of R_y to reconstruct the dataset and then marginalise it out. With W, the joint distribution can be estimated by learning the model for $P(X, Y, Z|W)$ from test-wise deleted data, further by plugging in the values of W in the dataset, as data samples from $P(W)$, and last by disregarding the input W and keeping the generated virtual data for (X, Y, Z) to marginalise W out. We can then test CI relations in the complete data when virtual data of X, Y, Z that follow the joint distribution is given.
Since we do not have samples from $P(X, Y, Z|W)$, we have to somehow generate them in step two of the algorithm. Therefore we can use our assumptions 3.5.2 and a further assumption which is that our data is normal distributed to apply linear regression and learning the

equivalent model $P(X, Y^\star, Z|W, R_y = 0)$ as :

$$X = \beta_1 W + \epsilon_1,\ Y = \beta_2 W + \epsilon_2,\ Z = \beta_3 W + \epsilon_3$$

where β denotes the parameter of the linear regression and ϵ the residual.
Now we can sample the input values from the probability distribution $P(W)$. One has to shuffle the values of W in the observed dataset such that $P(W^S|R_y = 0) = P(W^S)$ where W^S denotes the shuffled W, because inputting the test-wise deleted data of W and adding the residuals from the linear regression models would yield the data follow the conditional distribution $P(W|R_y = 0)$ instead of $P(W)$. This is done in step three. The next step is to replace W with $W^\star$ and to build again regression models

$$\hat{X} = \beta_1 W^S + \epsilon_1,\ \hat{Y} = \beta_2 W^S + \epsilon_2,\ \hat{Z} = \beta_3 W^S + \epsilon_3$$

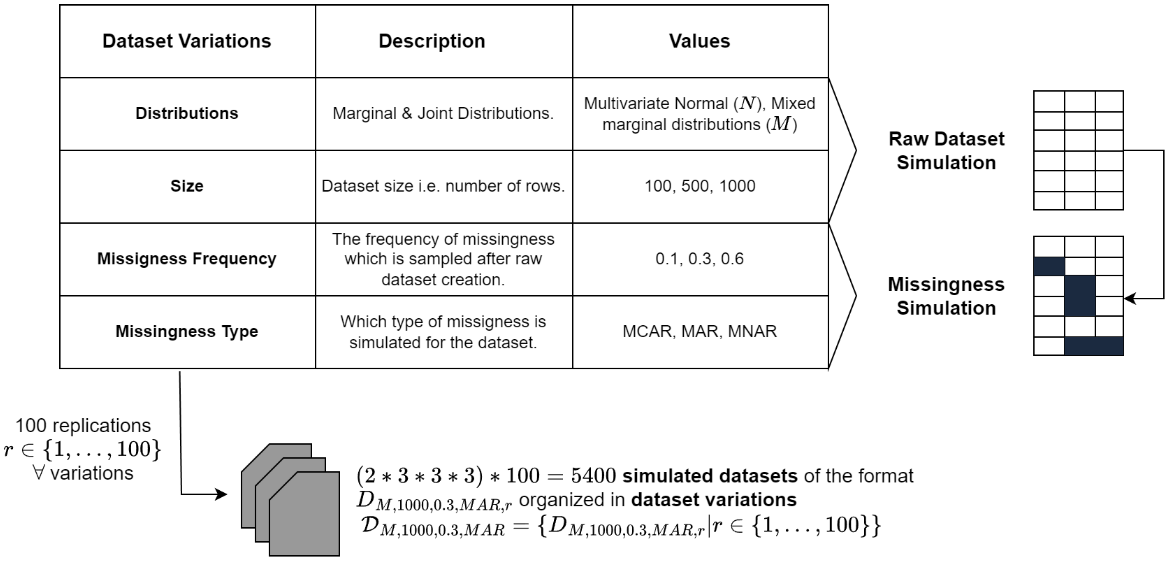
By plugging in the permutation-based correction algorithm 3 into the fourth step of the mvpc-algorithm 2 we have received at least in theory a better graph than the graph we would have obtained by using the simple TD-PC-algorithm and for sure a better graph than when applying list-wise deletion pc-algorithm. We will compare the results of the two main procedures, namely the pc- and the mvpc-algorithm, in the simulation study 4. (29) (19) (27)(4)

4 Simulation study

In order to examine if it is indeed also in practice possible to detect the three different types of missing values using graphical models and improve their imputation with this additional knowledge, a simulation study was conducted, which will be discussed in the next sections. The main focus will be to first examine if there are differences in the prediction quality between the pc-algorithm and the mvpc-algorithm. Secondly to check if the imputations for MNAR variables are biased not only in theory but also in practice and furthermore can be corrected by using weighted knn instead of traditional multiple imputation to impute the values.

Since especially the mvpc-algorithm only covers datasets which contain just normal distributed variables or binary variables, the obvious choice was to start with a multivariate normal distributed data generating process, for which all explanatory variables are independent. Though this is not a very realistic scenario and requires an extension. Therefore the second data generating process which was used aims to represent a more realistic dataset and will be used for the main analysis, even if it violates in theory some of the model assumptions like the normality assumption.

The following figures 7 8 9 provide an overview over the general structure of the simulation study. It can be separated in three main parts, namely the simulation of the data itself 7, the modelling and evaluation of the graphical models 8 and the imputations 9. In the following sections we will discuss these steps shown in the figures below in more detail.



Dataset Variations	Description	Values
Distributions	Marginal & Joint Distributions.	Multivariate Normal (N), Mixed marginal distributions (M)
Size	Dataset size i.e. number of rows.	100, 500, 1000
Missingness Frequency	The frequency of missingness which is sampled after raw dataset creation.	0.1, 0.3, 0.6
Missingness Type	Which type of missingness is simulated for the dataset.	MCAR, MAR, MNAR

$(2 * 3 * 3 * 3) * 100 = 5400$ **simulated datasets** of the format $D_{M,1000,0.3,MAR,r}$ organized in **dataset variations** $\mathcal{D}_{M,1000,0.3,MAR} = \{D_{M,1000,0.3,MAR,r} | r \in \{1, \dots, 100\}\}$

Figure 7: Scheme of main steps in the data simulation process

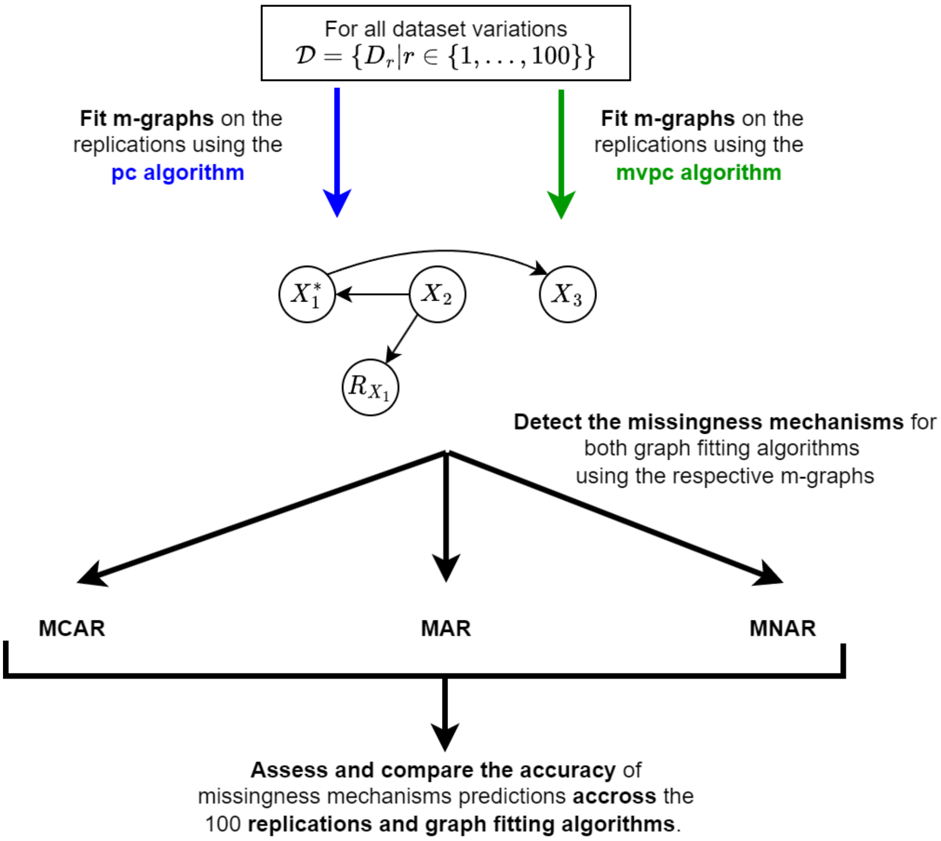

For all dataset variations
$\mathcal{D} = \{D_r | r \in \{1, \ldots, 100\}\}$
Fit m-graphs on the replications using the pc algorithm
Fit m-graphs on the replications using the mvpc algorithm
X_1^*
X_2
X_3
R_{X_1}
Detect the missingness mechanisms for both graph fitting algorithms using the respective m-graphs
MCAR
MAR
MNAR
Assess and compare the accuracy of missingness mechanisms predictions accross the 100 replications and graph fitting algorithms.

Figure 8: Scheme of main steps in the graphical modelling part of the simulation study

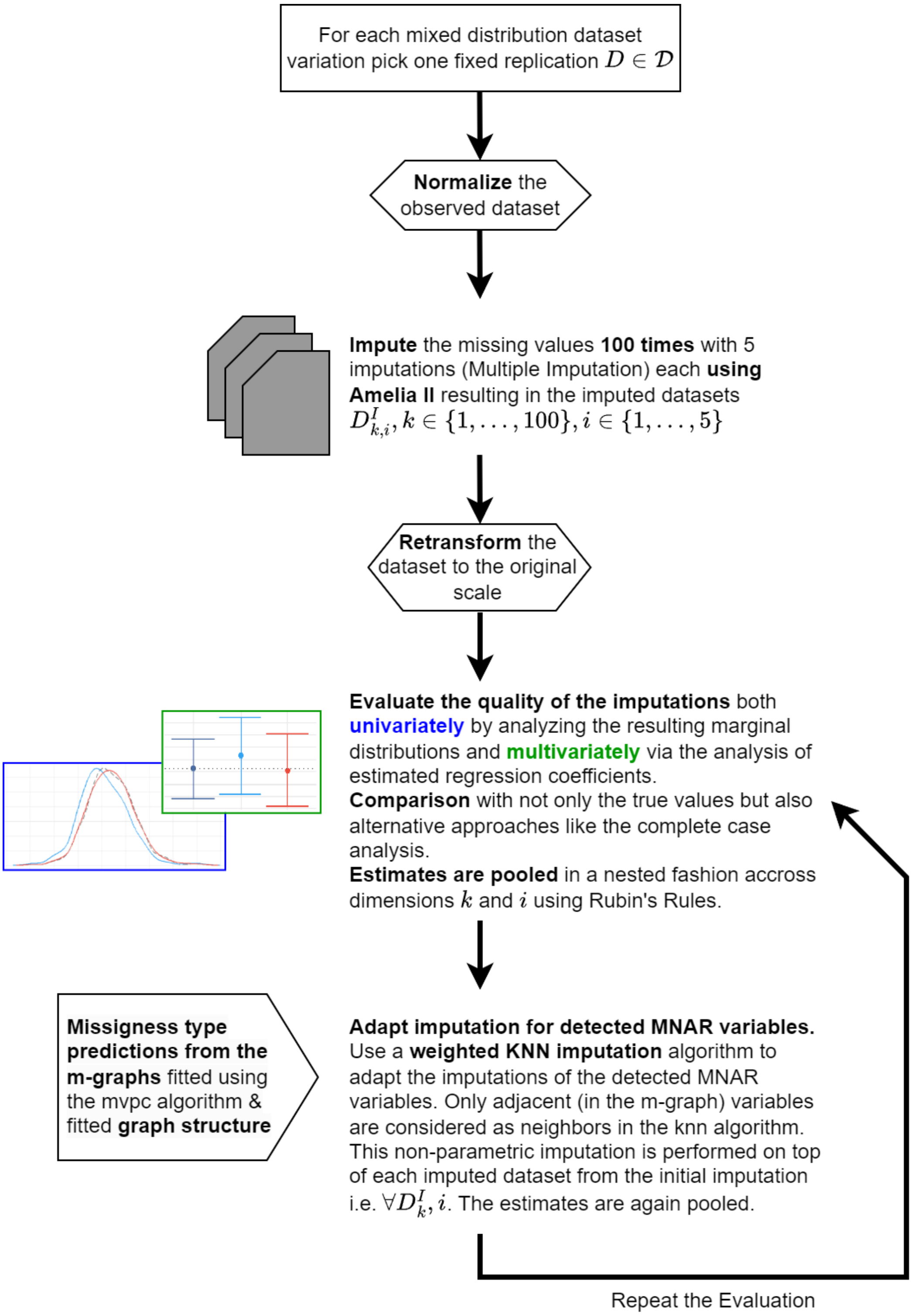

For each mixed distribution dataset variation pick one fixed replication $D \in \mathcal{D}$
Normalize the observed dataset
Impute the missing values **100 times** with 5 imputations (Multiple Imputation) each **using Amelia II** resulting in the imputed datasets $D^I_{k,i}, k \in \{1, \dots, 100\}, i \in \{1, \dots, 5\}$
Retransform the dataset to the original scale
Evaluate the quality of the imputations both **univariately** by analyzing the resulting marginal distributions and **multivariately** via the analysis of estimated regression coefficients.
Comparison with not only the true values but also alternative approaches like the complete case analysis.
Estimates are pooled in a nested fashion accross dimensions k and i using Rubin's Rules.
Missingness type predictions from the m-graphs fitted using the mvpc algorithm & fitted **graph structure**
Adapt imputation for detected MNAR variables. Use a **weighted KNN imputation** algorithm to adapt the imputations of the detected MNAR variables. Only adjacent (in the m-graph) variables are considered as neighbors in the knn algorithm. This non-parametric imputation is performed on top of each imputed dataset from the initial imputation i.e. $\forall D^I_k, i$. The estimates are again pooled.
Repeat the Evaluation

Figure 9: Scheme of main steps in the imputation part of the simulation study

4.1 Data simulation

You can find the dependence structure of the data simulation model illustrated in figure 10. We have one target variable y which we will need later when building the regression model and nine explanatory variables which are both categorical with two or three categories and continuous. The continuous ones follow different distributions as the normal distribution but also others like the beta or gamma distribution. It can also be seen that some explanatory variables are dependent from each other. The concrete dependency structures can be obtained from the formula shown in the figure. Following the formulas, 1000 samples were drawn from the different distributions for each variable. Later in the simulation study we will investigate also the influence of the dataset size by shortening it to 500 and 100 entries, respectively.

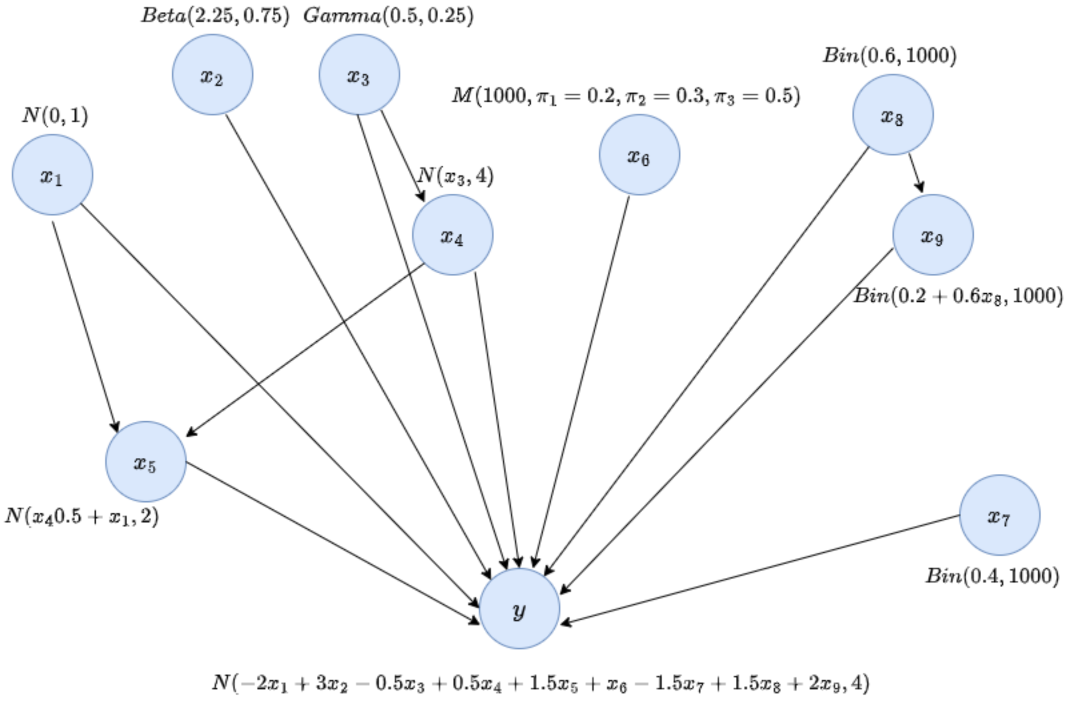


Figure 10: Chart of the simulated data, including the information about the distribution of the single variables and their connection to each other

In order to simulate afterwards the missing entries the R-package *simstudy* (7) was used. We simulated for each missing type a separate dataset, such that one dataset contains only one type of missingness, except of MNAR were it was needed due to the definition of MNAR to include also a MCAR/MAR variable. Further the *tidyverse* package (32) was used to create most of the visualisations using *ggplot2*. The procedure to produce MCAR variables was straightforward by randomly removing entries of the variables analogously to the ratio we want to be missing which was first 0.1 then 0.3 and 0.6, respectively. To be able to simulate MAR and MNAR scenarios one has to additionally calculate different sampling probabilities for the single entries before sampling which entries should become NA with these computed probabilities. To do that we first standardise the variable x on which the missing procedure will depend to avoid problems with different value ranges. After that we input the values into a sigmoid function to have a smooth transition between the probabilities of being missing or not. This looks as follows:

$$P(being\,sampled) = sigmoid(\,multiplier \cdot \frac{x - \hat{\mu}}{\hat{\sigma}})$$

First we tried small values for the multiplier around one to examine if the graphical model can recognise also weak MAR/MNAR relations, which worked quite well, later on the value was fixed at three to get a steeper sigmoid and hence a stronger relation between the dependent variable and the missing indicator in order to intensify the potential bias in the imputations. It turned out that as well the graphical models as the imputations were not influenced by this changes significantly. Therefore only results with multiplier = 3 will be shown.

The only difference between MAR simulation and MNAR simulation is that the variable x on which we depend the missingness process has to be fully observed in the first scenario whereas it has to be partially observed in the second scenario. Thus we made the dependent variable x MCAR for the MNAR category. Further some missing entries were produced by depending on two x variables. In this case we first fix on how the two variables should affect the missingness. For the y variable x_1, which is normal distributed, and x_7, which is binary, should affect the missing process equally strong. Therefore we sample 50% of the missing entries with putting in the x_1 variable in the sigmoid and 50% the x_7 while ensuring the desired missingness frequency and by taking into account that entries might overlap. For x_5 the missingness was influenced approximately by 80% of x_1 and by 20% by x_4. The other simulated dependencies can be seen in the graphs 13 14. Note that we use the notation of chapter 3.4, but here the missing indicator variable R is called $missing_{x_j}$ and j is the index of the corresponding variable.

The figures 11 12 show the MNAR process on the single observations level for one sample. The missing variable which we are interested in is on the x-axis and the MCAR variable which influences the missingness process is on the y-axis. The missing entries are shown in red. We can see that in both plots there are significantly more missing values in the upper part of the figure which means that the probability of an entry to be missing increases by higher values of the variable on the y-axis and hence is not MCAR. The missingness rate was fixed to 0.1 for this visualisation for the sake of clarity. Have a look at the Appendix A for the visualisation of the variables x_5, y and x_2 from the multivariate normal dataset. We will discuss the general structure of the multivariate normal data generatig process briefly in the next section. The plots for MAR look similar to those shown here, with the only difference that now the dependent variable on the y-axis is fully observed. In the MCAR scenario one would not see any patterns since the missingness is by definition completely random.

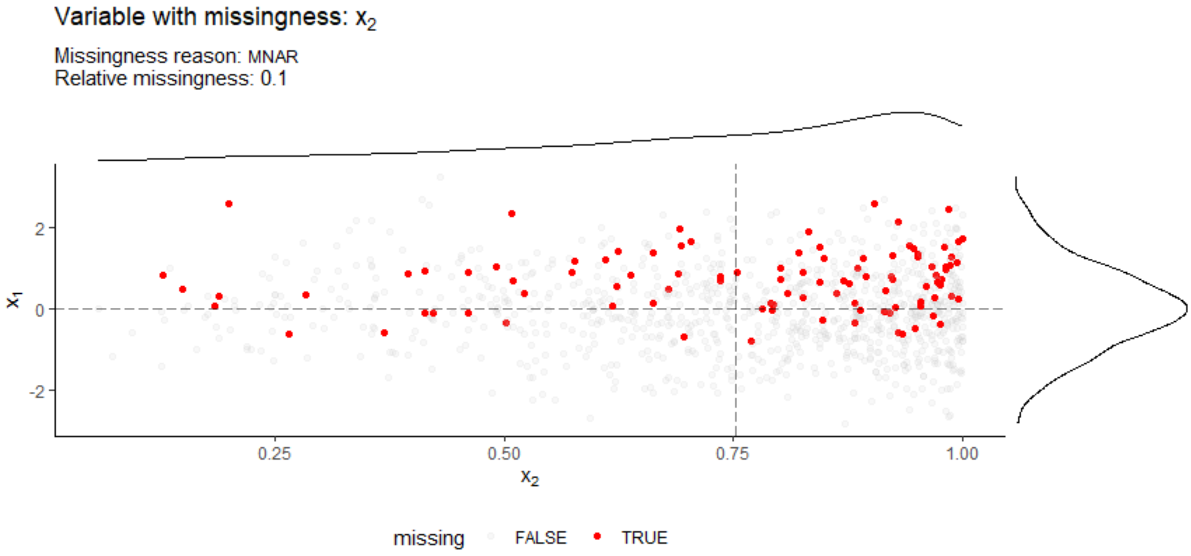


Figure 11: Graph that displays the relationship of the missing values of the Beta distributed variable x_2 on the x-axis and the normal distributed Variable x_1, that contains MCAR values, on the y-axis which influences the missing process of x_2 in such a way that it is MNAR.

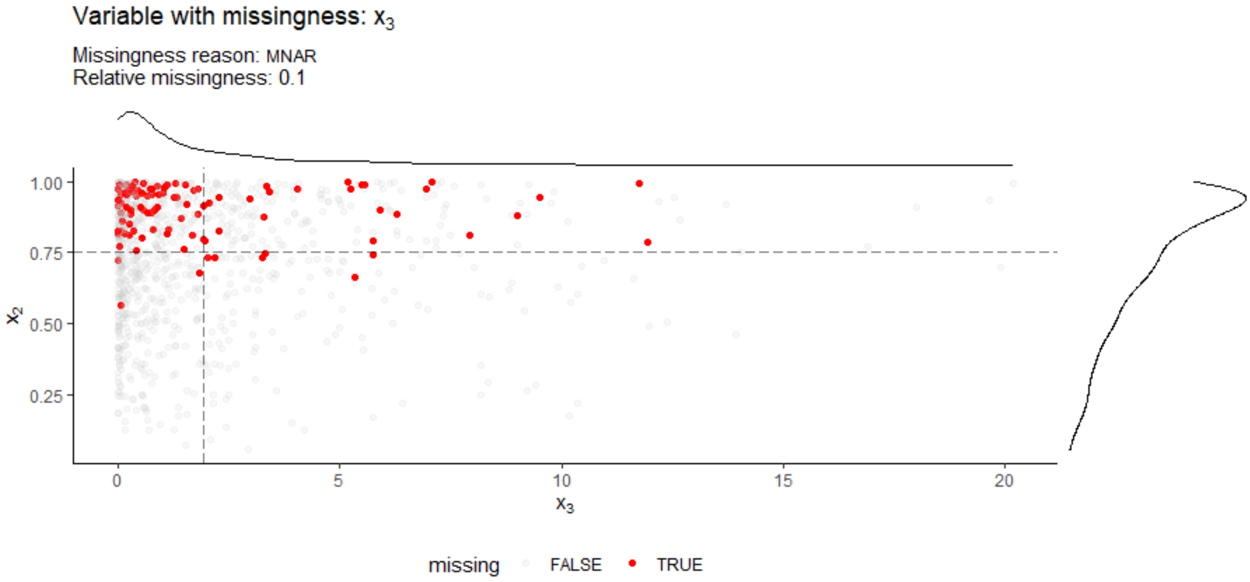


Figure 12: Graph that displays the relationship of the missing values of the Gamma distributed variable x_3 on the x-axis and the Beta distributed Variable x_2, that contains MNAR values, on the y-axis which influences the missing process of x_3 in such a way that it is MNAR.

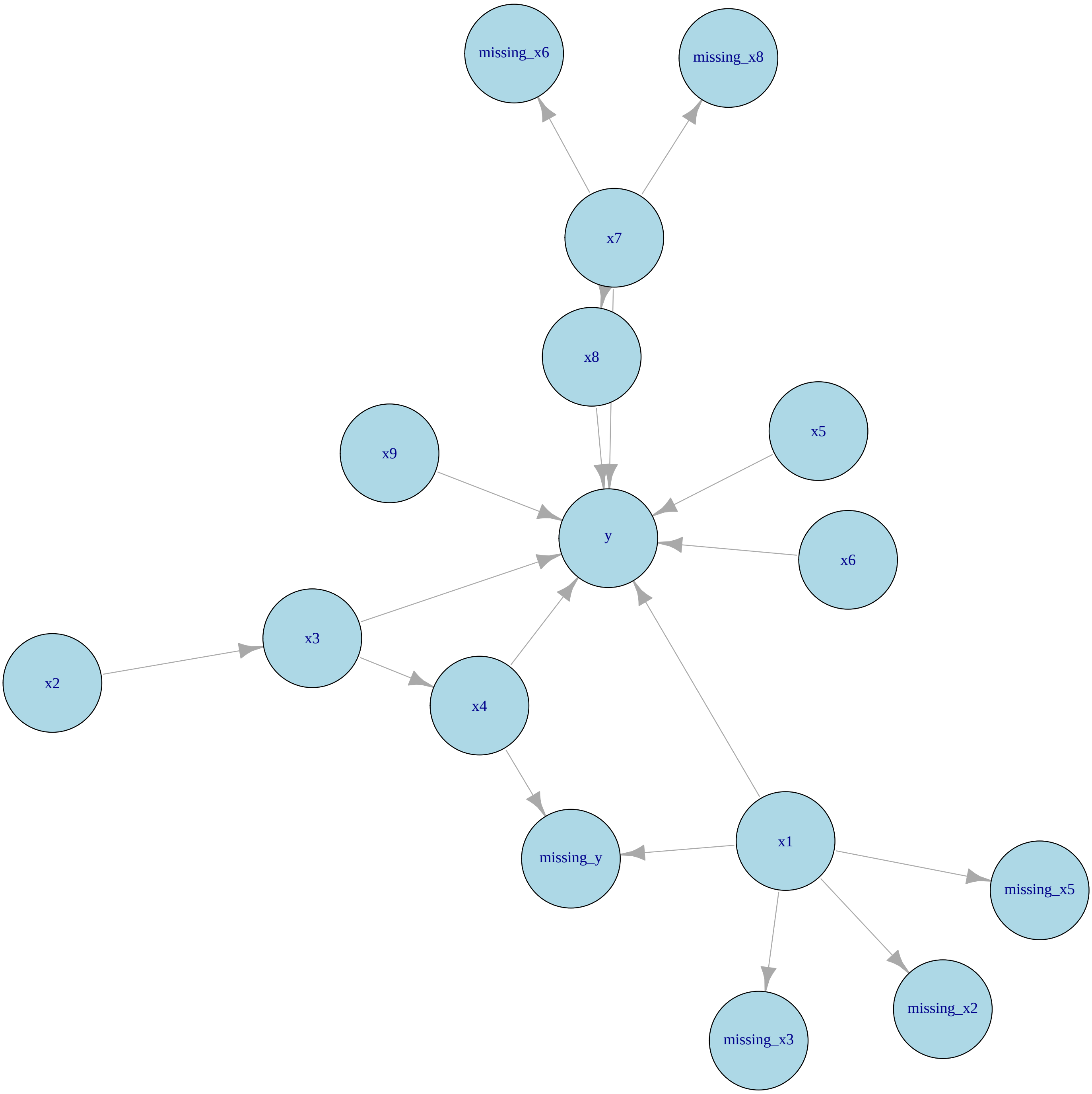


Figure 13: The graph shows what the graphical model should predict ideally in the MAR scenario.

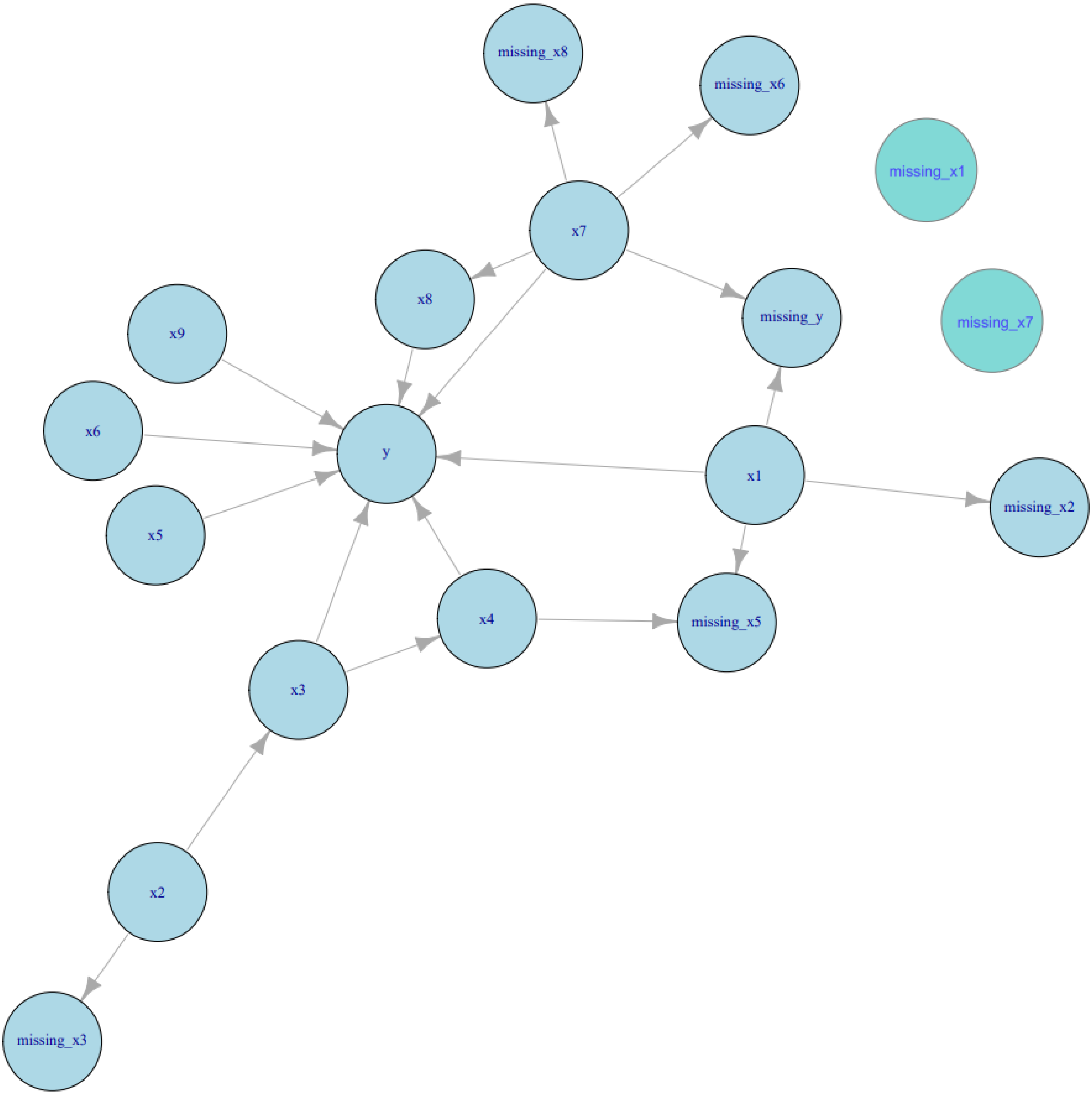


Figure 14: The graph shows what the graphical model should predict ideally in the MNAR scenario.

4.2 M-graphs: Modelling and Evaluation of the results

After having created the datasets we want to work with, we will look at the graphical models produced by the pc-algorithm and the mvpc-algorithm in this section. For the mvpc-algorithm the sligthly adapted code of Ruibo Tu was used for the modelling, which can be found at `https://github.com/TURuibo/mvpc`. The pc-algorithm is implemented in the R-package *pcalg* (16) (8). The main function *pc()* takes as one of the inputs the correlation matrix of the data which we calculated by computing the correlations of the pairwise complete observations to avoid problems with missing entries and to be at the same time as sample efficient as possible. We saw already in figure 13 and figure 14 how the graphs should ideally look like in the MNAR and MAR scenario if the *mvpc-* and *pc-algorithm* would have worked perfectly. Even if we can say that overall the algorithms worked quite well, there are nevertheless some missmodeled relations which we will look at now. For the sake of clarity we will not discuss every single graphical model but focus on the most interesting ones, which are those that differ strongly between the *mvpc-* and the *pc-algorithm* and made more mistakes in modelling the true underlying structure. The other graphical models can be found in the Appendix B or be reproduced by using the r-code provided at `https://github.com/Eleftheria1/Classifying-missing-values-with-graphical-models` as anything else done in the simulation study.

Exemplary evaluation of the m-graphs for one single dataset

We will start with the MCAR scenario. For the missingness rates 0.1 and 0.3 both algorithms worked quite well. The *pc-algorithm* made one mistake at each missingness rate whereas the *mvpc-algorithm* recognised everything correctly at the 0.1 scenario. When having a missingness rate of 0.6 we can see in figure 15 that the y variable and the x_8 variable were predicted to be MAR instead of MCAR, whereas the mvpc-algorithm in figure 16 classified everything correctly. Of course the overall graph structure is not quite good since it does not recognise a lot of relations between variables which have one but this is not the first aim here since we are only interested in classifying the correct types of missingness processes.

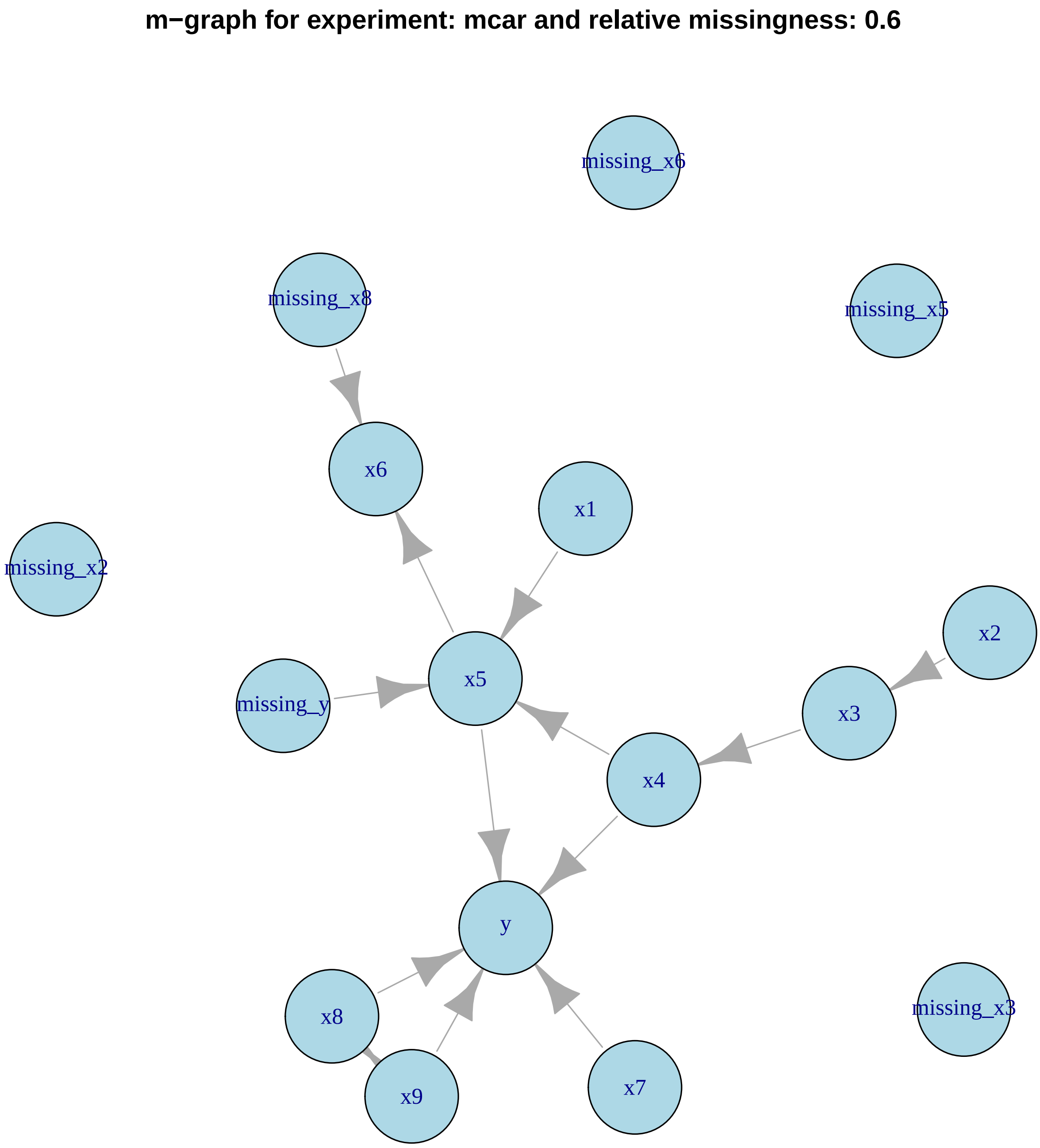


Figure 15: The graph displays the prediction of the graphical model when using the pc-algorithm when having simulated MCAR variables only and a missingness rate of 0.6 for each variable.

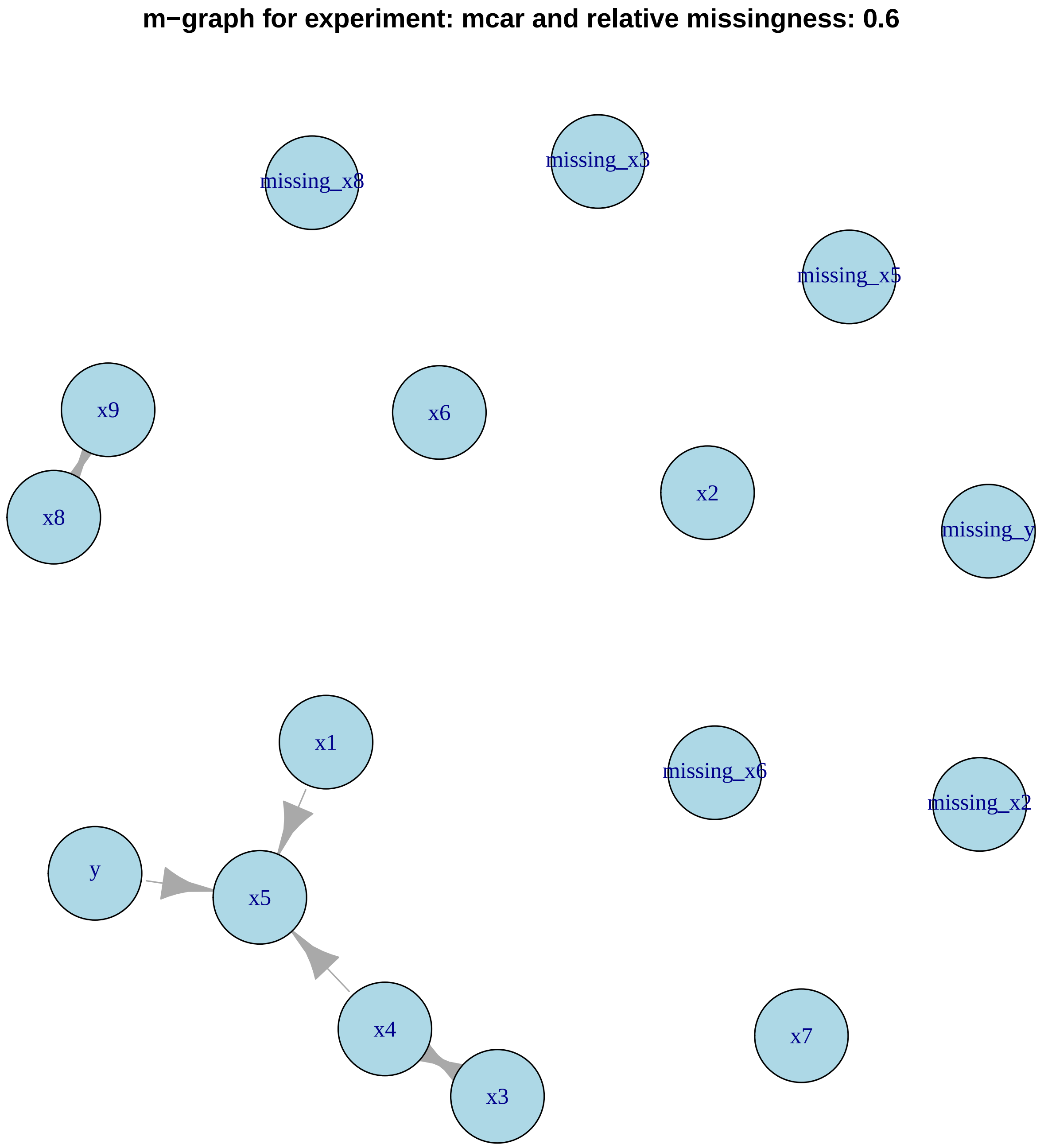


Figure 16: The graph displays the prediction of the graphical model when using the mvpc-algorithm when having simulated MCAR variables only and a missingness rate of 0.6 for each variable.

In the MAR scenario things went very well as for the mvpc as for the pc-algorithm for the missingness rates 0.1 and 0.6. As you can see in figure 17 the pc-algorithm predicted bidirected arrows for all variables which are connected to x_1 which yields an MNAR prediction and also the missingness indicator of x_5 is connected to another missing indicator namely x_6 which again let us assume that we have an MNAR process here. The mvpc-algorithm had in the 0.3 scenario the exactly same problems which can be seen in figure 59 in the Appendix.

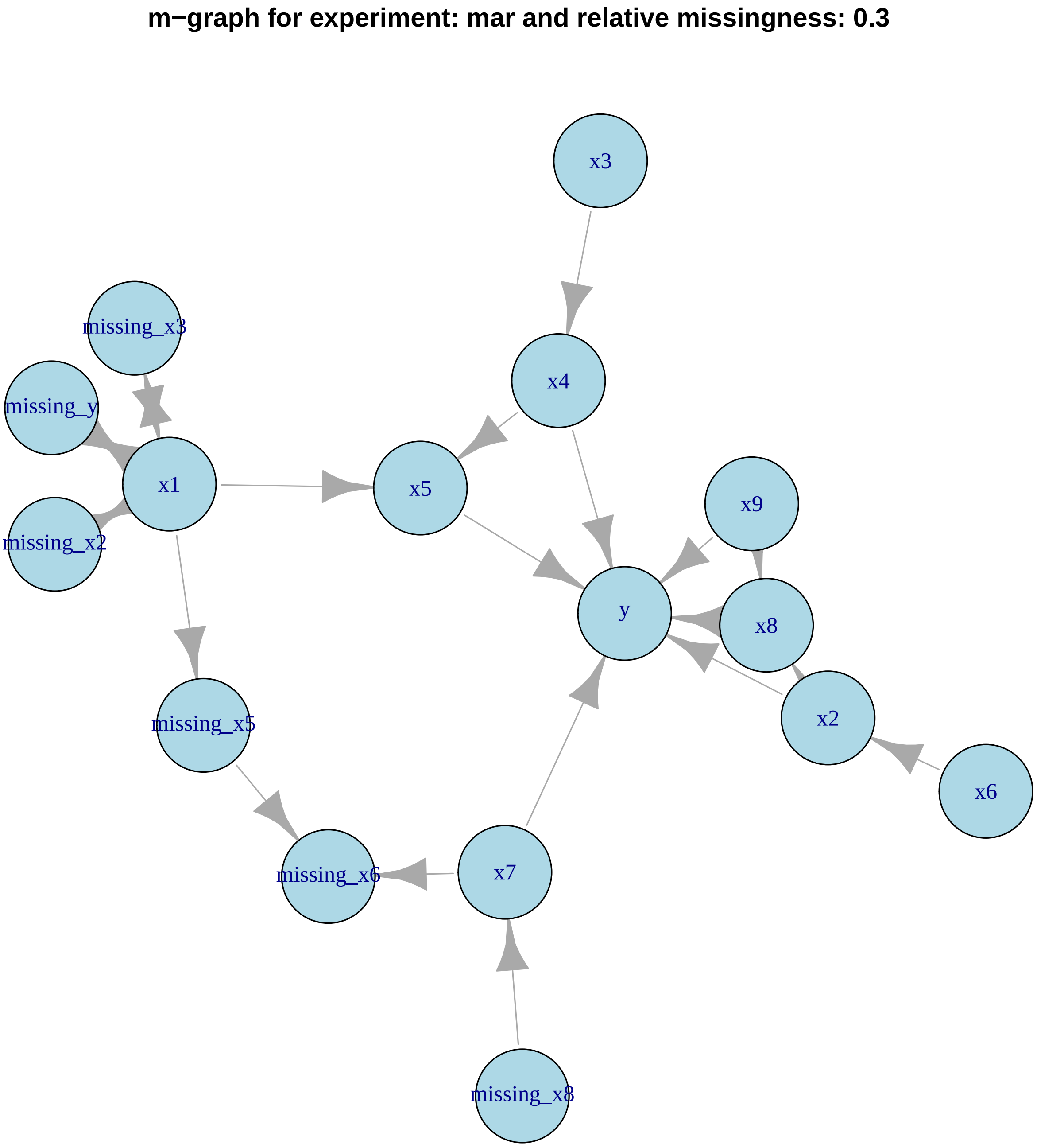


Figure 17: The graph displays the prediction of the graphical model when using the pc-algorithm when having simulated MAR variables only and a missingness rate of 0.3 for each variable.

On the other hand both algorithms worked nearly perfectly when having a missingness rate of 0.3 20 and 0.1 in the MNAR scenario but have some problems recognising one MNAR process and also one MCAR process with a missingness rate of 0.6 18 19. When using the pc-algorithm x_7 which is MCAR is predicted to be MNAR since it has a connection with x_8 which itself is missing, whereas the mvpc-algorithm predicted it correctly. But before concluding that the mvpc-algorithm did a better job in general we have to look at our target variable y which is in many applications of high interest for us and this variable is wrongly predicted as MCAR from the mvpc-algorithm which is undesirable. The pc-algorithm has also not recognised all the true relations of y correctly but at least it would yield the correctconclusion that y is in fact MNAR.

As you can see it is not easy to decide which algorithm works better since they perform both quite similar. Further we have to keep in mind that there is always some randomness in our model and we should not overinterprete the results of one replication.

This is why we will have a look at the averaged performance of those models when repeating the procedure 100 times with slightly different datasets caused due to this randomness and produced by replicating the same data model. Moreover we will change also the size of the dataset to investigate if the algorithms work also for smaller data.

In summary it can be said that the algorithms work in general quite good for this single dataset but nevertheless there are problems in some specific scenarios. Since some theoretical assumptions, especially the normality assumption, are violated, we have simulated also a dataset which contains only normal distributed variables of which only one is missing in the MCAR and MNAR scenario and two are missing in the MNAR scenario to make the MNAR relation possible. Further the covariates are independent from each other to make the structure of the graph as simple as possible to examine if the algorithms work far better under these *perfect* conditions which of course are not very realistic in practice. The theoretical graphical model for MNAR can be seen in figure 23. The theoretical structures for MAR and MCAR cases are the trivial reductions of this graph and are shown in Appendix B. On the whole the algorithms worked very good for this dataset, which is not very surprising but nevertheless also these graphs have for some setups problems with recognising the missing type correctly as in figure 22. Here the missing indicator variable should be connected to x_1 to be classified as MAR but it is not and hence classified as MCAR. We will see also in the following that the performance suffers strongly under the reduction of the sample size even if the dataset is in theory perfectly suitable for using this models. This is why we will not proceed with the analysis of this simplistic normal distributed dataset, since it does not perform much better than the mixed dataset and does not represent a realistic dataset in practice. If someone is still interested in the analysis of this dataset she/he might use the provided r-code to get the results for the imputations, which we will discuss later.

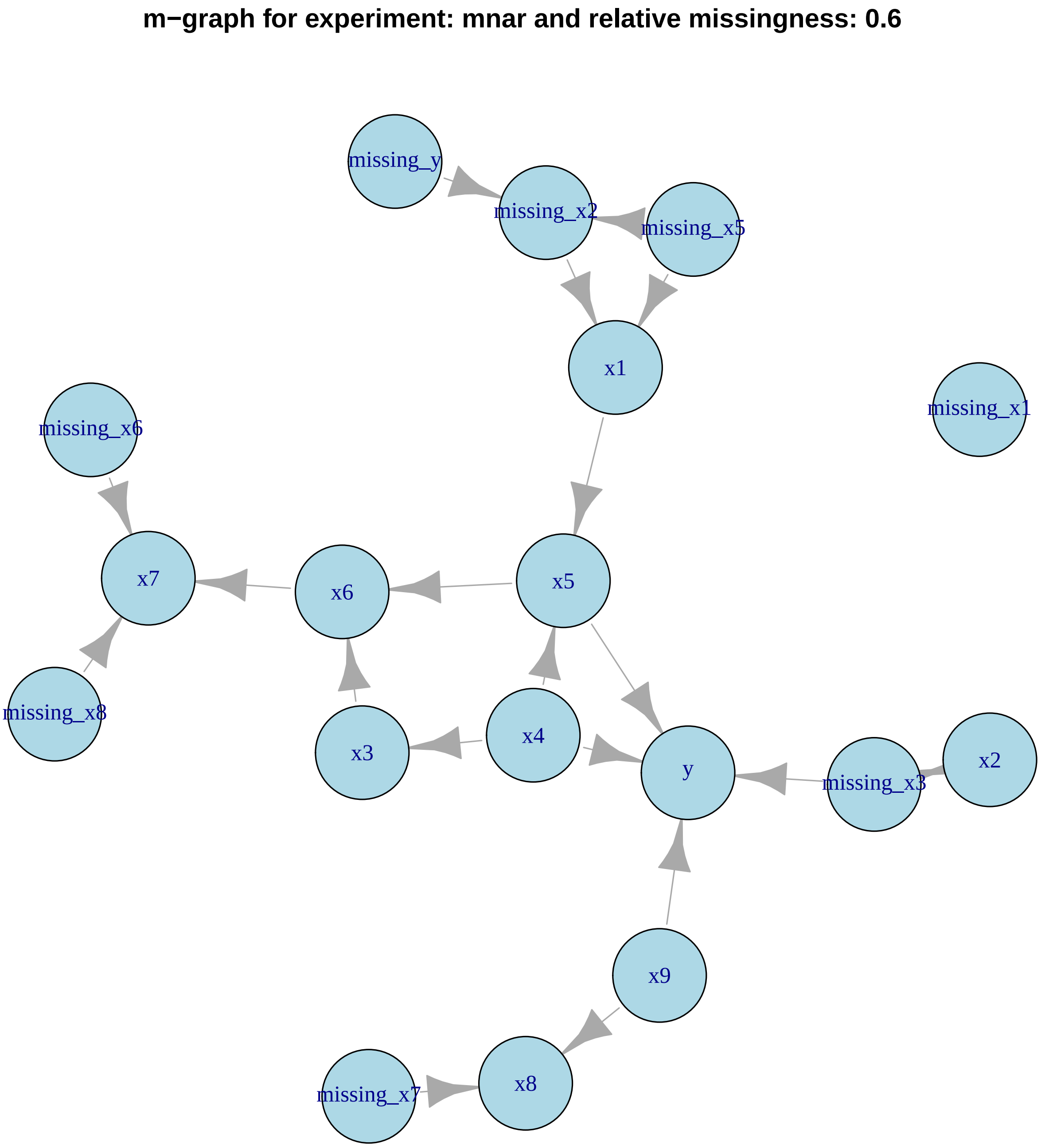


Figure 18: The graph displays the prediction of the graphical model when using the pc-algorithm when having simulated MNAR variables $x_2, x_3, x_5, x_6, x_8, y$ and MCAR variables x_1, x_7 and a missingness rate of 0.6 for each variable.

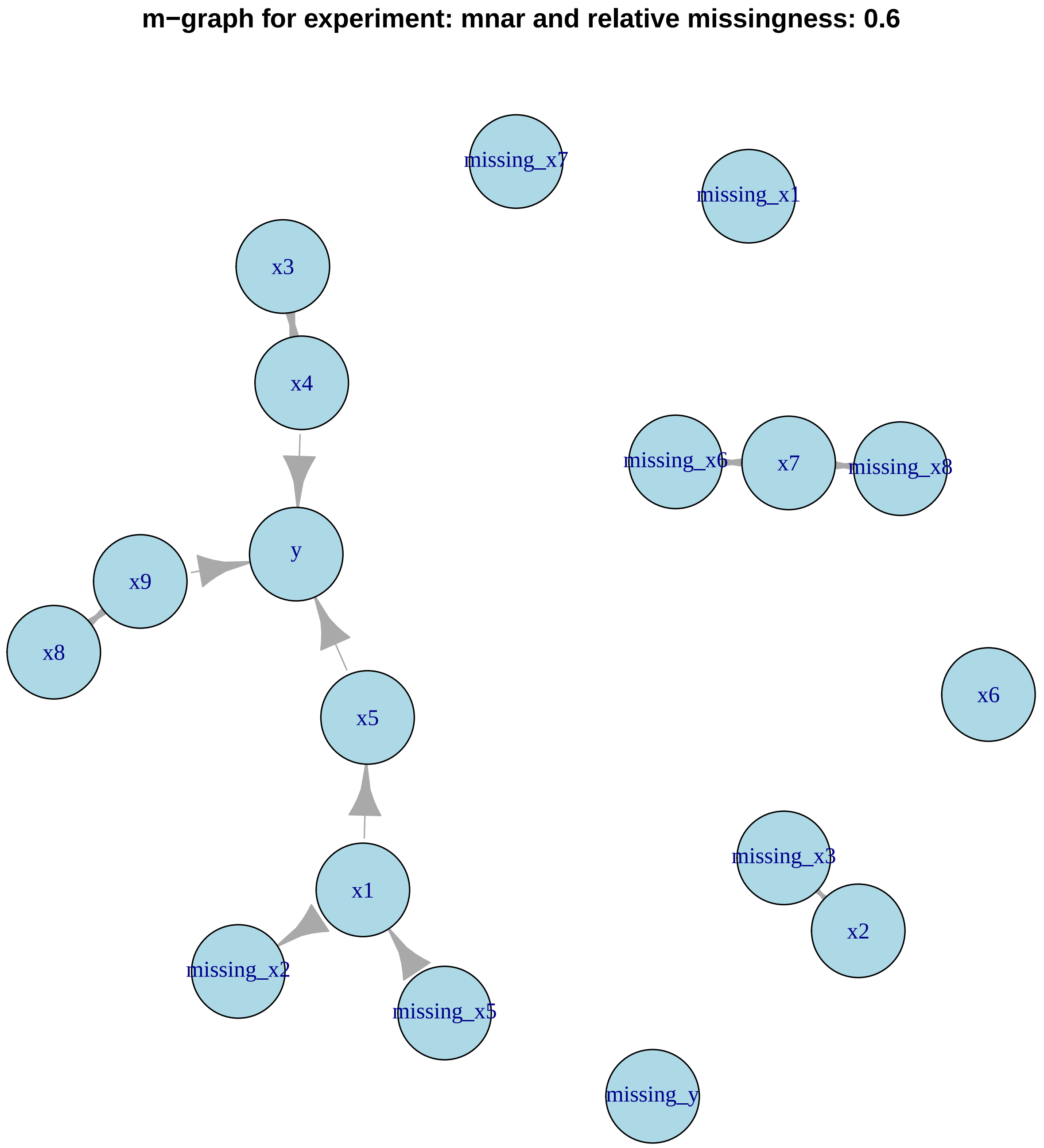


Figure 19: The graph displays the prediction of the graphical model when using the mvpc-algorithm when having simulated MNAR variables $x_2, x_3, x_5, x_6, x_8, y$ and MCAR variables x_1, x_7 and a missingness rate of 0.6 for each variable.

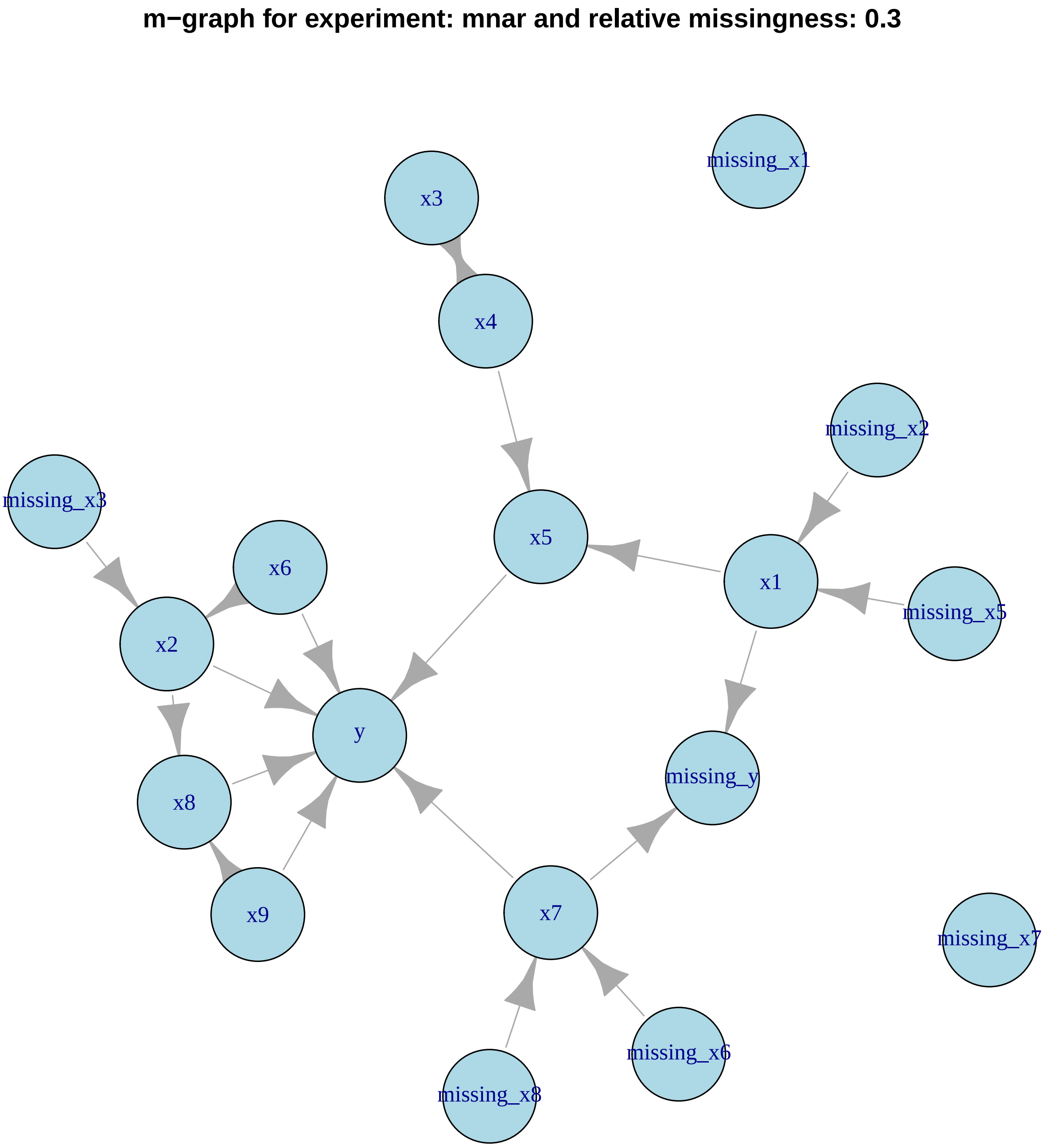


Figure 20: The graph displays the prediction of the graphical model when using the mvpc-algorithm when having simulated MNAR variables $x_2, x_3, x_5, x_6, x_8, y$ and MCAR variables x_1, x_7 and a missingness rate of 0.3 for each variable.

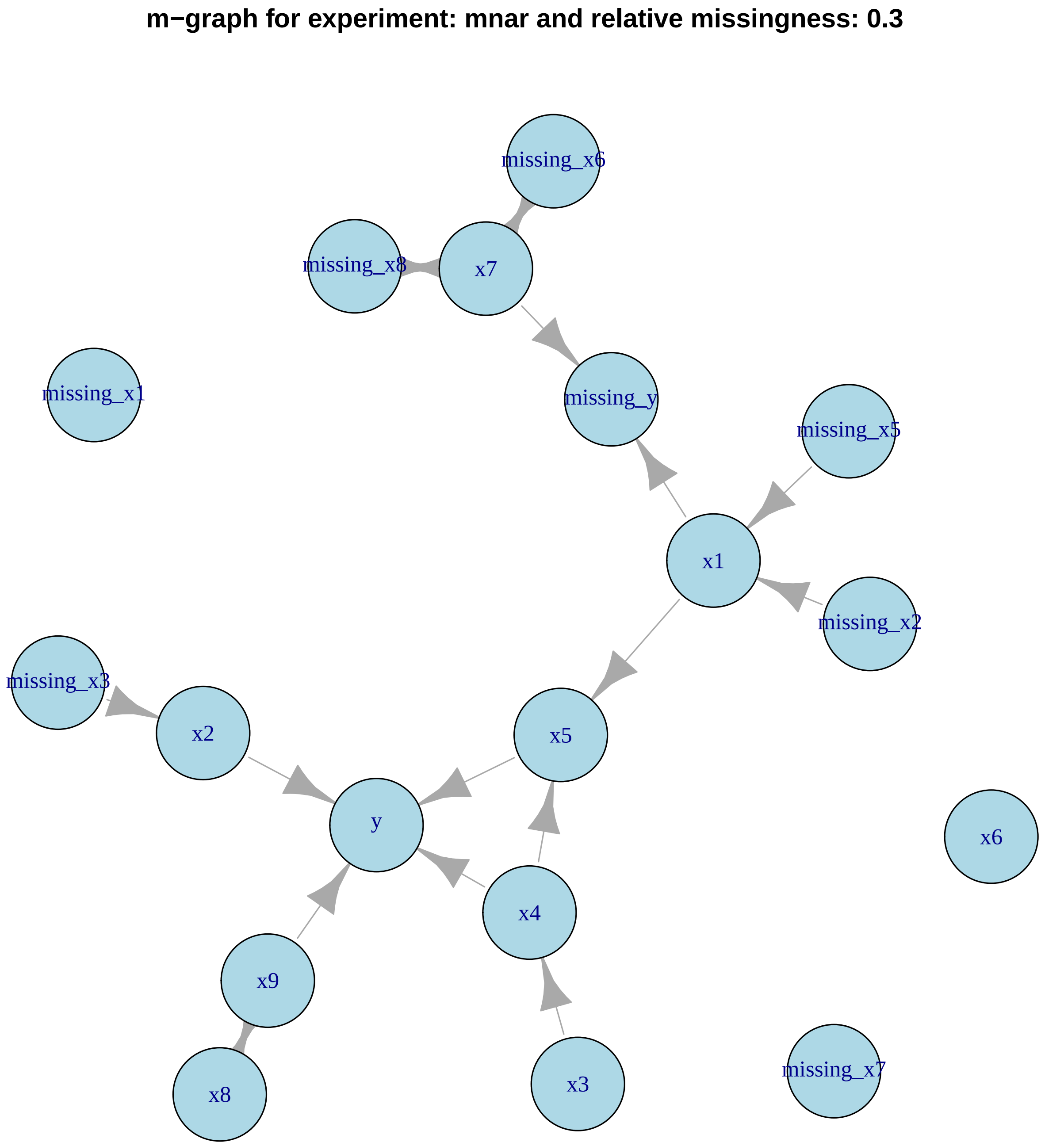


Figure 21: The graph displays the prediction of the graphical model when using the mvpc-algorithm when having simulated MNAR variables $x_2, x_3, x_5, x_6, x_8, y$ and MCAR variables x_1, x_7 and a missingness rate of 0.3 for each variable.

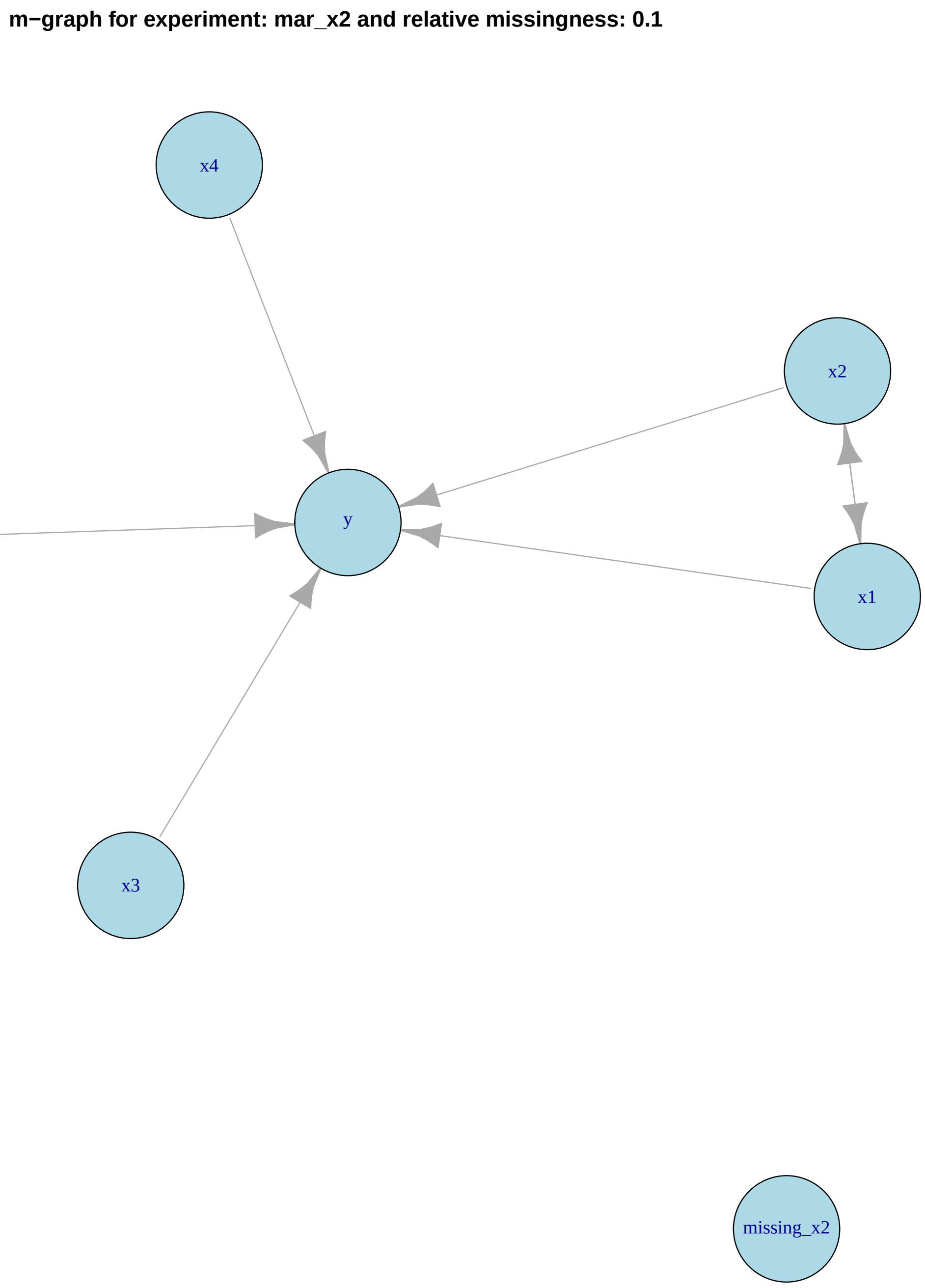

m−graph for experiment: mar_x2 and relative missingness: 0.1
x4
x2
x5
y
x1
x3
missing_x2

Figure 22: The graph shows the predicted graphical model under MAR in the multivariate normal dataset when having a missingness rate of 0.1 and having used as well the pc-algorithm as the mvpc-algorithm.

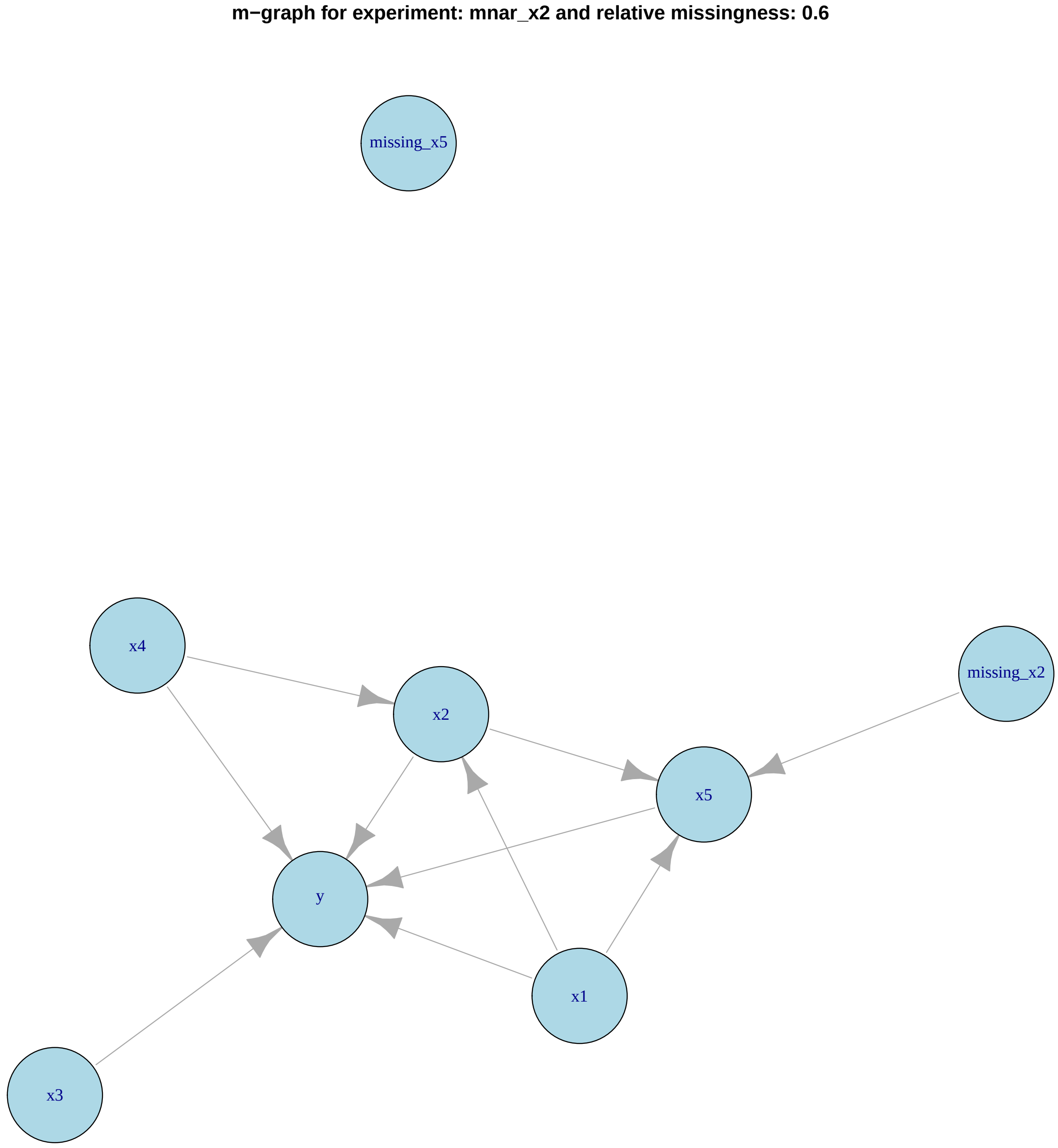

m−graph for experiment: mnar_x2 and relative missingness: 0.6
missing_x5
x4
x2
missing_x2
x5
y
x1
x3

Figure 23: The graph shows the predicted graphical model under MNAR in the multivariate normal dataset when having a missingness rate of 0.6 and having used the pc-algorithm.

Overall evaluation of the m-graphs

Figure 24 reinforces this statement. Also for the normal distributed dataset 100 replications were done to reduce the variance of the results due to randomness. The figure on the left shows the results for the pc-algorithm and that one on the right represents the results of the mvpc-algorithm. On the x-axis we have the three different missingness rates and on the y-axis the variables which have missing values. For each missing type one figure is provided. We start with the MAR scenario and end with the MNAR. Although we choose $n = 1000$ as size of the dataset only 40% of the MAR variables were recognised as MAR with a missingness rate of 0.1. The results for smaller sample sizes were even worse 25. There the mvpc-algorithm recognises nearly nothing correctly as MAR. We will have a look at the results for the mixed dataset soon but even without having seen these results we can say that using the normal distributed dataset does not lead to notably superior results.

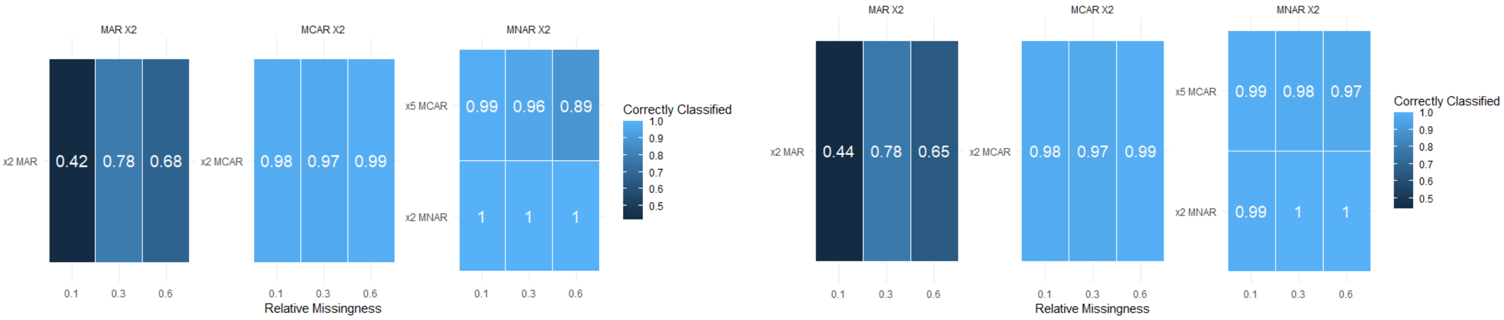


Figure 24: Averaged evaluation of the classification predictions of 100 replicated graphical models for the multivariate normal dataset when using the pc-algorithm on the left and the mvpc-algorithm on the right and a sample size of $n = 1000$.

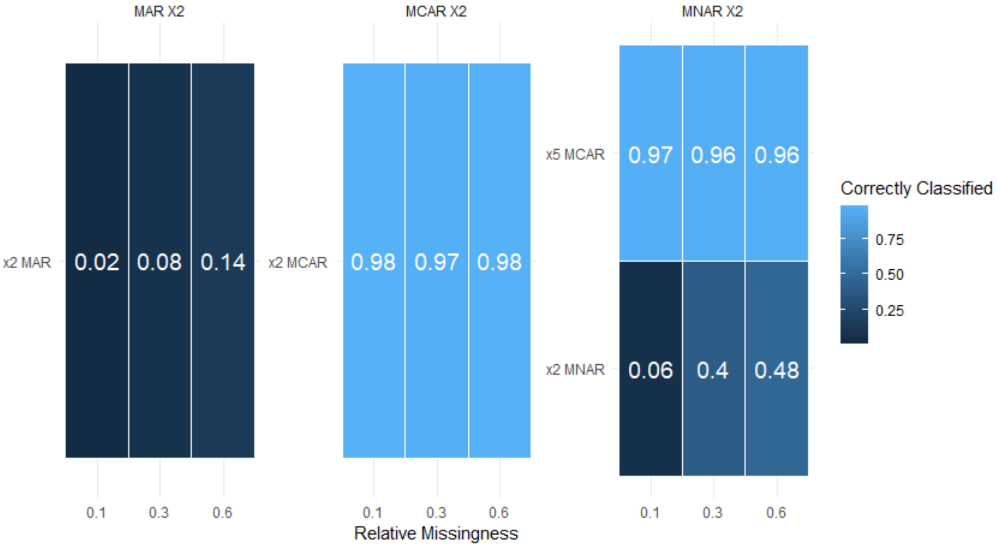


Figure 25: Averaged evaluation of the classification predictions of 100 replicated graphical models for the multivariate normal dataset when using the mvpc-algorithm and a sample size of $n = 100$.

Switching now back to the analysis of the mixed dataset we can see in figure 26 and figure 27 the results of the 100 replicated graphical models for the pc-algorithm and the mvpc-algorithm, respectively. Looking only at the colours we can see that the mvpc results are lighter which means an overall better performance. This can be observed especially at the MCAR scenario which is somehow obvious since the difference between the pc-algorithm and the mvpc is primarily that the mvpc-algorithm deletes some additional edges which yields more MCAR relations. The MNAR scenario is predicted by both algorithms very good which is of high importance to us since MNAR imputation has to be treated differently. The algorithms predict in this sense conservatively because nearly all MNAR are classified correctly but some MAR for example are also classified as MNAR. In some applications this trend might be better than the other way around, following the slogan "better safe than sorry" when imputing missing values and adapting the imputations for MNAR scenarios. In the MAR scenario the two categorical variables x_6 and x_8 were predicted nearly always false from the mvpc-algorithm and also the pc-algorithm seems to have some problems with these type of variables. Thus we can conclude that using continuous variables which are not normal distributed as x_2 and x_3 pose no serious problem but using categorical ones could lead to increasing errors at least in the MAR scenario. Further we observe that the predictions get worse when increasing the missingness rate which is not surprising since we have then less values to calculate the correlations for the independence tests in the graphical model. The results for the smaller datasets $n = 100$ and $n = 500$ can be found in the Appendix B. For $n = 100$ the modelling does not work reliably at all as also discussed in the normal dataset. For $n = 500$ the algorithms worked quite well for the MCAR and MNAR scenario and also the MAR scenario got good results apart from the categorical variables whose predictions got even worse than in the $n = 1000$ case.

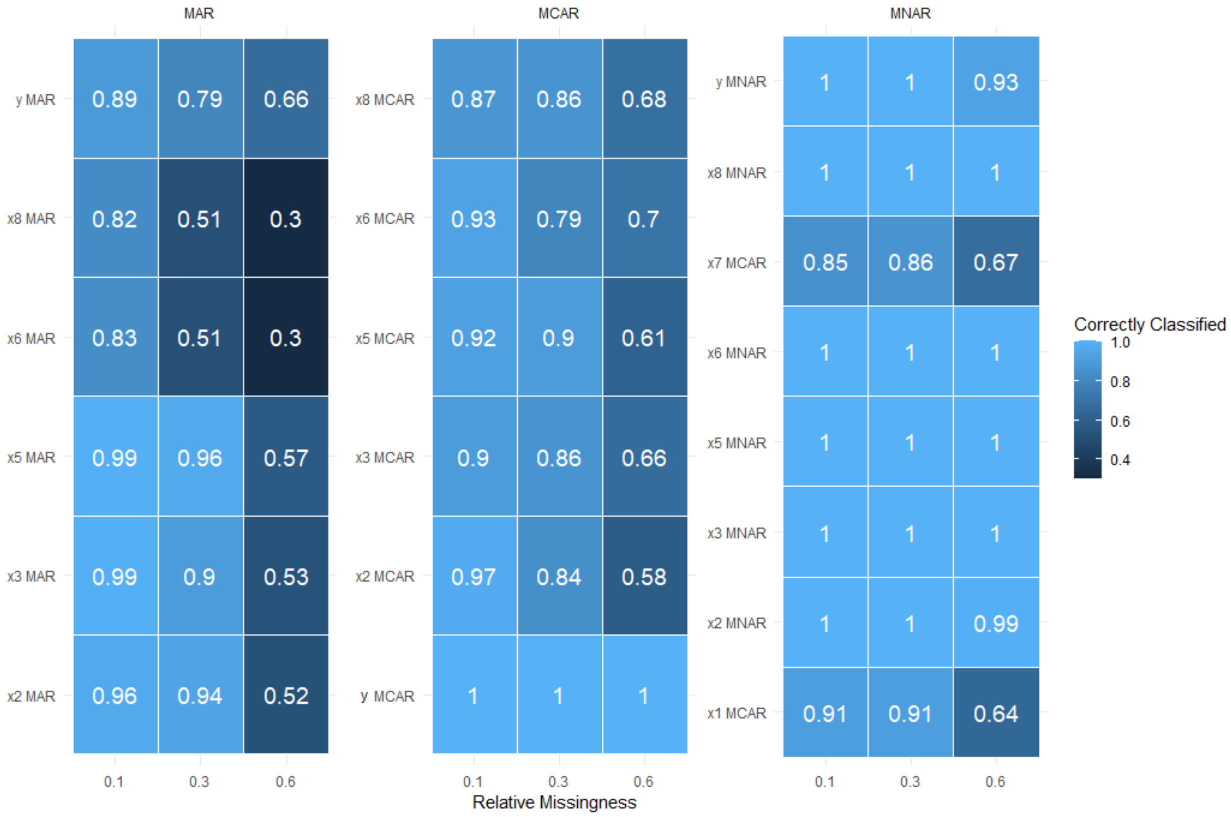


Figure 26: Averaged evaluation of the classification predictions of 100 replicated graphical models for the mixed dataset when using the standard pc-algorithm and a sample size of $n = 1000$.

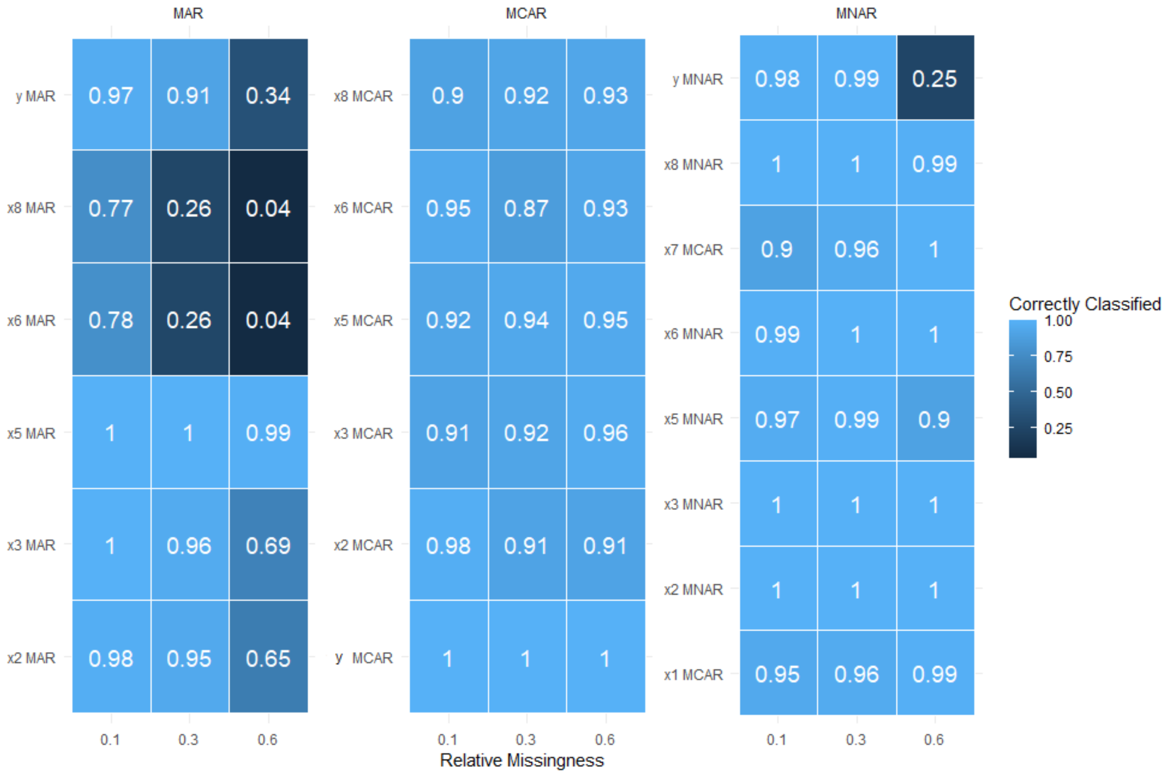


Figure 27: Averaged evaluation of the classification predictions of 100 replicated graphical models for the mixed dataset when using the mvpc-algorithm and a sample size of $n = 1000$.

Overall the quality of the predictions were satisfying and can be used with high confidence also in practice when having at least 500 observations to detect the missingness type at hand. The mvpc-algorithm did a slightly better work then the pc-algorithm but is not provided as a r-package and hence not very user friendly. The pc-algorithm on the other hand is implemented in the *pcalg* package and very easy to use. It depends on the single use case which algorithm is preferable to use since the differences in performance are not too high. This missing value type detection can be used for adapting the imputation as we will discuss in the following chapter but the information itself can already be interesting for the researcher. By knowing which variables are MNAR one can focus on these variables to investigate for instance why people are not willing to answer certain questions in surveys. This can yield e.g. in social science a better understanding of the communities we are living in and maybe also to an improvement based on this analysis, e.g an improved survey design.

4.3 Imputations

4.3.1 AMELIA

The information about the type of missingness can now be used to adapt the imputation process. First we impute everything with the expectation-maximization with bootstrapping algorithm implemented in the R-package Amelia II (13) although it produces in theory biased results when having MNAR data at hand and hence likely also in practice. We will examine this hypothesis with our simulation study and then adapt the imputation method of the as MNAR classified variables to get theoretically valid imputations. Since Amelia II assumes that our data is normally distributed we will apply transformations to our variables to make them fit better into the normality assumption. The algorithm works also without these transformations but of course performs better with them, which is why we decided to do them. The applied transformations will be shortly discussed in the following.
We perform a *ordered quantile* normalization for the variable x_2. This normalization technique is based off of a rank mapping to the normal distribution, which guarantees normally distributed transformed data and is especially suitable when the data at hand is heavily skewed as x_2 is. The transformation looks as follows:

$$g(x) = \Phi^{-1}(\frac{rank(x) - 0.5}{length(x)})$$

while x refers to the raw data(23). The chosen transformations are not arbitrary but selected according to calculations of the *bestNormalize* package (22). The incorporated function attempts to find and execute the best of all of potential normalizing transformations. It selects the best one on the basis of the Pearson P test statistic for normality. The transformation that has the lowest P, calculated on the transformed data, is selected. The Pearson test statistic is

$$P = \sum(C_i - E_i)^2/E_i$$

where C_i is the number of counted and E_i is the number of expected observations (under the hypothesis) in class i. The classes are build is such a way that they are equiprobable under the hypothesis of normality.
For x_3 we chose a simple square root transformation, whereas for the categorical variables we used the non standardized arcsinh transformation:

$$g(x) = log(x + sqrt(x^2 + 1))$$

(22). After having imputed the missing values an inverse transformation has to be done to obtain the original value ranges which is easily done with the predict method of the *bestNor-*

malize package.

As already discussed in the previous chapter simulations are always linked with randomness, which is why also for the imputations 100 replications were performed. In every of these 100 replications we imputed the data five times as characteristically done in multiple imputation. To evaluate the goodness of the imputation we will have a look at the marginal densities for continuous variables as well as histograms for the categorical ones on the one hand and on the other hand at the parameter estimates of the linear regression models. Those will be fitted firstly on the fully observed data rows, which is assumed to be the entire population, secondly on the listwise deleted data in the following also denoted as complete cases and thirdly the imputed dataset. We decided to not look at single imputation values as they do not tell us whether the relations within the available dataset are retained which is of interest when working with a dataset. Further it would be great if it would be feasible to compare the joint density before and after imputation but since this joint density is in most cases not accessible in closed form to us for now we stick with the marginal densities which serve only as indicators of the goodness of the imputations and thus should not be over interpreted since even if the densities look good for each variable the joint density might not be recovered as the relations between variables potentially were not modelled correctly. To be able to aggregate the five imputations into one graph we average over the five marginal densities which is a further a reason why the graphs should not be over interpreted as they do not represent exactly the one to one imputations. Nevertheless we have to somehow put our results into a clearly arranged format to make visualization possible. For the categorical variables the average over the class ratios in each imputation was taken to create the barplot.

Marginal view of imputations

Most of the categorical variables showed good marginal imputation results regarding to these barplots C but not as good as proceeding just with the complete case analysis which is of course very undesirable since we want to improve our results by imputing the missing values. The imputation of x_8 under MCAR with a missingness rate of 0.6 28 was very good which is not surprising since MCAR is the easiest missing type to impute. When looking at the results of the imputations for the MAR 29 and MNAR 30 scenario, things do not look so good anymore. Whereas the third category of variable x_6 was imputed nearly perfectly the imputations of the first two categories were quite bad since the first one is twice as high as it should be. Also for the x_7 variable the imputations are far apart from the original values. But it is interesting that MNAR imputation does not seem to perform worse than MAR imputation although it should in theory. In fact you can find several plots in the Appendix C where imputations in the MNAR scenario worked quite well.

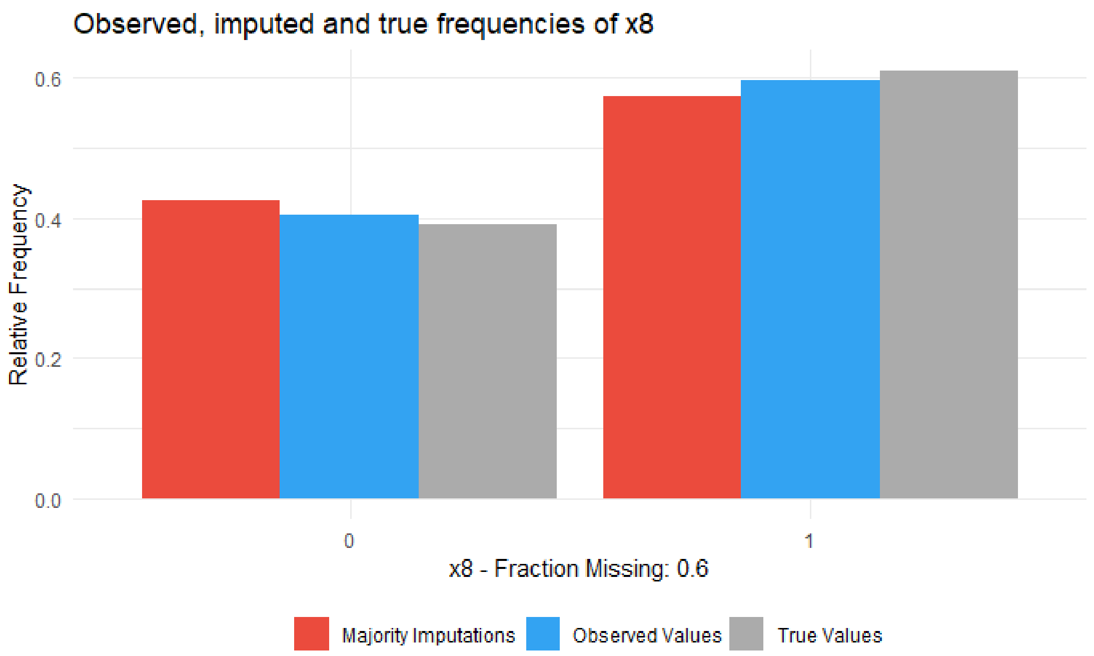


Figure 28: Histogram which displays the proportions between the categories of the categorical variable x_8 for the population, the observed and the imputed dataset with a missing ratio of 0.6 under MCAR.

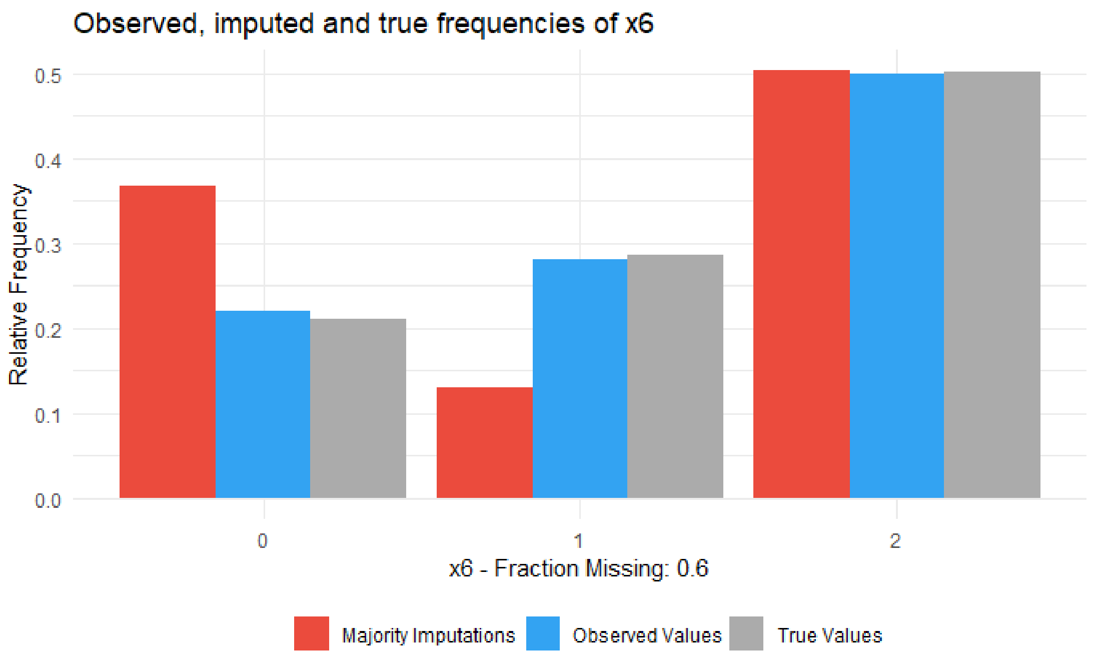


Figure 29: Histogram which displays the proportions between the categories of the categorical variable x_6 for the population, the observed and the imputed dataset with a missing ratio of 0.6 under MAR.

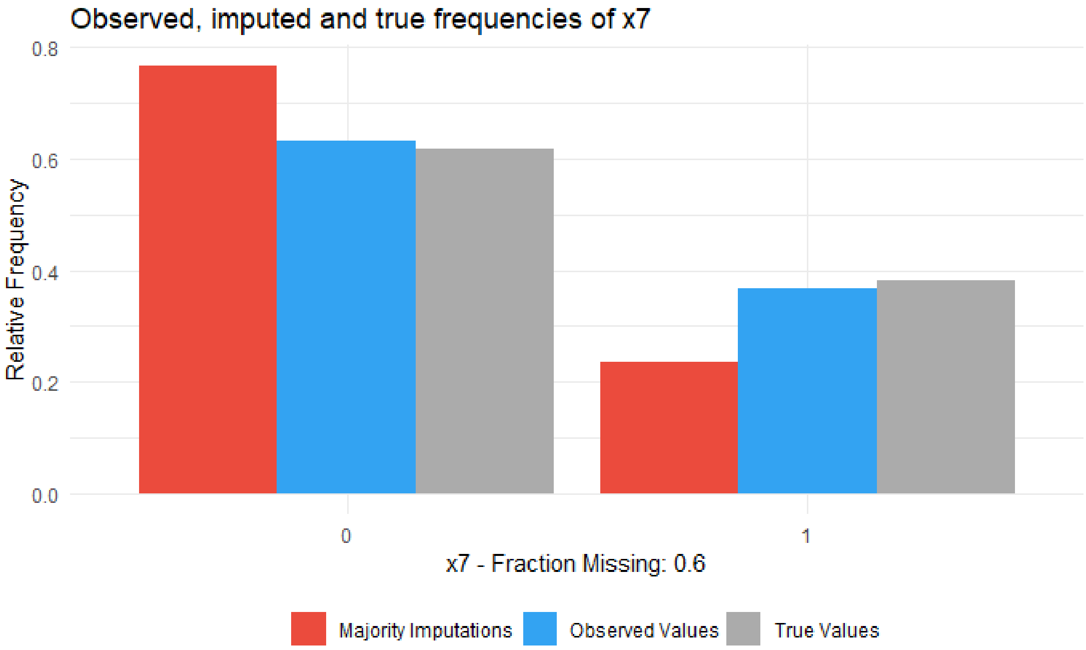


Figure 30: Histogram which displays the proportions between the categories of the categorical variable x_7 for the population, the observed and the imputed dataset with a missing ratio of 0.6 under MNAR.

Also for the continuous variables no significant differences between MNAR and the other types of missingness are visible in the imputations. The imputation for the variables x_1 and x_3 worked very good for all missingness rates although x_3 is heavily skewed and not normal distributed C. It has to be mentioned though that also the complete case analysis represents the margins of the data very good. Especially for x_2 the model had problems with the imputation when having a missingness rate of 0.6 in all three settings of missingness 31. Here the complete case analysis was far better than the imputed one. For comparison also the imputation without normalizing the variables before the imputation is shown in figure 32 to show that the normalization is indeed improving the model. The imputation of x_5 worked also well 34 and gives also better results then the complete case analysis for a missingness rate of 0.6. Most plots which are showed here are with a missingness rate of 0.6 since it is more challenging to impute variables with a lot of missing values and one can conclude from the results that if the imputation worked well with a missingness rate of 0.6 then it will most likely work in general also for a lower missingness rate. For the other plots have a look at the Appendix C.

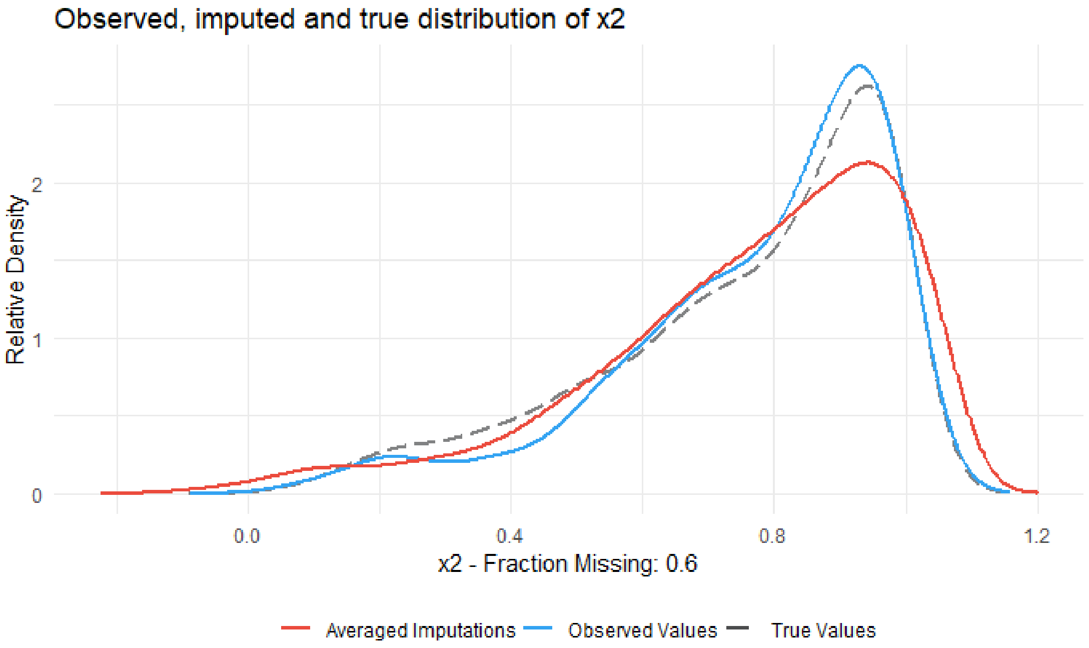


Figure 31: Density of x_2 before and after imputation with a missingness rate of 0.6 compared to the population based density under MNAR. Here we have additionally normalized the variable to improve imputation

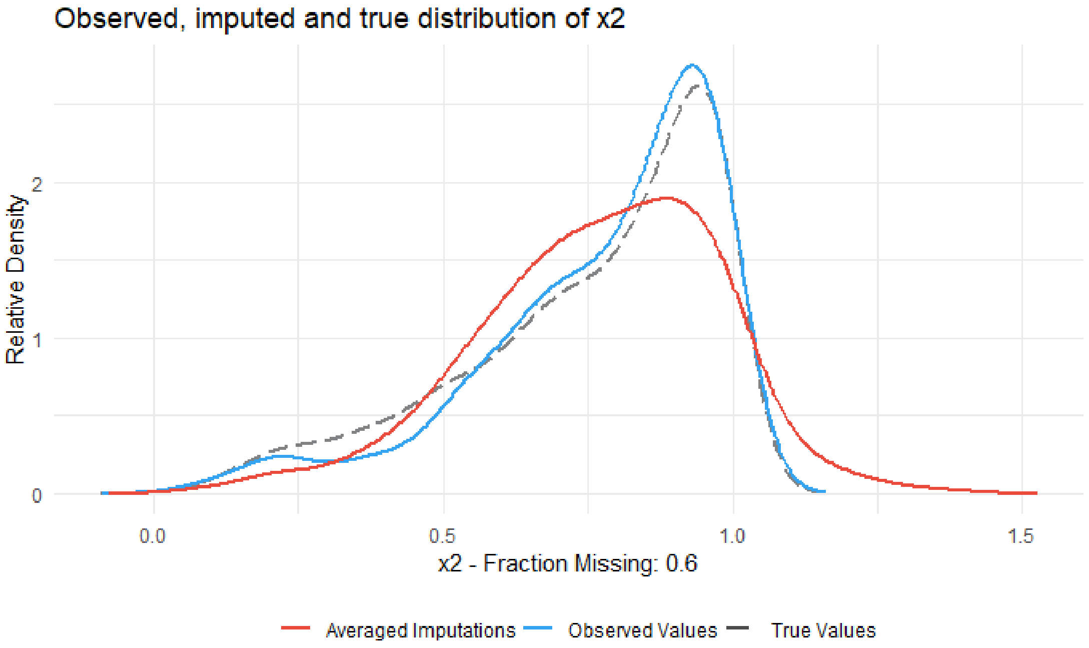


Figure 32: Density of x_2 before and after imputation with a missingness rate of 0.6 compared to the population based density under MNAR.

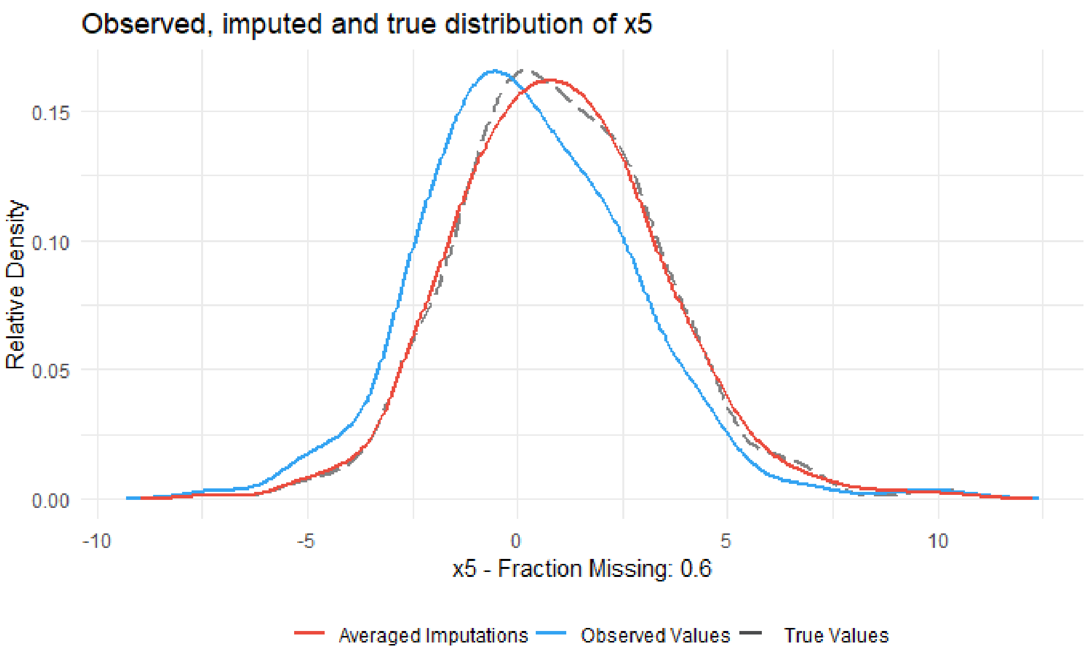


Figure 33: Density of x_5 before and after imputation with a missingness rate of 0.6 compared to the population based density under MAR.

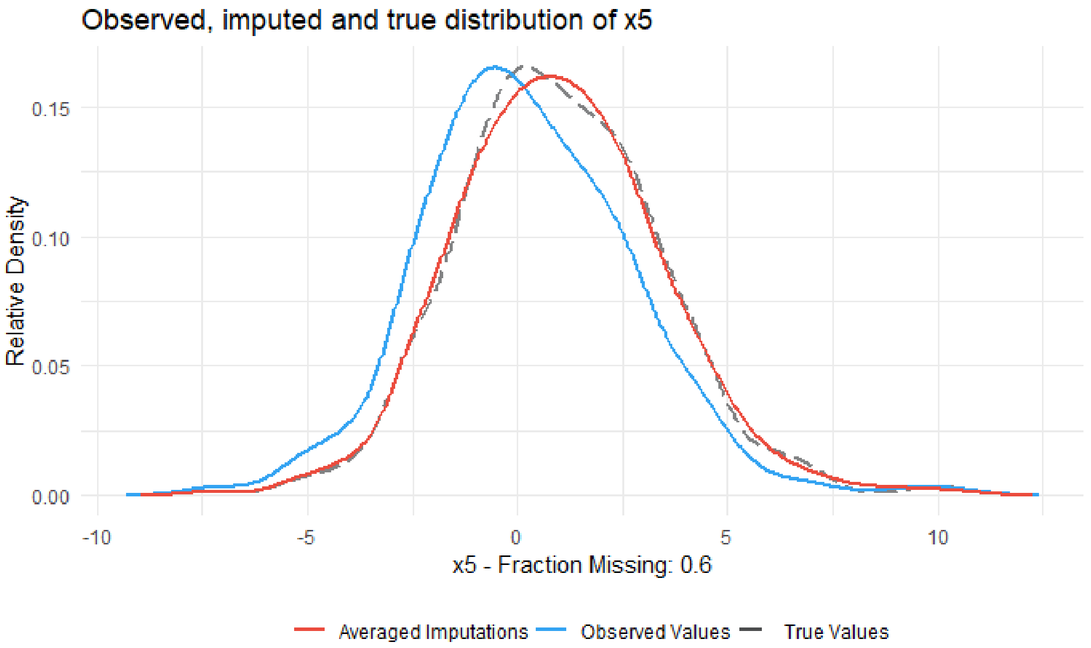


Figure 34: Density of x_5 before and after imputation with a missingness rate of 0.6 compared to the population based density under MAR.

Goodness of imputation via regression

As already discussed in the introduction of this section one should not over interpret this results since they do not cover any kind of relationship between the variables. This is why we will look now at the coefficients of the linear regression

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_{6_1} x_{6_1} + \beta_{6_2} x_{6_2} + \beta_7 x_7 + \beta_8 x_8 + \beta_9 x_9$$

which cover some of the mentioned relations. With this regression we want to predict our target variable y, the true simulated coefficients can be found in fiure 10. For the regression parameters we have to first pool the results of these five imputations according to Rubin's Rules before pooling the results for the 100 imputations. This was done with the *pool()* function from the *mice* (31) (3) package which is one of the most popular packages for multiple imputation especially when you are interested in using Markov Chain Monte Carlo approaches. After having pooled the coefficients themselves but also their variances, confidence intervals were produced which are displayed in the following graphics. We compare the confidence intervals from the imputations with those of the original dataset which represents our population without any missigness and the complete case analysis if we have at least 5 observations, else we take only the population based estimates for comparison. Note that the real values of the regression are taken to be the point estimates of the population estimates since we treat the dataset as the whole population and not as sample of the population. Theoretically one could compare the coefficients also with the theoretically real values which were used to create the population but we will not do that here since we want to portray the population with our simulation.

Overall the results showed that especially for high missingness rates the coefficients of the imputations are a lot closer to the population coefficients then the complete case analysis estimates. Not only the point estimates themselves are closer to the true values but also the variances are far smaller which means that we have less uncertainty in the estimation than when using the complete cases which is obvious since we have considerably less data for the complete case estimation. For missingness rates of 0.6 often it was not even possible to estimate coefficients since the complete cases were too few which is also the reason why we will not look at estimated coefficients for less then five observations. This estimations are very unstable and the variances so high that it does not make sense to interpret them. An example for a missingness rate of 0.6 for each variable which yields a total number of 35 complete observations in the MAR scenario can be found in figure 35. One can see that the confidence intervals of the complete cases are far bigger then those of the imputed coefficients. For the variable x_7 the estimated value from the complete cases is so far out of range that it is not even displayed. The 95% confidence intervals of the imputations from most of the variables cover the true value. An exception is the variable x_6 which might be caused due to the categorical nature of this variable. But especially for the continuous ones the imputation is better than the estimation using only the complete cases. Taking these results into account, which assess the relations between the different variables more realistically and hence indicate better whether the dataset as a whole is well imputed, show us that, as already mentioned, that the marginal densities we have looked at before should not be overinterpreted. Whereas the imputations seemed to be often even worse than the complete cases when considering only the marginal densities the estimation of the coefficients demonstrates the opposite.

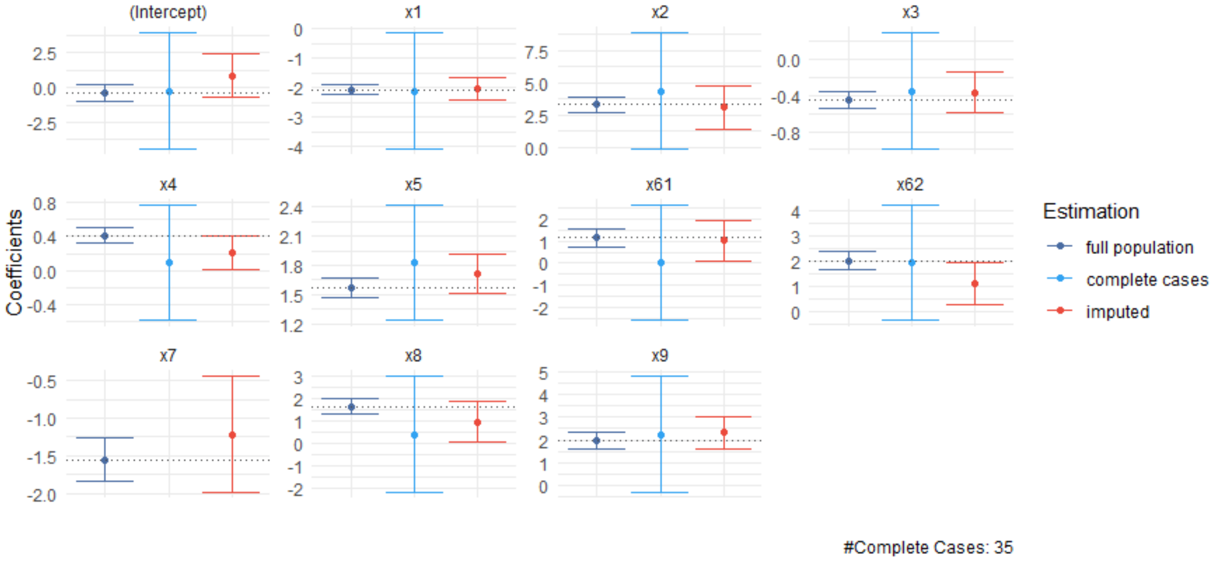


Figure 35: Pooled coefficients from Amelia imputation in the MAR scenario with 0.6 missingness in each variable compared to the list wise deletion calculated coefficients and the population based coefficients.

If only few entries are missing as in figure 36 it can be assumed that the complete cases can cover the relations between the variables still quite good. This can be seen also in the figure. There are not much differences between the complete case analysis and the imputed data analysis. The intervals are similar wide but nevertheless the point estimates of the imputed data are slightly more precise. It is interesting that this holds especially for the categorical variables, e.g the coefficient for x_9 was estimated perfectly with the imputed data.

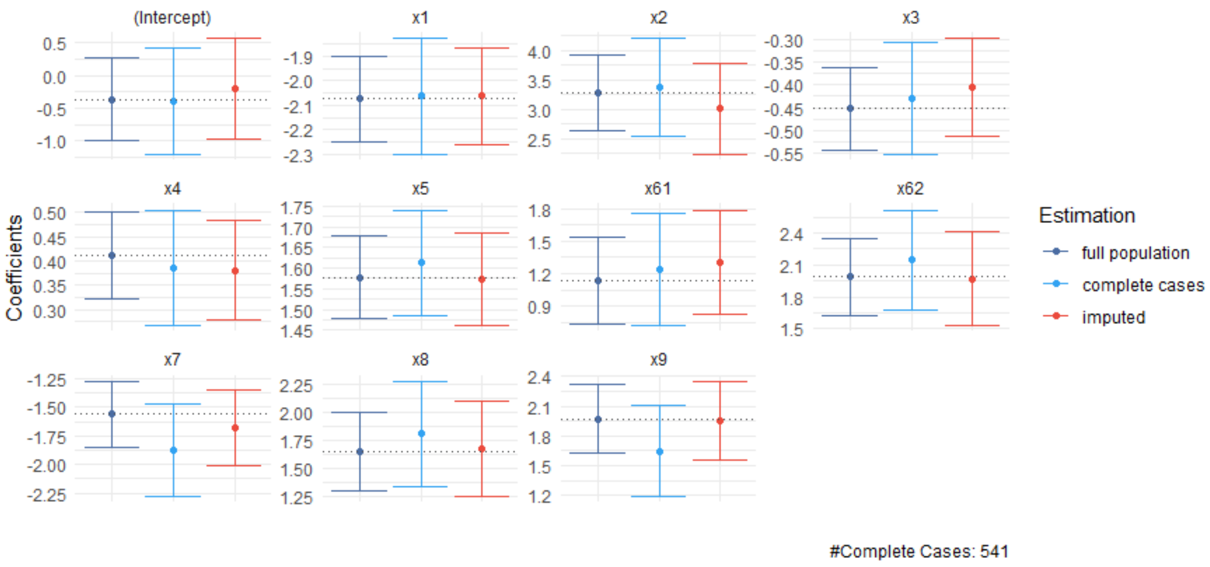


Figure 36: Pooled coefficients from Amelia imputation in the MAR scenario with 0.1 missingness in each variable compared to the list wise deletion calculated coefficients and the population based coefficients.

When looking at the MCAR scenario with a missingness rate of 0.3 which corresponds to an amount of 105 fully observed rows 37, the interpretation is similar to that of the 0.6 scenario from above. Also here the estimation of the imputed continuous variables worked better than for the categorical ones, e.g x_5 was estimated perfectly with the imputed values whereas it could be better for the complete cases. Moreover this time every confidence interval based on the imputed data covers the true value. Note that also the complete case analysis covers the true values for every variable but of course this is also due to the wide ranges of these confidence intervals. Now we want to compare the MCAR scenario with the MNAR scenario to see if the theoretically existing bias in the imputation is also visible in the simulation 38. It can be seen that e.g. the estimated coefficients for x_2 got worse but is still in the confidence interval as every other estimated coefficient. Also the estimates for the variables x_3 and x_5 are not as good as before but on the other hand the estimation for the coefficient of x_8 and x_9 improved. Note that in the MNAR scenario due to randomness more data points were partially observed than in the MCAR scenario, which makes a one to one comparison difficult. Nevertheless we can say that in general the results are not much worse for the MNAR scenario then for the MCAR scenario. Still it is never bad to search for alternatives which do not violate model assumptions because even if in this simulation the imputation for MNAR seems to work quite well with Amelia one has no guarantee that it will work also with another dataset since the theory has proofed that the imputations under MNAR are biased. This is why we will discuss also the knn imputation as a potential solution in the next section, which does not have any model assumptions that can be violated by MNAR data.

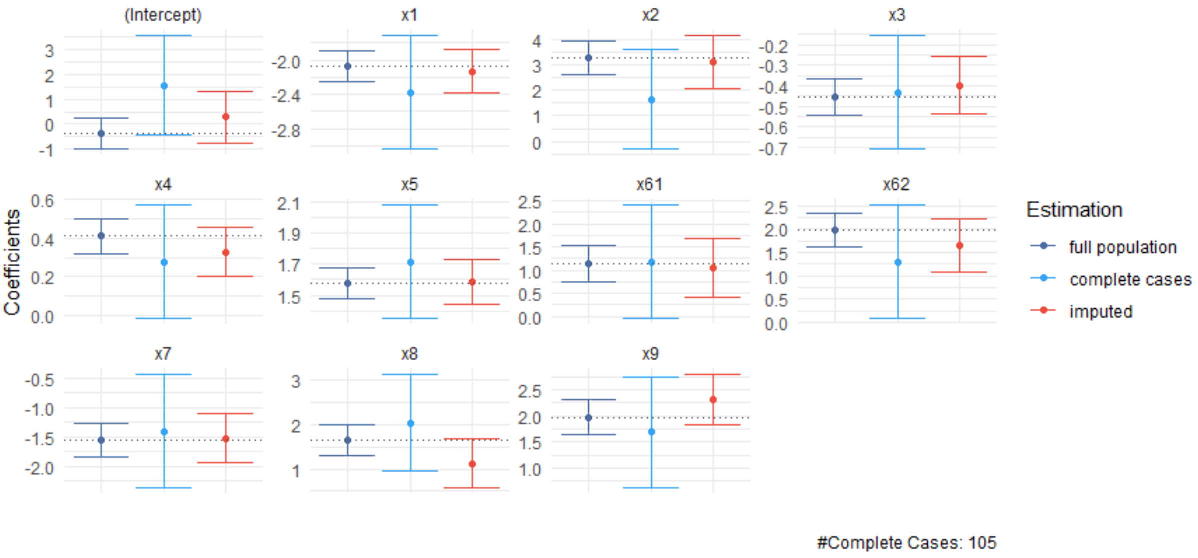


Figure 37: Pooled coefficients from Amelia imputation in the MCAR scenario with 0.3 missingness in each variable compared to the list wise deletion calculated coefficients and the population based coefficients.

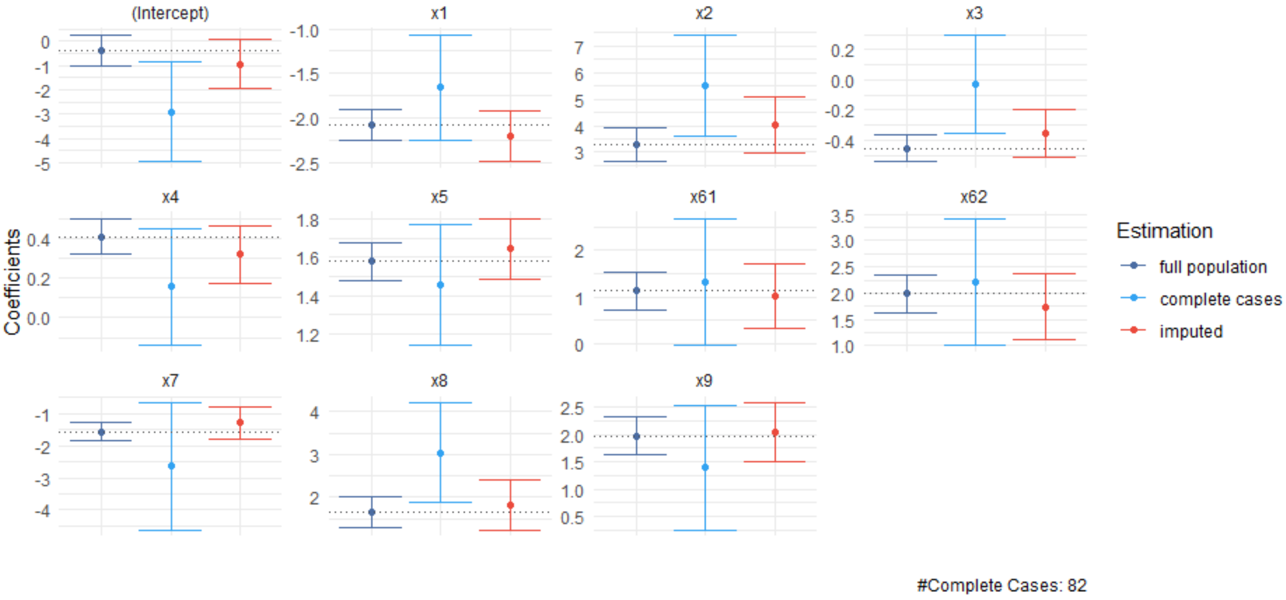


Figure 38: Pooled coefficients from Amelia imputation in the MNAR scenario with 0.3 missingness in each variable compared to the list wise deletion calculated coefficients and the population based coefficients.

4.3.2 Weighted knn

In this section we want to examine if the imputations can be improved when using a non parametric imputation method which has no model assumptions and hence does not violate those under MNAR. For the implementation the R-package *kknn* (25) was primarily used. It offers you the possibility to hand over the nominal variables and the columns which should be taken into account for the calculation of the distances. The distances were weighted according to the rules presented in section 2.2.2. The information retained from the graphical models was used to focus only on distances between variables which have some kind of relation, i.e edges in the graph to remove noises caused due to uncorrelated variables. Not only the variables that had edges with the variable of interest itself were included in the calculation but also these ones which had edges with the missing indicator variable of the corresponding variable. Further the epanechnikov kernel was used as an argument of the *kknn()* function. Note that until now we focused on the interpretation of the coefficients based on the true underlying scenarios MCAR, MAR and MNAR. Now we want to include our results from the classifications for each variable from the graphical models and impute only those variables with the knn method which were classified as MNAR and not those which are indeed MNAR. We will do so because we want to examine if the imputations improve when using the graphical models before to decide which imputation method is adequate. Imputing simply all true MNAR variables assumes that the graphical models exhibits no classification errors which is, as we have seen, not realistic, although a high accuracy is likely.

Further the preimputed data created by Amelia II was used for all variables which were not imputed in the current state as input for the function to make calculations between these variables and that one which should be imputed in this state possible. This was repeated until every variables which was classified as MNAR was imputed by the knn method.

Again we have to account for variances in the knn imputation due to randomness which is why we repeated also this imputation method also 100 times and pooled the results such that we are able to interpret them.

Starting with the comparison of the densities we can see that the knn imputations worsen significantly with increasing missing ratio 39 39 41, especially when comparing them to the Amelia imputations which seem to be more robust against these changes. For a missingness rate of 0.1 the knn method can compete with the Amelia imputation but for a missingness rate of 0.6 for sure not.

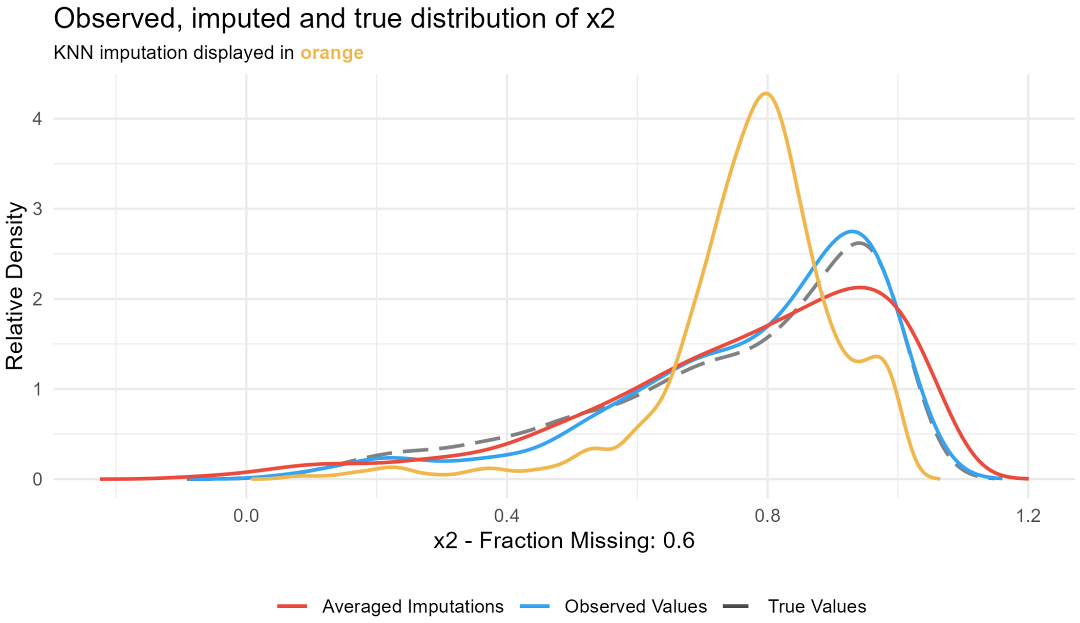


Figure 39: Averaged density of x_2 before and after imputation with Amelia and knn, respectively with a missingness rate of 0.6 compared to the population based density under MNAR.

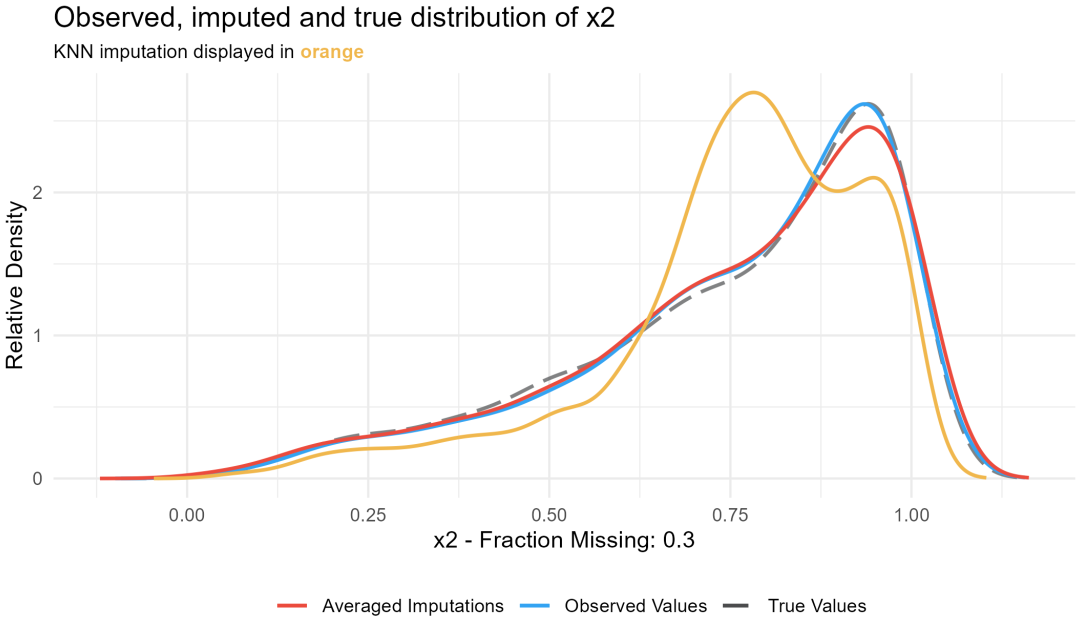


Figure 40: Averaged density of x_2 before and after imputation with Amelia and knn, respectively with a missingness rate of 0.3 compared to the population based density under MNAR.

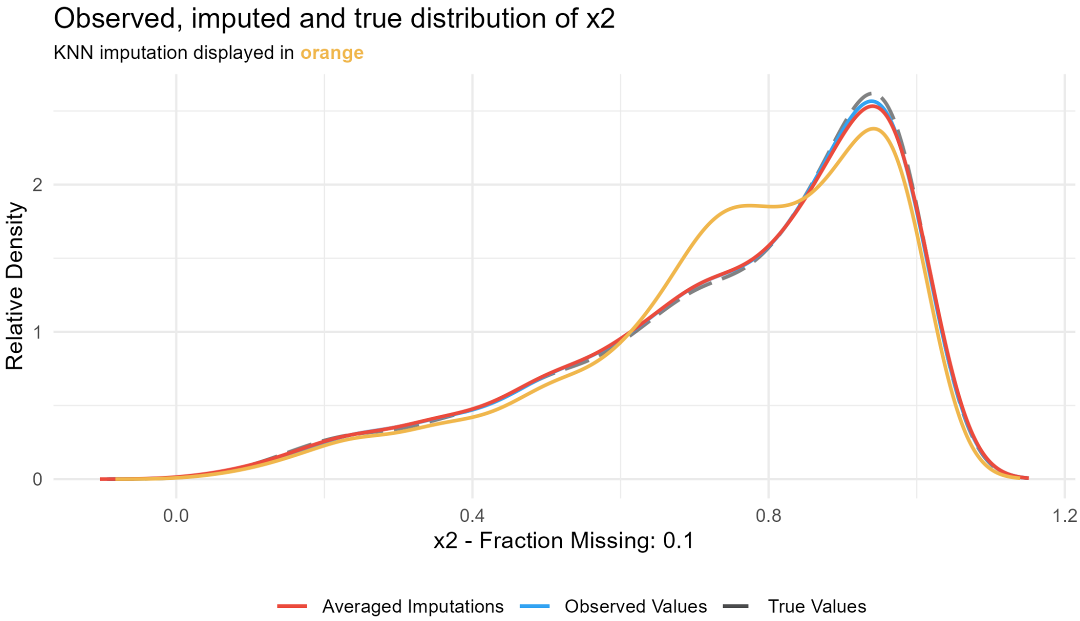


Figure 41: Averaged density of x_2 before and after imputation with Amelia and knn, respectively with a missingness rate of 0.1 compared to the population based density under MNAR.

Besides the imputation of the x_2 variable also the variables x_5 and y were not satisfyingly imputed by the knn method as visualised in figures 44 42. The values around the mean are clearly over estimated. Even the complete case density in the 0.6 missing scenario is far closer to the true marginal density than the knn density. But it has to be again noted that if one considers only the marginal densities a missingness rate of 0.6 means that exactly 60% of the variable which we are looking at is missing but if you are interesting in recovering the relations between more variables as done in the presented linear regression based assessment far more observations are not fully observed which is why we can assume that the joint density for the dataset would look much worse for the complete data cases than the imputed ones.

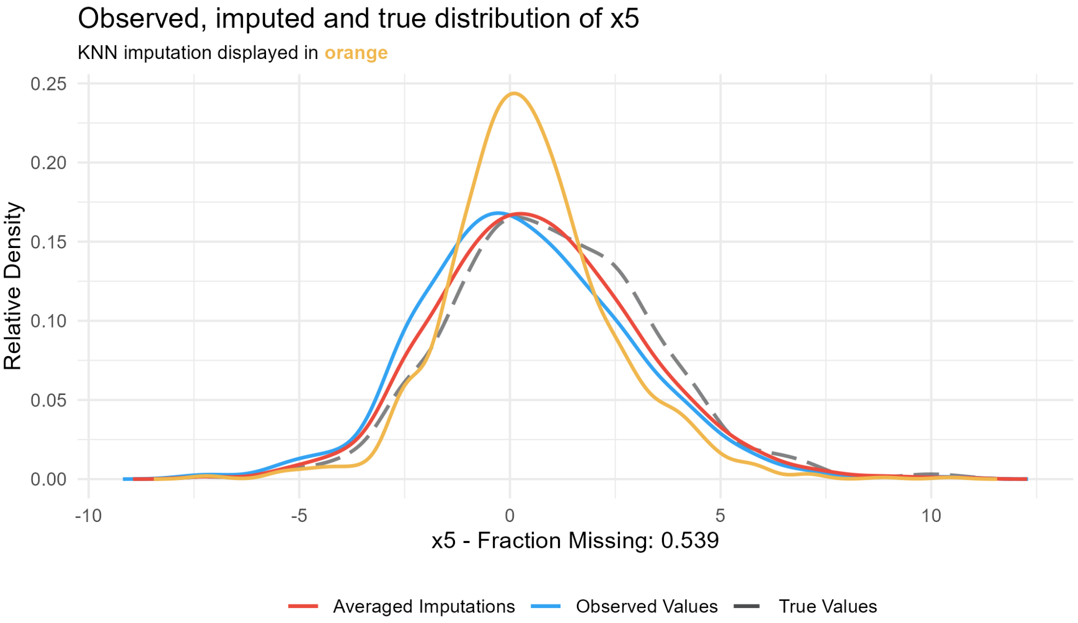


Figure 42: Averaged density of x_5 before and after imputation with Amelia and knn, respectively with a missingness rate of 0.6 compared to the population based density under MNAR.

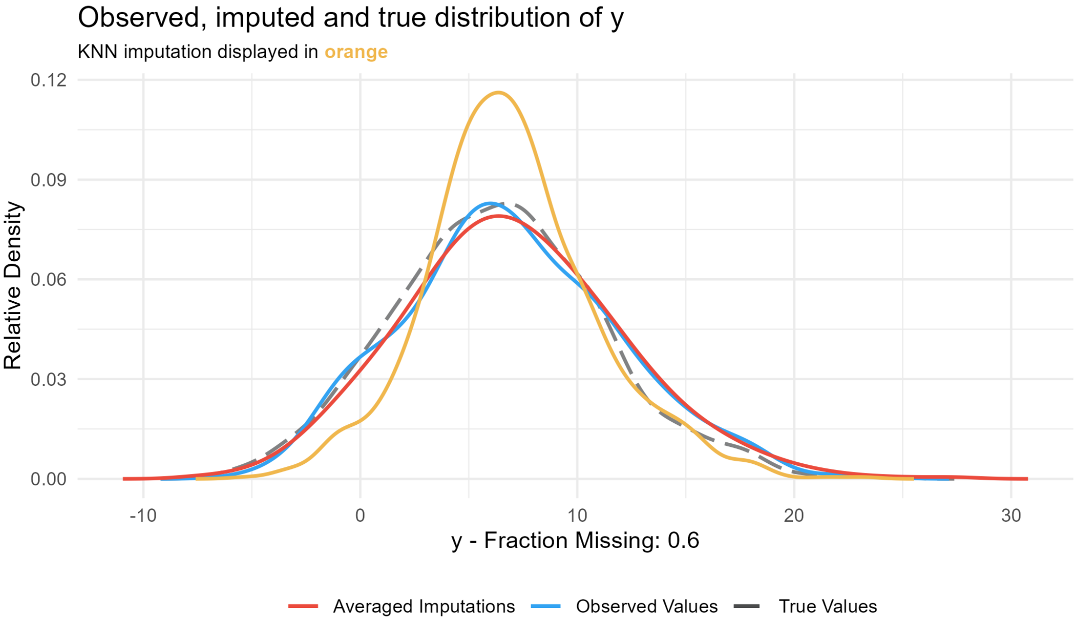


Figure 43: Averaged density of y before and after imputation with Amelia and knn, respectively with a missingness rate of 0.6 compared to the population based density under MNAR.

Even if the knn method has again overestimated the density around the mean for the variable x_3 it is competitive to the imputation of Amelia which has underestimates this area of the density. The problem with overestimating the density around the mean is probably due to the construction of the knn algorithm which calculates the distances and averages them afterwards such that values around the mean are more heavily influencing the resulting value of the missing entry.

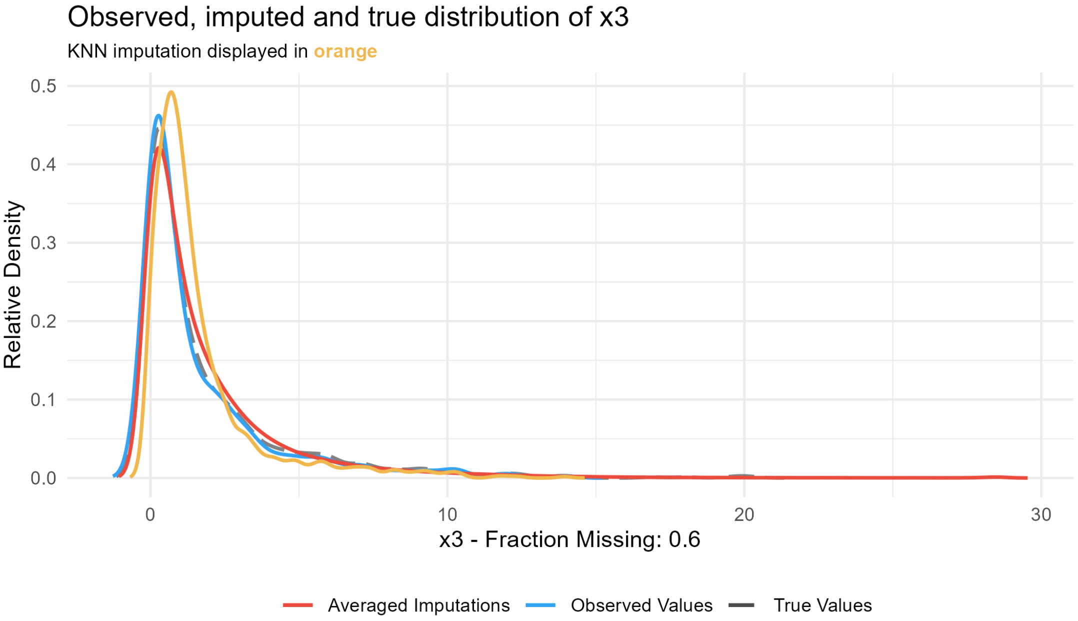


Figure 44: Averaged density of x_3 before and after imputation with Amelia and knn, respectively with a missingness rate of 0.6 compared to the population based density under MNAR.

The differences between the different estimates were very little for the categorical variables, which is why the results will be presented in the tables below, showing only the difference between the estimation and the true population value instead of looking at the barplots. Table 1 displays the averaged error of the estimated ratios for the MNAR variables x_6 and x_8 for the scenario where 10% of the variable entries are missing. Since x_8 is binary it is sufficient to look only at the error rate of the ratios for category 0. The error of category 1 is then exactly the same value with opposite sign. Variable x_6 has three categories hence we have to consider two of them. It is interesting that unlike for the continuous variables the knn algorithm worked well for the categorical variables in this scenario. At least if we compare the results with the Amelia imputation. The complete case analysis is even better than the knn imputation not only for a missingness rate of 0.1 but also for higher missingness rates 2 3. Even if more then the half of the entries is missing 3, the complete case analysis performed better than the imputation based analyses. This is surprising but maybe also misleading as we will see in the following, when looking at the regression coefficients which consider also the relations between the variables. In that case that we have a missingness rate of 0.3 the results of the knn imputation and the Amelia imputation are quite similar. You can notice that the results of the knn algorithm are still slightly better than those of the Amelia one whereas that changes when increasing the missingness rate to 0.6. The results are again comparable but now Amelia outperforms knn, especially for the estimated ratio of x_{6_2} and therefore also for the reference category which is not displayed.

In summary it can be said that the performance of the knn imputation suffers greatly by increasing the missingness rate both for the continuous and for the categorical variables whereas Amelia is more robust against these changes. Contrary to our expectations the performance of the complete case analysis did not suffer from increasing the missingness rate in the categorical variables. Nevertheless the results should be interpreted with care, since they do not cover the multivariate dependence between the variables and do not give indication of whether the joint density which is often of interest is estimated correctly or not.

	x_{6_1}	x_{6_2}	x_8
complete cases	0.0019	0.0024	-0.0044
Amelia	-0.025	0.048	0.011
knn	-0.003	0.024	0.012

Table 1: Table displays the difference between the imputed ratios and the true ratios of the categorical variables. The results of the three different approaches are listed row wise. The categorical variables which are analysed are displayed in the columns. The present missingness rate is 0.1.

	x_{6_1}	x_{6_2}	x_8
complete cases	-0.0070	0.0123	0.0014
Amelia	-0.091	0.145	0.024
knn	-0.085	0.15	0.018

Table 2: Table displays the difference between the imputed ratios and the true ratios of the categorical variables. The results of the three different approaches are listed row wise. The categorical variables which are analysed are displayed in the columns. The present missingness rate is 0.3.

	x_{6_1}	x_{6_2}	x_8
complete cases	-0.007	-0.002	-0.01
Amelia	-0.133	0.154	0.035
knn	-0.175	0.298	0.038

Table 3: Table displays the difference between the imputed ratios and the true ratios of the categorical variables. The results of the three different approaches are listed row wise. The categorical variables which are analysed are displayed in the columns. The present missingness rate is 0.6.

In the following we will proceed with examining if the relations between the variables were retained when using knn imputation by looking again at the coefficients of the linear regression model. It is interesting that one of the worst estimates is that one of x_4 which itself is not missing 45. This shows that the bad imputation of a variable can influence the estimate of another variable which might be well imputed or not even missing. This emphasises the statement that it is not enough to look only at the marginal densities. The confidence intervals are very tight but do not cover the true values for many variables as x_3, x_{6_2} and x_9. We can further see that those variables were treated as MNAR that are indeed MNAR, so the classification itself worked good in this example.

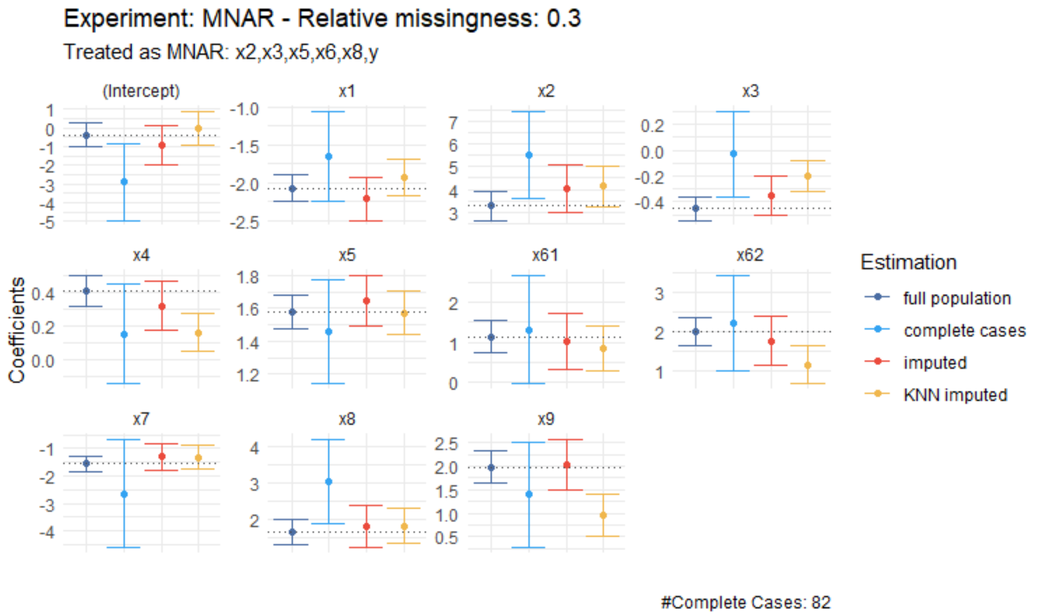


Figure 45: Pooled coefficients from Amelia imputation and knn imputation in the MNAR scenario with 0.3 missingness in each variable compared to the list wise deletion calculated coefficients and the population based coefficients. The variables which were predicted as MNAR from the graphical model are used for knn imputation

While figure 45 shows the results for a missingness rate of 0.3, only 10% of the observations for each variable are missing in figure 46. Since less data points are missing it is easier for the knn algorithm to impute the values correctly. Therefore the true values are covered by all confidence intervals. The confidence intervals of the knn results are slightly tighter than those of the Amelia imputation and of course also than those of the complete case analysis, as you can see e.g when comparing the estimated intervals of x_{6_1} or x_7. This means that the variance and hence also the uncertainty is smaller when using the knn algorithm. On the other hand we have again clearly better point estimates for the variables x_1, x_3, x_5, x_9 and the intercept when using Amelia imputation instead of knn imputation. Compared with the complete case analysis the point estimates are similar to those of the knn imputation. Taking into account that the intervals are much bigger when not imputing the missing values it is worth to use the knn imputation instead of doing a complete case analysis. If we increase the missingness rate to 0.6 47 it is also recommended to use knn imputation instead of complete case analysis since the remaining amount of fully observed values is 5 which does not make it possible to calculate meaningful estimates for the coefficients. Nevertheless Amelia outperforms knn again significantly. While the 95% intervals of the Amelia imputation cover all true values, only four out of eleven knn intervals cover the true values. Again we can observe that the intervals are much smaller when using knn imputation, which is also a reason why the probability of covering the true value when using Amelia is much higher. Nevertheless we also notice that the point estimates of Amelia are with exception of the estimates of x_2, x_4, x_9 always much closer to the true value then those of the knn imputation. Only the estimate of x_4 got nearly perfectly calculated by using knn imputation instead of Amelia imputation. Overall Amelia performs better than knn based extension for MNAR variables although it violates some model assumptions not even for a missingness rate of 0.6 but for all missingness rates. Note that in figure 47 x_1 was treated also as MNAR although it was simulated as MCAR since the graphical model classified this variable wrongly. Assuming that the graphical model has classified everything correctly yields figure 48. This figure shows comparable results as figure 47. Also here the true values is often not covered by the knn intervals and the point estimates are in the most cases worse than those of the Amelia imputation. It seems that the x_1 variable which was classified wrongly did not influence the results of the coefficients a lot. Especially for the estimate of x_1 itself the results are overlapping but as we have seen before a bad imputation of a variable can be unrecognised if we look only at the variable itself and not at the whole dataset, including their depenence etc.

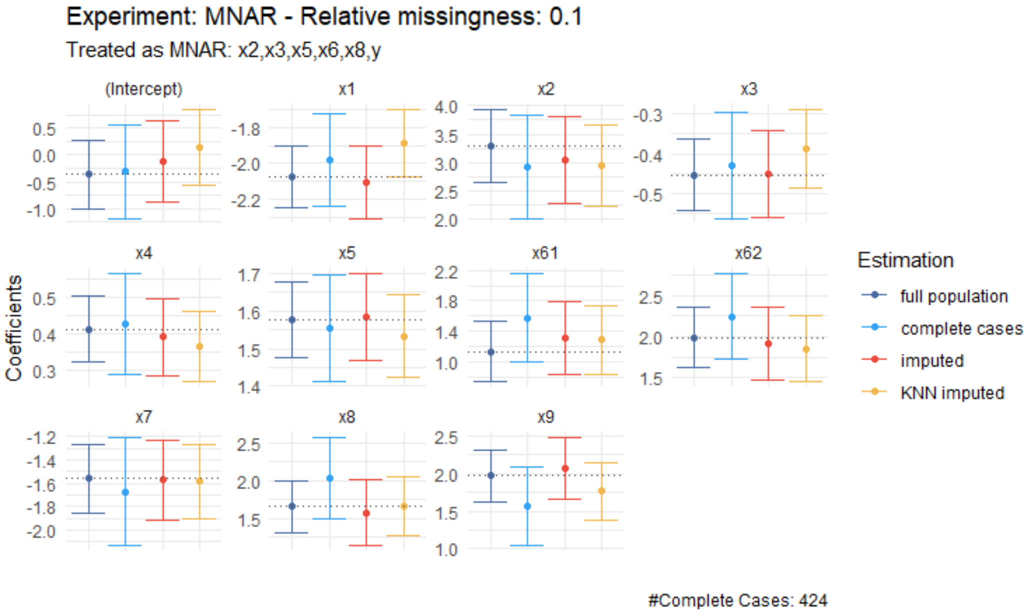
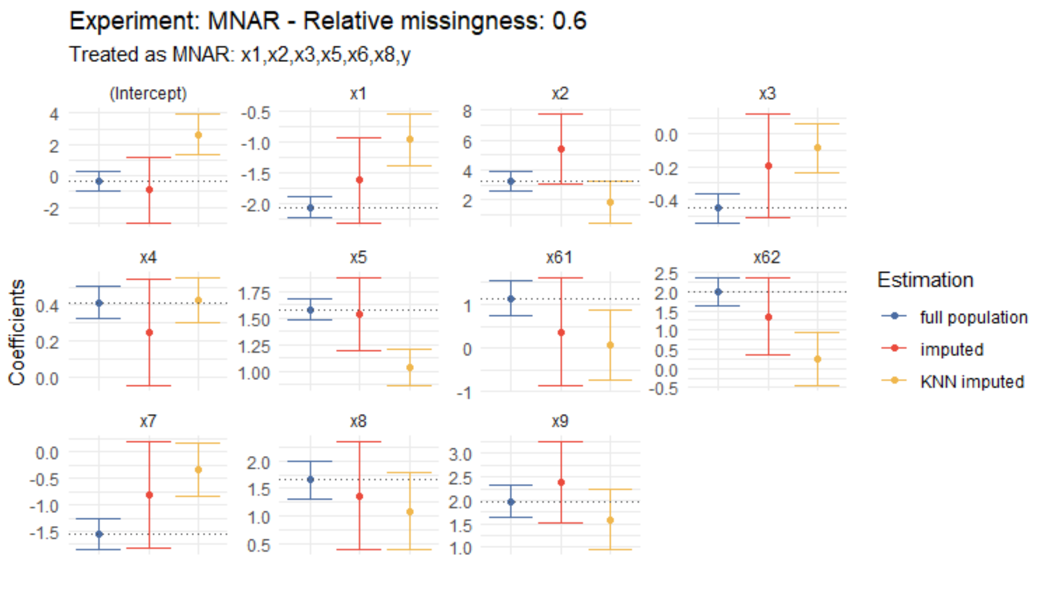
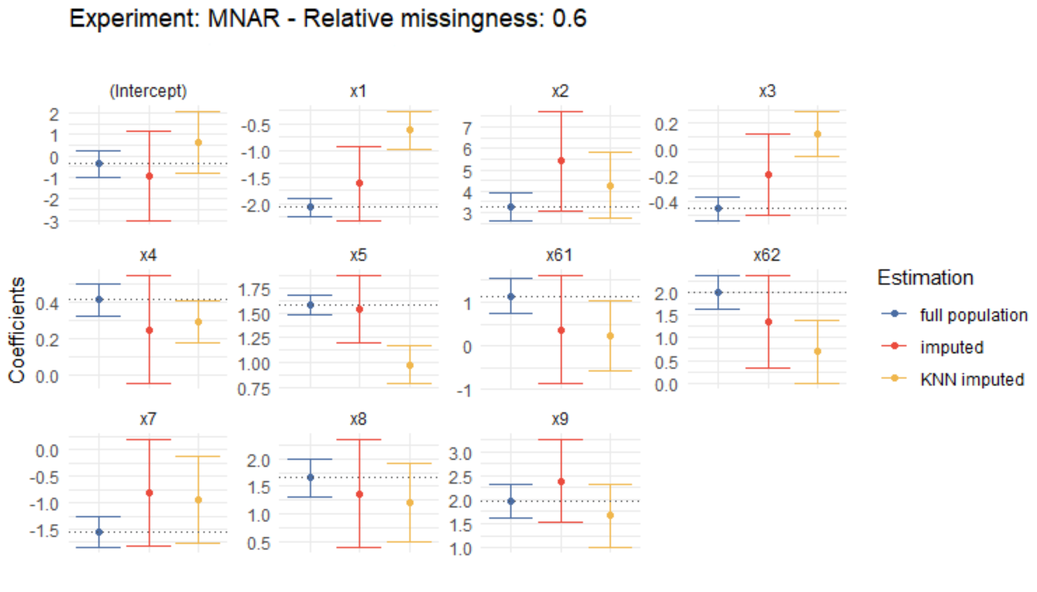


Figure 46: Pooled coefficients from Amelia imputation and knn imputation in the MNAR scenario with 0.1 missingness in each variable compared to the list wise deletion calculated coefficients and the population based coefficients. The variables which were predicted as MNAR from the graphical model are used for knn imputation



Due to only 5 complete cases the corresponding estimates are not displayed.

Figure 47: Pooled coefficients from Amelia imputation and knn imputation in the MNAR scenario with 0.6 missingness in each variable compared to the population based coefficients. The variables which were predicted as MNAR from the graphical model are used for knn imputation



Due to only 5 complete cases the corresponding estimates are not displayed.

Figure 48: Pooled coefficients from Amelia imputation and knn imputation in the MNAR scenario with 0.6 missingness in each variable compared to the population based coefficients. The variables which were simulated as MNAR from the graphical model are used for knn imputation

5 Conclusion

The importance of considering missing values in a data analysis is due to its potentially high amount of information unquestionable. The difficult question is not *if* we should incorporate missing data analysis in our data analysis but *how* we should do it. This work has presented two approaches which cover two different aspects of missing data analysis. Firstly we answered the question: *Why is the data missing?* by using graphical models and classifying the missing values into the three categories of missingness: MCAR, MAR, MNAR. Secondly we looked at the question: *How should we impute the missing values?* The second question can be answered also independent from the first one by just trying to find a generally best imputation methodfor all missigness types. We focused on combining the two questions and using the information about the missingness process to adapt the imputation if missingness is MNAR to work on this question.

Since Mohan et. al. have proven that traditional graphical models are biased when missing data is present and have provided an alternative graphical model which yields in theory better results, it seemed reasonable to compare these two approaches when classifying the missing value types. The results in the simulation study showed that the adapted mvpc-algorithm indeed achieved a better classification accuracy, especially for the MAR variables. Nevertheless also the results from the traditional pc-algorithm, which is compared to the mvpc-algorithm, that is not available as a R package, very easy to use, were satisfactory. The performance worsened slightly by increasing the missingness rate, but was still good. Severe problems have been observed when decreasing the sample size on the other hand. Independent of the chosen algorithm the classifications were unreliable and it is not advisable to use those. The results of the simulation study indicate that assuming that one has data with at least 500 observations using graphical models to detect the missing type is recommended.

Is it also recommended to use this information further for adapting the imputation method? In theory we have seen that MNAR imputations are biased when using model based imputation methods. In the simulation study this behaviour could not be reproduced from the analysis of the regression coefficients. Although we simulated the data to have strong MNAR relations by fixing the multiplier to a value of three 4.1 the estimated coefficients are comparably accurate compared to those estimated under MAR or MCAR. Of course this is not a guarantee for other datasets which is why even if the imputation quality did not seem to suffer under MNAR in this simulation study it would still be great to find an alternative imputation method which does not violate model assumptions and gives thus in theory unbiased results. Based on that we tried also an imputation method based on the weighted knn algorithm for the MNAR variables predicted by the graphical model. For most of the cases the knn imputation performed worse than Amelia imputation. However the confidence intervals of the analyzied regression coefficients were smaller which is desirable since this demonstrates less uncertainty. But in many cases they did not even cover the true values which is of course undesired, since it indicates stability issues in the estimation.

Even if the knn imputation does not violate model assumptions since it is an unparametric procedure and is hence in theory more adequate for the MNAR case, it does not seem like a good alternative to Amelia imputation in practice based on the results of this simulation study. Nevertheless several other simulation studies can be found in the literature which showed biased results for MNAR, which underlines the future relevance of finding an unbiased method for MNAR imputation. Maybe knn is not the best alternative or can be improved by doing a multiple steps knn or changing the weights in a more proper way. At least the simulation study has shown that probably all of the studies done in practice which mostly assume MAR or MCAR processes to impute their data are not as bad as expected from the theory. For now we can conclude that as the detection of missingness types using graphical models works with high accuracy in cases with adequate data size the open challenge remains in finding a rigorous imputation method for adapting the MNAR imputations that perform competitively with respect to standard imputation methods. Besides, the detection by itself can result in valuable insights for the researcher performing the analysis.

References

[1] Batista, Gustavo E. ; Monard, Maria C. u. a.: A study of K-nearest neighbour as an imputation method. In: *His* 87 (2002), Nr. 251-260, S. 48

[2] Beesley, Lauren J. ; Taylor, Jeremy M.: Accounting for not-at-random missingness through imputation stacking. In: *Statistics in Medicine* 40 (2021), Nr. 27, S. 6118–6132

[3] Buuren, Stef van ; Groothuis-Oudshoorn, Karin: *mice: Multivariate Imputation by Chained Equations in R*. `https://www.jstatsoft.org/v45/i03/`. Version: 2011

[4] Daniel, Rhian M. ; Kenward, Michael G. ; Cousens, Simon N. ; De Stavola, Bianca L.: Using causal diagrams to guide analysis in missing data problems. In: *Statistical methods in medical research* 21 (2012), Nr. 3, S. 243–256

[5] Do, Chuong B. ; Batzoglou, Serafim: What is the expectation maximization algorithm? In: *Nature biotechnology* 26 (2008), Nr. 8, S. 897–899

[6] Edwards, David: *Introduction to graphical modelling*. Springer Science & Business Media, 2012

[7] Goldfeld, Keith ; Wujciak-Jens, Jacob: simstudy: Illuminating research methods through data generation. In: *Journal of Open Source Software* 5 (2020), Nr. 54, 2763. `http://dx.doi.org/10.21105/joss.02763`. – DOI 10.21105/joss.02763

[8] Hauser, Alain ; Bühlmann, Peter: Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs. In: *Journal of Machine Learning Research* 13 (2012), 2409–2464. `https://jmlr.org/papers/v13/hauser12a.html`

[9] Hechenbichler, Klaus ; Schliep, Klaus: Weighted k-nearest-neighbor techniques and ordinal classification. (2004)

[10] Heymans, MW ; Eekhout, I: Applied missing data analysis with SPSS and (R) Studio. 2019. In: *R Bookdown: Amsterdam, Available online: https://bookdown.org/mwheymans/bookmi* (2019)

[11] Højsgaard, Søren ; Edwards, David ; Lauritzen, Steffen: *Graphical models with R*. Springer Science & Business Media, 2012

[12] Honaker, James ; King, Gary ; Blackwell, Matthew: Amelia II: A program for missing data. In: *Journal of statistical software* 45 (2011), S. 1–47

[13] Honaker, James ; King, Gary ; Blackwell, Matthew: Amelia II: A Program for Missing Data. In: *Journal of Statistical Software* 45 (2011), Nr. 7, S. 1–47. `http://dx.doi.org/10.18637/jss.v045.i07`. – DOI 10.18637/jss.v045.i07

[14] Kalisch, Markus ; Bühlman, Peter: Estimating high-dimensional directed acyclic graphs with the PC-algorithm. In: *Journal of Machine Learning Research* 8 (2007), Nr. 3

[15] Kalisch, Markus ; Hauser, A ; Maathuis, MH ; Mächler, Martin: An Overview of the pcalg Package for R. (2020)

[16] Kalisch, Markus ; Mächler, Martin ; Colombo, Diego ; Maathuis, Marloes H. ; Bühlmann, Peter: Causal Inference Using Graphical Models with the R Package pcalg. In: *Journal of Statistical Software* 47 (2012), Nr. 11, S. 1–26. `http://dx.doi.org/10.18637/jss.v047.i11`. – DOI 10.18637/jss.v047.i11

[17] Kleinke, Kristian ; Reinecke, Jost ; Salfrán, Daniel ; Spiess, Martin: *Applied multiple imputation*. Springer, 2020

[18] Lalande, Florian ; Doya, Kenji: Numerical data imputation: Choose kNN over deep learning. In: *Similarity Search and Applications: 15th International Conference, SISAP 2022, Bologna, Italy, October 5–7, 2022, Proceedings* Springer, 2022, S. 3–10

[19] Mohan, Karthika ; Pearl, Judea: Graphical models for processing missing data. In: *Journal of the American Statistical Association* 116 (2021), Nr. 534, S. 1023–1037

[20] Mustillo, Sarah ; Kwon, Soyoung: Auxiliary variables in multiple imputation when data are missing not at random. In: *The Journal of Mathematical Sociology* 39 (2015), Nr. 2, S. 73–91

[21] Pearl, Judea: *Causality*. Cambridge university press, 2009

[22] Peterson, Ryan A.: Finding Optimal Normalizing Transformations via bestNormalize. In: *The R Journal* 13 (2021), Nr. 1, S. 310–329. `http://dx.doi.org/10.32614/RJ-2021-041`. – DOI 10.32614/RJ–2021–041

[23] Peterson, Ryan A. ; Cavanaugh, Joseph E.: Ordered quantile normalization: a semiparametric transformation built for the cross-validation era. In: *Journal of Applied Statistics* 47 (2020), Nr. 13-15, S. 2312–2327. `http://dx.doi.org/10.1080/02664763.2019.1630372`. – DOI 10.1080/02664763.2019.1630372

[24] Petrazzini, Ben O. ; Naya, Hugo ; Lopez-Bello, Fernando ; Vazquez, Gustavo ; Spangenberg, Lucía: Evaluation of different approaches for missing data imputation on features associated to genomic data. In: *BioData mining* 14 (2021), Nr. 1, S. 1–13

[25] SCHLIEP, Klaus ; HECHENBICHLER, Klaus: *kknn: Weighted k-Nearest Neighbors*, 2016. `https://CRAN.R-project.org/package=kknn`. – R package version 1.3.1

[26] SPEED, Terence P. ; KIIVERI, Harri T.: Gaussian Markov distributions over finite graphs. In: *The Annals of Statistics* (1986), S. 138–150

[27] THOEMMES, Felix ; MOHAN, Karthika: Graphical representation of missing data problems. In: *Structural Equation Modeling: A Multidisciplinary Journal* 22 (2015), Nr. 4, S. 631–642

[28] TOMPSETT, Daniel M. ; LEACY, Finbarr ; MORENO-BETANCUR, Margarita ; HERON, Jon ; WHITE, Ian R.: On the use of the not-at-random fully conditional specification (NARFCS) procedure in practice. In: *Statistics in medicine* 37 (2018), Nr. 15, S. 2338–2353

[29] TU, Ruibo ; ZHANG, Cheng ; ACKERMANN, Paul ; MOHAN, Karthika ; KJELLSTRÖM, Hedvig ; ZHANG, Kun: Causal discovery in the presence of missing data. In: *The 22nd International Conference on Artificial Intelligence and Statistics* PMLR, 2019, S. 1762–1770

[30] VAN BUUREN, Stef: *Flexible imputation of missing data.* CRC press, 2018

[31] VAN BUUREN, Stef ; GROOTHUIS-OUDSHOORN, Karin: mice: Multivariate imputation by chained equations in R. In: *Journal of statistical software* 45 (2011), S. 1–67

[32] WICKHAM, Hadley ; AVERICK, Mara ; BRYAN, Jennifer ; CHANG, Winston ; MCGOWAN, Lucy D'Agostino ; FRANÇOIS, Romain ; GROLEMUND, Garrett ; HAYES, Alex ; HENRY, Lionel ; HESTER, Jim ; KUHN, Max ; PEDERSEN, Thomas L. ; MILLER, Evan ; BACHE, Stephan M. ; MÜLLER, Kirill ; OOMS, Jeroen ; ROBINSON, David ; SEIDEL, Dana P. ; SPINU, Vitalie ; TAKAHASHI, Kohske ; VAUGHAN, Davis ; WILKE, Claus ; WOO, Kara ; YUTANI, Hiroaki: Welcome to the {tidyverse}. 4 (2019), S. 1686. `http://dx.doi.org/10.21105/joss.01686`. – DOI 10.21105/joss.01686

[33] YOON, Jinsung ; JORDON, James ; SCHAAR, Mihaela: Gain: Missing data imputation using generative adversarial nets. In: *International conference on machine learning* PMLR, 2018, S. 5689–5698

Appendices

A Data Simulation

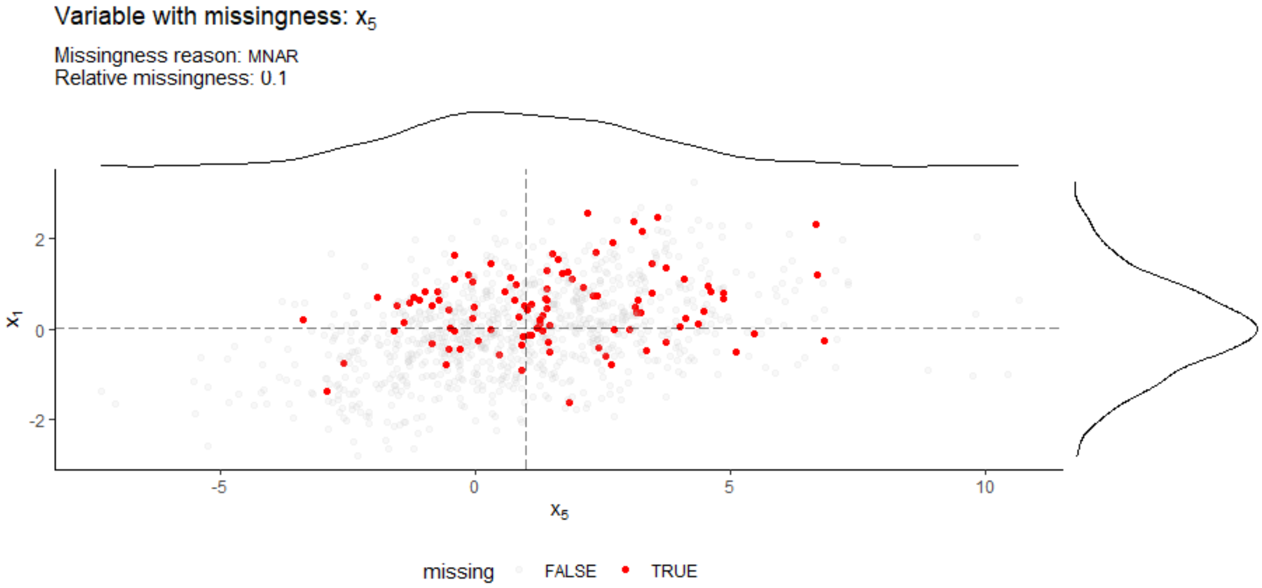


Figure 49: Graph that displays the relationship of the missing values of the normal distributed variable x_5 on the x-axis and the normal distributed Variable x_1, that contains MCAR values, on the y-axis which influences the missing process of x_5 in such a way that it is MNAR.

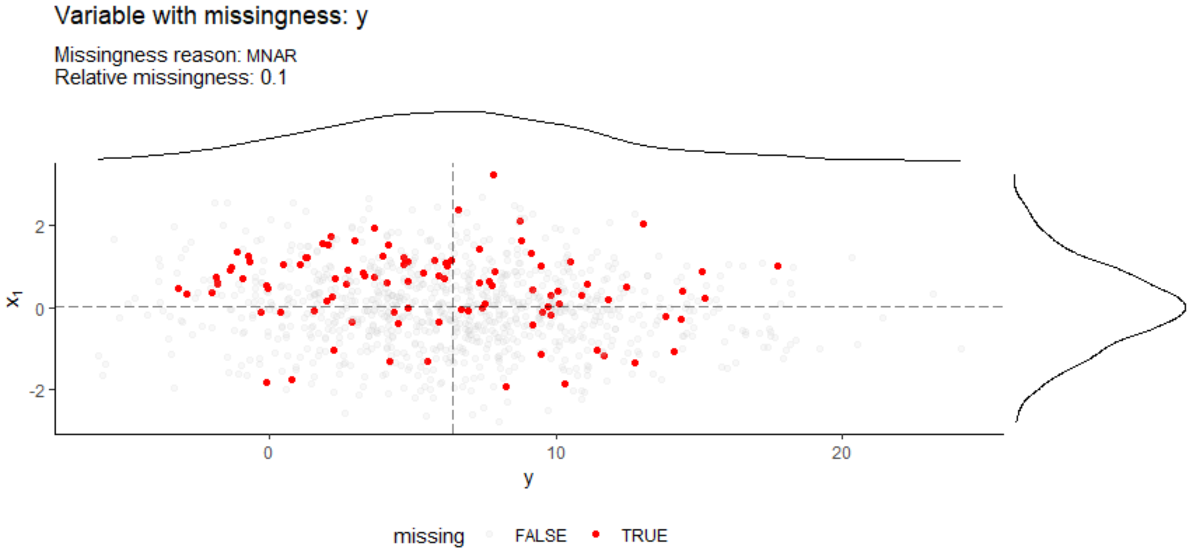


Figure 50: Graph that displays the relationship of the missing values of the normal distributed variable y on the x-axis and the normal distributed Variable x_1, that contains MCAR values, on the y-axis which influences the missing process of y in such a way that it is MNAR.

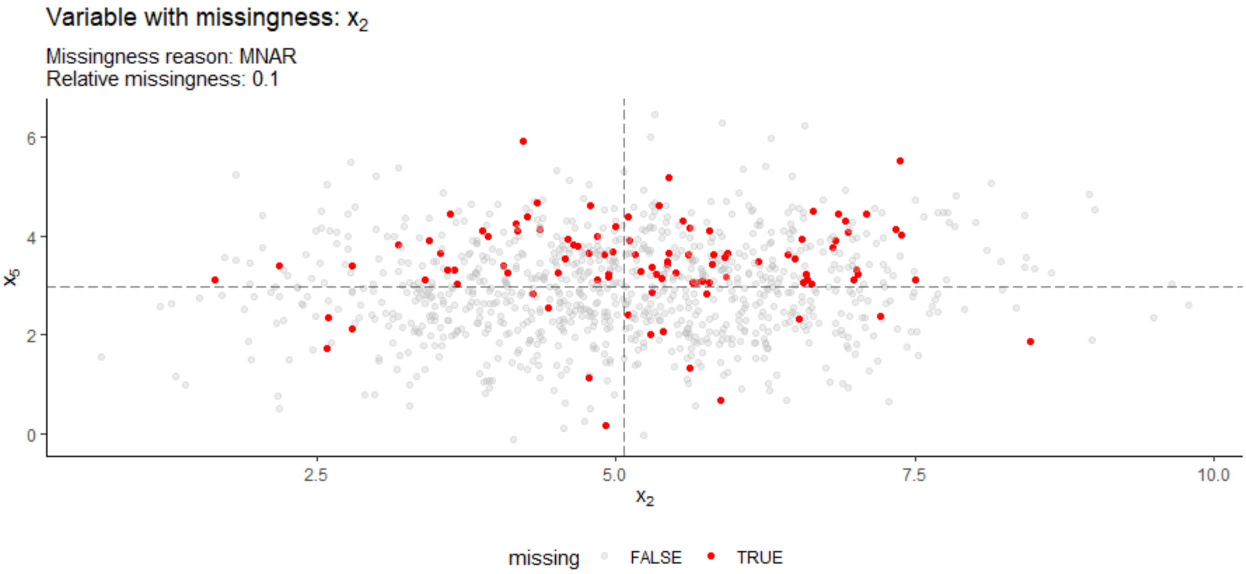


Figure 51: Graph that displays the relationship of the missing values of the normal distributed dataset with x_2 on the x-axis and x_5, that contains MCAR values, on the y-axis which influences the missing process of x_2 in such a way that it is MNAR.

B Graphical models

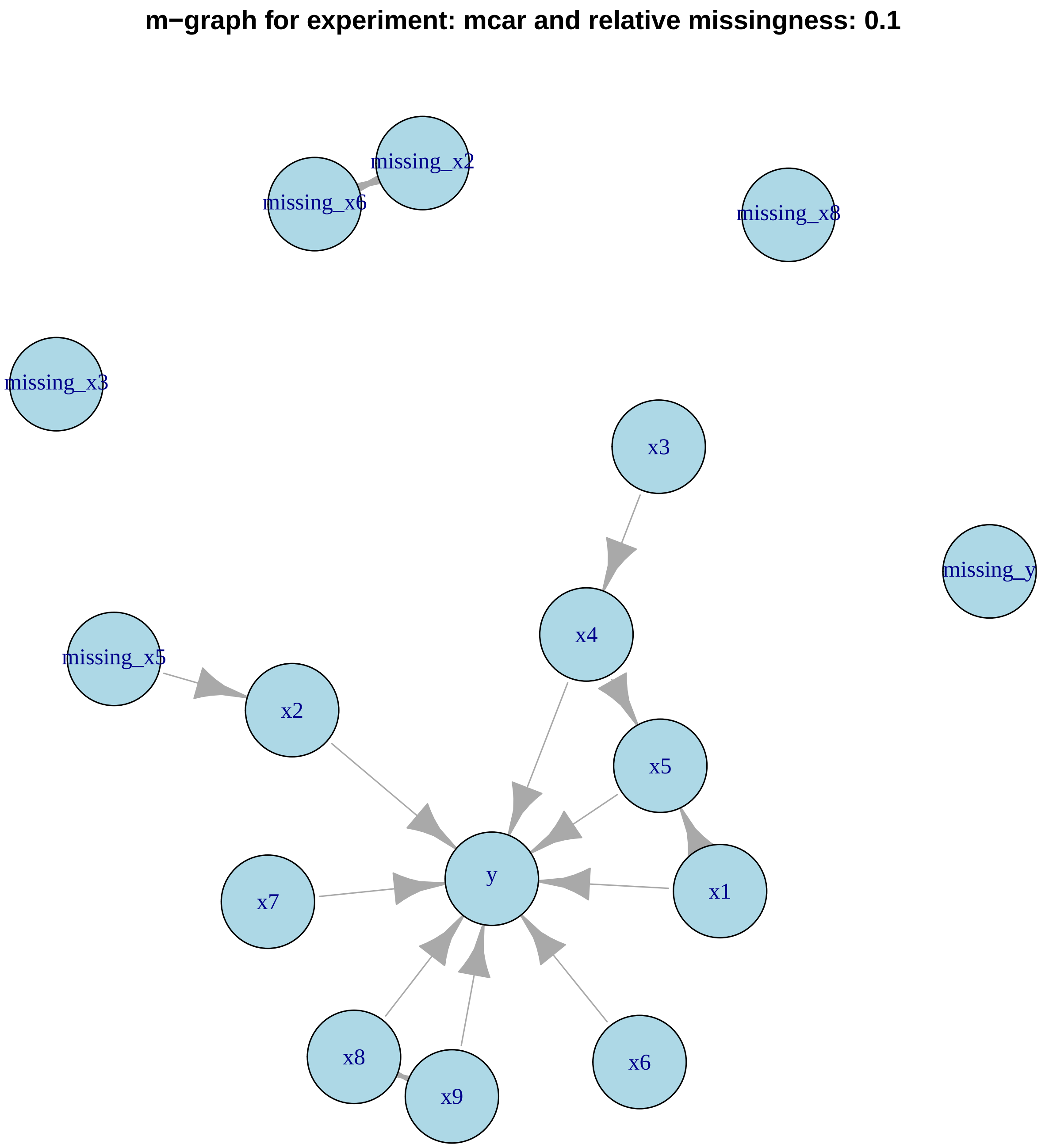


Figure 52: The graph displays the prediction of the graphical model when using the pc-algorithm when having simulated MCAR variables only and a missingness rate of 0.1 for each variable.

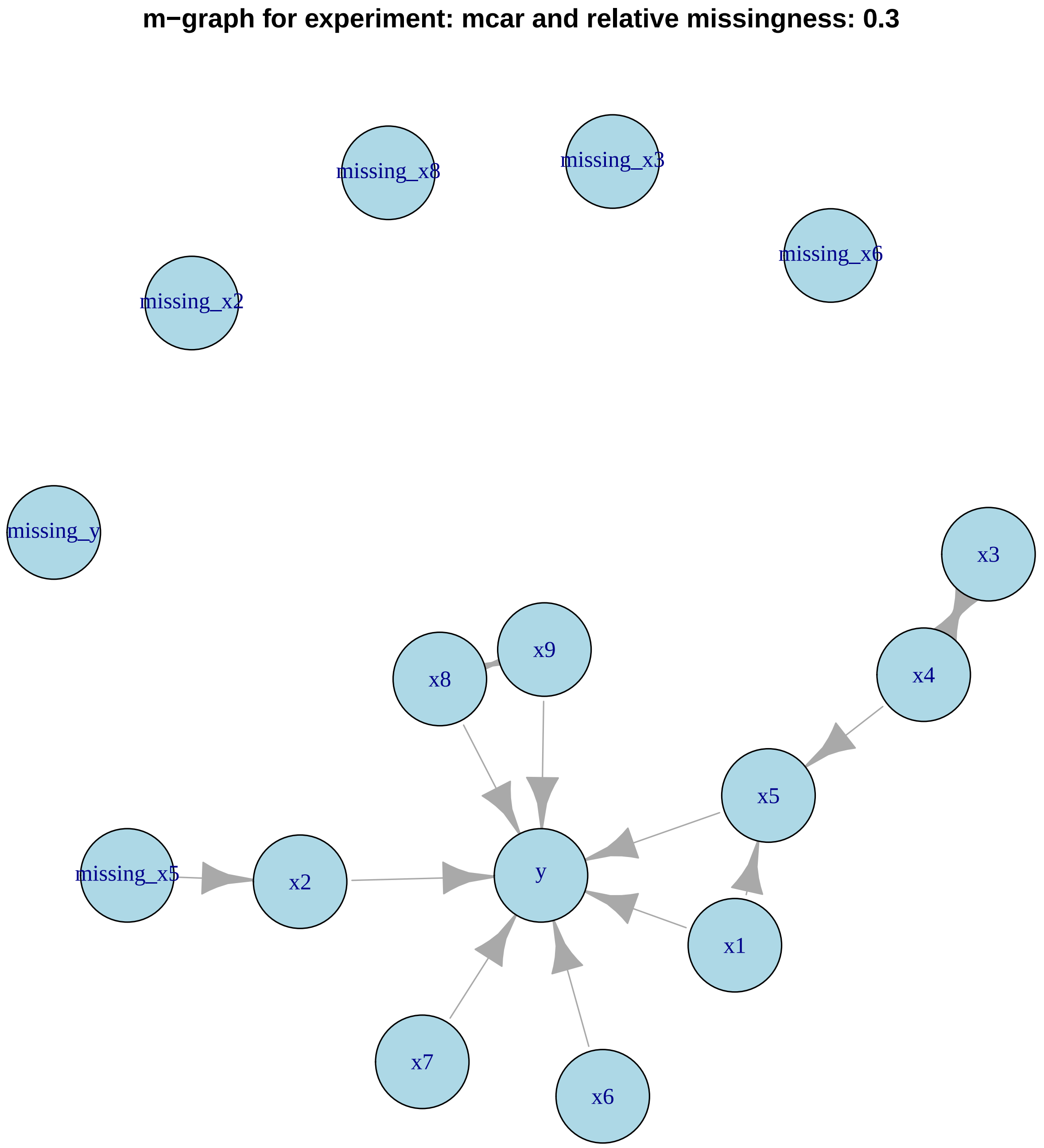


Figure 53: The graph displays the prediction of the graphical model when using the pc-algorithm when having simulated MCAR variables only and a missingness rate of 0.3 for each variable.

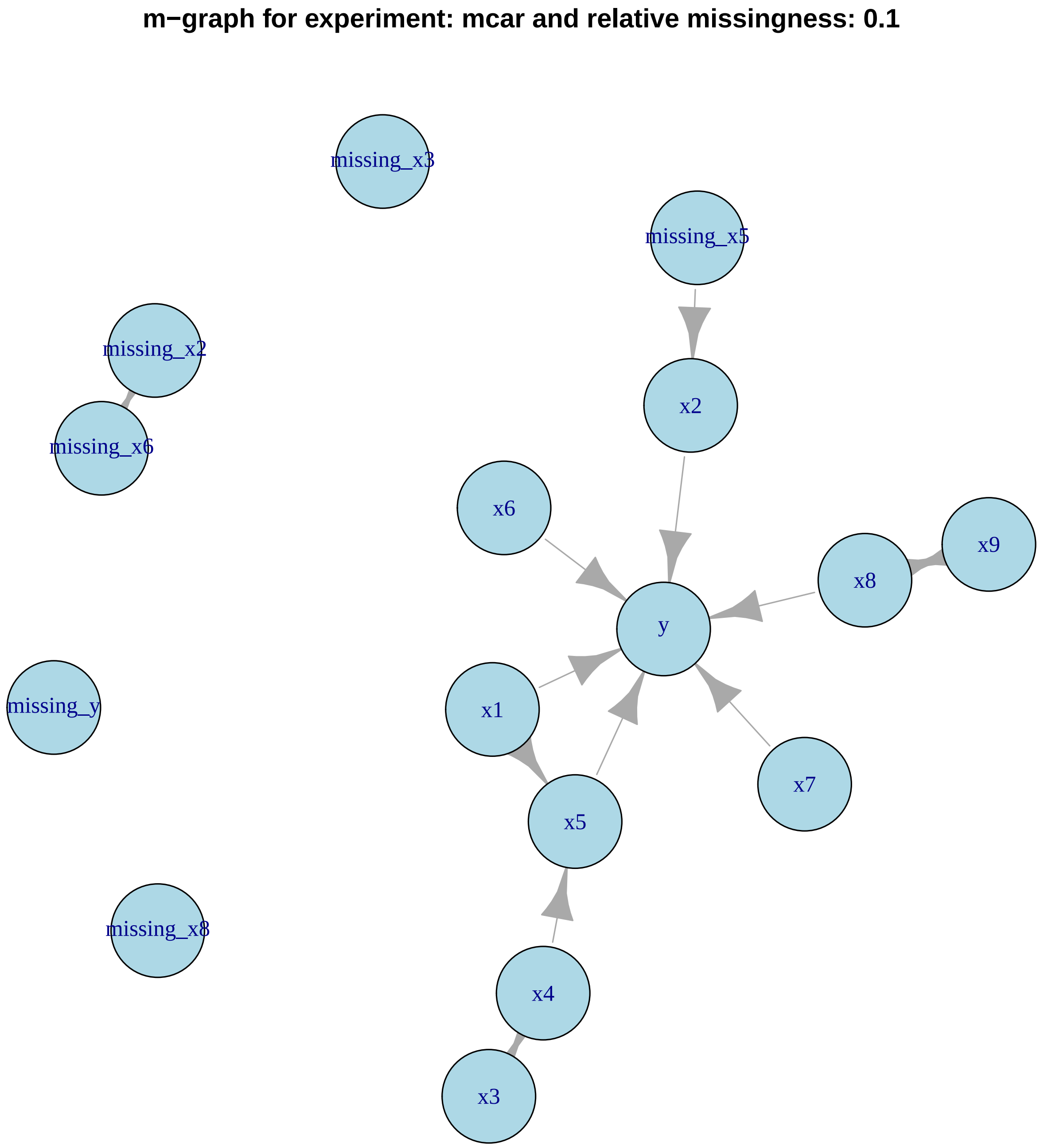


Figure 54: The graph displays the prediction of the graphical model when using the mvpc-algorithm when having simulated MCAR variables only and a missingness rate of 0.1 for each variable.

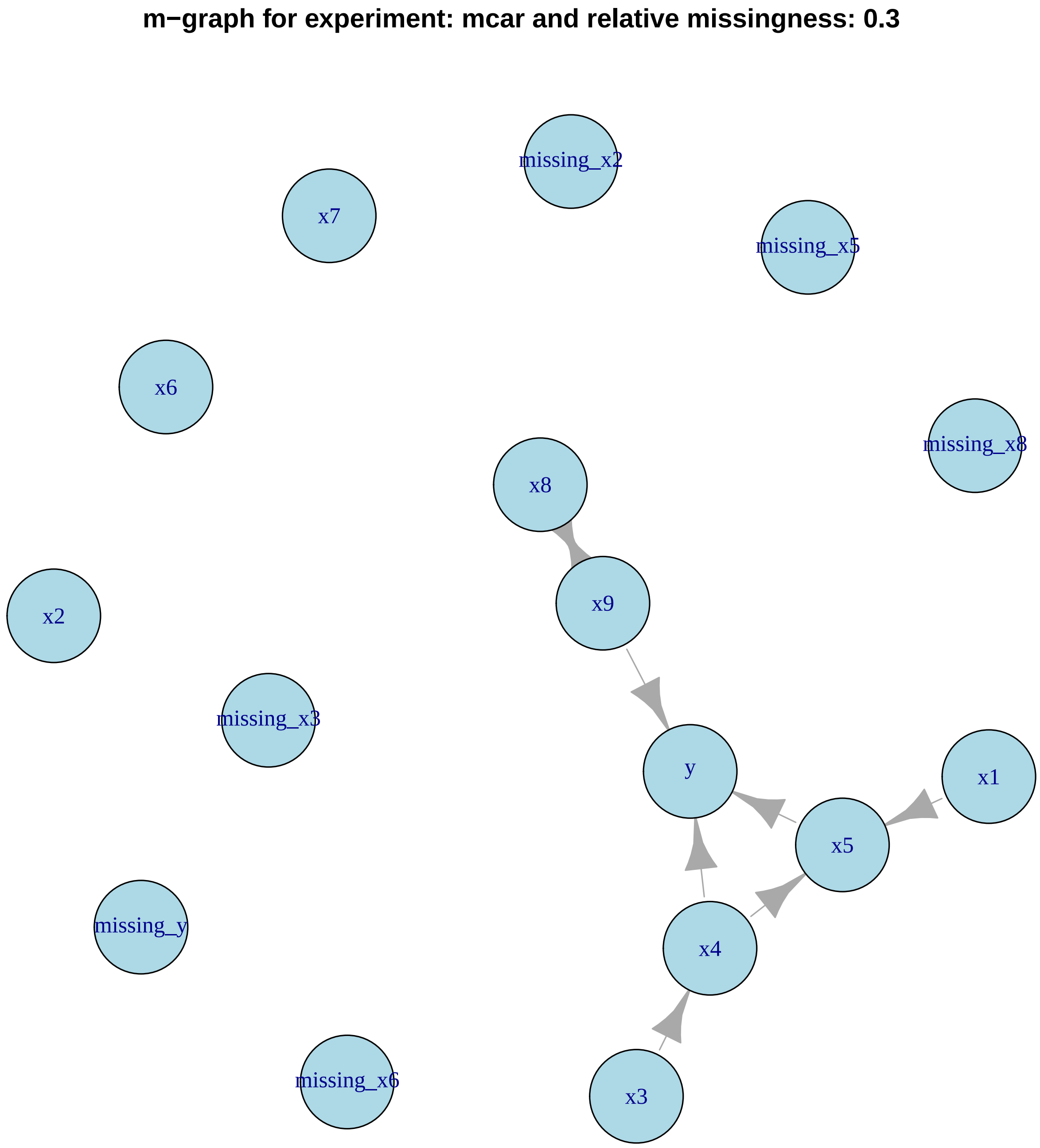


Figure 55: The graph displays the prediction of the graphical model when using the mvpc-algorithm when having simulated MCAR variables only and a missingness rate of 0.3 for each variable.

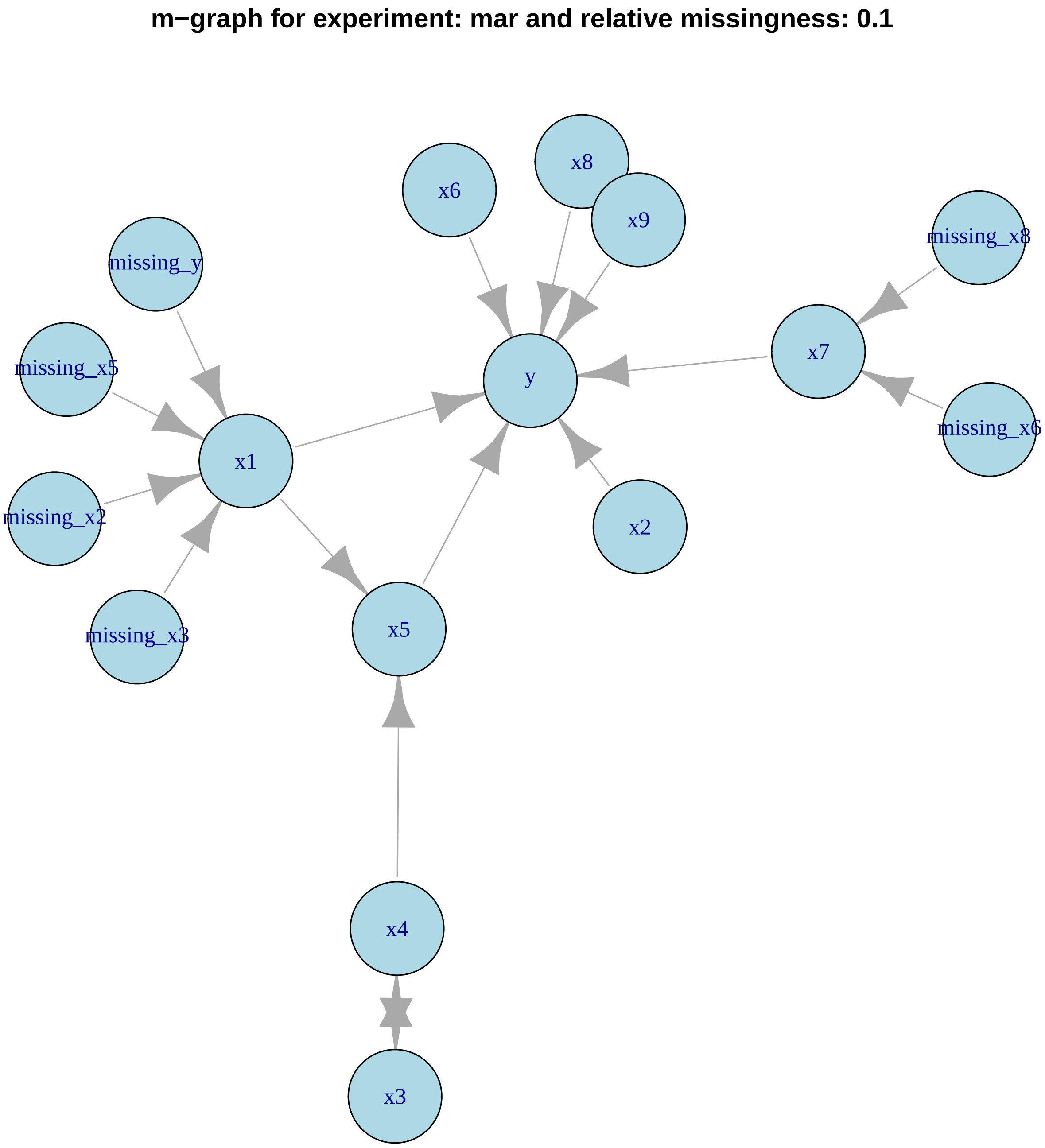


Figure 56: The graph displays the prediction of the graphical model when using the pc-algorithm when having simulated MAR variables only and a missingness rate of 0.1 for each variable.

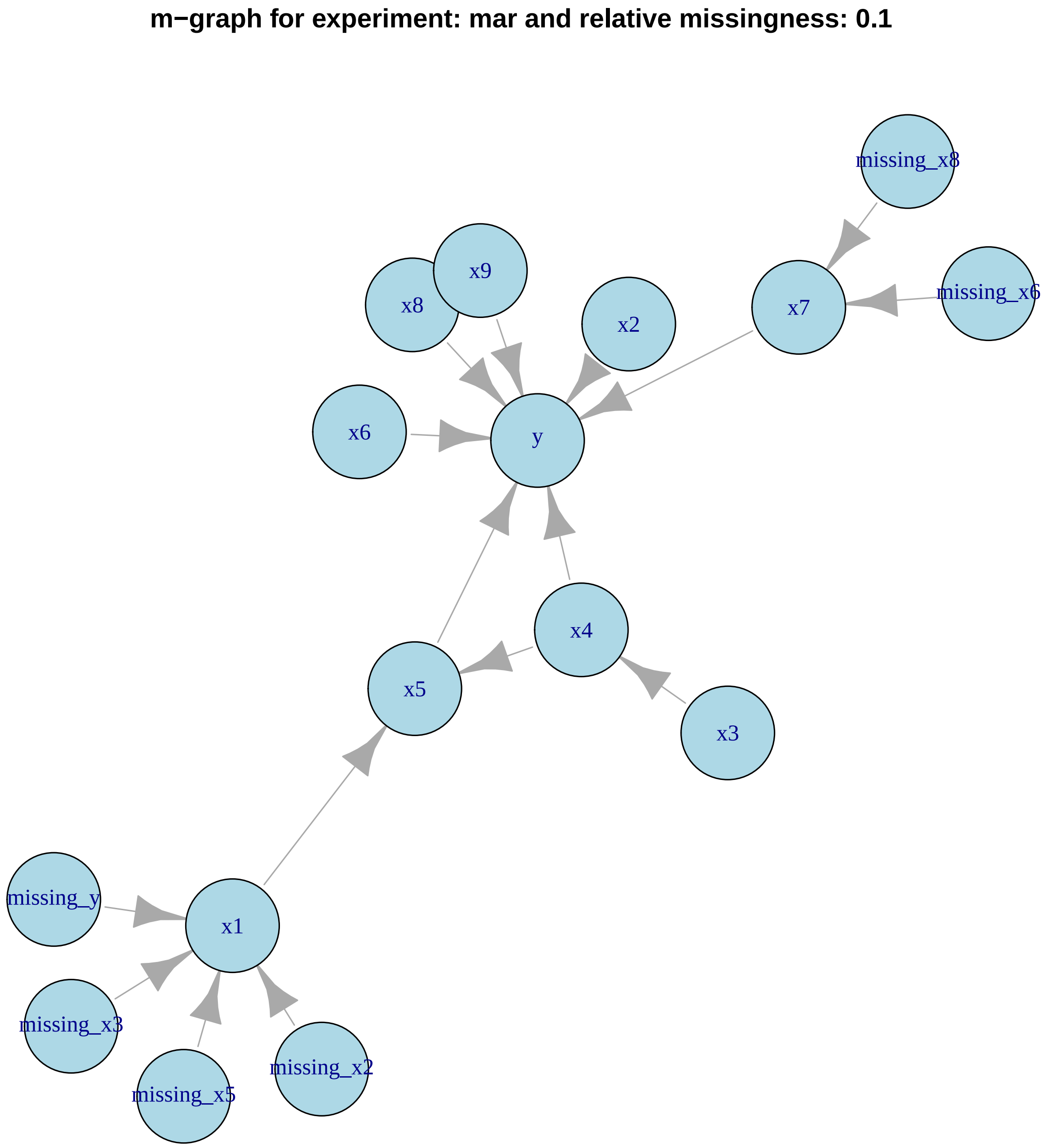


Figure 57: The graph displays the prediction of the graphical model when using the mvpc-algorithm when having simulated MAR variables only and a missingness rate of 0.1 for each variable.

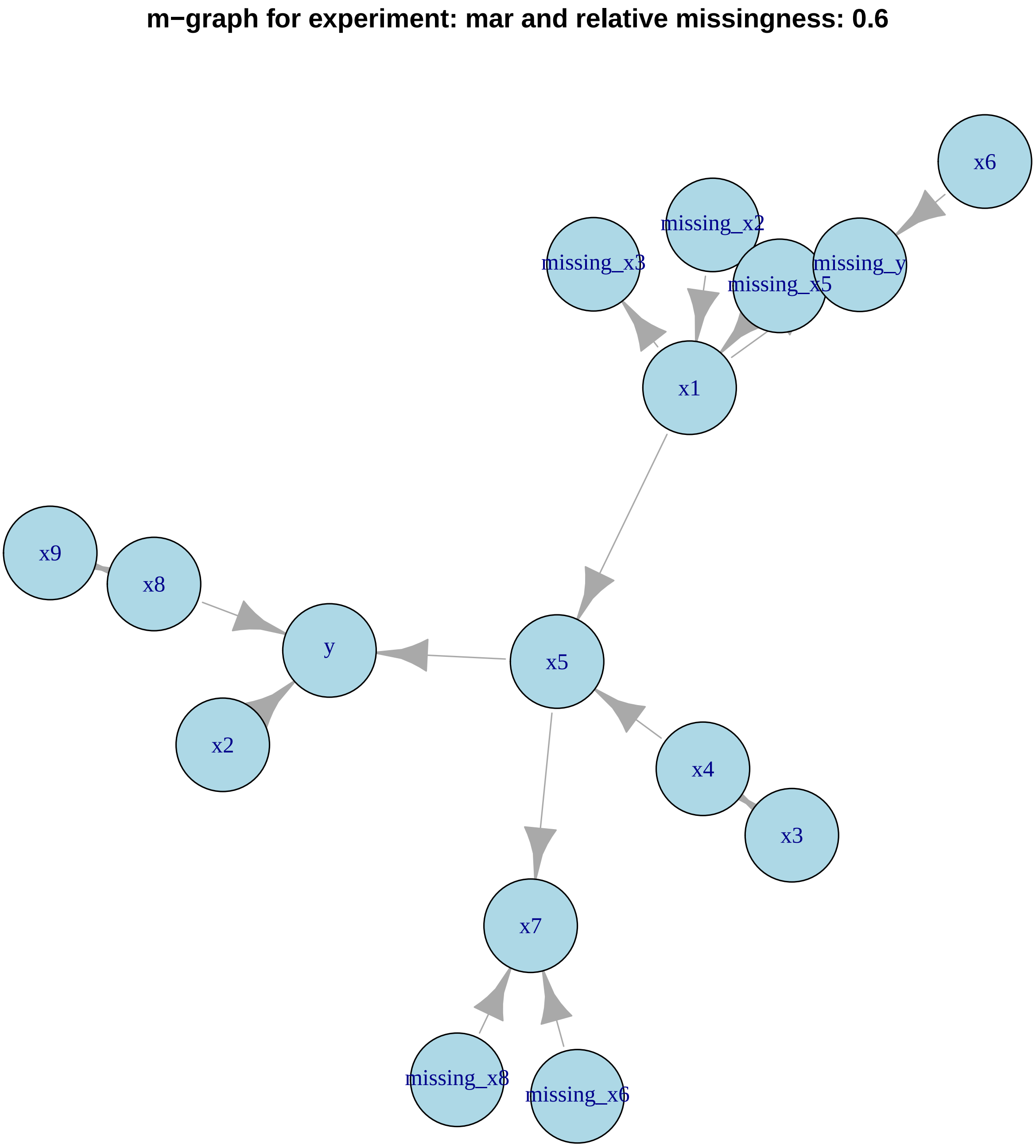


Figure 58: The graph displays the prediction of the graphical model when using the pc-algorithm when having simulated MAR variables only and a missingness rate of 0.6 for each variable.

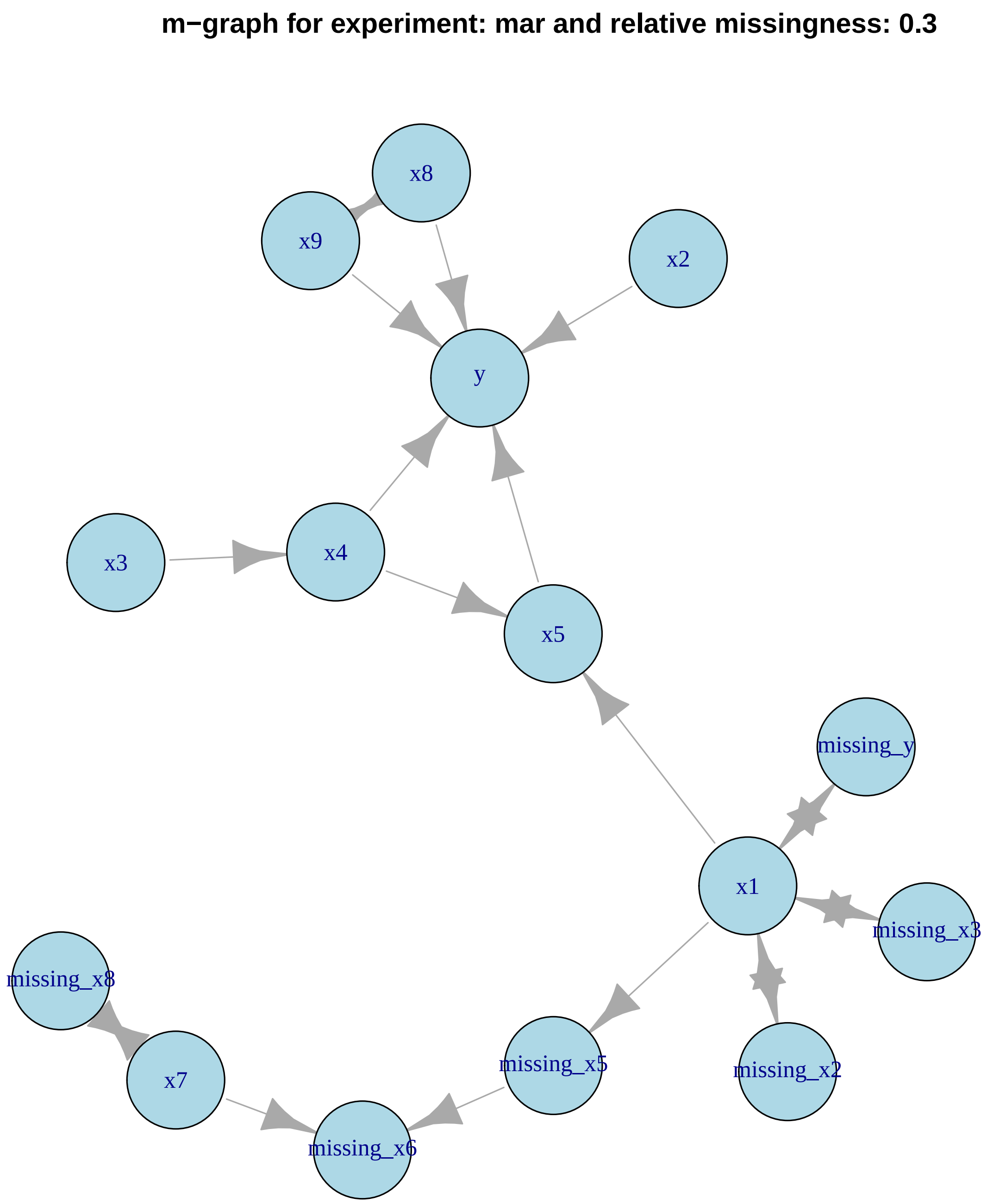


Figure 59: The graph displays the prediction of the graphical model when using the mvpc-algorithm when having simulated MAR variables only and a missingness rate of 0.3 for each variable.

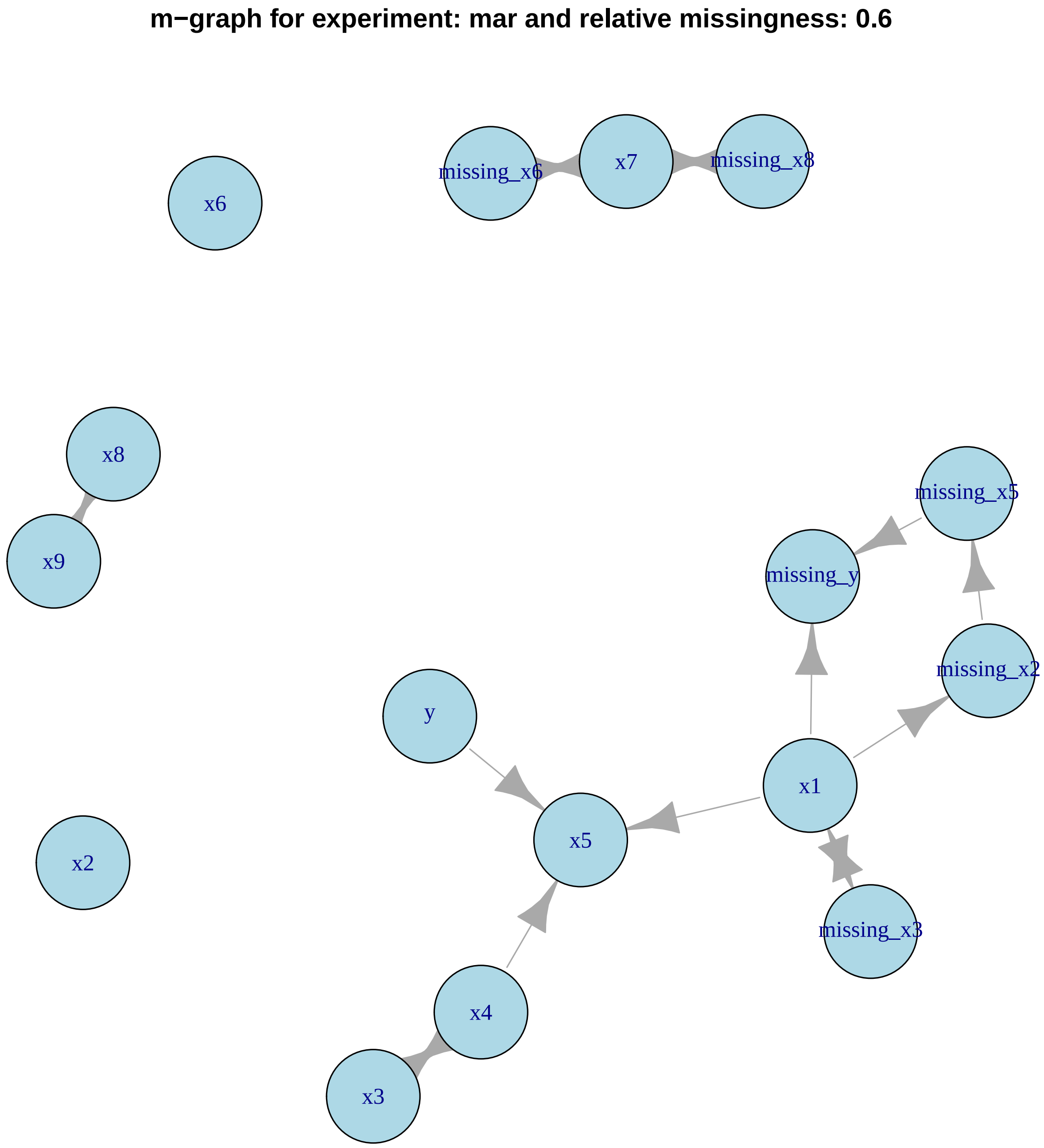


Figure 60: The graph displays the prediction of the graphical model when using the mvpc-algorithm when having simulated MAR variables only and a missingness rate of 0.6 for each variable.

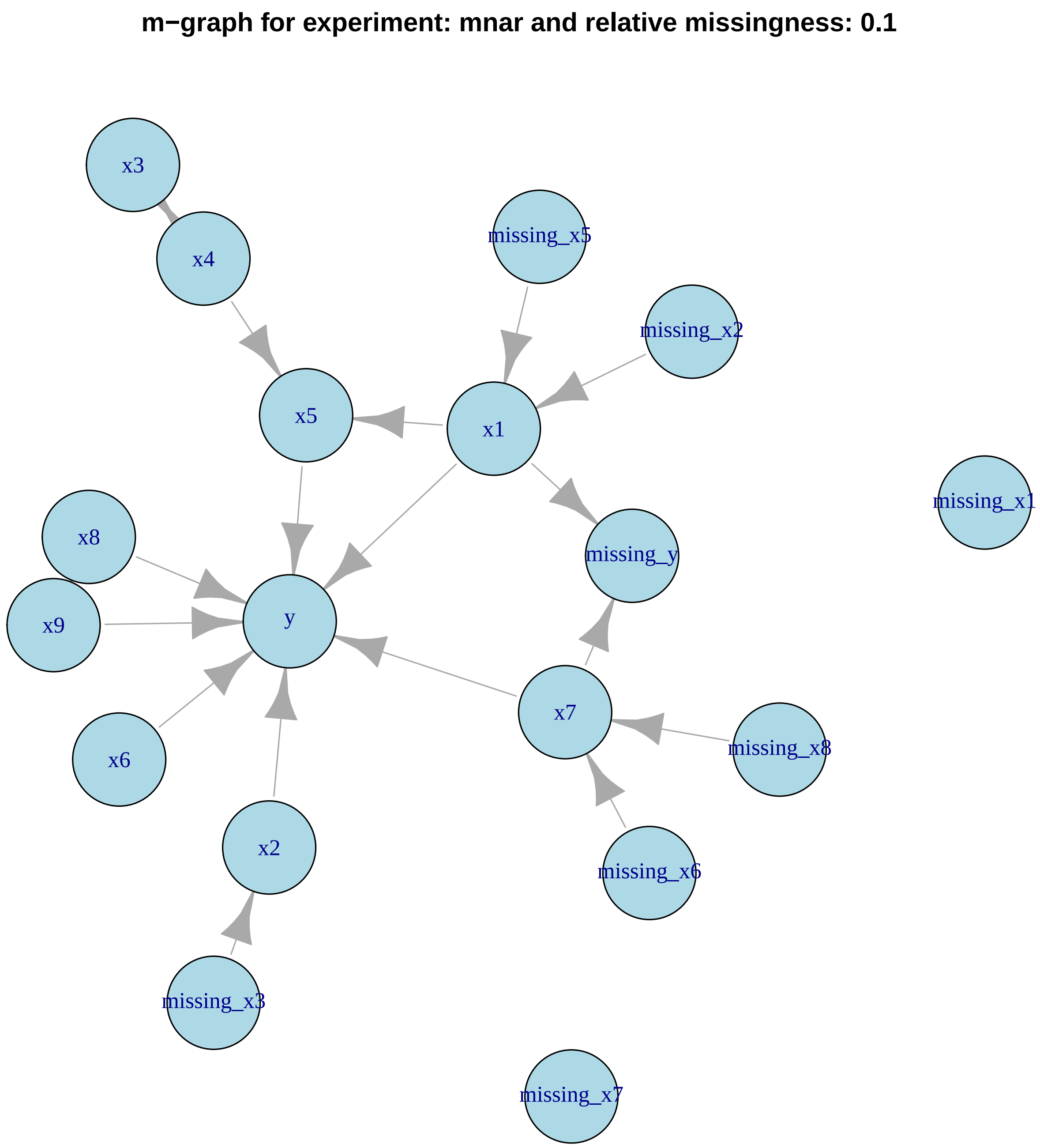


Figure 61: The graph displays the prediction of the graphical model when using the pc-algorithm when having simulated MNAR variables only and a missingness rate of 0.1 for each variable.

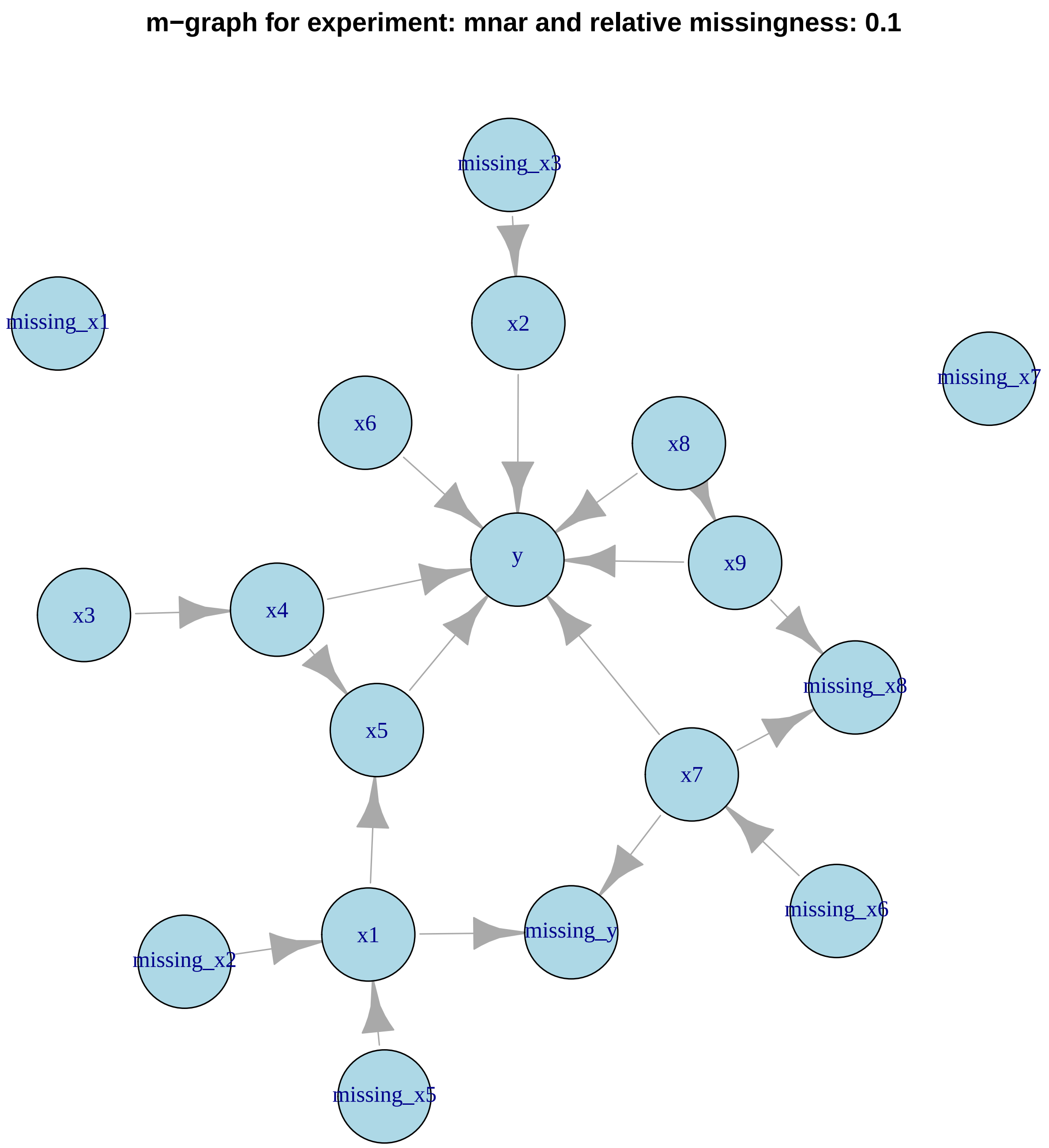


Figure 62: The graph displays the prediction of the graphical model when using the mvpc-algorithm when having simulated MNAR variables only and a missingness rate of 0.1 for each variable.

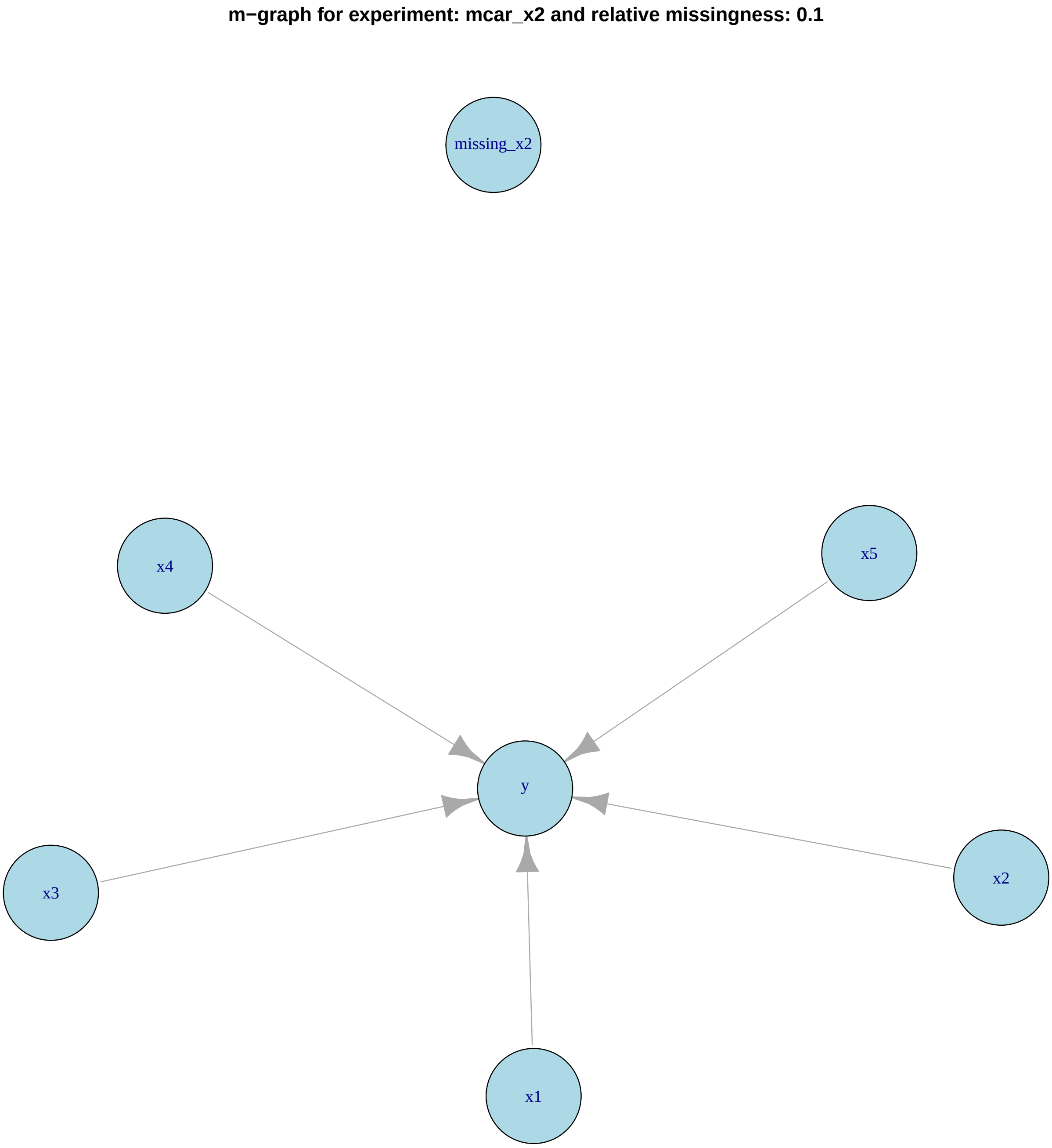

m−graph for experiment: mcar_x2 and relative missingness: 0.1
missing_x2
x4
x5
y
x3
x2
x1

Figure 63: The graph shows the correct graphical model under MCAR in the multivariate normal dataset, which was also predicted by both the pc-algorithm and the mvpc-algorithm when having a missingness rate of 0.1 and 0.6.

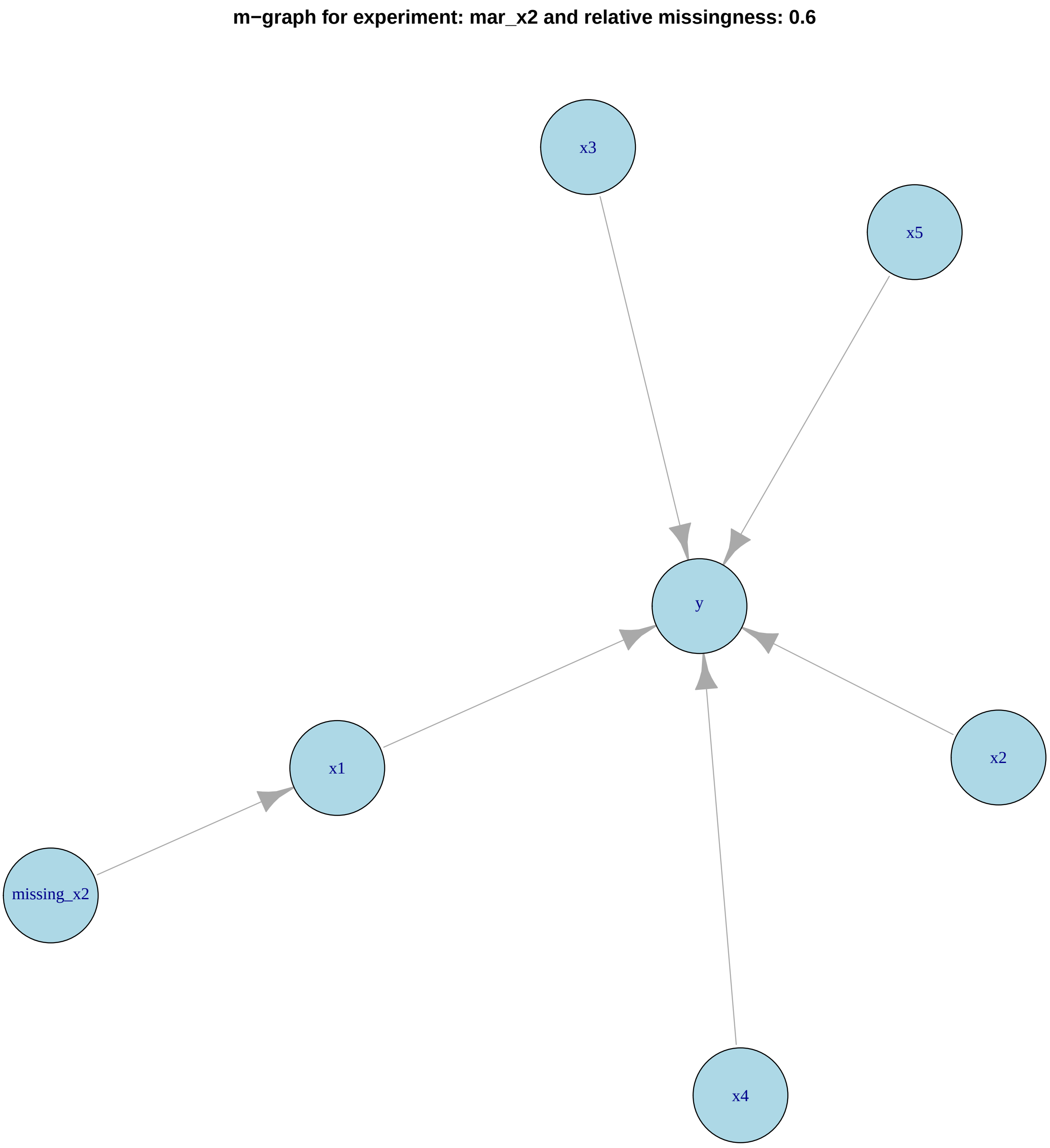

m−graph for experiment: mar_x2 and relative missingness: 0.6
x3
x5
y
x1
x2
missing_x2
x4

Figure 64: The graph shows the correct graphical model under MAR in the multivariate normal dataset, which was also predicted by both the pc-algorithm and the mvpc-algorithm when having a missingness rate of 0.3 and 0.6.

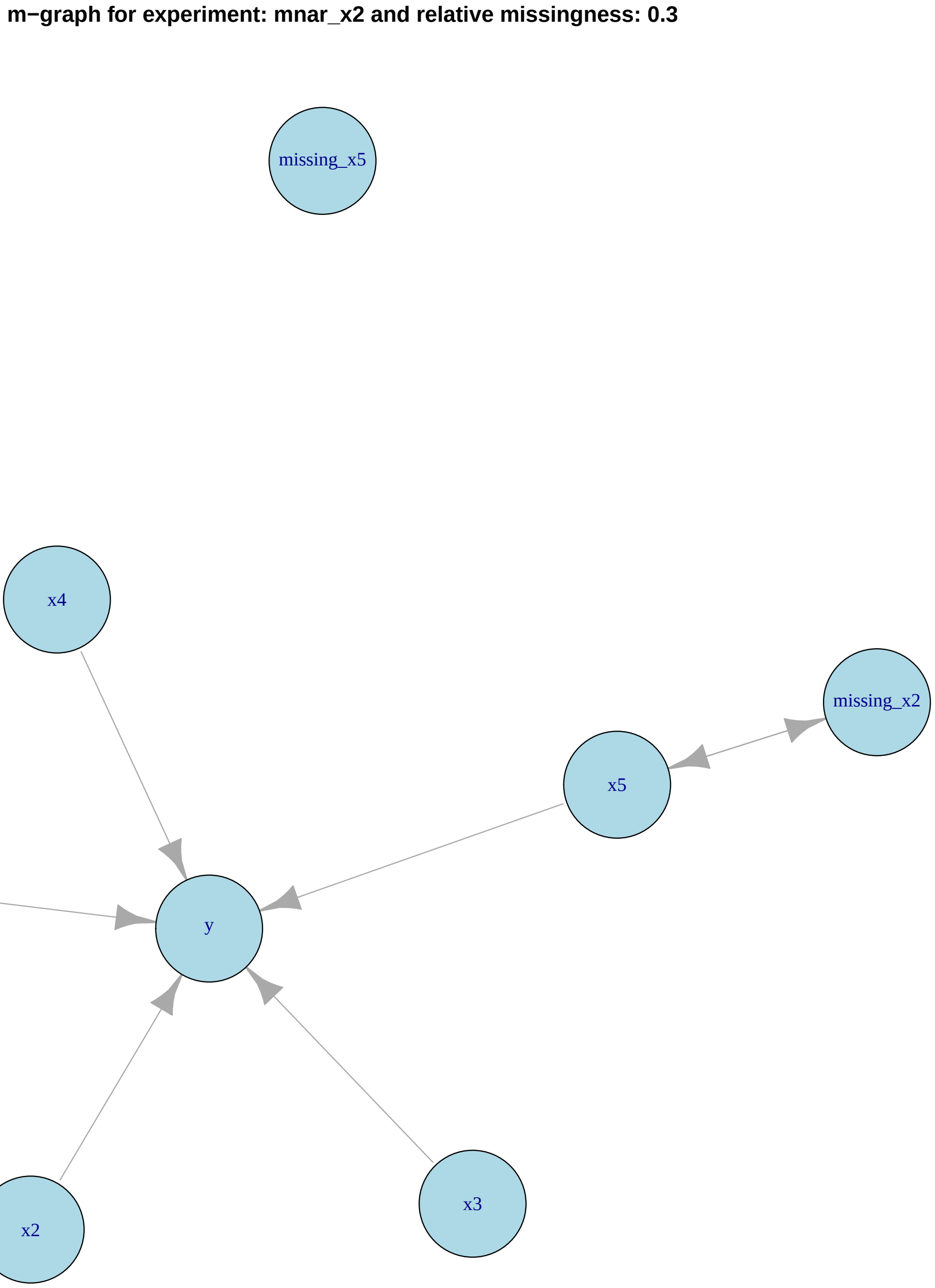

m−graph for experiment: mnar_x2 and relative missingness: 0.3
missing_x5
x4
missing_x2
x5
x1
y
x2
x3

Figure 65: The graph shows the correct graphical model under the MNAR scenario in the multivariate normal dataset, which was also predicted by both the pc-algorithm and the mvpc-algorithm when having a missingness rate of 0.3.

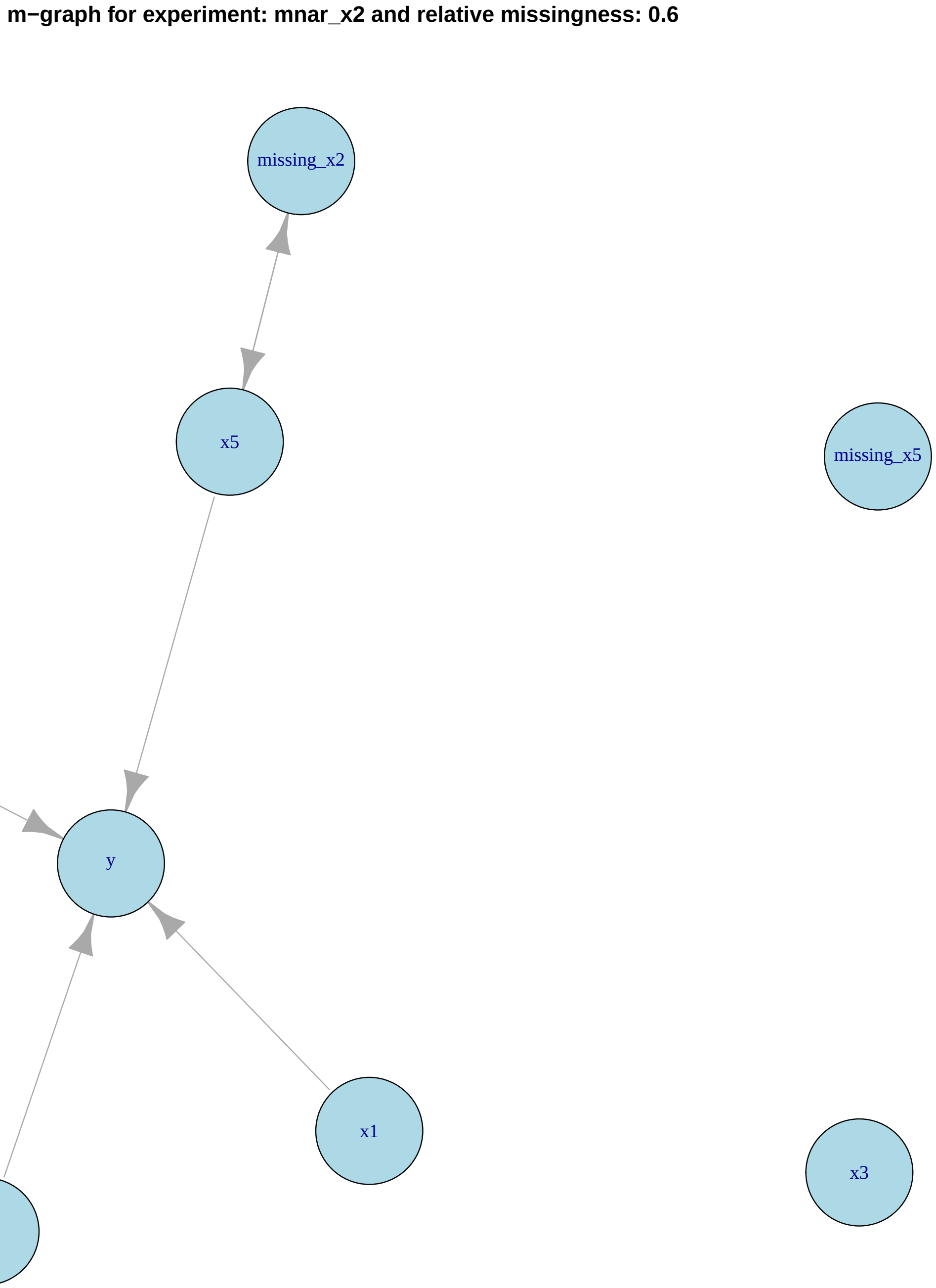

m−graph for experiment: mnar_x2 and relative missingness: 0.6
missing_x2
x5
missing_x5
x4
y
x1
x2
x3

Figure 66: The graph shows the predicted graphical model under MNAR in the multivariate normal dataset when having a missingness rate of 0.6 and having used the mvpc-algorithm.

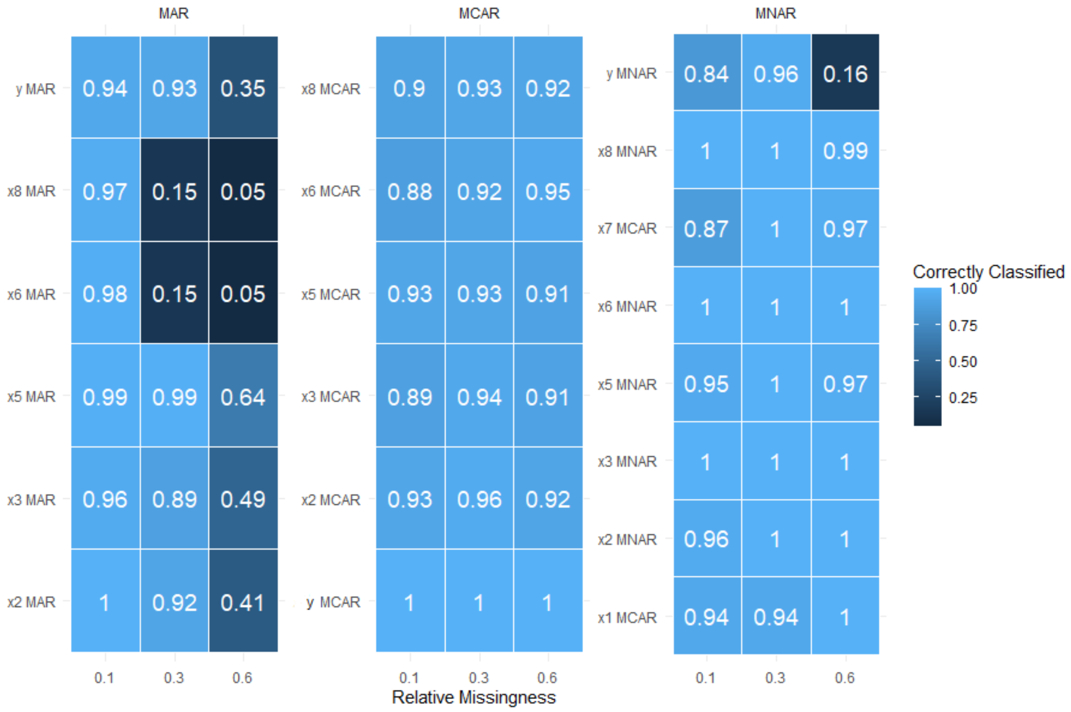


Figure 67: Averaged evaluation of the classification predictions of 100 replicated graphical models for the mixed dataset when using the mvpc-algorithm and a sample size of $n = 500$.

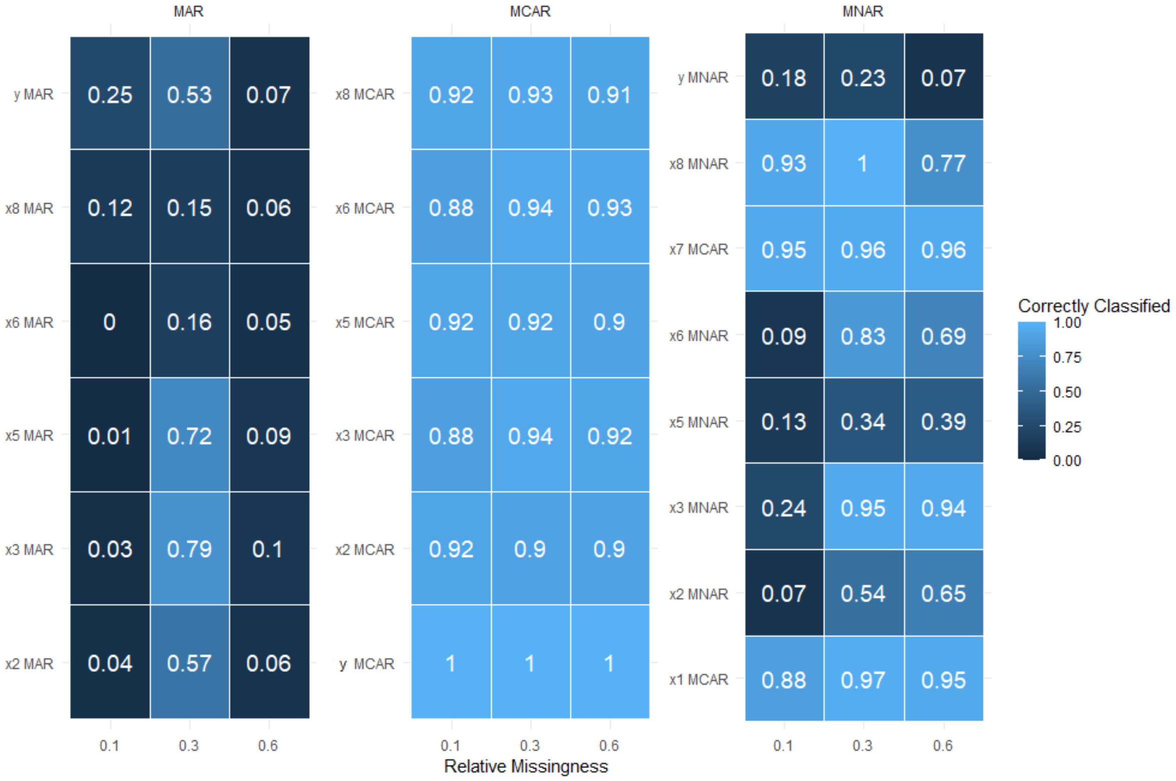


Figure 68: Averaged evaluation of the classification predictions of 100 replicated graphical models for the mixed dataset when using the mvpc-algorithm and a sample size of $n = 100$.

C Imputation

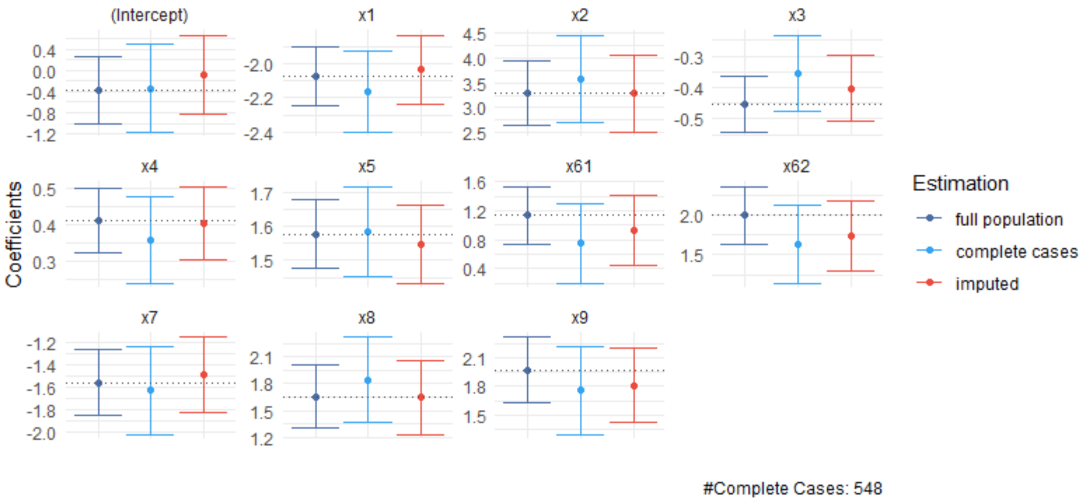


Figure 69: Pooled coefficients from Amelia imputation in the MCAR scenario with 0.1 missingness in each variable compared to the list wise deletion calculated coefficients and the population based coefficients.

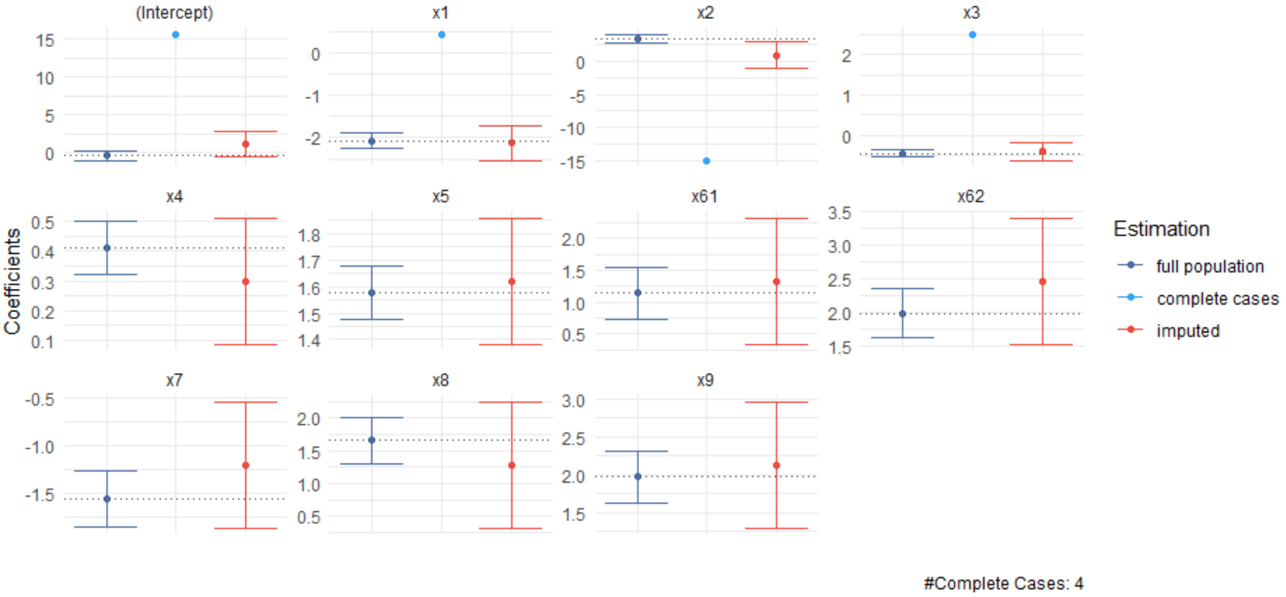


Figure 70: Pooled coefficients from Amelia imputation in the MCAR scenario with 0.6 missingness in each variable compared to the list wise deletion calculated coefficients and the population based coefficients.

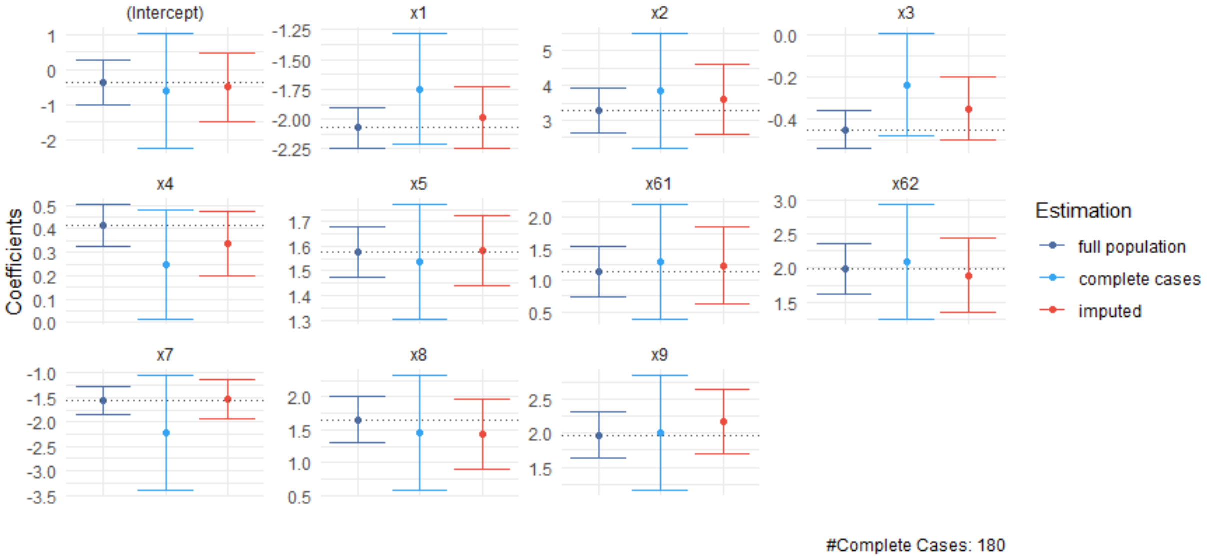


Figure 71: Pooled coefficients from Amelia imputation in the MAR scenario with 0.3 missingness in each variable compared to the list wise deletion calculated coefficients and the population based coefficients.

Declaration of Authorship

I hereby confirm that I have written the accompanying thesis by myself, without contributions from any sources other than those cited in the text. This also applies to all graphics, images and tables included in the thesis.

Eleftheria Papavasiliou

Munich, 10.08.2023