Technische Universität München

Department of Mathematics

Master's Thesis

Vine Copula Imputation

Stephan Zeisberger

Supervisor: Claudia Czado

Advisor: Eike Brechmann

Submission Date: 20.01.2014

I assure the single handed composition of this master's thesis only supported by declared resources.

Garching,

Zusammenfassung

In dieser Arbeit werden drei neue Methoden präsentiert und getestet, mit deren Hilfe ein lückenhafter Datensatz vervollständigt werden kann. Dieses Einfügen fehlender Werte nennt man Imputation. Alle drei Imputationsmethoden basieren auf der Modellierung von Daten mit Hilfe der sogenannten Vine Copulas. Diese Art der Modellierung ermöglicht sehr flexible Abhängigkeitsstrukturen zwischen mehreren zufälligen Ereignissen. Dies ist für gute Schätzungen der fehlenden Werte ein enormer Vorteil, da gerade diese Abhängigkeit genutzt wird um basierend auf den gegebenen Daten die nicht vorhandenen zu erlangen. Alle drei Methoden erlauben den allgemeinsten Fall der Vine Copulas, den R-vine, ausgestattet mit (verschiedenen) parametrischen, bivariaten Copulafamilien (z.B. Gauss, Gumbel, Clayton, ...).

Jede der untersuchten Vine Copula Imputationsmethoden schätzt anfangs ein Modell basierend auf den gegebenen Werten des Datensatzes. Zwei versuchen nun durch simulieren eines Schätzwertes gegeben den restlichen, bekannten Daten einen möglichst guten "Lückenfüller" zu erstellen um den Datensatz zu vervollständigen. Die übrig gebliebene Methode errechnet durch den bedingten Erwartungswert jeweils einen Imputationswert.

Es werden Algorithmen erarbeitet, die es dem Leser erleichtern, die präsentierte Theorie selbst in die Praxis umzusetzen. Unter anderem eine Mglichkeit, wie Werte eines bestimmten R-vines mit schon teilweise gegebenen Daten simuliert werden können. An einer Simulationsstudie mit verschiedenen Szenarien können Stärken und Schwächen der Vine Copula Imputationsmethoden im Vergleich zu schon existierenden Verfahren untersucht werden. Diese Simulationsstudie bewertet zugleich alle getesteten Imputationsmethoden.

Schließlich wird am Ende dieser Arbeit ein Datensatz aus einer medizinischen Studie mit fehlenden Daten untersucht. Dies bietet ebenfalls eine Möglichkeit zur Evaluation der verschiedenen Verfahren und zeigt gleichzeitig Herausforderungen, die mit den neu entwickelten Methoden in der Praxis einher gehen.

Einfache Methoden zur Imputation (simple imputation) für weniger komplexe Probleme sind schon länger bekannt. 1976 formulierte Rubin (see Rubin D. B., (1976)) ein Modell für unvollständige Datensätze und führte den Begriff missing at random (MAR) ein, der von diesem Zeitpunkt an in fast jeder Literatur über Imputation zu finden ist.

Die Theorie der Vine Copulas geht zurück auf Joe (see Joe H., (1996)). Er konstruierte multivariate Verteilungsfunktionen mit Hilfe von einfachen Bausteinen, die er "pair-copulas" nannte.

Die Zusammenführung dieser beiden Theorien (und verschiedener Erweiterungen) findet nun in dieser Arbeit statt und endet in drei separaten, zur Anwendung bereiten Imputationsverfahren.

Contents

List of Algorithms

Chapter 1

Motivation

In nearly every survey the presence of missing data occurs. For statistical analysis, this is a real issue, because incomplete data sets are challenging. It is not possible to apply common analyzing procedures. They presuppose observed and complete data. If there are only few values missing, one can consider just disregarding the incomplete observations and working with the complete cases. In many applications, for example in medical surveys, there are two reasons that do not make this procedure an alternative. One, there is often not a lot of data available and losing even a small amount of observations is hardly compensable. Two, it is very expensive to get the information, so collected data, on which money was spent, is not thrown out just because there are some parts missing.

Further problems occur with missing data. One has to distinguish between different types of missing values and why they are missing. There are a lot reasons why data might be incomplete. Following is an explanation of the different scenarios we study in this thesis.

1.1 Difficulties with Missing Data

There are many different forms of missing data in the world of surveys. Following the notation in Little R. J. A. (1987) (see Little R. J. A. (1987), pp. 14-17), we will mention several of them and describe what kind of problems we seek to solve with the techniques presented in this thesis.

At first one needs to distinguish between unit nonresponse and item nonresponse. Imagine a company wanting to test a new product and sending it, together with a questionnaire containing 10 questions, to 1000 test persons. Only 990 of the testers send the questionnaire back to the company. This is called unit nonresponse. Although it would be possible to solve, this kind of missing data is not our main focus. We are interested in solving the problem of item nonresponse. Again, imagine the situation as before with the difference that all 1000 questionnaire were sent back to the company, but one or more test persons haven't answered all of the 10 questions.

Now one can distinguish between various types of item nonresponse like in Rubin (1976): Missing Completely at Random (MCAR), Missing at Random (MAR) and Not Missing at Random (NMAR).

(**MCAR**): Suppose there was a study where 1000 persons were asked 10 questions about their last holiday experience. Because of a printing mistake, question one was not on every questionnaire, thus 5 of the 1000 people just could not answer the first question. Hence there is no observable reason for this lack of data with respect to the measured values. In a mathematical sense, suppose there is a 10-*dimensional* random variable $X = (X_1,\ldots,X_{10})$ with some missing data in X_1. Now we say the data is missing completely at random if the absence of X_1 does not depend on the values of $X_1,\ldots,X_{10}$ i.e.

$$P(X_1 \text{ is missing}|X_1,\ldots,X_{10}) = P(X_1 \text{ is missing}).$$

Of course there are reasons why the data are missing, but it does not depend on the values of interest.

(**MAR**): There is a very famous example in the literature (see for example Rubin D. B. 1987, pp. 5-6) where the missing at random case often occurs. When people are asked about their wage, the wealthier ones do not answer the question with a positive probability. Here, the reasons why they show this behavior is not of great interest, but how we can measure it. For example, richer people might live in certain districts or maybe have certain occupations. So one can conclude that, if a person who did not answer the wage question lives in a specific part of town, and has a specific job, she or he probably has a very high income. Again in mathematical terms, data on the random variable wage X_1 of a 3-*dimensional* random vector $(X_1 \text{ (wage)}, X_2 \text{ (district)}, X_3 \text{ (occupation)})$ is said to be missing at random if the absence completely depends on the random variables X_2 and X_3, i.e.

$$P(X_1 \text{ is missing}|X_1, X_2, X_3) = P(X_1 \text{ is missing}|X_2, X_3).$$

(**NMAR**): If data is neither MCAR nor MAR, then it belongs to the class of NMAR. Because in this thesis, the main point is to use dependencies between missing data and given values, this case is not of interest here. One example could be found (see for example Death Penalty Information Center, 2013) in the death penalty data from the USA. Until now (28.10.2013), 31 people have been executed during the year 2013. Only one of them is a woman. Imagine that sex were observed in the study and female gender was marked with an asterisk in the tables, and there were no sign for male. Caused by a misunderstanding between data collectors and statisticians, the male gender is treated as if it were missing now, because there is no sign in the tables. So only female sex is observed, with no reason identifiable in the further collected data, but with a high correlation with respect to the marginal values "sex".

1.2 Notation

It is necessary to set some notations before we start with the whole thematic of imputation.

- In theory,

$$\mathbf{X} = (X_1,\ldots,X_d) \in \mathbb{R}^d$$

denotes the random events, with possibly missing values.

$$n_1 \in \mathbb{N}$$

denotes the number of full cases in the observations.

$$m_i \in \mathbb{N}$$

denotes the number of missing values in the row $i = 1 \dots, n$ of the observations.

$$\mathbf{F} = (\mathbf{F}_1, \dots, \mathbf{F}_d) \in \mathbb{R}^{(n_1) \times d}, \qquad \mathbf{F}_i \in \mathbb{R}^{n_1}, \qquad i = 1, \dots, d$$

denotes the complete case matrix.

- in the examples

$$\mathbf{Data} = \begin{pmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_n \end{pmatrix} \in \mathbb{R}^{n \times d}, \qquad \mathbf{A}_i \in \mathbb{R}^d, \qquad i = 1, \dots, n$$

denotes the data matrix with possibly missing values.

$$\mathbf{Data}^I = \begin{pmatrix} \mathbf{A}_1^I \\ \vdots \\ \mathbf{A}_n^I \end{pmatrix} \in \mathbb{R}^{n \times d}, \qquad \mathbf{A}_i^I \in \mathbb{R}^d, \qquad i = 1, \dots, n$$

denotes the completely imputed data matrix without missing values.

$$\mathbf{F} = \begin{pmatrix} \mathbf{A}_{i_1} \\ \vdots \\ \mathbf{A}_{i_{n_1}} \end{pmatrix} \in \mathbb{R}^{n_1 \times d}, \qquad \mathbf{A}_{i_k} \in \mathbb{R}^d, \qquad k = 1, \dots, n_1$$

denotes the matrix with all rows, which have non missing observations.

$$I_F = \{i_1, \dots, i_{n_1}\}$$

is the set of indices of rows with no missing observations.

$$I_{F^C} = \{1, \dots, n\} \backslash I_F = \{i_1, \dots, i_{n_1}\}$$

is the set of indices of rows with missing observations.

$$\mathbf{A}_i = (a_{i1}, \dots, a_{id}), \qquad , i = 1, \dots, n$$

are the observed values on the original scale.

$$\mathbf{A}_i = (u_{i1}, \dots, u_{id}), \qquad , i = 1, \dots, n$$

are the observed values on the $[0, 1]$-scale, or U-scale.

1.3 Definitions

Definition 1 (AIC & BIC)**.** *Generic function calculating the Akaike information criterion for one or several model objects for which a log-likelihood value can be obtained, according to the formula*

$$-(-2 \times \textit{log-likelihood} + k \times \textit{npar}),$$

where npar represents the number of parameters in the fitted model, and $k = 2$ *for the usual AIC, or* $k = \log(n)$ *(n the number of observations) for the so-called BIC or SBC (Schwarz's Bayesian criterion).*

Definition 2 ($N(\mu, \sigma^2)$)**.** $N(\mu, \sigma^2)$ *denotes the normal distribution with mean* $\mu \in \mathbb{R}$ *and variance* $\sigma^2 > 0$*. Its density is given by*

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}, \qquad x \in \mathbb{R}$$

Definition 3 ($\chi^2(\nu)$-distribution)**.** *The* $\chi^2(\nu)$*-distribution, with* $\nu \in \mathbb{N}$ *degrees of freedom denotes the distribution function with density*

$$f(x) = \frac{1}{2^{\nu/2}\Gamma(\nu/2)} x^{\nu/2-1} e^{-x/2} \mathbf{1}_{\{x>0\}}, \qquad x \in \mathbb{R},$$

where Γ *denotes the Gamma function.*

Definition 4 ((non-central) t-distribution)**.** $t(\nu, \mu)$ *denotes the (non-central) Student's t-distribution with parameters* $\nu \in \mathbb{N}$ *(degrees of freedom parameter) and* $\mu \in \mathbb{R}$ *(non-centrality parameter). This distribution is composed of* $X \sim N(0,1)$*, and* $V \sim \chi^2(\nu)$*-distribution (independent of* X*), via*

$$\frac{(X + \mu)}{\sqrt{V/\nu}}.$$

Chapter 2

Commonly used Imputation Methods

There are a lot of different imputation methods used in practice. The reader should be aware that every imputed value is wrong, though some are better in view of the data analyzing process. Always keep in mind that we prepare the data for a statistical analysis and not to have the exact values. So the aim is to not distort the unknown multivariate distribution of the random variables given by the whole data set for example by changing the expected value or increasing or decreasing the standard deviation. The following is a brief overview of some imputation methods commonly used in practice with their strengths and weaknesses.

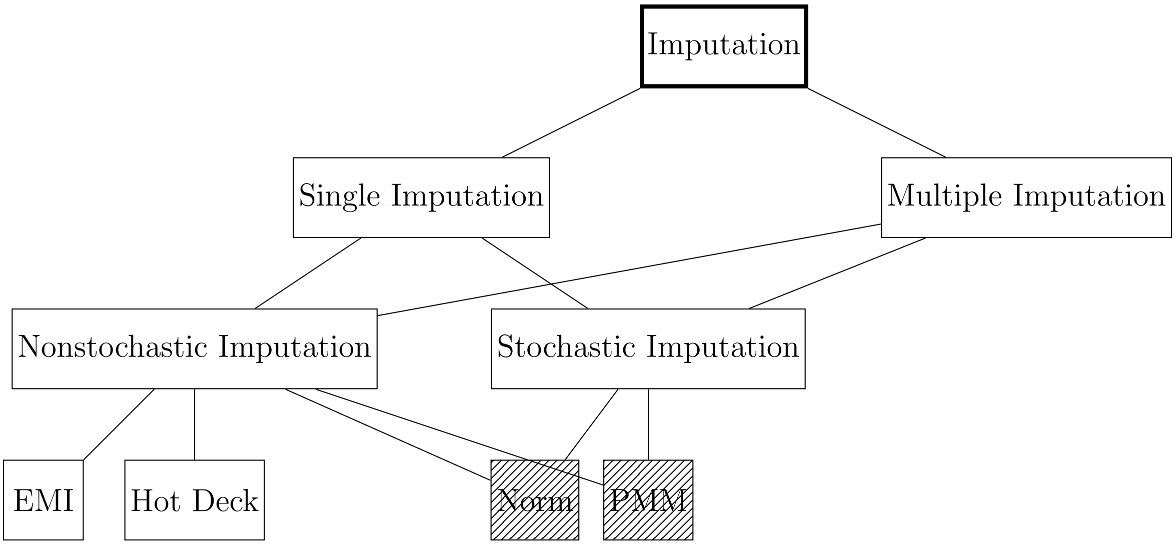


Figure 2.1: Different Imputation Methods, the dashed will be utilized later.

Since a later presented vine copula imputation method has some parallels to the Linear Regression with normally distributed error (Norm) approach, it is useful to compare it with the original idea. We will see that, in many cases, it is not sufficient to consider only linear relationships among the data. So it is necessary to enlarge the concept and allow for more types of dependencies, like asymmetric dependence structures, in the model.

The Predictive Mean Matching (PMM) is interesting, first, because it uses real values from the data as imputes, which is different to those vine copula imputation methods discussed later, which use either simulated values or expectations. A comparison with this procedure that combines parametric and nonparametric techniques is interesting, as it can show some strengths and weaknesses of the new imputation methods in some situations. Second, this approach performed well in further tests, and therefor is a good indicator for the success or failure of the newly developed methods.

2.1 Single Imputation

Single imputation means that, step by step, one fills in a precise (or random) value for each missing item to have a complete data set to analyze.

Example 1 (Single Imputation). *Say we have a* 4*-dimensional dataset matrix* $\mathbf{Data} \in \mathbb{R}^{5\times 4}$ *with* 5 *independent observations (*$\mathbf{A}_i \in \mathbb{R}^d$*,* $i = 1, \ldots, 5$*) of a* 4*-dimensional random vector* $\mathbf{X} = (X_1, \ldots, X_4)$*, and some missing data,*

$$\mathbf{Data} = \begin{pmatrix} a_{1,1} & - & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & - & - \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \\ a_{5,1} & a_{5,2} & a_{5,3} & a_{5,4} \end{pmatrix},$$

where the dashes belong to nonresponse. Using a single imputation method, one gets a single dataset $\mathbf{Data}^I$ *without missing values,*

$$\mathbf{Data}^I = \begin{pmatrix} a_{1,1} & a^*_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a^*_{3,3} & a^*_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \\ a_{5,1} & a_{5,2} & a_{5,3} & a_{5,4} \end{pmatrix},$$

where a^* *are the imputed values. Later on, we will differentiate between imputing only one nonresponse (***case 1***, like in the first row) and imputing two or more values (***case 2***, like in the third row).*

The main point is that imputation is applied only once. For methods with one precise value (nonstochastic single imputation), this means that no additional uncertainty is added to these imputed values. Any time the method is reiterated, the same results will occur. At first glance this is an attainable property, but it leads to a high underestimation of the variance in the analyzing procedure afterwards. If a random effect is added to the values (stochastic single imputation), an underestimation of the correlation between the multivariate data can occur. Nevertheless, it is often used in practice, because once a single imputation is done, the manipulated data can be handled as if it were complete, unfortunately with more or less incorrect results.

2.1.1 Nonstochastic Single Imputation

In a nonstochastic single imputation method, as mentioned above, the imputation values stay the same when repeating the procedure. If the method is applied m times on the same dataset **Data**, it follows that $a^{*1}_{i,j} = \ldots = a^{*m}_{i,j}$ for every missing value $a^{*}_{i,j}$, $i = 1, \ldots, n, j = 1, \ldots, d$.

Empirical Mean Imputation

One rather elementary approach is to simply use the empirical mean or median of the observed data for every margin to fill in the missing data. For the matrix $\mathbf{Data}^I$, this means that

Case 1. $a^*_{1,2} := \frac{1}{4}\sum_{i=2}^{5} a_{i,2}$,

Case 2a. $a^*_{3,3} := \frac{1}{4}\sum_{i=1,i\neq 3}^{5} a_{i,3}$ and

Case 2b. $a^*_{3,4} := \frac{1}{4}\sum_{i=1,i\neq 3}^{5} a_{i,4}$.

The advantages are obvious. Every time this method is used one gets the same values. Additionally, the marginal empirical expected value will not differ from the one computed before imputation. Further, one does not need to distinguish between one or more missing values in one row. But maybe this rather intuitive way is slightly too simple for our purpose. The most undesirable point, apart from those mentioned before, is that the dependence between the marginal distributions is highly underestimated.

Hot Deck Imputation

An approach that takes a closer look at the dependence structure between the marginals is the so called hot deck method, where, in its simplest form, one tries to find a "best matching partner" to the set of complete values in an incomplete array (for example see Little R. J. A., 1987, pp. 62-67). It is possible to use the current, but also data of a survey that was collected earlier. In a mathematical sense, one tries to find a complete vector in dimension n that minimizes the distance between the incomplete vector in a lower dimensional vector space with dimension n_1, where the number of missing values is $n - n_1$. If it is found, one completes the missing values in the incomplete vector with those of the complete. For the nonresponse in the matrix **Data**, this means: select the complete cases,

$$\mathbf{F} = \mathbf{Data}_{I_{FC}} = \begin{pmatrix} a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \\ a_{5,1} & a_{5,2} & a_{5,3} & a_{5,4} \end{pmatrix},$$

with $I_{FC} = \{2, 4, 5\}$. Choose a norm, for example the Euclidean norm.

Case 1. To get the value $a^*_{1,2}$, compute the distance for every vector in $\mathbf{F}$ to the vector $(a_{1,1}, a_{1,3}, a_{1,4})$ in the 3-*dimensional* submatrix $(\mathbf{F}_1, \mathbf{F}_3, \mathbf{F}_4)$, $d_i = \sum_{j=1,j\neq 2}^{4}(a_{i,j} - a_{1,j})^2$, for $i = 2, 4, 5$. Set $i = \arg\min_{i\in\{2,4,5\}}\{d_i\}$ and $a^*_{1,2} := a_{i,2}$.

Case 2a. For the value $a^*_{3,3}$, search in $\mathbf{F}$ for the closest vector in $(\mathbf{F}_1, \mathbf{F}_2)$ to $(a_{3,1}, a_{3,2})$, $d_i = \sum_{j=1}^{2}(a_{i,j} - a_{3,j})^2$, for $i = 2, 4, 5$. Set $i = \arg\min_{i\in\{2,4,5\}}\{d_i\}$ and $a^*_{3,3} := a_{i,3}$.

Case 2b. Last the value $a^*_{3,4}$. Search in $\mathbf{F}$ for the closest vector in $(\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3)$ to $(a_{3,1}, a_{3,2}, a^*_{3,3})$, $d_i = \left(\sum_{j=1}^{2}(a_{i,j} - a_{3,j})^2\right) + (a_{i,3} - a^*_{3,3})^2$, for $i = 2, 4, 5$. Set $i = \arg\min_{i\in\{2,4,5\}} \{d_i\}$ and $a^*_{3,4} := a_{i,4}$.

So if there is more than one missing value per row, the procedure has to be iterated until every nonresponse is filled with an imputation value. Again, this produces biased estimates of variances. In todays literature, different, more sophisticated types of hot deck approaches have been proposed which do not belong to the category of nonstochastic single imputation anymore. One example is a multiple imputation version in Rubin (1987).

2.1.2 Stochastic Single Imputation

A stochastic single imputation method tries to overcome the variance underestimation in the marginal by adding a stochastic error to the imputation value. One possible approach is fitting an appropriate stochastic model to the given data and simulating imputation values from the model. In the example (Example 1), fit a model $\mathbf{M}(\mathbf{F}, \theta)$ to the data in $\mathbf{F}$, where θ is the parameter vector of the model. Simulate from the values $x^*_{1,2} := X_2|X_1 = a_{1,1}, X_3 = a_{1,3}, X_4 = a_{1,4}, \theta = \hat{\theta}$ and $x^*_{3,3} := X_3|X_1 = a_{3,1}, X_2 = a_{3,2}, \theta = \hat{\theta}$ and $x^*_{3,4} := X_4|X_1 = a_{3,1}, X_2 = a_{3,2}, X_3 = a^*_{3,3}, \theta = \hat{\theta}$ with a simulation scheme for $\mathbf{M}$. Set $a^*_{1,2}$ equal to one simulation of $x^*_{1,2}$, $a^*_{3,3}$ equal to one simulation of $x^*_{3,3}$ and $a^*_{3,4}$ equal to one simulation of $x^*_{3,4}$.

Linear Regression

Like in many statistical areas, linear regression is also a possible solution for the missing data problem. Here, a linear regression model is fitted under the assumption of a continuous response variable with missing data (see Rubin D. B. 1987, pp. 166-167). The predictor variables are chosen from a set of linearly correlated effects. At this point, the information of missing data is used by simulating a new regression model based on the posterior parameters and their corresponding estimated distribution. Otherwise the variance of the model would be too small and would cause biased imputed values. The following steps explain the procedure more detailed:

Case 1. Only one column with missing values in **Data** $\in \mathbb{R}^{n\times d}$, with n independent observations $(\mathbf{A}_i \in \mathbb{R}^d, i = 1, \ldots, n)$ of a *d-dimensional* random vector $\mathbf{X} = (X_1, \ldots, X_d) \in \mathbb{R}^d$. W.l.o.g. only missing values in X_d.

1. Set X_d with $(n - n_1)$ missing values in **Data** as the response variable and the remaining $d - 1$ covariates $X_1, \ldots, X_{d-1}$ as predictor variables. Construct a linear model
$$X_d = \beta_0 + \beta_1 X_1 + \ldots, +\beta_{d-1}X_{d-1} + \epsilon,$$
only using complete cases for samples from X_d and $X_1, \ldots, X_{d-1}$. Let $\mathbf{F}_{-k}$ denote the matrix $\mathbf{F}$ with the k'th column removed. This yields to parameter estimates
$$\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \ldots, \hat{\beta}_{d-1})^T = ((\mathbf{1}, \mathbf{F}_{-d})^T(\mathbf{1}, \mathbf{F}_{-d}))^{-1}(\mathbf{1}, \mathbf{F}_{-d})^T\mathbf{F}_d$$

and

$$\hat{\sigma}_\epsilon^2 = \frac{(\mathbf{a}_d - \hat{\mathbf{a}}_d)^T(\mathbf{a}_d - \hat{\mathbf{a}}_d)}{n_1 - (d-1)}$$

only based on the complete case matrix $\mathbf{F}$, with $\hat{\mathbf{a}}_d = (\mathbf{1}, \mathbf{F}_{-d})\hat{\boldsymbol{\beta}}$, and $\mathbf{a}_d = \mathbf{F}_d$. Further, one gets the inverse of the $\mathbf{S}$-*matrix* with $\mathbf{S}^{-1} := ((\mathbf{1}, \mathbf{F}_{-d})^T(\mathbf{1}, \mathbf{F}_{-d}))^{-1}$.

2. As mentioned above, the variance has to be adjusted to overcome the estimation error through nonresponse. That is done via taking $\tilde{\sigma}_\epsilon^2 := \hat{\sigma}_\epsilon^2(n_1 - (d-1))/c$, where c is drawn from a $\chi^2_{n_1-(d-1)}$ random variable.

3. In a next step, the estimated parameters $\hat{\boldsymbol{\beta}}$ have to be adjusted, too, because the missing values also have random influence on it. Define $\tilde{\boldsymbol{\beta}} := \hat{\boldsymbol{\beta}} + \tilde{\sigma}_\epsilon(\mathbf{S}^{1/2})^T\mathbf{Z}$, where $\mathbf{S} := (\mathbf{S}^{1/2})^T\mathbf{S}^{1/2}$ and $\mathbf{Z}$ is a vector of d independent normally distributed random variables.

4. Every nonresponse $a_{i,d}$ is now imputed by the value $(\mathbf{1}, \mathbf{F}_{-d})_i\tilde{\boldsymbol{\beta}} + z_i\tilde{\sigma}_\epsilon$, with z_i being a $N(0,1)$ random variable.

For the value $a^*_{1,2}$ in the **Data** matrix in the previous example (Example 1), the procedure is the following:

Example 2 (**Case 1**). *Construct a linear model with data of all complete cases* $\mathbf{F}$*,*

$$a_{i,2} = \beta_0 + \beta_1 \times a_{i,1} + \beta_3 \times a_{i,3} + \beta_4 \times a_{i,4} + \epsilon_i, \qquad i = 2, 4, 5.$$

This yields to parameter estimates

$$\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_3, \hat{\beta}_4) \text{ and } \hat{\sigma}_\epsilon^2,$$

with

$$\mathbf{S}^{-1} :=((\mathbf{1}, \mathbf{F}_{-2})^T(\mathbf{1}, \mathbf{F}_{-2}))^{-1} = \left(\begin{pmatrix} 1 & a_{2,1} & a_{2,3} & a_{2,4} \\ 1 & a_{4,1} & a_{4,3} & a_{4,4} \\ 1 & a_{5,1} & a_{5,3} & a_{5,4} \end{pmatrix}^T \begin{pmatrix} 1 & a_{2,1} & a_{2,3} & a_{2,4} \\ 1 & a_{4,1} & a_{4,3} & a_{4,4} \\ 1 & a_{5,1} & a_{5,3} & a_{5,4} \end{pmatrix}\right)^{-1}.$$

Compute $\tilde{\sigma}_\epsilon^2$ *and* $\tilde{\boldsymbol{\beta}}$ *as mentioned and simulate a value* z *from a standard normal distribution. Set*

$$a^*_{1,2} := \tilde{\beta}_0 + \tilde{\beta}_1 \times a_{1,1} + \tilde{\beta}_3 \times a_{1,3} + \tilde{\beta}_4 \times a_{1,4} + z \times \tilde{\sigma}_\epsilon.$$

Case 2 More than one column with missing values in **Data** $\in \mathbb{R}^{n \times d}$, with n independent observations $(\mathbf{A}_i \in \mathbb{R}^d, i = 1, \ldots, n)$ of a *d-dimensional* random vector $\mathbf{X} = (X_1, \ldots, X_d) \in \mathbb{R}^d$.

1. For each column $j = 1, \ldots, d$, with nonresponse in **Data** do

2. For each row $i = 1, \ldots, n$, where X_j has a nonresponse in **Data**, set $X_{i_1}, \ldots, X_{i_{m_i}}$ with missing value in row i in **Data** as the response variables (where w.l.o.g. $i_1 := j$) and the $d - m_i$ covariates $X_{i_{m_i+1}}, \ldots, X_{i_d}$ without missing values in **Data** as predictor variables. Construct a linear model
$$X_{i_1} = \beta_0 + \beta_{i_{m_i+1}} X_{i_{m_i+1}} + \ldots + \beta_{i_d} X_{i_d} + \epsilon,$$
only using complete cases for samples from $X_1, \ldots, X_d$. This yields to parameter estimates
$$\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_{i_{m_i+1}}, \ldots, \hat{\beta}_{i_d})^T = ((\mathbf{1}, \mathbf{F}_{-D_i})^T (\mathbf{1}, \mathbf{F}_{-D_i}))^{-1} (\mathbf{1}, \mathbf{F}_{-D_i})^T \mathbf{F}_{i_1}$$
with $D_i := \{i_{m_i+1}, \ldots, i_d\}$ denotes the set of missing entries in row i in **Data**, and
$$\hat{\sigma}_\epsilon^2 = \frac{(\mathbf{a}_{i_1} - \hat{\mathbf{a}}_{i_1})^T (\mathbf{a}_{i_1} - \hat{\mathbf{a}}_{i_1})}{n_1 - |D_i|}$$
only based on the complete case matrix $\mathbf{F}$, with $\hat{\mathbf{a}}_{i_1} = (\mathbf{1}, \mathbf{F}_{-D_i}) \hat{\boldsymbol{\beta}}$, and $\mathbf{a}_{i_1} = \mathbf{F}_{i_1}$. Further one gets the inverse of the $\mathbf{S}$-*matrix* with $\mathbf{S}^{-1} := ((\mathbf{1}, \mathbf{F}_{-D_i})^T (\mathbf{1}, \mathbf{F}_{-D_i}))^{-1}$.

3. Again, the variance has to be adjusted to overcome the estimation error through nonresponse. That is done via taking $\tilde{\sigma}_\epsilon^2 := \hat{\sigma}_\epsilon^2 (n_1 - |D_i|)/c$, where c is drawn from a $\chi^2_{n_1 - (\mathbf{d-1})}$ random variable. Here the draw is really from a $\chi^2_{n_1-(d-1)}$ distribution, because instead of $|D_i|$, we consider $d-1$ predictor variables, but some are missing.

4. In a next step, the estimated parameters $\hat{\boldsymbol{\beta}}$ have to be adjusted, too, because the missing values also have random influence on it. Define $\tilde{\boldsymbol{\beta}} := \hat{\boldsymbol{\beta}} + \tilde{\sigma}_\epsilon (\mathbf{S}^{1/2})^T \mathbf{Z}$, where $\mathbf{S} := (\mathbf{S}^{1/2})^T \mathbf{S}^{1/2}$ and $\mathbf{Z}$ is a vector of d independent normally distributed random variables.

5. The nonresponse a_{i,i_1} is now imputed by the value $(\mathbf{1}, \mathbf{F}_{-D_i})_i \tilde{\boldsymbol{\beta}} + z_i \tilde{\sigma}_\epsilon$, with z_i being a $N(0,1)$ random variable.

6. Set $\mathbf{Data} = \mathbf{Data}^I$ with the imputed values, and reiterate with the next column, while the $\mathbf{F}$ matrix does not change.

For the values $a^*_{3,3}$ and $a^*_{3,4}$ in the **Data** matrix in the previous example (Example 1), the procedure is the following:

Example 3 (**Case 2**).

a. *Construct a linear model with data of all complete cases* $\mathbf{F}$*,*
$$a_{i,3} = \beta_0 + \beta_1 \times a_{i,1} + \beta_2 \times a_{i,2} + \epsilon_i, \qquad i = 2, 4, 5.$$
This yields to parameter estimates
$$\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2) \text{ and } \hat{\sigma}_\epsilon^2,$$

with

$$\mathbf{S}^{-1} :=((\mathbf{1},\mathbf{F}_{-\{3,4\}})^T(\mathbf{1},\mathbf{F}_{-\{3,4\}}))^{-1} = \left(\begin{pmatrix} 1 & a_{2,1} & a_{2,2} \\ 1 & a_{4,1} & a_{4,2} \\ 1 & a_{5,1} & a_{5,2} \end{pmatrix}^T \begin{pmatrix} 1 & a_{2,1} & a_{2,2} \\ 1 & a_{4,1} & a_{4,2} \\ 1 & a_{5,1} & a_{5,2} \end{pmatrix}\right)^{-1}.$$

Compute $\tilde{\sigma}_\epsilon^2$ and $\tilde{\boldsymbol{\beta}}$ as mentioned and simulate a value z_a from a standard normal distribution. Set

$$a^*_{3,3} := \tilde{\beta}_0 + \tilde{\beta}_1 \times a_{3,1} + \tilde{\beta}_2 \times a_{3,2} + z_a \times \tilde{\sigma}_\epsilon.$$

b. Construct a linear model with data of all complete cases $\mathbf{F}$*,*

$$a_{i,4} = \beta_0 + \beta_1 \times a_{i,1} + \beta_2 \times a_{i,2} + \beta_3 \times a_{i,3} + \epsilon_i, \qquad i = 2, 4, 5.$$

This yields to parameter estimates

$$\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3) \text{ and } \hat{\sigma}_\epsilon^2.$$

with

$$\mathbf{S}^{-1} :=((\mathbf{1},\mathbf{F}_{-4})^T(\mathbf{1},\mathbf{F}_{-4}))^{-1} = \left(\begin{pmatrix} 1 & a_{2,1} & a_{2,2} & x_{2,3} \\ 1 & a_{4,1} & a_{4,2} & x_{4,3} \\ 1 & a_{5,1} & a_{5,2} & x_{5,3} \end{pmatrix}^T \begin{pmatrix} 1 & a_{2,1} & a_{2,2} & a_{2,3} \\ 1 & a_{4,1} & a_{4,2} & a_{4,3} \\ 1 & a_{5,1} & a_{5,2} & a_{5,3} \end{pmatrix}\right)^{-1}.$$

Compute $\tilde{\sigma}_\epsilon^2$ and $\tilde{\boldsymbol{\beta}}$ as mentioned and simulate a value z_b from a standard normal distribution, independent of the value z_a. Set

$$a^*_{3,4} := \tilde{\beta}_0 + \tilde{\beta}_1 \times a_{3,1} + \tilde{\beta}_2 \times a_{3,2} + \tilde{\beta}_3 \times a^*_{3,3} + z_b \times \tilde{\sigma}_\epsilon.$$

Again, if there is more than one missing value in one row, the procedure will be applied iteratively. Note that, if one takes the value $\hat{\boldsymbol{\beta}}^T\mathbf{a}_i$ for imputation, linear regression belongs to the category of nonstochastic single imputation methods.

Predictive Mean Matching

The predictive mean matching (PMM) contains mainly the idea of the linear regression method, but with some slight distinction, leading to a significant difference in the impute values. The first three steps are the same as in the linear regression approach.

Case 1 Only one column with missing values in $\mathbf{Data} \in \mathbb{R}^{n\times d}$, with n independent observations ($\mathbf{A}_i \in \mathbb{R}^d, i = 1, \ldots, n$) of a *d-dimensional* random vector $\mathbf{X} = (X_1, \ldots, X_d) \in \mathbb{R}^d$. W.l.o.g. only missing values in X_d.

4. Every nonresponse $a_{i,d}$ in $\mathbf{Data}_{I_{FC}}$ is predicted by the value $(\mathbf{1}, \mathbf{F}_{-d})_i\tilde{\boldsymbol{\beta}} + z_i\tilde{\sigma}_\epsilon$.

5. For the missing $a_{i,d}$ find an observed candidate (a set of candidates) in $\mathbf{F}_d$ with closest predicted value (values), and take its observed value (draw in the set randomly an observed value) for imputation. So for all complete cases in $\mathbf{F}$, and the nonresponse case, set
$$\hat{a}_{r,d} = (\mathbf{1}, \mathbf{F}_{-d})_r \tilde{\boldsymbol{\beta}} + z_r \tilde{\sigma}_\epsilon, \qquad r \in \{r | \mathbf{A}_r \in \mathbf{F}\} \cup i$$
for z_r independent standard normal draws. Find d_i, with
$$d_i := \min_{r \in \{r | \mathbf{A}_r \in \mathbf{F}\}} \{|\hat{a}_{r,d} - \hat{a}_{i,d}|\}$$
and define i^*, such that
$$i^* := \arg \min_{r \in \{r | \mathbf{A}_r \in \mathbf{F}\}} \{|\hat{a}_{r,d} - \hat{a}_{i,d}|\}$$
and set $a^*_{i,d} := a_{i^*,d}$.

Example 4 (**Case 1**). *For the value $a^*_{1,2}$ in the example (Example 1), that means: Set*
$$\hat{a}_{r,2} = \tilde{\beta}_0 + \tilde{\beta}_1 \times a_{r,1} + \tilde{\beta}_3 \times a_{r,3} + \tilde{\beta}_4 \times a_{r,4} + z_r \times \tilde{\sigma}_\epsilon, \qquad r \in \{1, 2, 4, 5\},$$
for z_i independent standard normal draws. Find d_1, with
$$d_1 := \min_{r \in \{2,4,5\}} \{|\hat{a}_{r,1} - \hat{a}_{1,2}|\}$$
and define i^*, such that
$$i^* := \arg \min_{r \in \{2,4,5\}} \{|\hat{a}_{r,1} - \hat{a}_{1,2}|\}$$
and set $a^*_{1,2} := a_{i^*,2}$.

Case 2 More than one column with missing values in $\mathbf{Data} \in \mathbb{R}^{n \times d}$, with n independent observations ($\mathbf{A}_i \in \mathbb{R}^d, i = 1, \ldots, n$) of a *d-dimensional* random vector $\mathbf{X} = (X_1, \ldots, X_d) \in \mathbb{R}^d$.

5. Every nonresponse a_{i,i_1} in $\mathbf{Data}_{I_{FC}}$ is predicted by the value $(\mathbf{1}, \mathbf{F}_{-D_i})_i \tilde{\boldsymbol{\beta}} + z_i \tilde{\sigma}_\epsilon$.

6. For the missing a_{i,i_1} find an observed candidate (a set of candidates) in $\mathbf{F}_{i_1}$ with closest predicted value (values), and take its observed value (draw in the set randomly an observed value) for imputation. So for all complete cases in $\mathbf{F}$, and the nonresponse case, set
$$\hat{a}_{r,r_1} = (\mathbf{1}, \mathbf{F}_{-D_r})_r \tilde{\boldsymbol{\beta}} + z_r \tilde{\sigma}_\epsilon, \qquad r \in \{r | \mathbf{A}_r \in \mathbf{F}\} \cup i$$
for z_r independent standard normal draws. Find d_i, with
$$d_i := \min_{r \in \{r | \mathbf{A}_r \in \mathbf{F}\}} \{|\hat{a}_{r,r_1} - \hat{a}_{i,i_1}|\}$$
and define i^*, such that
$$i^* := \arg \min_{r \in \{r | \mathbf{A}_r \in \mathbf{F}\}} \{|\hat{a}_{r,r_1} - \hat{a}_{i,i_1}|\}$$
and set $a^*_{i,i_1} := a_{i^*,i_1}$.

Example 5 (**Case 2**).

a. *For the value* $a^*_{3,3}$. *Set*

$$\hat{a}_{r,3} = \tilde{\beta}_0 + \tilde{\beta}_1 \times a_{r,1} + \tilde{\beta}_2 \times a_{r,2} + z_r \times \tilde{\sigma}_\epsilon, \qquad r \in \{2,3,4,5\}.$$

for z_r *independent standard normal draws. Find* d_3*, with*

$$d_3 := \min_{r\in\{2,4,5\}} \{|\hat{a}_{r,3} - \hat{a}_{3,3}|\}$$

and define i^**, such that*

$$i^* := \arg \min_{r\in\{2,4,5\}} \{|\hat{a}_{r,3} - \hat{a}_{3,3}|\}$$

and set $a^*_{3,3} := a_{i^*,3}$.

b. *And for the value* $a^*_{3,4}$*, set*

$$\begin{aligned} \hat{a}_{r,4} &= \tilde{\beta}_0 + \tilde{\beta}_1 \times a_{r,1} + \tilde{\beta}_2 \times a_{r,2} + \tilde{\beta}_3 \times a_{r,3} + z_i \times \tilde{\sigma}_\epsilon, \qquad r \in \{2,4,5\}, \\ \hat{a}_{3,4} &= \tilde{\beta}_0 + \tilde{\beta}_1 \times a_{3,1} + \tilde{\beta}_2 \times a_{3,2} + \tilde{\beta}_3 \times a^*_{3,3} + z_3 \times \tilde{\sigma}_\epsilon. \end{aligned}$$

for z_r *independent standard normal draws,* $r \in \{2,3,4,5\}$. *Find* d_3*, with*

$$d_3 := \min_{r\in\{2,4,5\}} \{|\hat{a}_{r,4} - \hat{a}_{3,4}|\}$$

and define i^**, such that*

$$i^* := \arg \min_{r\in\{2,4,5\}} \{|\hat{a}_{r,4} - \hat{a}_{3,4}|\}$$

and set $a^*_{3,4} := a_{i^*,4}$.

This sounds similar to the hot deck approach, but the matching is done on the predicted values only. Note again the iteratively applied procedure if there is more than one value missing. Further, like in the linear regression approach, one can create a nonstochastic single imputation method if one uses $(\mathbf{1}, \mathbf{F}_{-D_i})_i\hat{\boldsymbol{\beta}}$ instead of $(\mathbf{1}, \mathbf{F}_{-D_i})_i\tilde{\boldsymbol{\beta}}+z_i\tilde{\sigma}_\epsilon$ as regression estimates.

2.2 Multiple Imputation

In the words of Rubin (Rubin D. B. 1987, p. 15): "Multiple imputation retains the virtues of single imputation and corrects its major flaws". What is the idea behind this? One imputes values with more than one single imputation method more than once (for example, one uses the three "best matching partners" in the hot deck approach, first with the data of the current survey and second with data collected earlier) to receive a distribution of probabilities. Then one treats each imputed data set as if it were one without missing values and analyzes it with respect to the important quantities like the expected value, variance or covariance. One takes the obtained parameters and analyzes them.

In a mathematical sense, choose L single imputation methods. Apply all L methods to the matrix with missing data $\mathbf{Data}$. This yields to L full matrices $\mathbf{Data}_l^I$, $l = 1, \ldots, L$ without nonresponse. Analyze every matrix $\mathbf{Data}_l^I$, $l = 1, \ldots, L$ for the parameter of interest θ and obtain L estimates of the parameter $\hat{\theta}_l$, $l = 1, \ldots, L$. Now there is data available to analyze the parameter of interest θ, for example for its expectation and its variance.

Nonstichastic single imputation methods can only be applied once to receive a full data matrix $\mathbf{Data}^I$, while stochastic single imputation methods are able to produce different imputation matrices when repeating them.

Chapter 3

Parametric Vine Copulae

The vine copula or pair-copula theory provides a very flexible and powerful approach to modeling multivariate data, i.e. constructing multivariate distribution functions, only using bivariate copulae and univariate distribution functions as building blocks. The bivariate copulae contain all information about the dependency structure between the data like the correlation matrix in the multivariate Gaussian distribution does, whereas the marginal distributions are modeled by one-dimensional distribution functions. As special cases, the vine copulae contain the multivariate Gaussian and the multivariate t distribution. But they still contain far more, like distributions with tail asymmetries, with combinations of different types of dependence and with different marginals. One major fault is that in almost all cases there is no closed form cdf (cumulative distribution function) available, but the good news is that there is always a closed form density using only parametric bivariate copulae. This is very helpful when applying maximum likelihood methods for parameter estimation.

3.1 Theory of Vines

In the following a theory is presented on how a general multivariate distribution function can be decomposed in terms of only bivariate copulae and one-dimensional distribution functions (Aas, Czado, Frigessi, Bakken (2009)).
Consider a *d-dimensional* random vector $\mathbf{X} = (X_1, \ldots, X_d)$ with given density function f, distribution function F and marginal distributions F_i, $i = 1, \ldots, d$. The density can be uniquely (up to variable permutation) factorized by

$$f(x_1, \ldots, x_d) = f(x_d) f(x_{d-1}|x_d) f(x_{d-2}|x_{d-1}, x_d) \cdots f(x_1|x_2, \ldots, x_d). \tag{3.1}$$

The idea is to combine this decomposition with the famous Sklar's theorem (Sklar, 1959)

$$F(x_1, \ldots, x_d) = C(F_1(x_1), F_2(x_2), \ldots, F_d(x_d)),$$

which, in words, states that every distribution function can be expressed in terms of its *d-dimensional* copula C and the (maybe different) marginal distribution functions F_i, $i = 1 \ldots, d$. Likewise, the density has the expression

$$f(x_1, \ldots, x_d) = c(F_1(x_1), \ldots, F_d(x_d)) f_1(x_1) \cdots f_d(x_d),$$

where all marginals have to be strictly increasing and continuous and c is the copula density. This is just d times an application of the chain rule for derivatives.
With the recursion formula

$$f(x|\mathbf{v}) = c_{XV_j|\mathbf{V}_{-j}}(F(x|\mathbf{v}_{-j}), F(v_j|\mathbf{v}_{-j}))f(x|\mathbf{v}_{-j}), \tag{3.2}$$

and the expression for the conditional distribution function shown by Joe (1996)

$$F(x|\mathbf{v}) = \frac{\partial C_{XV_j|\mathbf{V}_{-j}}(F(x|\mathbf{v}_{-j}), F(v_j|\mathbf{v}_{-j}))}{\partial F(v_j|\mathbf{v}_{-j})}, \tag{3.3}$$

where $C_{XV_j|\mathbf{V}_{-j}}$ and $c_{XV_j|\mathbf{V}_{-j}}$ are the copula and the copula density of (X, V_j) given $\mathbf{V}_{-j}$ respectively, and $\mathbf{v}_{-j}$ is the vector $\mathbf{v}$ with the jth element missing, it is now possible to derive a functional form of the multivariate density in terms of bivariate copulae and univariate distribution functions only.
For simplifying model selection procedures and calculations, we assume that the copula $C_{XV_j|\mathbf{V}_{-j}}$ does not depend on $\mathbf{V}_{-j}$. This is called the simplifying assumption. Notation

$$\begin{aligned} C_{XV_j;\mathbf{V}_{-j}}(x, v_j) =& C_{XV_j|\mathbf{V}_{-j}}(x, v_j), \text{ and} & (3.4)\\ C_{X|V_j;\mathbf{V}_{-j}}(x|v_j) =& \frac{\partial C_{XV_j|\mathbf{V}_{-j}}(x, v_j)}{\partial v_j}. & (3.5) \end{aligned}$$

It is said to be the copula associated with the bivariate distribution and its derivative respectively. For simplicity sometimes

$$\begin{aligned} C_{i_1 i_2; i_3 \ldots i_k}(u_{i_1}, u_{i_2}), \text{ and} & \quad (3.6)\\ C_{i_1|i_2; i_3 \ldots i_k}(u_{i_1}|u_{i_2}), & \quad (3.7) \end{aligned}$$

denotes the copula for the bivariate distribution function U_{i_1}, U_{i_2} given $U_{i_3}, \ldots, U_{i_k}$ and for its partial derivative, where $U_1, \ldots, U_k \sim U[0, 1]$. The same holds for the density function c.
For notation in later sections it is useful to introduce the definition of the function $h(x, v)$, to represent a conditional distribution function $F(x|v)$ where

$$h(x, v) = C_{X|V} = \frac{\partial C_{XV}(x, v)}{\partial v}, \tag{3.8}$$

and $x, v \in [0, 1]$.
It is easily seen that this factorization is not unique, because there are many possibilities to step through the recursion formula of the density (see Morales Napoles et al. (2010)). So Bedford and Cooke (2001b, 2002) introduced an organization scheme, helping to structure the precise copula. They invented a graphical way called *the regular vine* (or *R-vine*).
In this thesis the assumption of an underlying regular vine (R-vine) holds which is defined using graph theory (Kurowicka and Cooke (2006)).

Definition 5 (R-vine tree specification)**.** *Let Γ be a graph $(E(\Gamma), V(\Gamma))$ with $E(\Gamma)$ the set of edges and $V(\Gamma)$ the set of vertexes. Γ is a regular vine on d elements with $E(\Gamma) = E_1 \cup \ldots \cup E_{d-1}$ if*

1. *$\Gamma = \{T_1, \ldots, T_{d-1}\}$ is a forest consisting of $d - 1$ trees.*

2. *T_1 is a connected tree with nodes $N_1 = \{1, \dots, d\}$ and edges E_1. For $i = 2, \dots, d-1$, T_i is a tree with nodes $N_i = E_{i-1}$.*
3. *Proximity condition: For $i = 2, \dots, d-1$, for $\{A, B\} \in E_i$, $|(A\Delta B)| = 2$, with $A\Delta B := (A\backslash B) \cup (B\backslash A)$ denotes the symmetric difference.*

The following example should help to illustrate this idea.

Example 6. *Considering a special density factorization* (3.1) *of the random vector* $(X_1, \dots, X_5)$ *one gets*

$$f(x_1, \dots, x_5) = f(x_5)f(x_3|x_5)f(x_2|x_3, x_5)f(x_4|x_2, x_3, x_5)f(x_1|x_2, x_3, x_4, x_5).$$

With the recursion formula for the density (3.2) *one derives the expressions for the factors only using copula densities:*

$$\begin{aligned}
f(x_5) =& f(x_5)\\
f(x_3|x_5) =& c_{35}(F(x_3), F(x_5))f(x_3)\\
f(x_2|x_3, x_5) =& c_{25;3}(F(x_2|x_3), F(x_5|x_3))f(x_2|x_3)\\
=& c_{25;3}(F(x_2|x_3), F(x_5|x_3))c_{23}(F(x_2), F(x_3))f(x_2)\\
f(x_4|x_2, x_3, x_5) =& c_{45;23}(F(x_4|x_2, x_3), F(x_5|x_2, x_3))f(x_4|x_2, x_3)\\
=& c_{45;23}(F(x_4|x_2, x_3), F(x_5|x_2, x_3))c_{24;3}(F(x_2|x_3), F(x_4|x_3))f(x_4|x_3)\\
=& c_{45;23}(F(x_4|x_2, x_3), F(x_5|x_2, x_3))c_{24;3}(F(x_2|x_3), F(x_4|x_3))\\
& c_{34}(F(x_3), F(x_4))f(x_4)\\
f(x_1|x_2, x_3, x_4, x_5) =& c_{15;234}(F(x_1|x_2, x_3, x_4), F(x_5|x_2, x_3, x_4))f(x_1|x_2, x_3, x_4)\\
=& c_{15;234}(F(x_1|x_2, x_3, x_4), F(x_5|x_2, x_3, x_4))\\
& c_{14;23}(F(x_1|x_2, x_3), F(x_4|x_2, x_3))f(x_1|x_2, x_3)\\
=& c_{15;234}(F(x_1|x_2, x_3, x_4), F(x_5|x_2, x_3, x_4)\\
& c_{14;23}(F(x_1|x_2, x_3), F(x_4|x_2, x_3))c_{13;2}(F(x_1|x_2), F(x_3|x_2))f(x_1|x_2)\\
=& c_{15;234}(F(x_1|x_2, x_3, x_4), F(x_5|x_2, x_3, x_4)\\
& c_{14;23}(F(x_1|x_2, x_3), F(x_4|x_2, x_3))c_{13;2}(F(x_1|x_2), F(x_3|x_2))\\
& c_{12}(F(x_1), F(x_2))f(x_1).
\end{aligned}$$

Applying (3.3) *yields the final result. This special choice of factorization corresponds to the R-vine tree structure in Figure* 3.1.

There are two special cases of an R-vine structure, called the Canonical vine (C-vine) and the D-vine. First, the C-vine has the representation

$$f(x) = \prod_{k=1}^{d} f(x_k) \times \prod_{i=1}^{d-1}\prod_{j=1}^{d-i} c_{i,i+j;1:(i-1)}(F(x_i|x_1, \dots, x_{i-1}), F(x_{i+j}|x_1, \dots, x_{i-1})),$$

and second, the D-vine is defined as having the form

$$f(x) = \prod_{k=1}^{d} f(x_k) \times \prod_{i=1}^{d-1}\prod_{j=1}^{d-i} c_{j,j+i;(j+1):(j+i-1)}(F(x_j|x_{j+1}, \dots, x_{j+i-1}), F(x_{j+i}|x_{j+1}, \dots, x_{j+i-1})).$$

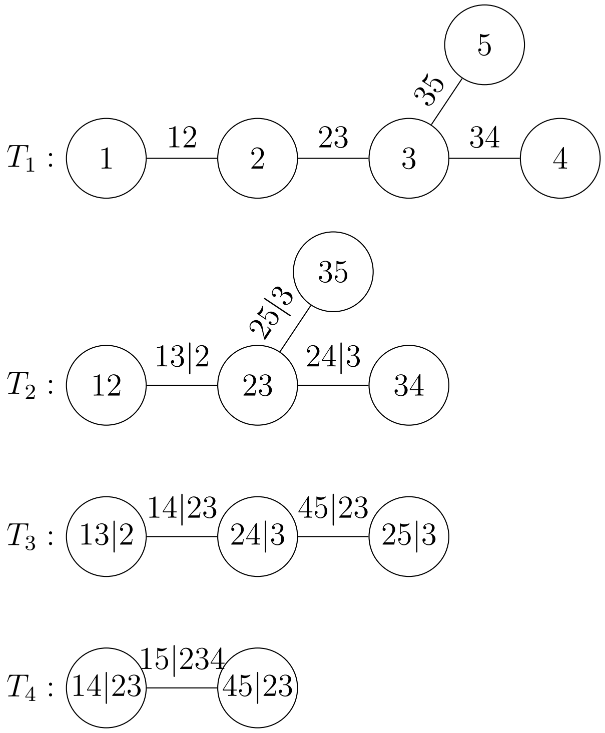


Figure 3.1: 5-*dimensional* R-vine structure.

D-vines have trees like path, while C-vines have stars as trees.

Conditional CDF's and inverses.

Later, for simulation, we need to compute conditional distribution functions and their inverses. One example for each, C-vine and D-vine, should be enough for demonstration.

Example 7 (Conditional Distribution Function & Inverse (C-vine))**.** *Let* $\mathbf{X} = (X_1, \dots, X_4)$ *be a* 4*-dimensional random vector, with an underlying C-vine structure. With the recursion formula (Equation 3.3) one can compute the conditional distribution function* $F_{4|123}(x_4|x_1, x_2, x_3)$*,*

$$\begin{aligned}
F_{4|123}(x_4|x_1,x_2,x_3) =& C_{4|3;12}(F(x_4|x_1,x_2)|F(x_3|x_1,x_2)) \\
F(x_4|x_1,x_2) =& C_{4|2;1}(F(x_4|x_1)|F(x_2|x_1)) \\
F(x_3|x_1,x_2) =& C_{3|2;1}(F(x_3|x_1)|F(x_2|x_1)) \\
F(x_4|x_1) =& C_{4|1}(F(x_4)|F(x_1)) \\
F(x_3|x_1) =& C_{3|1}(F(x_3)|F(x_1)) \\
F(x_2|x_1) =& C_{2|1}(F(x_2)|F(x_1)) \\
\Longrightarrow & \\
F_{4|123}(x_4|x_1,x_2,x_3) =& C_{4|3;12}(C_{4|2;1}(C_{4|1}(F(x_4)|F(x_1))|C_{2|1}(F(x_2)|F(x_1)))| \\
& C_{3|2;1}(C_{3|1}(F(x_3)|F(x_1))| \\
& C_{2|1}(F(x_2)| \\
& F(x_1))))
\end{aligned}$$

with inverse

$$\begin{aligned}
F^{-1}_{4|123}(y_4|x_1,x_2,x_3) =& F^{-1}_{4|12}(C^{-1}_{4|3;12}(y_4|F(x_3|x_1,x_2))|x_1,x_2)\\
F^{-1}_{4|12}(y_4|x_1,x_2) =& F^{-1}_{4|1}(C^{-1}_{4|2;1}(y_4|F(x_2|x_1))|x_1)\\
F^{-1}_{4|1}(y_4|x_1) =& F^{-1}_{4}(C^{-1}_{4|1}(y_4|F(x_1)))\\
\Longrightarrow&\\
F^{-1}_{4|123}(y_4|x_1,x_2,x_3) =& F^{-1}_{4}(C^{-1}_{4|1}(C^{-1}_{4|2;1}(C^{-1}_{4|3;12}(y_4|\\
& C_{3|2;1}(C_{3|1}(F(x_3)|F(x_1))|C_{2|1}(F(x_2)|F(x_1))))|\\
& C_{2|1}(F(x_2)|F(x_1)))|\\
& F(x_1))).
\end{aligned}$$

The same holds true for a permutation of $\pi(1,2,3,4) = (1,2,4,3)$.

Example 8 (Conditional Distribution Function & Inverse (D-vine))**.** *Let* $\mathbf{X} = (X_1,\ldots,X_4)$ *be a* 4*-dimensional random vector, with an underlying D-vine structure. With the recursion formula (Equation 3.3) one can compute the conditional distribution function* $F_{4|123}(x_4|x_1,x_2,x_3)$*,*

$$\begin{aligned}
F_{4|123}(x_4|x_1,x_2,x_3) =& C_{4|1;23}(F(x_4|x_2,x_3)|F(x_1|x_2,x_3))\\
F(x_4|x_2,x_3) =& C_{4|2;3}(F(x_4|x_3)|F(x_2|x_3))\\
F(x_1|x_2,x_3) =& C_{1|3;2}(F(x_1|x_2)|F(x_3|x_2))\\
F(x_4|x_3) =& C_{4|3}(F(x_4)|F(x_3))\\
F(x_2|x_3) =& C_{2|3}(F(x_2)|F(x_3))\\
F(x_1|x_2) =& C_{1|2}(F(x_1)|F(x_2))\\
F(x_3|x_2) =& C_{3|2}(F(x_3)|F(x_2))\\
\Longrightarrow&\\
F_{4|123}(x_4|x_1,x_2,x_3) =& C_{4|1;23}(C_{4|2;3}(C_{4|3}(F(x_4)|F(x_3))|C_{2|3}(F(x_2)|F(x_3)))|\\
& C_{1|3;2}(C_{1|2}(F(x_1)|F(x_2))|\\
& C_{3|2}(F(x_3)|\\
& F(x_2))))
\end{aligned}$$

with inverse

$$\begin{aligned}
F^{-1}_{4|123}(y_4|x_1,x_2,x_3) =& F^{-1}_{4|23}(C^{-1}_{4|1;23}(y_4|F(x_1|x_2,x_3))|x_2,x_3)\\
F^{-1}_{4|23}(y_4|x_2,x_3) =& F^{-1}_{4|3}(C^{-1}_{4|2;3}(y_4|F(x_2|x_3))|x_3)\\
F^{-1}_{4|3}(y_4|x_3) =& F^{-1}_{4}(C^{-1}_{4|3}(y_4|F(x_3)))\\
\Longrightarrow&\\
F^{-1}_{4|123}(y_4|x_1,x_2,x_3) =& F^{-1}_{4}(C^{-1}_{4|3}(C^{-1}_{4|2;3}(C^{-1}_{4|1;23}(y_4|\\
& C_{1|3;2}(C_{1|2}(F(x_1)|F(x_2))|C_{3|2}(F(x_3)|F(x_2))))|\\
& C_{1|2}(F(x_1)|F(x_2)))|\\
& F(x_3))).
\end{aligned}$$

The same holds true for a permutation of $\pi(1,2,3,4) = (4,3,2,1)$.

For statistical inference, one needs to store an R-vine copula consisting of all the corresponding pair-copula types and parameters properly. This was explained for example in (Dimann J., Brechmann E. C., Czado C., Kurowicka D., (2013)). They are simply using one lower triangular array for coding the tree structure, one for determining the type and one for fixing the parameter for all one-parametric bivariate copula families. Clearly for multi-parametric families one needs further arrays to store the additional information.

Definition 6 (Array Constraint Set)**.** *For an d-dimensional R-vine, let $M = (m_{ij})_{i,j=1,\dots,d}$ be a lower triangular array. The i-th constraint set for M is*

$$\mathcal{C}_M(i) = \{(\{m_{i,i}, m_{k,i}\}, D) | k = i+1, \dots, d, D = \{m_{k+1,i}, \dots m_{d,i}\}\}$$

for $i = 1, \dots, d-1$. If $k = d$ set $D = \emptyset$. The constraint set for array M is the union $\mathcal{C}_M = \mathcal{C}_M(1) \cup \dots \cup \mathcal{C}_M(d-1)$. For the elements of the constraint set $(\{m_{i,i}, m_{k,i}\}, D) \in \mathcal{C}_M$ call $\{m_{i,i}, m_{k,i}\}$ the conditioned set and D the conditioning set.

For demonstrating the use of this definition, just compare the following example of an lower triangular array $\hat{M}$ with Figure 3.1.

Example 9 (Lower Triangular Array)**.** *The constraint set always consists of one diagonal element and an element in the same column below this entry together with all entries following in that column.*

$$\hat{M} = \begin{pmatrix} \mathbf{1} & & & & \\ 5 & 5 & & & \\ \mathbf{4} & 4 & 4 & & \\ \mathbf{3} & 2 & 2 & 3 & \\ \mathbf{2} & 3 & 3 & 2 & 2 \end{pmatrix},$$

e.g (14|23) as can be found in T_4 in Figure 3.1. In the next step to defining an R-vine array, one needs some proximity sets to ensure the proximity condition required in definition 5.

Definition 7 (Proximity Sets)**.**

$$P_M(i) := \{(m_{i,i}, D) | k = i+1, \dots, d, D = \{m_{k,i}, \dots m_{d,i}\}\},$$
$$\tilde{P}_M(i) := \{(m_{k,i}, D) | k = i+1, \dots, d, D = \{m_{i,i}\} \cup \{m_{k+1,i}, \dots m_{d,i}\}\}.$$

So one can finally define an R-vine array.

Definition 8 (R-Vine Array)**.** *A lower triangular array $M = (m_{ij})_{i,j=1,\dots,d}$ is called an R-vine array if for $i = 1, \dots, d-1$ and for all $k = i+1, \dots, d-1$ there is a $j \in \{i+1, \dots, n-1\}$ with*

$$(m_{k,i}, \{m_{k+1,i}, \dots, m_{d,i}\}) \in P_M(j) \ \text{or} \ \in \tilde{P}_M(j).$$

3.2 Simulation and Estimation

Two of our proposed imputation methods work via the simulation of the nonresponse given the values that are not missing like in the linear regression case. Hence it is crucial to have a simulation as well as an estimation scheme for both a vine copula model and for the marginals available. In the following section the inversion method to stimulate univariate random variables and the algorithm of Dissmann will be presented (Dissmann J., Brechmann E. C., Czado C., Kurowicka D., (2013)). For R-vine simulation, that is the most general class of vine copulae. Afterwards, some estimation methods for the model will be explained. As there are only parametric copulae and margins considered, the methods will be reduced to parametric and semi-parametric estimators.

3.2.1 Marginal Simulation

The most common method to generate any univariate random variable is the Inversion Sampling Method. It works the following way:

Theorem 1 (Inversion Sampling)**.** *Let F be a continuous and increasing distribution function on $\mathbb{R}$ with generalized inverse F^{-1} defined by*

$$F^{-1} : (0,1) \to \mathbb{R}, \qquad F^{-1}(x) := \inf\{y \in \mathbb{R} : F(y) \geq x\}.$$

If $U \sim U[0,1]$ is a uniformly distributed random variable on the interval $[0,1]$, then $F^{-1}(U)$ has distribution function F. Also if X has distribution function F, then $F(X)$ is uniformly distributed on $[0,1]$.

Proof 1 (Inversion Theorem)**.** *The first statement follows after noting that $\forall x \in \mathbb{R}$,*

$$\begin{aligned}\mathbb{P}(F^{-1}(U) \leq x) &= \mathbb{P}(\inf\{y \in \mathbb{R} : F(y) \geq U\} \leq x)\\ &= \mathbb{P}(U \leq F(x))\\ &= F(x).\end{aligned}$$

Further note that $F \circ F^{-1}$ is the identity and F^{-1} is strictly increasing. Then the second statement follows from the fact that $\forall u \in (0,1)$,

$$\begin{aligned}\mathbb{P}(F(X) \leq u) &= \mathbb{P}(X \leq F^{-1}(u))\\ &= F \circ F^{-1}(u)\\ &= u.\end{aligned}$$

□

So the procedure works by generating a uniformly distributed random variable $U \sim U[0,1]$ and applying the generalized inverse (or quantile function) of the distribution function that one wants to simulate $F^{-1}(U)$.

3.2.2 R-vine Simulation

Generating a sample of an *d-dimensional* R-vine is based on the Rosenblatt transformation for continuous multivariate distributions that provides a procedure to sample from a *d-dimensional* multivariate distribution function.

Theorem 2 (Multivariate Inversion Sampling). *Let F be the d-dimensional distribution function of the random vector $(X_1, \dots X_d)$ with continuous univariate marginal distribution functions $F_1, \dots, F_d$. Further let*

$$F_{2|1}, F_{3|12}, \dots, F_{d|1,\dots,d-1}$$

be the corresponding conditional distribution functions with generalized inverses

$$F^{-1}_{2|1}, F^{-1}_{3|12}, \dots, F^{-1}_{d|1,\dots,d-1}.$$

To generate a random vector $(X_1, \dots X_d) \sim F$, one has to simulate $U_1, \dots, U_d$ iid uniform on $[0, 1]$ and set

$$\begin{aligned}
X_1 &= F_1^{-1}(U_1)\\
X_2 &= F^{-1}_{2|1}(U_2|X_1)\\
X_3 &= F^{-1}_{3|12}(U_3|X_1, X_2)\\
&\vdots\\
X_d &= F^{-1}_{d|1,\dots,d-1}(U_d|X_1, \dots, X_{d-1}).
\end{aligned}$$

Now the vector $(X_1, \dots X_d)$ has the distribution function F.

Clearly, a permutation of the numbers $1, 2, \dots, d$ is possible.

Proof 2 (Sampling Multivariate Distribution Functions). *Let the setting be as in the theorem, then the case $d = 1$ is proved above in the Inversion Theorem. For the case $d = k - 1 \to d = k$,*

$$\begin{aligned}
&\mathbb{P}(X_1 \le x_1, X_2 \le x_2, \dots, X_k \le x_k)\\
&= \mathbb{P}(X_1 \le x_1, X_2 \le x_2, \dots, X_{k-1} \le x_{k-1}, X_k = F^{-1}_{k|1,\dots,k-1}(U_k|X_1, \dots, X_{k-1}) \le x_k)\\
&= \mathbb{P}(X_1 \le x_1, X_2 \le x_2, \dots, X_{k-1} \le x_{k-1}, U_k \le F_{k|1,\dots,k-1}(x_k|X_1, \dots, X_{k-1}))\\
&= \int_{-\infty}^{x_1} \dots \int_{-\infty}^{x_{k-1}} \mathbb{P}(U_k \le F_{k|1,\dots,k-1}(x_k|y_1, \dots, y_{k-1})|X_1 = y_1, \dots, X_{k-1} = y_{k-1})\\
&\quad dF_{1\dots k-1}(y_1, \dots, y_{k-1})\\
&= \int_{-\infty}^{x_1} \dots \int_{-\infty}^{x_{k-1}} F_{k|1,\dots,k-1}(x_k|y_1, \dots, y_{k-1}) dF_{1\dots k-1}(y_1, \dots, y_{k-1})\\
&= F_{1\dots k}(x_1, \dots x_k).
\end{aligned}$$

□

With the expression for the conditional distribution function (3.3) and the definition of the *h-function* (3.8) it is now possible to determine $F(x_{i_k}|x_{i_1} \dots x_{i_{k-1}})$ for a "well chosen" permutation $\pi((1, 2, \dots, d)) = (i_1, \dots, i_d)$ for every $k = 2, \dots, d$. For example for the canonical vine choose, without loss of generality, π as the identity and

$$F(x_k|x_1 \dots x_{k-1}) = \frac{\partial_{k-1} C_{k,k-1|1,\dots,k-2}\left(F(x_k|x_1, \dots x_{k-2}), F(x_{k-1}|x_1, \dots x_{k-2})\right)}{\partial F(x_{k-1}|x_1, \dots x_{k-2})}.$$

For the D-vine again choose π as the identity and

$$F(x_k|x_1 \dots x_{k-1}) = \frac{\partial_1 C_{k,1|2,\dots,k-1}\left(F(x_k|x_2,\dots x_{k-1}), F(x_1|x_2,\dots x_{k-1})\right)}{\partial F(x_1|x_2,\dots x_{k-1})}.$$

For simulating from a regular vine, Dissmann et al. (2013) developed an algorithm that chooses first a "proper" permutation for the indexing of the random variables and then steps through the sampling scheme given above. This involves a clever storing order of the *h-functions* and their inverses in two $d \times d$ arrays, where only the lower diagonals are used. For convenience it is assumed that the diagonal entries of M are ordered from d to 1, i.e., $m_{k,k}$=$d-k+1$. If this is not the case, just rename the random vector and get an equivalent vine representation with an ordered M.

Example 10 (Convenience Ordering)**.** *For example for the vector* $(X_1, X_2, X_3, X_4, X_5)$ *and the corresponding* $\hat{M}$ *from example* 9 *one gets a renamed vector* $(\hat{X}_5, \hat{X}_1, \hat{X}_2, \hat{X}_3, \hat{X}_4)$*, with* $\hat{X}_5 = X_1, \dots, \hat{X}_4 = X_5$ *with the array representation*

$$\hat{M} = \begin{pmatrix} 5 & & & & \\ 4 & 4 & & & \\ 3 & 3 & 3 & & \\ 2 & 1 & 1 & 2 & \\ 1 & 2 & 2 & 1 & 1 \end{pmatrix}.$$

At the same time this convenience ordering turns out to be a "proper" index permutation for simulation. To proceed, one last matrix has to be introduced, called the maximum array. It ensures that the algorithm inserts the right arguments in the *h-functions* and their inverses and is defined as follows.

Definition 9 (Maximum Array)**.** *The maximum array* $\mathbb{M} = (\mathbf{m}_{i,k})_{i,k=1\dots,d}$ *of a R-vine array* $M = (m_{i,k})_{i,k=1\dots,d}$ *is defined by* $\mathbf{m}_{i,k} := \max\{m_{i,k}, \dots, m_{d,k}\}$ *for all* $k = 1\dots, d$ *and* $i = k, \dots, d$.

In words, the element $\mathbf{m_{i,k}}$ is the maximum over all entries in column k from i up to the d'th row.

Example 11. *The maximum array* $\hat{\mathbb{M}}$ *of the R-vine array* $\hat{M}$ *is then by definition*

$$\hat{\mathbb{M}} = \begin{pmatrix} 5 & & & & \\ 4 & 4 & & & \\ 3 & 3 & 3 & & \\ 2 & 2 & 2 & 2 & \\ 1 & 2 & 2 & 1 & 1 \end{pmatrix}.$$

Additionally, two more lower triangular arrays have to be specified to store information about types and parameters of the bivariate copulae. That is $T = (t_{i,j})_{i,j=1,\dots,d}$ for the type (e.g. Normal, t, Clayton, etc.) and $P = (p_{i,j})_{i,j=1,\dots,d}$ for the parameter (for copula families with more than one parameter, additional arrays are needed) of each bivariate copulae.

Taking into account the convenience ordering, one can finally use Dissman's algorithm to get a simulation from a *d-dimensional* random vector $X = (U_1, \dots, U_d)$ with R-vine array $M \in \mathbb{N}^{d \times d}$, maximum array $\mathbb{M} \in \mathbb{N}^{d \times d}$, pair copula family specification array $T \in \mathbb{N}^{d \times d}$ and with corresponding parameter array $P \in \mathbb{R}^{d \times d}$.

```
Data: R-vine specification in array form, i.e., $M$,$T$,$P$, where $m_{k,k} = d - k + 1$,
      $k = 1, \dots, d$.
Result: Random observations $(x_1, \dots, x_d) \in [0,1]^d$ from the $R$-vine specification.
Let $u_1, \dots, u_d$ be independent uniform samples.
Allocate $V^{direct} = (v_{i,k}^{direct} | i, k = 1, \dots, d)$.
Allocate $V^{indirect} = (v_{i,k}^{indirect} | i, k = 1, \dots, d)$.
Set $(v_{d,1}^{direct}, v_{d,2}^{direct}, \dots, v_{d,d}^{direct}) = (u_1, u_2, \dots, u_d)$.
Let $\mathbb{M} = (\mathbf{m}_{i,k} | i, k = 1, \dots, d)$ with $\mathbf{m}_{i,k} = \max\{m_{i,k}, \dots, m_{d,k}\}$ for all
$k = 1, \dots, d-1$ and $i = k, \dots, d$.
$x_1 = v_{d,d}^{direct}$
for $k = d-1, \dots, 1$ do
    for $i = k+1, \dots, d$ do
        if $\mathbf{m}_{i,k} = m_{i,k}$ then
            Set $z_{i,k}^{(2)} = v_{i,d-\mathbf{m}_{i,k}+1}^{direct}$.
        else
            Set $z_{i,k}^{(2)} = v_{i,d-\mathbf{m}_{i,k}+1}^{indirect}$.
        end
        Set $v_{d,k}^{direct} = h^{-1}(v_{d,k}^{direct}, z_{i,k}^{(2)} | t_{i,k}, p_{i,k})$
    end
    $x_{d-k+1} = v_{d,k}^{direct}$
    for $i = d, \dots, k+1$ do
        Set $z_{i,k}^{(1)} = v_{i,k}^{direct}$
        Set $v_{i-1,k}^{direct} = h(z_{i,k}^{(1)}, z_{i,k}^{(2)} | t_{i,k}, p_{i,k})$ and $v_{i-1,k}^{indirect} = h(z_{i,k}^{(2)}, z_{i,k}^{(1)} | t_{i,k}, p_{i,k})$.
    end
end
return $(x_1, \dots, x_d)$.
```

Algorithm 1: Simulation of an R-vine specification

3.2.3 R-Vine Simulation with Given Values

The fundamentals for this issue are already given from the preceding chapter. For simulating with given values one supposedly has to do less work. In practice, however, almost the same ideas are needed. Algorithm 1 generates a sample from a *d-dimensional* random vector $(U_1, \dots, U_d)$ with a given R-vine structure. Now the aim is to enlarge this concept and generate a sample from a random vector $(U_1, \dots, U_d)$ with a given R-vine structure and given values $U_1 = u_1, \dots, U_r = u_r$, for $r < d$. Now the following corollary of Theorem 2 solves the problem theoretically.

Corollary 1 (Multivariate Inversion Sampling with Given Values)**.** *Let the conditions of Theorem 2 hold and additionally let $X_1 = x_1, \dots, X_r = x_r$ being the already given values from the random vector $(X_1, \dots, X_r, X_{r+1}, \dots, X_d)$. To generate a sample of the random vector $(X_1, \dots X_d) \sim F$, one has to simulate $U_{r+1}, \dots, U_d$ iid uniform on* $[0, 1]$ *and set*

$$\begin{aligned}
&X_1 = x_1\\
&\vdots\\
&X_r = x_r\\
&X_{r+1} = F^{-1}_{r+1|1\dots r}(U_{r+1}|X_1, \dots, X_r)\\
&\vdots\\
&X_d = F^{-1}_{d|1,\dots,d-1}(U_d|X_1, \dots, X_{d-1}).
\end{aligned}$$

Now the vector $(X_1, \dots X_d)$ is a sample from a random vector $\mathbf{X}$*, which is distributed according to F.*

Note that one needs the given values in a particular order. The procedure can only be applied if the first r values are given. Otherwise one has to rename the variables.
Algorithm 1 already creates the needed inverses, but additionally samples $(X_1, \dots, X_r)$ under the corresponding R-vine model. So only the steps where these now given values are created have to be corrected to ensure the right result. Adjustments have do be done only in the first r iterations over the columns, where in each iteration over the row, the $v^{direct}_{d,k}$ is replaced by the nested inverse (line 14). Further the x_{d-k+1} must not be replaced (line 16), because it is already given. This can be better illustrated by an example:

Example 12. *Given the R-vine matrix like in Example 10*

$$\hat{M} = \begin{pmatrix} 5 & & & & \\ 4 & 4 & & & \\ 3 & 3 & 3 & & \\ 2 & 1 & 1 & 2 & \\ 1 & 2 & 2 & 1 & 1 \end{pmatrix},$$

with corresponding maximum array

$$\hat{\mathbb{M}} = \begin{pmatrix} 5 & & & & \\ 4 & 4 & & & \\ 3 & 3 & 3 & & \\ 2 & 2 & 2 & 2 & \\ 1 & 2 & 2 & 1 & 1 \end{pmatrix},$$

algorithm 1 creates the following 4 *matrices: Setting (including line 6)*

$$V^{direct} = \begin{pmatrix} \square & & & & \\ \square & \square & & & \\ \square & \square & \square & & \\ \square & \square & \square & \square & \\ u_1 & u_2 & u_3 & u_4 & x_1 \end{pmatrix}, \quad V^{indirect} = \begin{pmatrix} \square & & & & \\ \square & \square & & & \\ \square & \square & \square & & \\ \square & \square & \square & \square & \\ \square & \square & \square & \square & \square \end{pmatrix},$$

$$z^{(1)} = \begin{pmatrix} \square & & & & \\ \square & \square & & & \\ \square & \square & \square & & \\ \square & \square & \square & \square & \\ \square & \square & \square & \square & \square \end{pmatrix}, \quad z^{(2)} = \begin{pmatrix} \square & & & & \\ \square & \square & & & \\ \square & \square & \square & & \\ \square & \square & \square & \square & \\ \square & \square & \square & \square & \square \end{pmatrix},$$

step $k = 4$

$$V^{direct} = \begin{pmatrix} \square & & & & \\ \square & \square & & & \\ \square & \square & \square & & \\ \square & \square & \square & C_{2|1} & \\ u_1 & u_2 & u_3 & x_2 & x_1 \end{pmatrix}, \quad V^{indirect} = \begin{pmatrix} \square & & & & \\ \square & \square & & & \\ \square & \square & \square & & \\ \square & \square & \square & C_{1|2} & \\ \square & \square & \square & \square & \square \end{pmatrix},$$

$$z^{(1)} = \begin{pmatrix} \square & & & & \\ \square & \square & & & \\ \square & \square & \square & & \\ \square & \square & \square & \square & \\ \square & \square & \square & x_2 & \square \end{pmatrix}, \quad z^{(2)} = \begin{pmatrix} \square & & & & \\ \square & \square & & & \\ \square & \square & \square & & \\ \square & \square & \square & \square & \\ \square & \square & \square & x_1 & \square \end{pmatrix},$$

step $k = 3$

$$V^{direct} = \begin{pmatrix} \square & & & & \\ \square & \square & & & \\ \square & \square & C_{3|2;1} & & \\ \square & \square & C_{3|1} & C_{2|1} & \\ u_1 & u_2 & x_3 & x_2 & x_1 \end{pmatrix}, \quad V^{indirect} = \begin{pmatrix} \square & & & & \\ \square & \square & & & \\ \square & \square & C_{2|3;1} & & \\ \square & \square & C_{1|3} & C_{1|2} & \\ \square & \square & \square & \square & \square \end{pmatrix},$$

$$z^{(1)} = \begin{pmatrix} \square & & & & \\ \square & \square & & & \\ \square & \square & \square & & \\ \square & \square & C_{3|1} & \square & \\ \square & \square & x_3 & x_2 & \square \end{pmatrix}, \quad z^{(2)} = \begin{pmatrix} \square & & & & \\ \square & \square & & & \\ \square & \square & \square & & \\ \square & \square & C_{2|1} & \square & \\ \square & \square & x_2 & x_1 & \square \end{pmatrix},$$

step $k = 2$

$$V^{direct} = \begin{pmatrix} \square & & & & \\ \square & C_{4|3;12} & & & \\ \square & C_{4|1;2} & C_{3|2;1} & & \\ \square & C_{4|2} & C_{3|1} & C_{2|1} & \\ u_1 & x_4 & x_3 & x_2 & x_1 \end{pmatrix}, \quad V^{indirect} = \begin{pmatrix} \square & & & & \\ \square & C_{3|4;12} & & & \\ \square & C_{1|4;2} & C_{2|3;1} & & \\ \square & C_{2|4} & C_{1|3} & C_{1|2} & \\ \square & \square & \square & \square & \square \end{pmatrix},$$

$$z^{(1)} = \begin{pmatrix} \square & & & & \\ \square & \square & & & \\ \square & C_{4|1;2} & \square & & \\ \square & C_{4|2} & C_{3|1} & \square & \\ \square & x_4 & x_3 & x_2 & \square \end{pmatrix}, \quad z^{(2)} = \begin{pmatrix} \square & & & & \\ \square & \square & & & \\ \square & C_{3|1;2} & \square & & \\ \square & C_{1|2} & C_{2|1} & \square & \\ \square & x_2 & x_2 & x_1 & \square \end{pmatrix},$$

step $k = 1$

$$V^{direct} = \begin{pmatrix} C_{5|4;123} & & & & \\ C_{5|3;12} & C_{4|3;12} & & & \\ C_{5|2;1} & C_{4|1;2} & C_{3|2;1} & & \\ C_{5|1} & C_{4|2} & C_{3|1} & C_{2|1} & \\ x_5 & x_4 & x_3 & x_2 & x_1 \end{pmatrix}, \quad V^{indirect} = \begin{pmatrix} C_{4|5;123} & & & & \\ C_{3|5;12} & C_{3|4;12} & & & \\ C_{2|5;1} & C_{1|4;2} & C_{2|3;1} & & \\ C_{1|5} & C_{2|4} & C_{1|3} & C_{1|2} & \\ \square & \square & \square & \square & \square \end{pmatrix},$$

$$z^{(1)} = \begin{pmatrix} \square & & & & \\ C_{5|3;12} & \square & & & \\ C_{5|2;1} & C_{4|1;2} & \square & & \\ C_{5|1} & C_{4|2} & C_{3|1} & \square & \\ x_5 & x_4 & x_3 & x_2 & \square \end{pmatrix}, \quad z^{(2)} = \begin{pmatrix} \square & & & & \\ C_{4|3;12} & \square & & & \\ C_{3|2;1} & C_{3|2;1} & \square & & \\ C_{2|1} & C_{1|2} & C_{2|1} & \square & \\ x_1 & x_2 & x_2 & x_1 & \square \end{pmatrix}.$$

The main point is that if values $x_1, \ldots, x_r$, for $r < 5$ are given, one just has to leave out the x_k computations for $k = 1, \ldots, r$, because the nested conditioned copulae are depending on $x_1 \ldots, x_{k-1}, x_k$ only. For $k = r + 1, \ldots, 5$ continue as in Algorithm 1.

So the result is an algorithm consisting of two parts. One just filling in the needed so called ”backward substeps” ($k = 1, \ldots, r$), and one doing ”forward-” and ”backward substeps” and really simulating returned values ($k = r + 1, \ldots, d$).

```
Data: Given values $x_1, \dots, x_{d-r}$, an R-vine specification in array form, i.e.,
      $M$,$T$,$P$, where $m_{k,k} = d - k + 1$, $k = 1, \dots, d$.
Result: Random observations $(x_{d-r+1}, \dots, x_d)$ from the $R$-vine specification.
Let $u_1, \dots, u_r$ be independent uniform samples.
Allocate $V^{direct} = (v^{direct}_{i,k} | i, k = 1, \dots, d)$.
Allocate $V^{indirect} = (v^{indirect}_{i,k} | i, k = 1, \dots, d)$.
Set $(v^{direct}_{d,1}, v^{direct}_{d,2}, \dots, v^{direct}_{d,d}) = (u_1, u_2, \dots, u_r, x_{d-r}, x_{d-r-1} \dots, x_1)$.
Let $\mathbb{M} = (\mathbf{m}_{i,k} | i, k = 1, \dots, d)$ with $\mathbf{m}_{i,k} = \max\{m_{i,k}, \dots, m_{d,k}\}$ for all
$k = 1, \dots, d - 1$ and $i = k, \dots, d$.
for $k = d - 1, \dots, d - r$ do
    for $i = k + 1, \dots, d$ do
        if $\mathbf{m}_{i,k} = m_{i,k}$ then
            Set $z^{(2)}_{i,k} = v^{direct}_{i,d-\mathbf{m}_{i,k}+1}$.
        else
            Set $z^{(2)}_{i,k} = v^{indirect}_{i,d-\mathbf{m}_{i,k}+1}$.
        end
    end
    for $i = d, \dots, k + 1$ do
        Set $z^{(1)}_{i,k} = v^{direct}_{i,k}$
        Set $v^{direct}_{i-1,k} = h(z^{(1)}_{i,k}, z^{(2)}_{i,k} | t_{i,k}, p_{i,k})$ and $v^{indirect}_{i-1,k} = h(z^{(2)}_{i,k}, z^{(1)}_{i,k} | t_{i,k}, p_{i,k})$.
    end
end
for $k = d - r - 1, \dots, 1$ do
    for $i = k + 1, \dots, d$ do
        if $\mathbf{m}_{i,k} = m_{i,k}$ then
            Set $z^{(2)}_{i,k} = v^{direct}_{i,d-\mathbf{m}_{i,k}+1}$.
        else
            Set $z^{(2)}_{i,k} = v^{indirect}_{i,d-\mathbf{m}_{i,k}+1}$.
        end
        Set $v^{direct}_{d,k} = h^{-1}(v^{direct}_{d,k}, z^{(2)}_{i,k} | t_{i,k}, p_{i,k})$
    end
    $x_{d-k+1} = v^{direct}_{d,k}$
    for $i = d, \dots, k + 1$ do
        Set $z^{(1)}_{i,k} = v^{direct}_{i,k}$
        Set $v^{direct}_{i-1,k} = h(z^{(1)}_{i,k}, z^{(2)}_{i,k} | t_{i,k}, p_{i,k})$ and $v^{indirect}_{i-1,k} = h(z^{(2)}_{i,k}, z^{(1)}_{i,k} | t_{i,k}, p_{i,k})$.
    end
end
return $(x_1, \dots, x_d)$.
```

Algorithm 2: Simulation of an R-vine specification with given values

Chapter 4

Vine Copula Imputation

Now we should have sufficient theory to start with the actual project, the imputation method via vine copulae. With this concept, it is possible to exploit much more general dependence structures, like asymmetric dependencies between multivariate random variables. This is far more general than, for example, the linear regression procedure, only allowing linear dependencies. Further there is a simulation scheme available that allows for stochastic imputation. With the known density function, it is also possible to create a nonstochastic method with an underlying vine copula model.
Let us start with an example in dimension 4 to illustrate the common thread:

Example 13. *Let there be a* 4*-dimensional random variable* $\mathbf{U} = (U_1, \ldots, U_4)$ *with uniformly distributed margins (*$U_i \sim U[0,1]$*,* $i = 1, \ldots, 4$*). Further there have been data collected* n *times (*$\mathbf{u}_i = (u_{i1}, \ldots, u_{i4})$*,* $u_i \overset{iid}{\sim} \mathbf{U}$*,* $i = 1, \ldots, n$*),*

Case 1. *in the first scenario with nonresponse only in the first variable* U_1*,*
$\mathbf{u}_1 = (-, u_{12}, u_{13}, u_{14})$

Case 2. *in the second scenario only in the second margin* U_2*,*
$\mathbf{u}_2 = (u_{21}, -, u_{23}, u_{24})$

Case 3. *in the third case only in the third variable* U_3*,*
$\mathbf{u}_3 = (u_{31}, u_{32}, -, u_{34})$

Case 4. *and last, only in the fourth variable* U_4*.*
$\mathbf{u}_4 = (u_{41}, u_{42}, u_{43}, -)$

Assume that U *follows a* C*-vine structure with estimated pair copulae* C_{12}*,* C_{13}*,* C_{14}*,* $C_{23;1}$*,* $C_{24;1}$*,* $C_{34;12}$*.*
With the simulation scheme mentioned in Section 3.2, it is easy to derive an imputation value that takes into account all available information, for every nonresponse in the third or in the fourth scenario, with missing value either in the third or in the fourth variable. That is for every missing value in the fourth case, do:

Case 4*a. Simulate* $V_4 \overset{iid}{\sim} U[0,1]$*.*

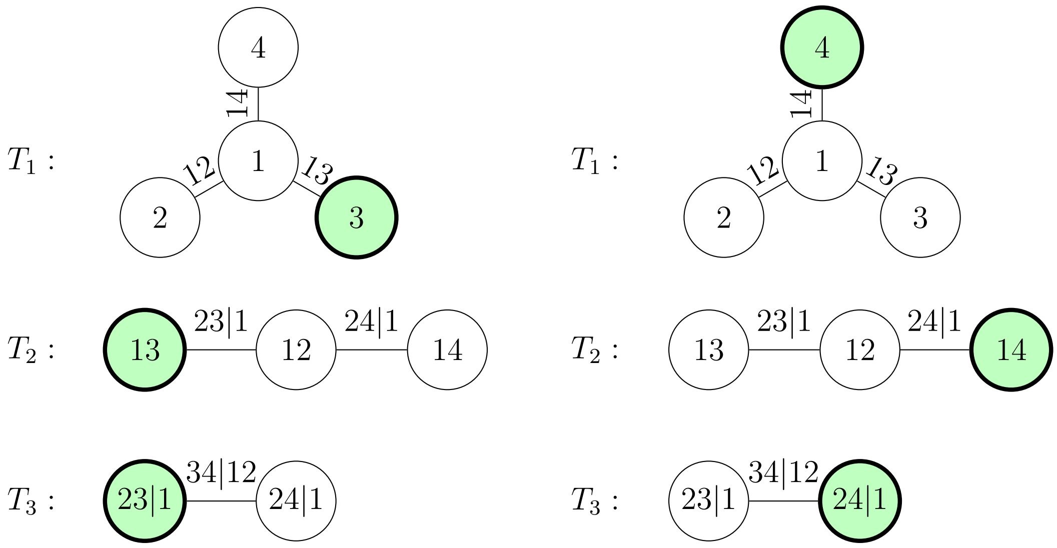


Figure 4.1: 4-*dimensional* C-vine structure with the vertexes labeled that contain the third or the fourth variable. These are the two examples where it is possible to impute with taking into account all available information (for **Case 3** and **Case 4**).

__Case 4__b. Compute the inverse of the conditional distribution function with Equation 3.3 (see Example 7), and set

$$\begin{aligned} u_4^* &= F^{-1}_{4|123}(V_4|u_1,u_2,u_3) \\ &= C^{-1}_{4|1}\left(C^{-1}_{4|2;1}\left(C^{-1}_{4|3;12}\left(V_4|C^{-1}_{3|2;1}\left(C^{-1}_{3|1}\left(u_3|u_1\right)|u_2\right)\right)|C^{-1}_{2|1}\left(u_1,u_2\right)\right)|u_1\right), \end{aligned}$$

and for every missing value in the third case, do:

__Case 3__a. Simulate $V_3 \overset{iid}{\sim} U[0,1]$.

__Case 3__b. Set

$$\begin{aligned} u_3^* &= F^{-1}_{3|124}(V_3|u_1,u_2,u_4) \\ &= C^{-1}_{3|1}\left(C^{-1}_{3|2;1}\left(C^{-1}_{3|4;12}\left(V_3|C^{-1}_{4|2;1}\left(C^{-1}_{4|1}\left(u_4|u_1\right)|u_2\right)\right)|C^{-1}_{2|1}\left(u_1,u_2\right)\right)|u_1\right), \end{aligned}$$

where $F^{-1}_{i_1|i_2i_3i_4}$ is the general inverse of the distribution function of U_{i_1} given U_{i_2}, U_{i_3} and U_{i_4}. Further $C^{-1}_{i_1|i_2;j_1\dots j_k} := (\partial_{i_2} C_{i_1,i_2;j_1\dots j_k})^{-1}$. This is straightforward, because every copula needed in the computation above is known from the estimation. Problems appear with the nonresponse in u_1 and u_2, so for scenario one and two. There are three possibilities for the imputation in u_1 considering most of the available information, but none of them is satisfying in the sense that we can use all of them:

__Case 1__a. Simulate $V_1 \sim U[0,1]$.

Case 1*b*. *Set*

$$u_1^* = F_{1|2}^{-1}(V_1|u_2) = C_{1|2}^{-1}(V_1|u_2),$$

or

$$u_1^* = F_{1|3}^{-1}(V_1|u_3) = C_{1|3}^{-1}(V_1|u_3),$$

or

$$u_1^* = F_{1|4}^{-1}(V_1|u_4) = C_{1|4}^{-1}(V_1|u_4),$$

notation as above. One can not use the other copulae in closed form, because they are conditioned on the first variable i.e. the first variable has to be known which is not the case. Almost the same problem appears for nonresponse in u_2 only. Here there are two possibilities:

Case 2*a*. *Simulate $V_2 \sim U[0,1]$.*

Case 2*b*. *Set*

$$u_2^* = F_{2|13}^{-1}(V_2|u_1,u_3) = C_{2|1}^{-1}\left(C_{2|3:1}^{-1}\left(V_2|C_{3|1}(u_3|u_1)\right)|u_1\right),$$

or

$$u_2^* = F_{2|14}^{-1}(V_2|u_1,u_4) = C_{2|1}^{-1}\left(C_{2|4:1}^{-1}\left(V_2|C_{4|1}(u_4|u_1)\right)|u_1\right),$$

again with the same notation.

Theorem 3 (Conditional CDF's for C-vines)**.**

- ***univariate*** *In general, it is possible for a C-vine to derive the conditional distribution function for the last two values only, i.e.*

$$\begin{aligned} F_{d|1:d-1} &= C_{d|1:d-1} = \partial_{d-1}C_{d,d-1|1:d-2}, \\ F_{d-1|1:(d-2),d} &= C_{d-1|1:(d-2),d} = \partial_d C_{d,d-1|1:d-2}, \end{aligned}$$

which are available in closed form in the C-vine structure. Therefor it is possible to simulate $U_i|\mathbf{U}_{-i} = \mathbf{u}_{-i}$ for $i \in \{d-1, d\}$, only.

- ***bivariate***

$$\begin{aligned} F_{d-1|1:d-2} &= C_{d-1|1:d-2} = \partial_{d-2}C_{d-1,d-2|1:d-3}, \\ F_{d-2|1:(d-3),d-1} &= C_{d-2|1:(d-3),d-1} = \partial_{d-1}C_{d-1,d-2|1:d-3}, \\ F_{d|1:d-2} &= C_{d|1:d-2} = \partial_{d-2}C_{d,d-2|1:d-3}, \\ F_{d-2|1:d-3,d} &= C_{d-2|1:d-3,d} = \partial_d C_{d,d-2|1:d-3} \end{aligned}$$

are again available in closed form for $d-2$ conditioning variables only. Combining this with the univariate case, it is possible to simulate $(U_i, U_j)|\mathbf{U}_{-\{i,j\}}$ only for $i, j \in \{d-2, d-1, d\}$, $i \neq j$.

- ***general*** *For d-k conditioning variables there are the conditioned cdf's available*

$$\begin{aligned} F_{i|1:(i-1),i+j} &= C_{i|1:(i-1),i+j} = \partial_{i+j} C_{i,i+j|1:i-1}, \\ F_{i|1:(i-1),i+j} &= C_{i+j|1:i} = \partial_{i} C_{i,i+j|1:i-1}, \qquad i = d-k, \qquad j = 1,\dots,d-i \end{aligned}$$

So possible simulations are $(U_{i_1},\dots,U_{i_k}|U_{-\{i_1,\dots,i_k\}})$ only for $i_1,\dots,i_k \in \{d-k,\dots,d\}$, for i_l $l = 1\dots,k$ pairwise unequal.

Theorem 4 (Conditional CDF's for D-vines)**.**

- ***univariate*** *In general, it is possible for a D-vine to derive the conditional distribution function for the first and the last value only, i.e.*

$$\begin{aligned} F_{d|1:d-1} &= C_{d|1:d-1} = \partial_1 C_{1,d|2:d-1}, \\ F_{1|2:d} &= C_{1|2:d} = \partial_d C_{1,d|2:d-1}, \end{aligned}$$

which are available in closed form in the D-vine structure. Therefor it is possible to simulate $U_i|\mathbf{U}_{-i} = \mathbf{u}_{-i}$ for $i \in \{1, d\}$, only.

- ***bivariate***

$$\begin{aligned} F_{d|2:d-1} &= C_{d|2:d-1} = \partial_2 C_{2,d|3:d-1}, \\ F_{2|3:d} &= C_{2|3:d} = \partial_d C_{2,d|3:d-1}, \\ F_{d-1|1:d-2} &= C_{d-1|1:d-2} = \partial_1 C_{1,d-1|2:d-2}, \\ F_{1|2:d-1} &= C_{1|2:d-1} = \partial_{d-1} C_{1,d-1|2:d-2} \end{aligned}$$

are again available in closed form for $d-2$ conditioning variables only. Combining this with the univariate case, it is possible to simulate $(U_i, U_j)|\mathbf{U}_{-\{i,j\}}$ only for $\{i,j\} \in \{\{1,2\},\{1,d\},\{d-1,d\}\}$.

- ***general*** *For d-k conditioning variables, the conditioned cdf's are available*

$$\begin{aligned} F_{j+i|j:(j+i-1)} &= C_{j+i|j:(j+i-1)} = \partial_j C_{j,j+i|(j+1):(j+i-1)}, \\ F_{j|j+1:(j+i)} &= C_{j|j+1:(j+i)} = \partial_{j+i} C_{j,j+i|(j+1):(j+i-1)}, \qquad i = d-k, \qquad j = 1,\dots,d-i \end{aligned}$$

So possible simulations for $k > 1$ are $(U_{i_1},\dots,U_{i_k}|U_{-\{i_1,\dots,i_k\}})$ only for $\{i_1,\dots,i_k\} \in \{\{1,d-k+2,\dots,d\},\{1,\dots,k_1,d-k_2+1,\dots,d|k_1+k_2=k\},\{1,\dots,k-1,d\}\}$.

Facing this issue, one could think about the following two approaches that are discussed in detail next. They only use closed form density functions and impute values with simulation of the missing data given the observed. First, for each missing value combination we try to find the "best" fitting model with the constraint that all nonresponse data have to be imputed such that every available information is included, called **Vine Copula Regression Imputation (CopReg)**. Second, we follow the procedure demonstrated in the 4-*dimensional* example in the way that we try to find the "best" fitting vine copula model and impute considering "the most possible" information available, called **Vine Copula Fitting Imputation (CopFit)**.

Taking another look at the problem in Example (13) of not being able to simulate the random variable U_1 given U_2, U_3, U_4, one could think about integrating out the given density function to get the conditional expected value $\mathbb{E}(U_1|U_2 = u_2, U_3 = u_3, U_4 = u_4)$ instead of using closed form cdf's. This method is straightforward for every vine structure as long as the continuity assumption holds true. Later on, this third approach is called **Vine Copula Expectation Imputation (CopExp)**.

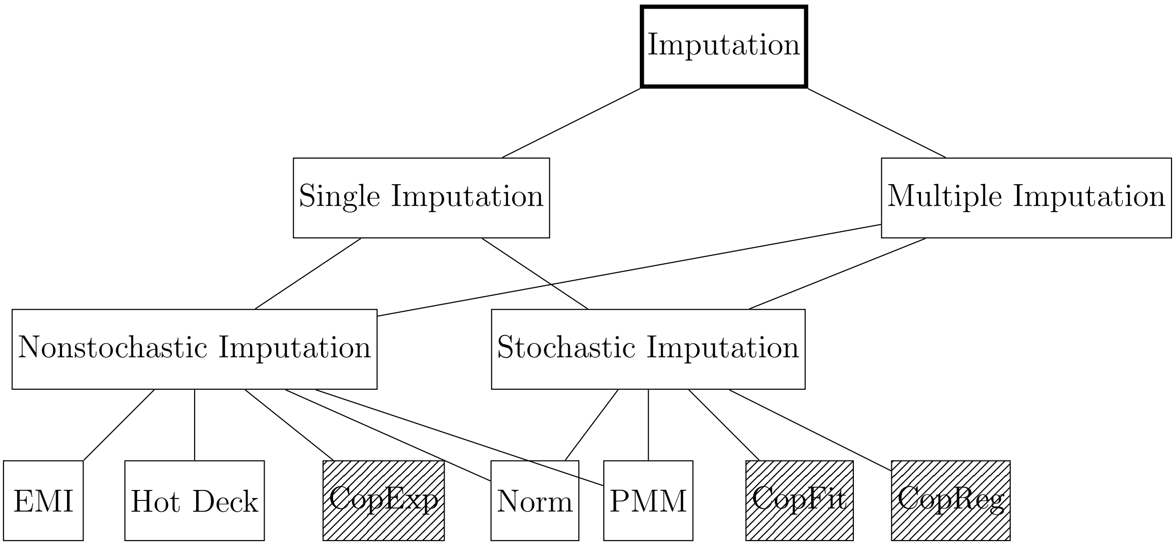


Figure 4.2: Overview of common imputation methods with the newly proposed copula based imputation methods (dashed).

4.1 Vine Copula Regression Imputation (CopReg)

This method picks up the idea of a classical regression model as the nomenclature suggests. Likewise, one determines response variables where information is needed (missing variable), and predictor variables that contain this information (given values). After fixing the affiliation of each, the "best" fitting vine copula model among the "appropriate" ones has to be found to impute the missing data with a simulation scheme.

Case 1 (Only one missing per row): So first, consider a matrix **Data** $\in \mathbb{R}^{n\times d}$ with n independent observations ($\mathbf{u}_i \in \mathbb{R}^d$, $i = 1, \ldots, n$) of a *d-dimensional* random vector $\mathbf{U} = (U_1, \ldots, U_d) \in \mathbb{R}^d$, $U_j \sim U[0,1]$, $j = 1, \ldots, d$ with, w.l.o.g. nonresponse in the first r rows of the first observation vector , i.e.

$$\mathbf{Data} = \begin{pmatrix} - & u_{12} & \ldots & u_{1d} \\ \vdots & \vdots & & \vdots \\ - & u_{r2} & \ldots & u_{rd} \\ u_{r+1,1} & u_{r+1,2} & \ldots & u_{r+1,d} \\ \vdots & \vdots & & \vdots \\ u_{n1} & u_{n2} & \ldots & u_{nd} \end{pmatrix}.$$

Now the procedure is the following:

Step 1. Determine the complete case matrix

$$\mathbf{F} = \begin{pmatrix} u_{r+1,1} & \ldots & u_{r+1,d} \\ \vdots & & \vdots \\ u_{n1} & \ldots & u_{nd} \end{pmatrix}.$$

Step 2. Determine the response variables ($= \{U_1\}$) and the predictor variables ($= \{U_2, \ldots, U_d\}$).

Step 3. Fit an R-vine model to **F** under the constraint, that $U_1|U_2, \ldots, U_d$ can be simulated. This means that for only one missing value per row (see Example 13), the nonresponse variable has to be a leaf in any tree in the structure of the vine trees (see Figure 4.1).

Step 4. Simulate $u_{11}^* \sim U_1|U_2 = u_{12}, \ldots, U_d = u_{1d}, \ldots, u_{r1}^* \sim U_1|U_2 = u_{r2}, \ldots, U_d = u_{rd}$ from the R-vine model.

Case 2 (More than one missing per row): Second, consider the same matrix **Data**, now w.l.o.g. with the first $m-1$ values missing in the first row, i.e.

$$\mathbf{Data} = \begin{pmatrix} - & \ldots & - & u_{1m} & \ldots & u_{1d} \\ u_{21} & \ldots & \ldots & \ldots & \ldots & u_{2d} \\ \vdots & & & & & \vdots \\ u_{n1} & \ldots & \ldots & \ldots & \ldots & u_{nd} \end{pmatrix}.$$

Now the procedure is the following:

Step 1. Determine the complete case matrix

$$F = \begin{pmatrix} u_{21} & \dots & u_{2d} \\ \vdots & & \vdots \\ u_{n1} & \dots & u_{nd} \end{pmatrix}.$$

Step 2. Determine the response variables ($= \{U_1, \dots, U_{m-1}\}$) and the predictor variables ($= \{U_m, \dots, U_d\}$).

Step 3.1. Fit an R-vine model to F under the constraint that $U_1|U_m, \dots, U_d$ can be simulated. This means that the R-vine model for the vector $(U_1, U_m \dots . U_d)$ has the constraint that the nonresponse variable U_1 has to be a leaf in any tree in the structure of the vine trees (see Figure 4.1).

Step 4.1. Simulate $u_{11}^* \sim U_1|U_m = u_{1m}, \dots, U_d = u_{1d}$ from the R-vine model.

Step 3.2. Fit an R-vine model to F under the constraint that $U_1|U_m, \dots, U_d$ and $U_2|U_1, U_m, \dots, U_d$ can be simulated. This is an extension of the R-vine model in Step 3.1, by adding the variable U_2.

Step 4.2. Simulate $u_{12}^* \sim U_2|U_1 = u_{11}^*, U_m = u_{1m}, \dots, U_d = u_{1d}$ from the R-vine model.

Step 3.m. Fit an R-vine model to F under the constraint, that $U_1|U_m, \dots, U_d$ and $U_i|U_1, \dots, U_{i-1}, U_m, \dots, U_d$, $i = 2, \dots, m-1$ can be simulated. This is an extension of the R-vine model in 3.$m-1$, by adding the variable U_{m-1}. So the R-vine model has to fulfill the condition that first the nonresponse variable U_{m-1} has to be a leaf in any tree in the structure of the vine trees. If we look at the structure of the vine trees without U_{m-1}, the variable U_{m-2} has to be such a leaf. This has to be possible up to variable U_1.

Step 4.m. Simulate $u_{1,m-1}^* \sim U_{m-1}|U_1 = u_{11}^*, \dots, U_{m-2} = u_{1,m-2}^*, U_m = u_{1m}, \dots, U_d = u_{1d}$ from the R-vine model.

If there is nonresponse in more than one row, we use the method on each row $\mathbf{A}_i \in \mathbf{Data}_{I_{FC}}$, $\forall i \in I_{FC}$, separately, while the $\mathbf{F}$ matrix does not change. Note that with changing the imputation order, the R-vine fit also changes.
As mentioned before, the "best" fitting model here is the one that Dissmann's tree by tree estimation suggests (of course one can use her or his preferred estimation scheme). Now the question of what "appropriate" vine copula models are so that every given information can be used to predict the nonresponse is almost answered. Clearly it does not matter what kind of bivariate copula families are chosen and which parameters they are equipped with. But looking back at the example from the beginning of Chapter 4, it clearly shows that the tree structure of the vine has to be restricted to a subclass of R-vine copulae. The restrictions are:

1. For only one missing value per row, as said before, the nonresponse variable has to be a leaf in any tree in the structure of the vine trees (see Figure 4.1).

2. For two or more missing values per row, at least one nonresponse variable in this row has to be a leaf in any tree in the structure of the vine trees. If we look at the structure of the vine trees without this leaf, again at least one nonresponse variable in this row has to be a leaf in any tree in the new structure of the vine trees etc. In matrix notation: if there are m out of n given variables in one row, it has to be possible to find a nonresponse variable with the property that it does not take place in the lower triangular R-vine array $(\mathbf{M})_{ij=1,\dots n}$ at places m_{ij} for all $i > j+1$. Then if this leaf nonresponse variable is deleted (in the sense that the whole column with its number on top and additionally every place the number occurs in is deleted), again it has to be possible to find such a nonresponse variable in this row etc. This has to be possible for $n-m-1$ deletions.

The following example will help to make this more intuitive:

Example 14 (Vine Copula Imputation for D-vines)**.** *Let* $\mathbf{U} = (U_1, \dots, U_5)$ *be a 5-dimesional random variable with uniformly distributed margins* $\{U_i \sim U[0,1],\ i = 1, \dots, 5\}$*. Again there have been data collected n times* $\{\mathbf{u}_i = (u_{i1}, \dots, u_{i5}),\ \mathbf{u}_i \overset{iid}{\sim} \mathbf{U},\ i = 1, \dots, n\}$ *now with nonresponse in two of the five variables. Assume that* $\mathbf{U}$ *follows a D-vine structure with estimated pair copulae* C_{12}, C_{23}, C_{34}, C_{45}, $C_{13;2}$, $C_{24;3}$, $C_{35;4}$, $C_{14;23}$, $C_{25;34}$, $C_{15;234}$.

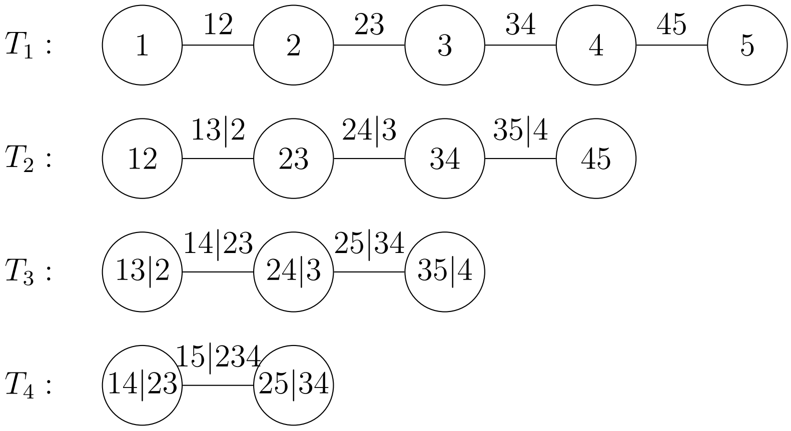


Figure 4.3: 5-*dimensional* D-vine structure.

Now we assume that there are 2 *missing values in one row of a vector* $\mathbf{u}_j$*. This could* *happen for* 10 *different cases:*

$$\begin{aligned}
\mathbf{u}_j &= (-,-,u_3,u_4,u_5),\\
\mathbf{u}_j &= (-,u_2,-,u_4,u_5),\\
\mathbf{u}_j &= (-,u_2,u_3,-,u_5),\\
\mathbf{u}_j &= (-,u_2,u_3,u_4,-),\\
\mathbf{u}_j &= (u_1,-,-,u_4,u_5),\\
\mathbf{u}_j &= (u_1,-,u_2,-,u_5),\\
\mathbf{u}_j &= (u_1,-,u_3,u_4,-),\\
\mathbf{u}_j &= (u_1,u_2,-,-,u_5),\\
\mathbf{u}_j &= (u_1,u_2,-,u_4,-),\\
\mathbf{u}_j &= (u_1,u_2,u_3,-,-).
\end{aligned}$$

Everything is fine as long as only the values of the pairs

Case 1. (U_1, U_2),

Case 2. (U_1, U_5), *or*

Case 3. (U_4, U_5)

are missing, since

Case 1. *First do the imputation procedure in* U_2 *then in* U_1*, i.e. simulate* V_1, V_2 *iid from a uniformly distributed random variable and impute*

$$\begin{aligned}
u_2^* =&C_{2|3}^{-1}(C_{2|4;3}^{-1}(C_{2|5;34}^{-1}(V_2|C_{5|3;4}(C_{5|4}(u_5|u_4)|C_{3|4}(u_3|u_4)))|C_{3|4}(u_3|u_4))|u_3),\\
u_1^* =&C_{1|2}^{-1}(C_{1|3;2}^{-1}(C_{1|4;23}^{-1}(C_{1|5;234}^{-1}(V_1|\\
&\quad C_{5|4;23}(C_{5|4;3}(C_{5|4}(u_5|u_4)|C_{4|3}(u_4|u_3))|C_{2|4;3}(C_{2|3}(u_2^*|u_3)|C_{3|4}(u_3|u_4))))|\\
&\quad C_{4|2;3}(C_{4|3}(u_4|u_3)|C_{2|3}(u_2^*|u_3)))|\\
&\quad C_{2|3}(u_2^*|u_3))|\\
&\quad u_2^*).
\end{aligned}$$

The inverses are again computed with the recursion formula of equation 3.3. So for the first imputation value u_2^**:*

$$\begin{aligned}
C(u_2|u_3,u_4,u_5) =&C_{2|5;34}(C(u_2|u_3,u_4)|C(u_5|u_3,u_4))\\
C(u_2|u_3,u_4) =&C_{2|4;3}(C(u_2|u_3)|C(u_3|u_4))\\
C(u_5|u_3,u_4) =&C_{5|3;4}(C(u_5|u_4)|C(u_3|u_4))\\
C(u_2|u_3) =&C_{2|3}(u_2|u_3)\\
C(u_3|u_4) =&C_{3|4}(u_3|u_4)\\
C(u_5|u_4) =&C_{5|4}(u_5|u_4)\\
&\Longrightarrow\\
C(u_2|u_3,u_4,u_5) =&C_{2|5;34}(C_{2|4;3}(C_{2|3}(u_2|u_3)|C_{3|4}(u_3|u_4))|\\
&C_{5|3;4}(C_{5|4}(u_5|u_4)|\\
&C_{3|4}(u_3|\\
&u_4))).
\end{aligned}$$

Now one can compute the inverses

$$
\begin{aligned}
C^{-1}(y_2|u_3,u_4,u_5) =& C^{-1}_{2|34}(C^{-1}_{2|5;34}(y_2|C(u_5|u_3,u_4))|u_3,u_4)\\
C^{-1}(y_2|u_3,u_4) =& C^{-1}_{2|3}(C^{-1}_{2|4;3}(y_2|C(u_3|u_4))|u_3)\\
\Longrightarrow&\\
C^{-1}(y_2|u_3,u_4,u_5) =& C^{-1}_{2|3}(C^{-1}_{2|4;3}(C^{-1}_{2|5;34}(y_2|\\
& C_{5|3;4}(C_{5|4}(u_5|u_4)|C_{3|4}(u_3|u_4))|\\
& C_{3|4}(u_3|u_4))|\\
& u_3).
\end{aligned}
$$

For the second imputation value u_1^*:

$$
\begin{aligned}
C(u_1|u_2,u_3,u_4,u_5) =& C_{1|5;234}(C(u_1|u_2,u_3,u_4)|C(u_5|u_2,u_3,u_4))\\
C(u_1|u_2,u_3,u_4) =& C_{1|4;23}(C(u_1|u_2,u_3)|C(u_4|u_2,u_3))\\
C(u_5|u_2,u_3,u_4) =& C_{5|2;34}(C(u_5|u_3,u_4)|C(u_2|u_3,u_4))\\
C(u_1|u_2,u_3) =& C_{1|3;2}(C(u_1|u_2)|C(u_3|u_2))\\
C(u_4|u_2,u_3) =& C_{4|2;3}(C(u_4|u_3)|C(u_2|u_3))\\
C(u_5|u_3,u_4) =& C_{5|3;4}(C(u_5|u_4)|C(u_3|u_4))\\
C(u_2|u_3,u_4) =& C_{2|4;3}(C(u_2|u_3)|C(u_3|u_4))\\
C(u_1|u_2) =& C_{1|2}(u_1|u_2)\\
C(u_3|u_2) =& C_{3|2}(u_3|u_2)\\
C(u_4|u_3) =& C_{4|3}(u_4|u_3)\\
C(u_2|u_3) =& C_{2|3}(u_2|u_3)\\
C(u_5|u_4) =& C_{5|4}(u_5|u_4)\\
C(u_3|u_4) =& C_{3|4}(u_3|u_4)\\
\Longrightarrow&\\
C(u_1|u_2,u_3,u_4,u_5) =& C_{1|5;234}(C_{1|4;23}(C_{1|3;2}(C_{1|2}(u_1|u_2)|C_{3|2}(u_3|u_2))|\\
& C_{4|2;3}(C_{4|3}(u_4|u_3)|C_{2|3}(u_2|u_3)))|\\
& C_{5|2;34}(C_{5|3;4}(C_{5|4}(u_5|u_4)|C_{3|4}(u_3|u_4))|\\
& C_{2|4;3}(C_{2|3}(u_2|u_3)|C_{3|4}(u_3|u_4))))
\end{aligned}
$$

Now one can compute the inverses:

$$\begin{aligned}
C^{-1}(y_1|u_2,u_3,u_4,u_5) =& C^{-1}_{1|234}(C^{-1}_{1|5;234}(y_1|C(u_5|u_2,u_3,u_4))|u_2,u_3,u_4)\\
C^{-1}(y_1|u_2,u_3,u_4) =& C^{-1}_{1|23}(C^{-1}_{1|4;23}(y_1|C(u_4|u_2,u_3))|u_2,u_3)\\
C^{-1}(u_1|u_2,u_3) =& C^{-1}_{1|2}(C_{1|3;2}(y_1|C(u_3|u_2))|u_2)\\
&\Longrightarrow\\
C^{-1}(y_1|u_2,u_3,u_4,u_5) =& C^{-1}_{1|2}(C_{1|3;2}(C^{-1}_{1|4;23}(C^{-1}_{1|5;234}(y_1|\\
& C_{5|2;34}(C_{5|3;4}(C_{5|4}(u_5|u_4)|C_{3|4}(u_3|u_4))|\\
& C_{2|4;3}(C_{2|3}(u_2|u_3)|C_{3|4}(u_3|u_4))))|\\
& C_{4|2;3}(C_{4|3}(u_4|u_3)|C_{2|3}(u_2|u_3)))|\\
& C_{2|3}(u_2|u_3))|\\
& u_2).
\end{aligned}$$

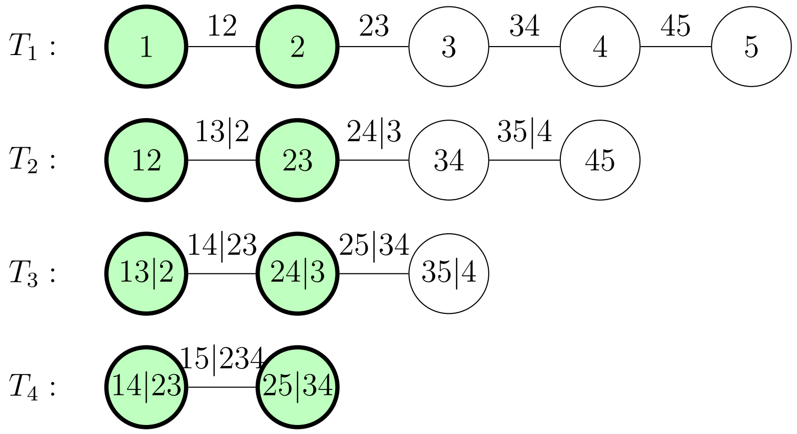


Figure 4.4: **Case 1.** 5-*dimensional* D-vine structure with the vertexes labeled that contain the first or the second variable. If they are both missing, imputation is possible with the copula regression method under this D-vine model.

Case 2. *The order does not matter in the sense that one can simulate with every given value. Either start with imputing u_1 or u_5. For example, start with u_1, i.e. simulate V_1, V_5 iid from a uniformly distributed random variable and impute*

$$\begin{aligned}
u_1^* =& C^{-1}_{1|2}(C^{-1}_{1|3;2}(C^{-1}_{1|4;23}(V_1|C_{4|2;3}(C_{4|3}(u_4|u_3)|C_{2|3}(u_2|u_3)))|C_{3|2}(u_3|u_2))|u_2),\\
u_5^* =& C^{-1}_{5|4}(C^{-1}_{5|3;4}(C^{-1}_{5|4;23}(C^{-1}_{5|4;123}(V_5|\\
& C_{1|4;23}(C_{1|3;2}(C_{1|2}(u_1^*|u_2)|C_{3|2}(u_3|u_2))|C_{4|2;3}(C_{4|3}(u_4|u_3)|C_{3|2}(u_3|u_2))))|\\
& C_{2|4;3}(C_{2|3}(u_2|u_3)|C_{4|3}(u_4|u_3)))|\\
& C_{4|3}(u_4|u_3))|\\
& u_4).
\end{aligned}$$

The inverses are computed as in case 1, with the permutation $\pi(1,2,3,4,5) = (5,1,2,3,4)$.

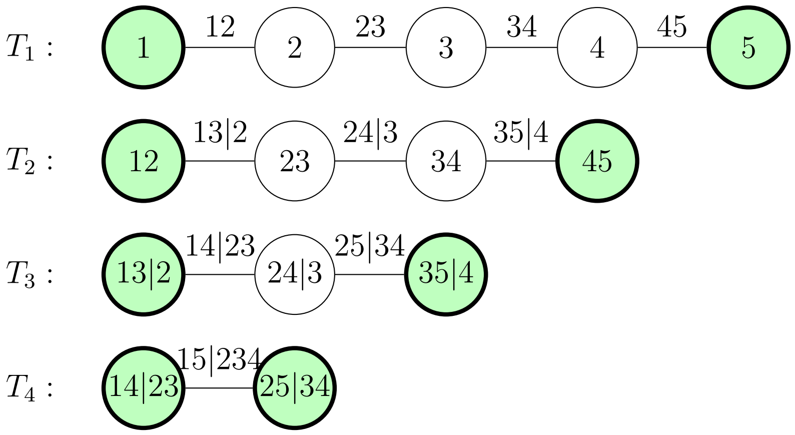


Figure 4.5: **Case 2.** 5-*dimensional* D-vine structure with the vertexes labeled that contain the first or the fifth variable. If they are both missing, imputation is possible with the copula regression method under this D-vine model.

Case 3. *Here start with the imputation in u_4 followed by u_5, i.e. simulate V_4, V_5 iid from a uniformly distributed random variable and impute*

$$\begin{aligned}
u_4^* =&C^{-1}_{4|3}(C^{-1}_{4|2;3}(C^{-1}_{4|3;12}(V_4|C_{1|3;2}(C_{1|2}(u_1|u_2)|C_{3|2}(u_3|u_2)))|C_{2|3}(u_2|u_3))|u_3),\\
u_5^* =&C^{-1}_{5|4}(C^{-1}_{5|3;4}(C^{-1}_{5|4;23}(C^{-1}_{5|4;123}(V_5|\\
&\quad C_{1|4;23}(C_{1|3;2}(C_{1|2}(u_1|u_2)|C_{3|2}(u_3|u_2))|C_{4|2;3}(C_{4|3}(u_4^*|u_3)|C_{3|2}(u_3|u_2))))|\\
&\quad C_{2|4;3}(C_{2|3}(u_2|u_3)|C_{4|3}(u_4^*|u_3)))|\\
&\quad C_{4|3}(u_4^*|u_3))|\\
&\quad u_4^*).
\end{aligned}$$

The inverses are computed as in case 1, with the permutation $\pi(1,2,3,4,5) = (5,4,3,2,1)$.

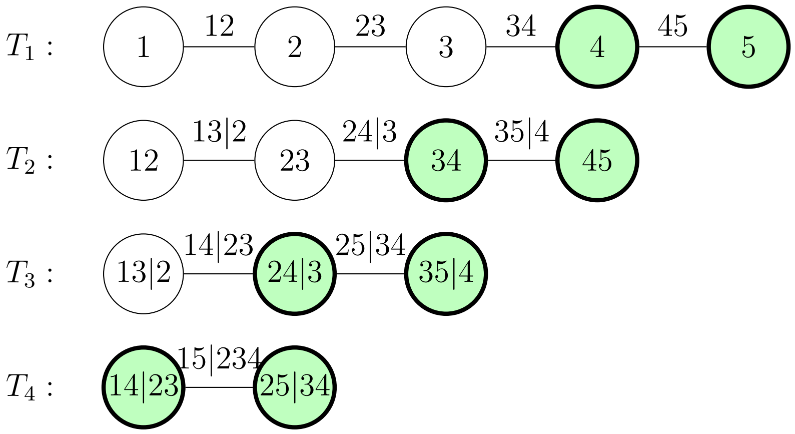


Figure 4.6: **Case 3.** 5-*dimensional* D-vine structure with the vertexes labeled that contain the fourth or the fifth variable. If they are both missing, imputation is possible with the copula regression method under this D-vine model.

In all these situations one can find a specific missing value with the property of being a leaf in every tree in the vine structure. After deletion again one finds a nonresponse with this characteristic.
*The problem changes significantly if the pair (U_1, U_4) is missing, which we call **Case 4**.*
In a first step, U_1 has the right property to be such a leaf. But already in the first tree U_4 fails to fulfill this criterion.
And not surprisingly, it is not possible to impute U_1 or U_4 with all information available. Lets try it for demonstration and simulate V_1, V_4 iid from a uniformly distributed random variable.

Case 4. *Try to compute*

$$\begin{aligned} u_1^* =& C_{1|2}^{-1}(C_{1|3;2}^{-1}(C_{1|4;23}^{-1}(V_1|C_{4|2;3}(C_{4|3}(u_4|u_3)|C_{2|3}(u_2|u_3)))|C_{3|2}(u_3|u_2))|u_2), or\\ u_4^* =& C_{4|3}^{-1}(C_{4|2;3}^{-1}(C_{4|1;23}^{-1}(V_4|C_{1|3;2}(C_{1|2}(u_1|u_2)|C_{3|2}(u_3|u_2)))|C_{2|3}(u_2|u_3))|u_3), \end{aligned}$$

but neither the first, nor the second value is possible to compute, because u_4 is missing in the first equation and u_1 in the second equation. So fitting a new D-vine model after changing the order of, for example, U_4 and U_5 "$\pi(U_1, U_2, U_3, U_4, U_5) \to (U_1, U_2, U_3, U_5, U_4)$" would lead us to Case 2.

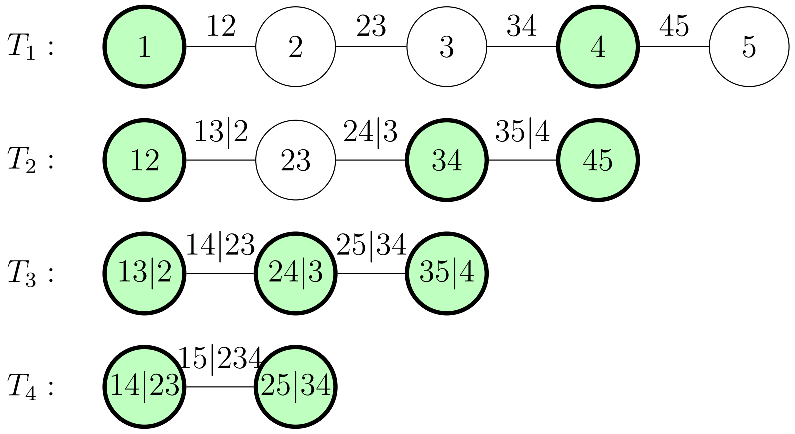


Figure 4.7: **Case 4.** 5-*dimensional* D-vine structure with the vertexes labeled that contain the first or the fourth variable. If they are both missing, imputation is not possible with the copula regression method under this D-vine model.

Now look at the general case i.e. for an R-vine structure. If there is only one missing value, it is possible to impute with all available information if and only if the nonresponse random variable is not in the conditioned set of any bivariate copula of the vine. This case occurs if and only if the nonresponse variable is a leaf in any tree in the structure of the vine trees. If there is more than one missing value, just iterate this procedure and treat the imputed values as given.

4.2 Vine Copula Fitting Imputation (CopFit)

As explained, this approach seems quite natural. However, only at first glance. Consider a matrix

$$\mathbf{Data} = \begin{pmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_n \end{pmatrix} \in [0,1]^{n \times d},$$

with n independent observations ($\mathbf{A}_i \in [0,1]^d$, $i = 1, \dots, n$) of a *d-dimensional* random vector $\mathbf{U} = (U_1, \dots, U_d)$, $U_j \sim U[0,1]$, $j = 1, \dots, d$ with some nonresponse in $n - n_1$ of the n observation vectors. To estimate the "best" vine copula model, only consider the complete cases $\mathbf{F}$ in $\mathbf{Data}$, with

$$\mathbf{F} = \begin{pmatrix} \mathbf{A}_{i_1} \\ \vdots \\ \mathbf{A}_{i_{n_1}} \end{pmatrix} \in [0,1]^{n_1 \times d},$$

only containing completely observed data. Now it is well known (i.e. with Dissmann's algorithm) how to estimate the tree structure of the vine model and all bivariate copula families with all additional parameters required. So the procedure is the following ("only one missing observation case" is omitted, because of having almost the same steps):

Step 1. Determine the complete case matrix $\mathbf{F}$ with respect to $\mathbf{Data}$.

Step 2. Fit an R-vine model to $\mathbf{F}$ without further constraints.

Step 3. For each row $\mathbf{A}_i$, for $i \in I_{FC} = \{1, \dots, n\} \backslash \{i_1, \dots, i_{n_1}\}$ with missing data, impute the m_i missing values with the help of each given value by using r sub-vine structures, as explained below.

Step 3.1. W.l.o.g the first m_i values are missing (otherwise restructure the data in each Step 3). Let r denote the number of sub-vine structures. They are determined with Algorithm 3 (Finding sub-vine structures), explained later. Simulate $u_{i1}^{*1} \sim U_1|D_1, \dots, u_{i1}^{*r} \sim U_1|D_r$, where

$$D_l \subset D := \{U_{m_i+1} = u_{i,m_i+1}, \dots, U_d = u_{id}\}, \qquad l = 1, \dots, r$$

are subsets of the whole conditioning set according to the sub-vine structures. Set $u_{i1}^* := f(u_{i1}^{*1}, \dots, u_{i1}^{*r})$, where $f : [0,1]^r \to [0,1]$ is some function explained later (e.g. the arithmetic mean $f(u_{i1}^{*1}, \dots, u_{i1}^{*r}) = 1/r \times \sum_{l=1}^{r} u_{i1}^{*l}$).

Step 3.2. Simulate $u_{i2}^{*1} \sim U_2|D_1, \dots, u_{i2}^{*r} \sim U_2|D_r$, where

$$D_l \subset D := \{U_1 = u_{i1}^*, U_{m_i+1} = u_{i,m_i+1}, \dots, U_d = u_{id}\}, \qquad l = 1, \dots, r$$

and set $u_{i2}^* := f(u_{i2}^{*1}, \dots, u_{i2}^{*r})$

Step 3.m_i. Simulate $u_{im_i}^{*1} \sim U_{m_i}|D_1, \dots, u_{im_i}^{*r} \sim U_{m_i}|D_r$, where

$$D_l \subset D := \{U_1 = u_{i1}^*, \dots, U_{m_i-1} = u_{i,m_i-1}^*, U_{m_i+1} = u_{i,m_i+1}, \dots, U_d = u_{id}\}, \qquad l = 1, \dots, r$$

and set $u_{im_i}^* := f(u_{im_i}^{*1}, \dots, u_{im_i}^{*r})$.

Note that the r can change from Step 3.1 to Step $3.m_i$. Further note that

$$\bigcup_{l=1}^{r} D_l = D$$

in each step, but the sets D_l, $l = 1, \dots, r$ are not necessarily disjoint. Again if more than one value per row is missing, as seen, the procedure is used iteratively.
A clear advantage of this method is that from now on, one always works with the "best" fitting vine copula model in the sense of a goodness of fit criteria (for example the AIC). Unfortunately, one has to be very careful with the imputation procedure considering the vine structure given now. From the previous example (Example 13) it should be clear that for some nonresponse variables there are several, sometimes not perfectly satisfying (in the sense that one could not use the whole response data available) possibilities to impute the missing values. So two questions arise: The first one (a) is about how one can get every imputation possibility available to get a value closest to reality and the second one (b) tries to answer which option should be taken (i.e. how to select the function f).
a) From the previous section (vine copula regression imputation) it is known that one only can use every information available to impute a missing value if and only if the variable is a leaf in every tree T_i, $i = 1, \dots, d-1$ of the vine tree structure. If this is the case, everything is fine and there is just one rational (the optimal one) possibility to impute the value via a simulation, taking into account every given $d-1$ observation in the whole *d-dimensional* vector. Unfortunately, this is only the case for (mostly only few) specific values. For all other missing data one has to find a sub-vine structure where the variable of interest is such a leaf.

```
Data: vine tree specification in array form, i.e., $M$, where $m_{k,k} = d - k + 1$,
      $k = 1, \dots, d$, the number of the missing value $r$, a binary vector $bi$
      (= $NULL$ at the beginning) and a list of matrices $List_M$ (= $NULL$ at the
      beginning).
Result: L sub-vine tree specifications in array form.
if $m_{i,j} = r$ for some $i \geq j + 2$, then
    search for all columns $j = j_1, \dots, j_J$ with entries $m_{i,j} = r$ for $i \geq j + 2$. For the
    column $j_{max}$, with $m_{i_{max},j_{max}} = r$, s.t. $i_{max} := \max\{i | m_{i,j} = r, \text{ with } i \geq j + 2\}$
    (that is the column, where $r$ is the lowest entry of all columns $j = j_1, \dots, j_J$),
    do
    1. $bi = (bi, 0)$, set $M_{bi} = M$ , set the column $j_{max}$ and all entries with
    $(m_{bi})_{i,j} = (m_{bi})_{j_{max},j_{max}}$ equal to $NULL$, repeat algorithm with $M_{bi}$
    and 2. $bi = (bi, 1)$, set $M_{bi} = M$, set all columns in $M_{bi}$, with
    $(m_{bi})_{k,k} \in \{(m_{bi})_{n,j_{max}} | n = j_{max} + 1, \dots, i_{max}\}$, $k = 1, \dots, d$ and all entries
    $(m_{bi})_{i,j} \in \{(m_{bi})_{n,j_{max}} | n = j_{max} + 1, \dots, i_{max}\}$, $i, j = 1, \dots, d$ equal to $NULL$,
    repeat algorithm with $M_{bi}$
else
    Delete all $NULL$'s and set $List_M = (List_M, M_{bi})$
end
return $List_M = (M_1, \dots, M_L)$.
```

Algorithm 3: Finding sub-vine structures

Theorem 5 (Algorithm "Finding sub-vine structures"). *The algorithm "Finding sub-vine* *structures" collects all possible (maximal) sub-vine structures where the missing value is such a leaf.*

Proof 3 (Algorithm "Finding sub-vine structures")**.** *A variable is a leaf in every tree of the vine tree structure if and only if there is no copula involved where this variable is in the conditioned set. Now in the R-vine matrix, for each column there are exactly two possibilities to reduce the matrix, such that the variable is not in the conditioned set anymore.*

1. *Delete the whole column that contains the variable in a conditioned set, i.e. where the variable is an entry below the sub-diagonal. Additionally, one has to delete all entries with the diagonal entry of this column in the whole matrix. Reducing the matrix in this way is equal to deleting the diagonal entry variable of the column in every tree in the R-vine tree structure.*

2. *In the column of the matrix, delete all variables between the diagonal entry and the missing variable in the whole Matrix. This is done via deleting all columns which have those as diagonal entries and additionally every entry where they occur. Reducing the matrix in such a way is equal to deleting all variables between the diagonal entry and the missing variable in every tree in the R-vine tree structure.*

□

Example 15 (Finding sub-vine structures)**.** *Assume a missing observation in node 1, given the R-vine matrix*

$$M = \begin{pmatrix} 5 & & & & \\ 3 & 4 & & & \\ 4 & 3 & 3 & & \\ 2 & 1 & 1 & 2 & \\ 1 & 2 & 2 & 1 & 1 \end{pmatrix},$$

with the tree specification shown in Figure 4.8. Now use Algorithm 3 to find all sub-vine structures where 1 *is a leaf in every tree in the R-vine tree structure:*

1. $$M_0 = \begin{pmatrix} \square & & & & \\ \square & 4 & & & \\ \square & 3 & 3 & & \\ \square & 1 & 1 & 2 & \\ \square & 2 & 2 & 1 & 1 \end{pmatrix},$$

   Delete the first column, because "1" is an entry below the sub-diagonal. Additionally, one has to remove every entry with a "5" (is not the case in this example). Since "1" is still in a conditioned set (second column), one has to repeat the algorithm with the reduced matrix. But first, the second step of the algorithm.

2. $$M_1 = \begin{pmatrix} 5 & & & & \\ \square & \square & & & \\ \square & \square & \square & & \\ \square & \square & \square & \square & \\ 1 & \square & \square & \square & 1 \end{pmatrix},$$

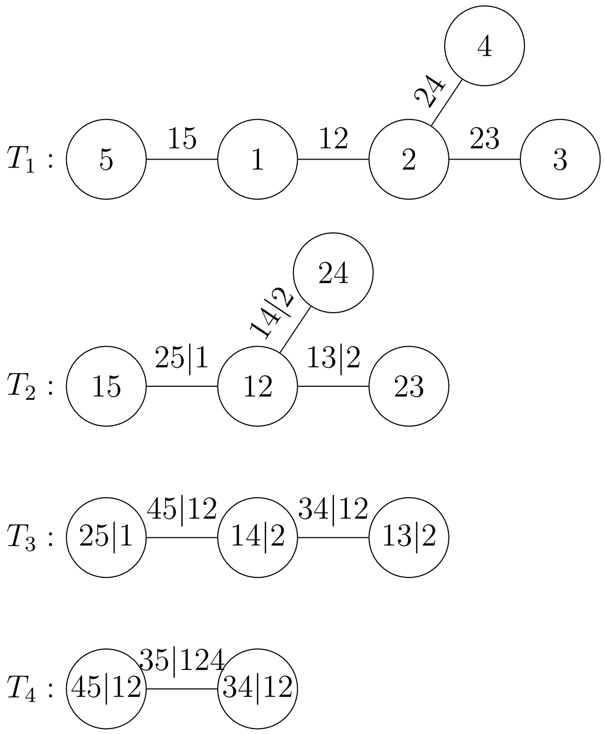


Figure 4.8: 5-*dimensional* R-vine structure.

Delete the entries between "5" and "1", so "1" is not in a conditioned set anymore i.e. the entries 2,3,4. Additionally, one has to remove the whole columns where those are the diagonal entries and all other entries, where those numbers occur (is not the case in this example). Now "1" is not in a conditioned set anymore.

1.1.

$$M_{00} = \begin{pmatrix} \square & & & & \\ \square & \square & & & \\ \square & \square & 3 & & \\ \square & \square & 1 & 2 & \\ \square & \square & 2 & 1 & 1 \end{pmatrix},$$

Delete the second column, because "1" is an entry below the sub-diagonal. Additionally, one has to remove every entry with a "4" (is not the case in this example). Now "1" is not in a conditioned set anymore.

1.2.

$$M_{01} = \begin{pmatrix} \square & & & & \\ \square & 4 & & & \\ \square & \square & \square & & \\ \square & 1 & \square & 2 & \\ \square & 2 & \square & 1 & 1 \end{pmatrix}.$$

Delete the entries between "4" and "1" in the second column, so "1" is not in a conditioned set anymore i.e entry "3". Additionally, one has to remove the whole *column where "3" is the diagonal entry and all other entries, where this number occurs (is not the case in this example). Now "1" is not in a conditioned set anymore.*

This results in 3 *different matrices:*

$$M_1 = \begin{pmatrix} 5 & \\ 1 & 1 \end{pmatrix}, \qquad M_{00} = \begin{pmatrix} 3 & & \\ 1 & 2 & \\ 2 & 1 & 1 \end{pmatrix}, \textit{ and} \qquad M_{01} = \begin{pmatrix} 4 & & \\ 1 & 2 & \\ 2 & 1 & 1 \end{pmatrix}.$$

Easily seen, in each R-vine array there is no copula conditioned on 1*. Additionally, they are maximal, i.e. it is not possible to add a vertex from the original R-vine array. So if there is the data vector with only the first value missing* $u_j = (-, u_{j2}, u_{j3}, u_{j4}, u_{j5})$*, the sets* D_l*,* $l = 1, 2, 3$ *are the following:*

$$D_1 = \{U_5 = u_{j5}\}, \qquad D_2 = \{U_2 = u_{j2}, U_3 = u_{j3}\}, \qquad D_3 = \{U_2 = u_{j2}, U_4 = u_{j3}\}.$$

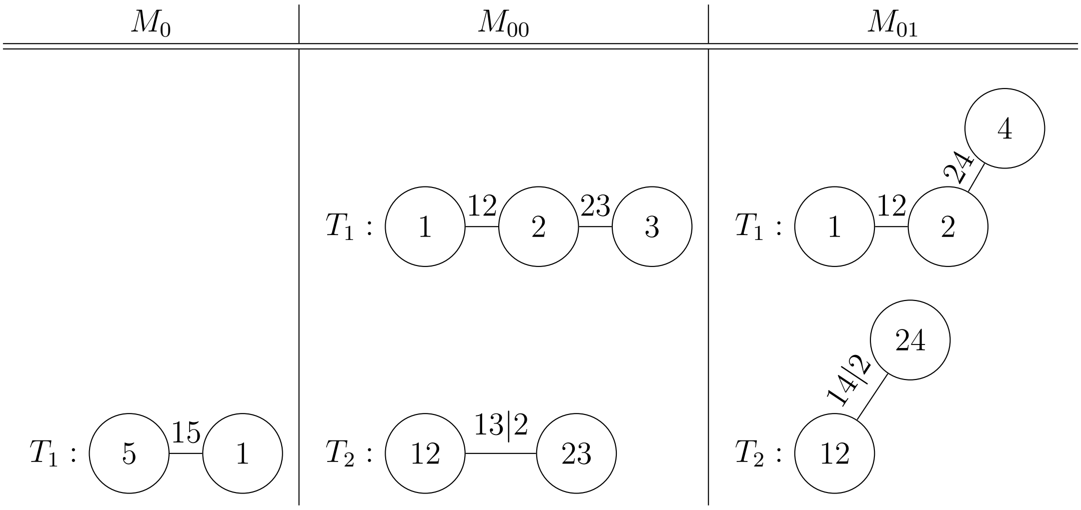


Table 4.1: Possible sub-vine structures where 1 is a leaf on each tree of the sub R-vine array (according to Figure 4.8).

b) Often, if there are several possibilities, it is useful to not only apply one of them, but try to get information from every or sometimes a small subset of them. This rule can also be applied in this method via taking a weighted sum of all available, or a chosen subset, of options to impute which can be demonstrated by Example 13.

Example 16 (Example 13 continued (a))**.** *Assume missing data in the first column of* **Data***, only. There is the problem of how to impute variable* U_1 *given* U_2, U_3, U_4 *in the given C-vine structure with bivariate copulae* C_{12}*,* C_{13}*,* C_{14}*,* $C_{23|1}$*,* $C_{24|1}$*,* $C_{34|12}$*.*

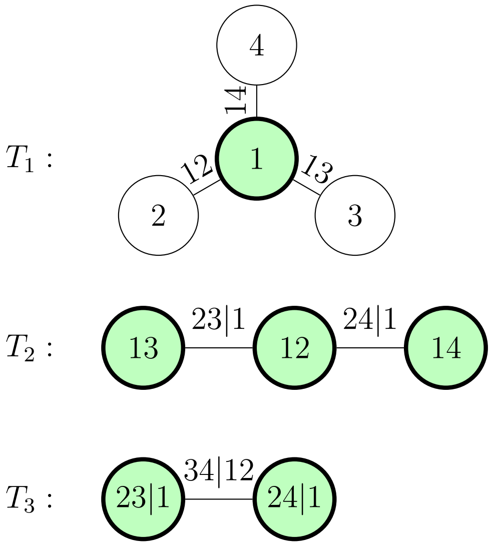


Figure 4.9: 4-*dimensional* C-vine structure with the vertexes labeled that contain the first variable, which is considered to be the missing observation. This is one example where it is not possible to impute with taking into account all available information.

The three possibilities are

Case 1*a. Simulate* $V_1 \sim U[0,1]$*.*

Case 1*b. Set*

$$u_1^* = F_{1|2}^{-1}(V_1|u_2) = C_{1|2}^{-1}(V_1|u_2),$$

or

$$u_1^* = F_{1|3}^{-1}(V_1|u_3) = C_{1|3}^{-1}(V_1|u_3),$$

or

$$u_1^* = F_{1|4}^{-1}(V_1|u_4) = C_{1|4}^{-1}(V_1|u_4).$$

Now, if one wants to use each of the three options, then one choice is to use an equally weighted sum, i.e.

Case 1*a. Simulate* $V_1, V_2, V_3 \sim U[0,1]$ *jointly independent.*

Case 1*b. Set*

$$u_1^* = \frac{1}{3}\left(C_{1|2}^{-1}(V_1|u_2) + C_{1|3}^{-1}(V_2|u_3) + C_{1|4}^{-1}(V_3|u_4)\right).$$

Alternatively, it is feasible to just use one method that is subjectively best. In the copula model assumption, high Kendall's tau values can be an indicator for a good imputation possibility, because it measures the dependency between missing and given data. High dependence (either positive or negative) is more likely to lead into a good imputation guess. Again a demonstration will help to understand the idea.

Example 17 (Example 13 continued (b))**.** *Again, let only the first column have missing observations. There are the three options to impute variable U_1 given U_2, U_3, U_4, but now look at the highest Kendall's tau value (in absolute terms):*

Case 1*a_1. Choose $i_{max} \in \{2,3,4\}$ s.t.*

$$\tau(C_{1i_{max}}) = \max\{|\tau(C_{1i})|, i \in \{2,3,4\}\}.$$

Case 1*a_2. Simulate $V_1 \sim U[0,1]$.*

Case 1*b. Set*

$$u_1^* = F_{1|i_{max}}^{-1}(V_1|u_{i_{max}}) = C_{1|i_{max}}^{-1}(V_1|u_{i_{max}}).$$

4.3 Vine Copula Expectation Imputation (CopExp)

The following procedure differs from the others because there is no simulation of a random variable involved. This implies that one can only do nonstochastic single imputations (without random influences), i.e. if one repeats the imputations with the same estimation methods, every time the same results will occur. This approach is comparable to the nonstochastic linear regression method. But, as said, instead of just using linear dependencies, here there will be a more flexible model estimation with probably asymmetric dependence structures. Later in the tests, this method will be extended in a way such that, if it is possible to simulate the missing value with all given values in the estimated vine structure, then simulating the nonresponse is preferred instead of conditioned expectation to avoid the variance underestimation.
As said before, the goal is to calculate the conditional expectation $\mathbb{E}[X_i|\mathbf{X}_{-i} = \mathbf{x}_{-i}]$, where $\mathbf{X}_{-i} = (X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_d)$ is the random vector without the i'th component. For this purpose it is very helpful to have the density function $f_{i|\{1,\ldots,d\}\setminus\{i\}}(x_i|\mathbf{x}_{-i})$ available. Consider the same matrix

$$\mathbf{Data} = \begin{pmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_n \end{pmatrix} \in [0,1]^{n\times d},$$

with n independent observations ($\mathbf{A}_i \in [0,1]^d$, $i = 1, \ldots, n$) of a *d-dimensional* random vector $\mathbf{U} = (U_1, \ldots, U_d)$, $U_j \sim U[0,1]$, $j = 1, \ldots, d$ with some nonresponse in $n - n_1$ of the n observation vectors, like before, in Section 4.2. To estimate the "best" vine copula model, again only consider the complete cases $\mathbf{F}$ in $\mathbf{Data}$, with

$$\mathbf{F} = \begin{pmatrix} \mathbf{A}_{i_1} \\ \vdots \\ \mathbf{A}_{i_{n_1}} \end{pmatrix} \in [0,1]^{n_1\times d},$$

only containing completely observed data. Then the procedure is (again, the "only one missing observation case" is omitted):

Step 1. Determine the complete case matrix $\mathbf{F}$ with respect to $\mathbf{Data}$.

Step 2. Fit an R-vine model to $\mathbf{F}$ without further constraints.

Step 3. For each row $\mathbf{A}_i$, for $i \in I_{FC} = \{1, \ldots, n\}\setminus\{i_1, \ldots, i_{n_1}\}$ with missing data, impute the m_i missing values separately.

Step 3.1. W.l.o.g the first m_i values of the i'th row $\mathbf{A}_i$ are missing (otherwise relabel the variables in this step). Set

$$u_{i1}^* := \mathbb{E}[U_1|U_{m_i+1} = u_{i,m_i+1}, \ldots, U_d = u_{id}]$$

Step 3.2. Set

$$u_{i2}^* := \mathbb{E}[U_2|U_1 = u_{i1}^*, U_{m_i+1} = u_{i,m_i+1}, \ldots, U_d = u_{id}]$$

Step 3.m_i. Set

$$u^*_{im_i} := \mathbb{E}[U_{m_i}|U_1 = u^*_{i1}, \dots, U_{m_i} = u^*_{i,m_i-1}, U_{m_i+1} = u_{i,m_i+1}, \dots, U_d = u_{id}].$$

Again, the method produces imputation values for each row $\mathbf{A}_i$ with missing values iteratively.

How to compute those conditional expectations is explained by the theorems below.

Theorem 6 (Conditional Density Function). *Let $f_{1,\dots,d}(\mathbf{x})$, and $f_{\{1,\dots,d\}\setminus\{i\}}(\mathbf{x}_{-i})$ be the continuous density functions of the d-dimensional random vector $\mathbf{X}$ and the $(d-1)$-dimensional random vector $\mathbf{X}_{-i}$ respectively. Then, for the (continuous) conditional density function, the following holds:*

$$f_{i|\{1,\dots,d\}\setminus\{i\}}(x_i|\mathbf{x}_{-i}) = \frac{f_{1,\dots,d}(\mathbf{x})}{f_{\{1,\dots,d\}\setminus\{i\}}(\mathbf{x}_{-i})}.$$

The same holds true for the conditional copula density function $c_{i|\{1,\dots,d\}\setminus\{i\}}(u_i|\mathbf{u}_{-i})$.

Now in the vine copula case, not every density function will always be available in closed form. So the next step is to get them via integrating out higher dimensional density functions.

Theorem 7 (Lower Dimensional Density Function). *Let $f_{1,\dots,d}(\mathbf{x})$ be the continuous density function of the d-dimensional random vector $\mathbf{X}$. Then, for the $(d-1)$-dimensional random vector $\mathbf{X}_{i-1}$, it holds that the corresponding (continuous) density is given by*

$$f_{\{1,\dots,d\}\setminus\{i\}}(\mathbf{x}_{-i}) = \int_{-\infty}^{\infty} f_{1,\dots,d}(\mathbf{x})dx_i.$$

Obviously, the same holds true for the copula density function $c_{\{1,\dots,d\}\setminus\{i\}}(\mathbf{u}_{-i})$.

At this point one can easily compute (in the most cases numerically) conditional expectations of one missing random variable X_i given $\mathbf{X}_{-i} = \mathbf{x}_{-i}$ with the help of the known closed form density and the two theorems above.

Theorem 8 (Conditional Expectation). *Let $f_{1,\dots,d}(\mathbf{x})$ and $f_{\{1,\dots,d\}\setminus\{i\}}(\mathbf{x}_{-i})$ be the continuous density functions of the d-dimensional random vector $\mathbf{X}$ and the $(d-1)$-dimensional random vector $\mathbf{X}_{-i}$. Then, for the conditional expectation, it holds that*

$$\mathbb{E}[X_i|\mathbf{X}_{-i} = \mathbf{x}_{-i}] = \frac{\int_{-\infty}^{\infty} x_i f_{1,\dots,d}((\mathbf{x}_{-i}, x_i))dx_i}{\int_{-\infty}^{\infty} f_{1,\dots,d}((\mathbf{x}_{-i}, x_i))dx_i}.$$

If there is more than one value missing, it is more suitable to apply the method sequentially than independently. There is a simple example to support this statement.

Example 18. *Set two random variables $X_1, X_2 \sim U[0,1]$ independently uniformly distributed on $[0,1]$ and a third random variable $X_3 := (1 - X_2 + X_1)\mathbf{1}_{\{X_2 > X_1\}} + (X_1 - X_2)\mathbf{1}_{\{X_2 \leq X_1\}}$. It is easy to show that X_3 is again uniform on $[0,1]$ and independent of X_2 and X_1, since the uniform distribution on this interval is negatively symmetric (i.e.*

$-X_2 \sim U[-1,0]$) and X_1 is independent of X_2. Now there is the case that $X_2 = 0.7$ is the given value with X_1 and X_3 missing, i.e. $(-, 0.7, -) = (x_1, x_2, x_3) \sim (X_1, X_2, X_3)$. If one computes expectations independently, the results are not satisfying since

$$\begin{aligned} x_1^* &:= \mathbb{E}[X_1|X_2 = x_2] = \mathbb{E}[X_1] = 0.5 \\ x_3^* &:= \mathbb{E}[X_3|X_2 = x_2] = \mathbb{E}[X_3] = 0.5 \end{aligned}$$

does not fulfill the property of $x_3 = 1 - 0.7 + 0.5 = 0.8$. With a sequential imputation of the missing values one gets

$$\begin{aligned} x_1^* &:= \mathbb{E}[X_1|X_2 = x_2] = \mathbb{E}[X_1] = 0.5 \\ x_3^* &:= \mathbb{E}[X_3|X_2 = 0.7, X_1 = 0.5] = (1 - 0.7 + 0.5) = 0.8, \end{aligned}$$

or, with the relation $X_1 = (X_3 + X_2 - 1)\mathbf{1}_{\{X_3 > 1 - X_2\}} + (X_3 + X_2)\mathbf{1}_{\{(X_3 \leq 1 - X_2)\}}$

$$\begin{aligned} x_3^* &:= \mathbb{E}[X_3|X_2 = x_2] = \mathbb{E}[X_3] = 0.5 \\ x_1^* &:= \mathbb{E}[X_1|X_2 = 0.7, X_3 = 0.5] = (0.5 + 0.7 - 1) = 0.8. \end{aligned}$$

This shows that firstly, sequential imputation keeps the special structure of the three random variable and secondly, that the order of imputation matters. Later in the test studies, the order is in such a way that simulation with all given values can be applied as fast as possible in the estimated R-vine structure. This will decrease the runtime of the algorithm immensely.

4.4 Comparison of the three Vine Copula Imputation Methods

	Vine Copula Regression Imputation (CopReg)	**Vine Copula Fitting Imputation (CopFit)**	**Vine Copula Expectation Imputation (CopExp)**
1) Method	Stochastic	Stochastic	Nonstochastic
2) Model fit	based on **F**	based on **F**	based on **F**
3) # of diff. models	several ($< 2^d$) models	one model	one model
4) Imputation via	one cond. simulation	several (r) cond. simulations	one cond. expectation

Table 4.2: Comparison of the three Vine Copula Imputation Methods.

Chapter 5

Simulation Study for the Performance of Copula Imputation Methods

5.1 Simulation Methods considered & Simulation Assumptions

Method	Abbreviation
Vine Copula Regression Imputation	**(CopReg)**
Vine Copula Fitting Imputation with real vine structure	**(CopFit)**
Vine Copula Fitting Imputation with vine structure according to the number of missing values in each column	**(CopFit2)**
Vine Copula Expectation Imputation	**(CopExp)**
Linear Regression	**(Norm)**
Predictive Mean Matching	**(PMM)**
Complete Case	**(Del)**

Table 5.1: Methods considered in the simulation study.

The theoretical aspects have been discussed and at this point the practical part begins. Until now, several different approaches have been investigated, using vine copulae to generate meaningful imputation data and obtain a complete data set. The next step is to determine which of the developed methods performs best according to selected criteria.

To get an idea of how well the approach of copula theory comes off in comparison to commonly used imputation methods, already existing procedures are integrated in the test, too. Here the **Predictive Mean Matching (PMM)** and the **Linear Regression (Norm)** methods are added.

In each of the three vine copula approaches, some degree of freedom is involved which has to be fixed here.

Copula Fitting Imputation (CopFit & CopFit2)

1. Here use the arithmetic mean of simulated values of all sub vine structures (with maximal possible number of known values) as the imputed value.

2.1. Once, for comparison, with the real (best fitting) C-vine structure **(CopFit)** and

2.2. once with a C-vine that allows to impute the marginal with the order from least missing to most missing values **(CopFit2)**. So the more values are marginally missing, the more likely it is to simulate with taking into account all available information in the C-vine structure, and therefor the better are the simulated imputation values for this marginal.

3. For both, the order of imputation is chosen from marginal 1 to 4 in the underlying C-vine structure, so the imputation values depend on this order.

Copula Regression Imputation (CopReg)

1. Only C-vine structures (overall 12) are allowed and

2. the imputation order is chosen such that for the best fitting C-vine, where only complete cases are allowed, marginal 1 is the first and marginal 4 the last candidate. This order has the advantage that the more given values can be included directly for imputation (in the best fitting C-vine structure), the later the nonresponse will be filled in there. This ensures, with each further imputation per row, an improvement in the chosen dependence structure.

Copula Expectation Imputation (CopExp)

1. The same C-vine and the same order as in the CopFit2 case is applied, which leads to a better run time.

2. The integration is done after marginals are transformed to N(0,1) margins to get smoother integration functions and with the help of an adaptive multivariate integration algorithm over hypercubes (R-function "adaptIntegrate" in the R-package "cubature").

3. If the C-vine structure allows for simulation of the missing values given all known values, than simulation is preferred. This is possible for the cases $(U_1, U_2, U_3, -)$, $(U_1, U_2, -, U_4)$, $(U_1, U_2, -, -)$, $(U_1, -, U_3, -)$, $(U_1, -, -, U_4)$, and $(U_1, -, -, -)$.

5.2 Simulation Setup

If a model is used, it is common to first use simulation to get an idea of how practical a newly invented scheme is. It is possible to consider various scenarios under known conditions to try out some of the strengths and weaknesses of some different approaches. For this purpose it is necessary to determine some important conditions.

- The simulation scheme.
- The number of simulations.
- The different conditions under which the simulation will be performed.

Since a simulation scheme for a vine copula dependence structure and a method for marginal transformations have been presented previously which additionally allow for many variabilities in creating multivariate random samples, it makes sense to use this procedure as the simulation scheme to generate data sets with missing values. It was decided to generate one hundred 4-*dimensional* data sets from the C-vine with density

$$\begin{aligned} c(u_1,u_2,u_3,u_4) =& c_{3,4;1,2}(C_{3|1;2}(u_3|u_2), C_{4|2;1}(u_4|u_2))\times \\ & c_{2,4;1}(C_{2|1}(u_2|u_1), C_{4|1}(u_4|u_1)) \times c_{2,3;1}(C_{2|1}(u_2|u_1), C_{3|1}(u_3|u_1))\times \\ & c_{1,2}(u_1,u_2) \times c_{1,3}(u_1,u_3) \times c_{1,4}(u_1,u_4), \end{aligned}$$

with missing values in each of the 4 variables. This was done under the following 6 binary combination possibilities (overall 64 combinations):

1. Length of the data sets:
   a) 500 or
   b) 1000

2. Marginally missing values
   a) (5%,5%,5%,5%) or
   b) (10%,1%,2%,5%)

3. Type of missing:
   a) uniform (random) **(MCAR)** or
   b) $(1-\alpha)$-quantile (the highest values) missing **(MAR)**

4. Marginal distributions:
   a) ($Exp(5)$,$t(3,0)$,$Exp(4)$,$t(2,0)$) **(Exp&t-dist)** or
   b) ($N(2,5)$,$t(3,0)$,$N(11,4)$,$N(-5,10)$) **(Norm&t-dist)**,
   where the expectations are denoted by $(\eta_1,\ldots,\eta_4)$

5. Copula families $(c_{1,2},c_{1,3},c_{1,4},c_{2,3;1},c_{2,4;1},c_{3,4;1,2})$:
   a) (Gauss,Gauss,Frank,Frank,Gauss,Gauss) or
   b) (Clayton,Clayton,Gumbel,Gumbel,Gumbel,Gauss)

6. Strength of dependencies in terms of Kendall's tau values (this is equivalent to differentiating between the parameters of the copula families, since there is a one to one relationship between the parameter of the families used and their Kendall's tau value):
   a) (0.7,0.4,0.6,0.5,0.3,0.2) **(High)** or
   b) (0.3,0.2,0.2,0.1,0.1,0.01) **(Low)**,
   denoted by $(\tau_1, \ldots, \tau_6)$

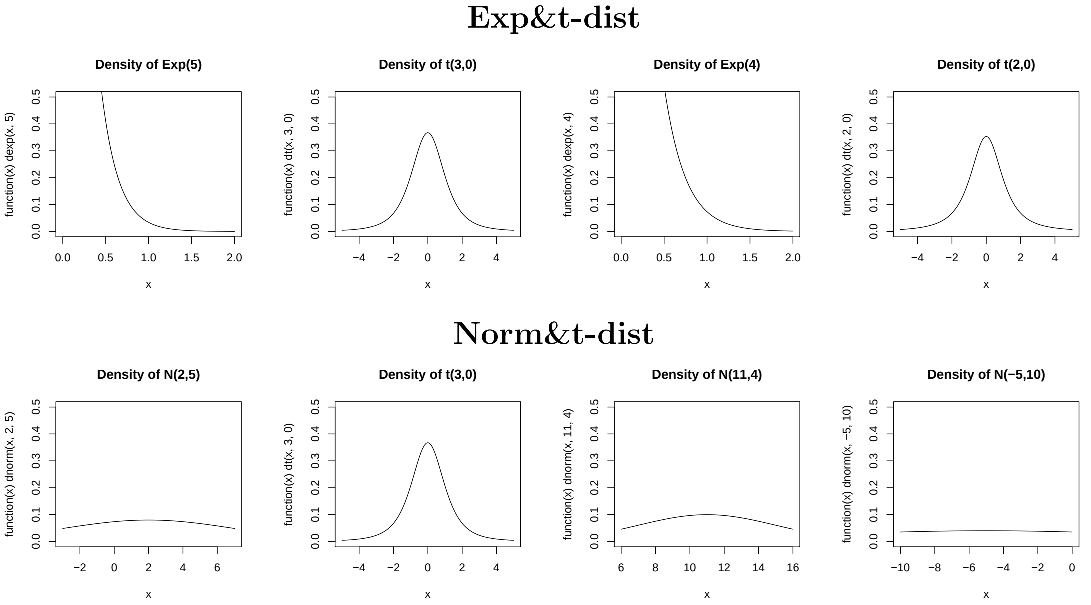


Figure 5.1: Plot of the density functions used for the marginals in the simulation. The plots are scaled with a width of 10 on the x-axis for Normal and t-distributions, and with a width of 2 for the Exp-distributions.

5.3 Performance Criteria

After imputation, the parameter of each (now fixed) family in the true C-vine structure was estimated by inversion, using estimated Kendall's tau values (using the corresponding bijection) and was compared to the true value from the simulation. The comparison was done via the **Euclidean norm**. This whole procedure was repeated with the parameters of the marginal distribution functions estimated via maximum likelihood under the assumption that the true distribution function was known. This may sound like a not always satisfied assumption, but finding well-fitting one-dimensional distributions is in many common situations well researched. Finally, in addition to the quality of the imputed data, an important aspect is the runtime, which was measured also. So goodness of success criteria are:

- deviation of the copula family parameters,
- deviation of the marginal distribution parameters, and

- runtime of the imputation method.

To compare the invented imputation techniques, a ranking system was used that, for each of the 100 data sets, rates the methods from 1 to #methods (in this case = 7) for each criterion separately:

- copula Kendall's tau value fitting for **High** Kendall's tau values in the **MCAR** cases,
- copula Kendall's tau value fitting for **Low** Kendall's tau values in the **MCAR** cases,
- copula Kendall's tau value fitting for **High** Kendall's tau values in the **MAR** cases,
- copula Kendall's tau value fitting for **Low** Kendall's tau values in the **MAR** cases,
- marginal expected value fitting for **Exp&t-dist** in the **MAR** cases,
- marginal expected value fitting for **Norm&t-dist** in the **MAR** cases,
- and runtime.

The arithmetic mean of the individual ranks decides about the rank for the different criteria. For this we need the definition of a *rank*-function.

Definition 10 (*rank*-function)**.**

$$rank_{\mathbf{x}}(x_i) := \frac{1}{|\{k|x_{i_1} \leq \ldots \leq x_{i_k} = x_i \leq \ldots \leq x_{i_d}\}|} \sum_{\{k|x_{i_1}\leq\ldots\leq x_{i_k}=x_i\leq\ldots\leq x_{i_d}\}} k$$

is the rank of entry x_i in the vector $\mathbf{x} = (x_1, \ldots, x_d)$, for $i \in \{1, \ldots, d\}$.

So for ”copula Kendall's tau value fitting for high Kendall's tau values in the MCAR case”, we have for data set $i = 1, \ldots, 100$, pair copula term $j = 1, \ldots, 6$, method $m = 1, \ldots, 7$ and scenario $s = 1, \ldots, 16$, compute

$$\begin{aligned} d_j^{m,s} :=& \frac{1}{100} \sum_{i=1}^{100} (\hat{\tau}_{i,j}^{m,s} - \tau_j^s)^2, \qquad j = 1, \ldots, 6, m = 1, \ldots, 7, s = 1, \ldots, 16 \\ & \text{average of data set distance. Set } \mathbf{d_j^s} := (d_j^{1,s}, \ldots, d_j^{7,s}) \\ rr^{m,s} :=& \frac{1}{6} \sum_{j=1}^{6} rank_{\mathbf{d}_j^s}(d_j^{m,s}), \qquad m = 1, \ldots, 7, s = 1, \ldots, 16 \\ & \text{average over pair copula term ranks. Set } \mathbf{rr^s} := (rr^{1,s}, \ldots, rr^{7,s}) \\ r^{m,s} :=& rank_{\mathbf{rr}^s}(rr^{m,s}), \qquad m = 1, \ldots, 7, s = 1, \ldots, 16. \\ & \text{rank of the individual method for each scenario } s. \end{aligned}$$

Set now

$$\mathbf{r}^s := (r^{1,s}, \ldots, r^{7,s}) \in [1, 7]^7, \qquad s = 1, \ldots, 16,$$

with $\sum_{m=1}^{7} r^{m,s} = \sum_{m=1}^{7} m = 28$. Set

$$\begin{aligned} rr^m :=& \frac{1}{16}\sum_{s=1}^{16} r^{m,s}, \qquad m = 1,\ldots,7, \\ & \text{average over scenarios. Set } \mathbf{rr} := (rr^1,\ldots,rr^7) \\ r^m :=& rank_{\mathbf{rr}} rr^m, \qquad m = 1,\ldots,7 \text{ and} \\ \mathbf{r} :=& (r^1,\ldots,r^7) \in [1,7]^7 \end{aligned}$$

as the overall result for the criterion "copula Kendall's tau value fitting for high Kendall's tau values in the MCAR case". A weighted mean (good fit is worth more than fast imputation in times of very fast computers) of the three criteria decides about the overall rank. For marginal parameter fit, case 3a (MCAR case) was omitted, because the results would have been almost identical for all methods.

5.4 Results

The tables below consist of the evaluation results. They are categorized in two different types of missing cases (simulated **MAR** and **MCAR** cases) and in three different observation cases (observing **High** and **Low** Kendall's tau values and observing marginal parameters for **Exp&t-dist** and for **Norm&t-dist**). So overall there are 64 scenarios investigated and they are arranged in 6 block scenarios giving 16 members per block scenario. These cases give an overview of how the individual methods have performed. The block scenarios are chosen in a natural way such that one can observe the performance for the dependence structure and for the marginal parameters. Further, maybe there is a difference in the type of missing cases. Runtime is evaluated separately without categorization. The first table shows the exact simulation categories in the 6 block scenarios, so the reader can really observe every situation individually.

There are also the plots of the Euclidean norm distance for the estimated Kendall's tau values and the marginal parameters added. There, one can observe the deviation of the distances for each case separately. If the whole dataset could not be filled out with imputation values for some reason, the distance was set to NA (missing) and the corresponding method was punished by the last place in this evaluation step. Note also that the y-axis has different ranges for Kendall's tau values and marginal parameters.

comparing copula fit τ		**assessing marginal** η
MCAR, **High**	**MAR**, **High**	**MAR**, **Exp&t-dist**
1) 1b,2b,**3a**,4a,5b,**6a**	33) 1b,2b,**3b**,4a,5b,**6a**	65) 1b,2b,**3b**,**4a**,5b,6a
2) 1b,2a,**3a**,4a,5b,**6a**	34) 1b,2a,**3b**,4a,5b,**6a**	66) 1b,2a,**3b**,**4a**,5b,6a
3) 1b,2b,**3a**,4a,5a,**6a**	35) 1b,2b,**3b**,4a,5a,**6a**	67) 1b,2b,**3b**,**4a**,5a,6a
4) 1b,2a,**3a**,4a,5a,**6a**	36) 1b,2a,**3b**,4a,5a,**6a**	68) 1b,2a,**3b**,**4a**,5a,6a
5) 1a,2b,**3a**,4a,5b,**6a**	37) 1a,2b,**3b**,4a,5b,**6a**	69) 1a,2b,**3b**,**4a**,5b,6a
6) 1a,2a,**3a**,4a,5b,**6a**	38) 1a,2a,**3b**,4a,5b,**6a**	70) 1a,2a,**3b**,**4a**,5b,6a
7) 1a,2b,**3a**,4a,5a,**6a**	39) 1a,2b,**3b**,4a,5a,**6a**	71) 1a,2b,**3b**,**4a**,5a,6a
8) 1a,2a,**3a**,4a,5a,**6a**	40) 1a,2a,**3b**,4a,5a,**6a**	72) 1a,2a,**3b**,**4a**,5a,6a
9) 1b,2b,**3a**,4b,5b,**6a**	41) 1b,2b,**3b**,4b,5b,**6a**	73) 1b,2b,**3b**,**4a**,5b,6b
10) 1b,2a,**3a**,4b,5b,**6a**	42) 1b,2a,**3b**,4b,5b,**6a**	74) 1b,2a,**3b**,**4a**,5b,6b
11) 1b,2b,**3a**,4b,5a,**6a**	43) 1b,2b,**3b**,4b,5a,**6a**	75) 1b,2b,**3b**,**4a**,5a,6b
12) 1b,2a,**3a**,4b,5a,**6a**	44) 1b,2a,**3b**,4b,5a,**6a**	76) 1b,2a,**3b**,**4a**,5a,6b
13) 1a,2b,**3a**,4b,5b,**6a**	45) 1a,2b,**3b**,4b,5b,**6a**	77) 1a,2b,**3b**,**4a**,5b,6b
14) 1a,2a,**3a**,4b,5b,**6a**	46) 1a,2a,**3b**,4b,5b,**6a**	78) 1a,2a,**3b**,**4a**,5b,6b
15) 1a,2b,**3a**,4b,5a,**6a**	47) 1a,2b,**3b**,4b,5a,**6a**	79) 1a,2b,**3b**,**4a**,5a,6b
16) 1a,2a,**3a**,4b,5a,**6a**	48) 1a,2a,**3b**,4b,5a,**6a**	80) 1a,2a,**3b**,**4a**,5a,6b
MCAR, **Low**	**MAR**, **Low**	**MAR**, **Norm&t-dist**
17) 1b,2b,**3a**,4a,5b,**6b**	49) 1b,2b,**3b**,4a,5b,**6b**	81) 1b,2b,**3b**,**4b**,5b,6a
18) 1b,2a,**3a**,4a,5b,**6b**	50) 1b,2a,**3b**,4a,5b,**6b**	82) 1b,2a,**3b**,**4b**,5b,6a
19) 1b,2b,**3a**,4a,5a,**6b**	51) 1b,2b,**3b**,4a,5a,**6b**	83) 1b,2b,**3b**,**4b**,5a,6a
20) 1b,2a,**3a**,4a,5a,**6b**	52) 1b,2a,**3b**,4a,5a,**6b**	84) 1b,2a,**3b**,**4b**,5a,6a
21) 1a,2b,**3a**,4a,5b,**6b**	53) 1a,2b,**3b**,4a,5b,**6b**	85) 1a,2b,**3b**,**4b**,5b,6a
22) 1a,2a,**3a**,4a,5b,**6b**	54) 1a,2a,**3b**,4a,5b,**6b**	86) 1a,2a,**3b**,**4b**,5b,6a
23) 1a,2b,**3a**,4a,5a,**6b**	55) 1a,2b,**3b**,4a,5a,**6b**	87) 1a,2b,**3b**,**4b**,5a,6a
24) 1a,2a,**3a**,4a,5a,**6b**	56) 1a,2a,**3b**,4a,5a,**6b**	88) 1a,2a,**3b**,**4b**,5a,6a
25) 1b,2b,**3a**,4b,5b,**6b**	57) 1b,2b,**3b**,4b,5b,**6b**	89) 1b,2b,**3b**,**4b**,5b,6b
26) 1b,2a,**3a**,4b,5b,**6b**	58) 1b,2a,**3b**,4b,5b,**6b**	90) 1b,2a,**3b**,**4b**,5b,6b
27) 1b,2b,**3a**,4b,5a,**6b**	59) 1b,2b,**3b**,4b,5a,**6b**	91) 1b,2b,**3b**,**4b**,5a,6b
28) 1b,2a,**3a**,4b,5a,**6b**	60) 1b,2a,**3b**,4b,5a,**6b**	92) 1b,2a,**3b**,**4b**,5a,6b
29) 1a,2b,**3a**,4b,5b,**6b**	61) 1a,2b,**3b**,4b,5b,**6b**	93) 1a,2b,**3b**,**4b**,5b,6b
30) 1a,2a,**3a**,4b,5b,**6b**	62) 1a,2a,**3b**,4b,5b,**6b**	94) 1a,2a,**3b**,**4b**,5b,6b
31) 1a,2b,**3a**,4b,5a,**6b**	63) 1a,2b,**3b**,4b,5a,**6b**	95) 1a,2b,**3b**,**4b**,5a,6b
32) 1a,2a,**3a**,4b,5a,**6b**	64) 1a,2a,**3b**,4b,5a,**6b**	96) 1a,2a,**3b**,**4b**,5a,6b

Table 5.2: Different simulation scenarios for the tables below (64 unique). For example (1b,2b,3a,4a,5b,6a) corresponds to data set length 1000, marginal missing values $(10\%, 1\%, 2\%, 5\%)$), type of missing is uniform, marginal distributions $(Exp(5)$,$t(3,0)$,$Exp(4)$,$t(2,0))$, copula families (Clayton,Clayton,Gumbel,Gumbel,Gumbel,Gauss) and Kendall's tau values (0.7,0.4,0.6,0.5,0.3,0.2).

5.4.1 Comparing copula fit in block MCAR, High

sc	TauCopFit	TauCopFit2	TauCopExp	**TauCopReg**	TauNorm	TauPMM	TauDel
1)	4	5	6	1	7	2	3
2)	5	3	6	2	7	4	1
3)	5	4	6	1	7	3	2
4)	3.5	3.5	7	1	6	5	2
5)	5	3	6	1	7	4	2
6)	4	5	6	1.5	7	3	1.5
7)	3	5	6	1	7	4	2
8)	5	3	6.5	2	6.5	4	1
9)	4	5	7	1	6	2.5	2.5
10)	4	6	7	2	5	3	1
11)	5	3	7	1	6	4	2
12)	4.5	4.5	7	2	6	3	1
13)	5.5	3.5	5.5	1	7	3.5	2
14)	3	4	7	2	6	5	1
15)	5.5	4	5.5	1	7	3	2
16)	5	4	7	1	6	3	2
	total	total	total	total	total	total	total
	5	4	6	**1**	7	3	2

Table 5.3: Ranks $\mathbf{r}^s$ of the different methods over the sixteen scenarios in block **MCAR, High**.

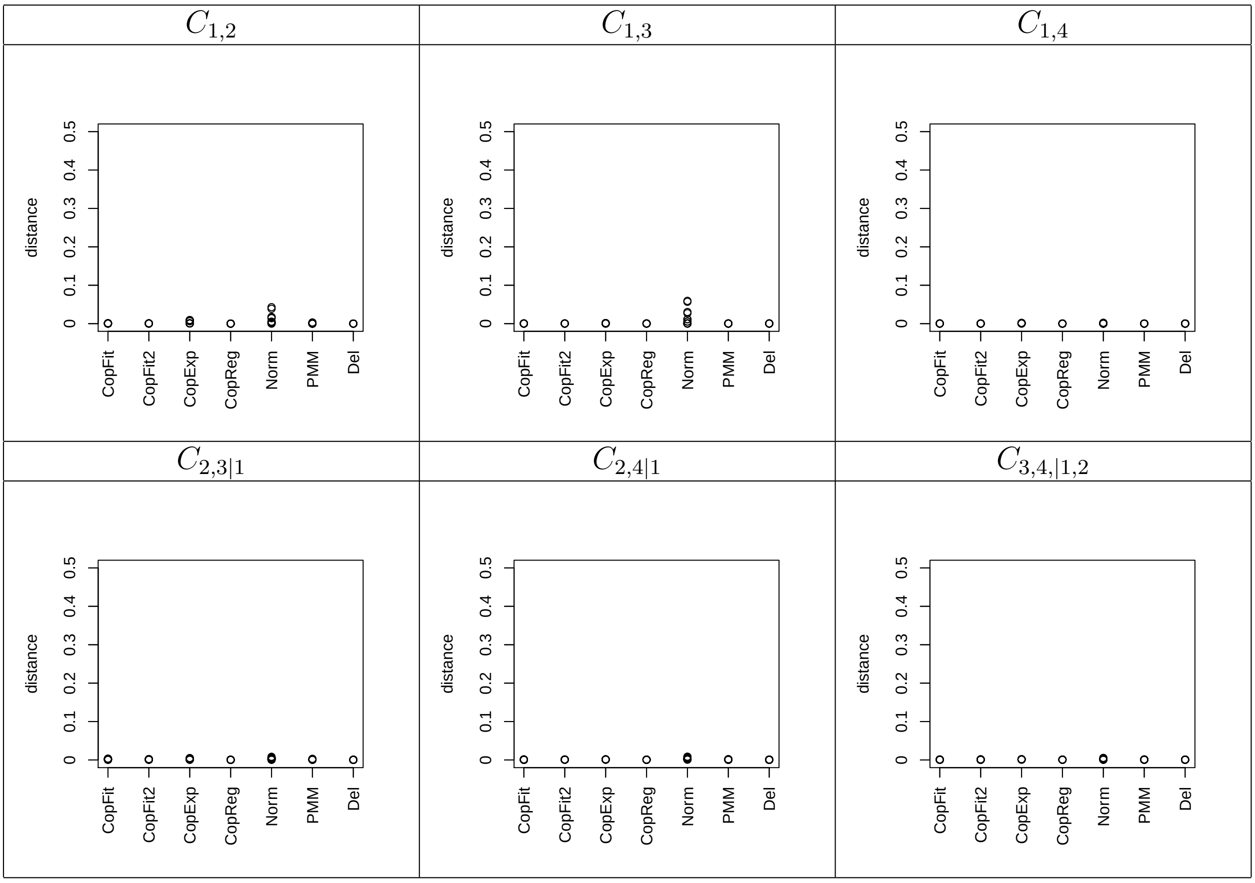


Figure 5.2: Plot of the Euclidean norm distance $d_j^{m,s}$ of the Kendall's tau values $\tau_j^{m,s}$ in block **MCAR, High**. For every pair copula term j and each method m, 16 dots are plotted for the scenarios s in this block.

As seen in the distance plot, the results are very close. For almost every method the dependence structure after imputation does not change significantly from the real one. For the scenarios with simulated **MCAR** data, every invented method should be able to find well-fitting imputation values, because with such data, one could also just consider complete cases. And with imputation we try to perform better than the deletion of observations. Also seen, the **High** dependencies with sometimes nonlinear structure are more challenging for Linear Regression (Norm).

5.4.2 Comparing copula fit in block MCAR, Low

sc	**TauCopFit**	TauCopFit2	TauCopExp	TauCopReg	TauNorm	TauPMM	TauDel
17)	3	2	6	5	7	1	4
18)	1	6	7	5	4	2	3
19)	1	5	3.5	6	7	3.5	2
20)	1	5	6	4	7	2.5	2.5
21)	2	7	5.5	5.5	2	4	2
22)	1	6	4	2.5	7	5	2.5
23)	2	5.5	7	5.5	4	1	3
24)	3	4	1.5	6	7	5	1.5
25)	4	7	5.5	5.5	2	3	1
26)	1	7	6	5	2	4	3
27)	5	6	3.5	7	2	1	3.5
28)	4	7	6	2.5	1	2.5	5
29)	2	7	6	5	4	1	3
30)	5	2	3	6	1	7	4
31)	2	4.5	7	6	1	4.5	3
32)	4	6.5	5	6.5	2	3	1
	total	total	total	total	total	total	total
	1	7	5	6	4	3	2

Table 5.4: Ranks $\mathbf{r}^s$ of the different methods over the sixteen scenarios in block **MCAR, Low**.

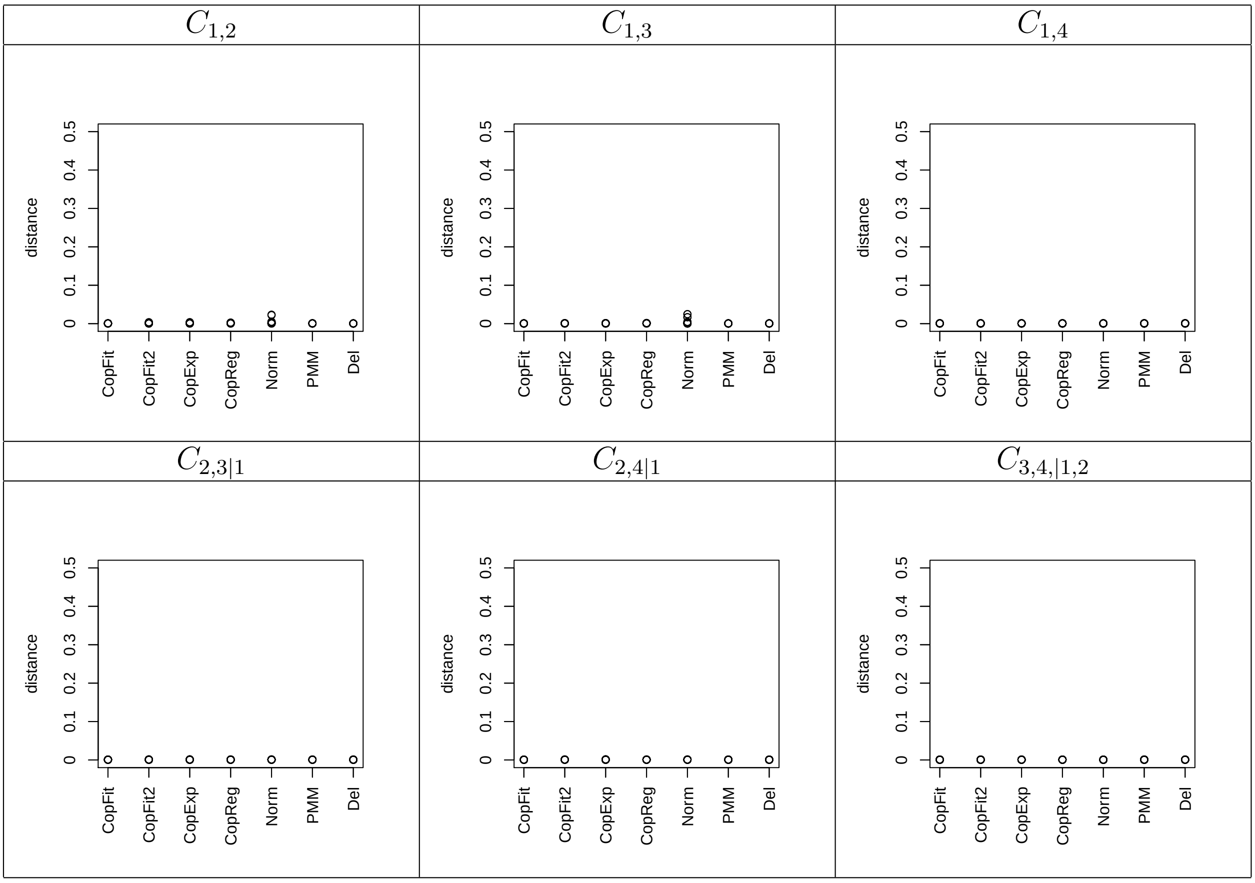


Figure 5.3: Plot of the Euclidean norm distance $d_j^{m,s}$ of the Kendall's tau values $\tau_j^{m,s}$ in block **MCAR, Low**. For every pair copula term j and each method m, 16 dots are plotted for the scenarios s in this block.

In this simulation block, there are almost the same results for all seven approaches in dependence modeling. With reducing the dependence between the margins, it is less grave if the imputation method suggests "wrong" imputation values. So it is not surprising that there is no big difference between the seven approaches.

5.4.3 Comparing copula fit in block MAR, High

sc	TauCopFit	TauCopFit2	TauCopExp	TauCopReg	TauNorm	TauPMM	**TauDel**
33)	5	4	7	2	6	3	1
34)	3	4.5	6	2	7	4.5	1
35)	3	5	7	1	6	4	2
36)	3	5	6.5	2	6.5	4	1
37)	3	5	7	1	6	4	2
38)	3	7	6	2	5	4	1
39)	5	4	7	2	6	3	1
40)	3	5	6	1	7	4	2
41)	6	5	7	2	4	1	3
42)	4	5	7	2.5	6	1	2.5
43)	4	5	7	3	6	2	1
44)	1.5	6	7	4	5	1.5	3
45)	3	4.5	7	4.5	6	2	1
46)	3.5	7	6	3.5	5	2	1
47)	3.5	5	7	3.5	6	2	1
48)	5	4	7	1	6	3	2
	total	total	total	total	total	total	total
	4	5	7	2	6	3	**1**

Table 5.5: Ranks $\mathbf{r}^s$ of the different methods over the sixteen scenarios in block **MAR, High**.

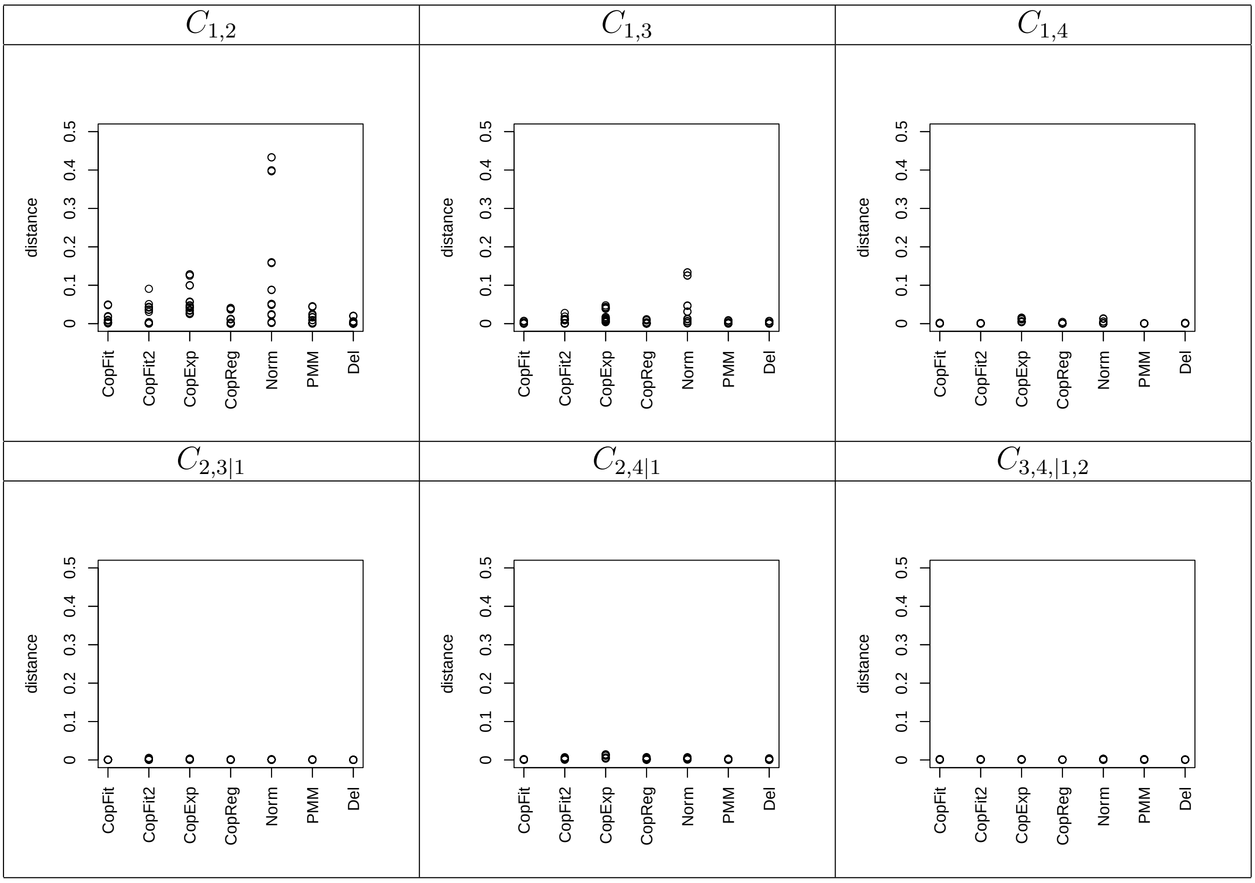


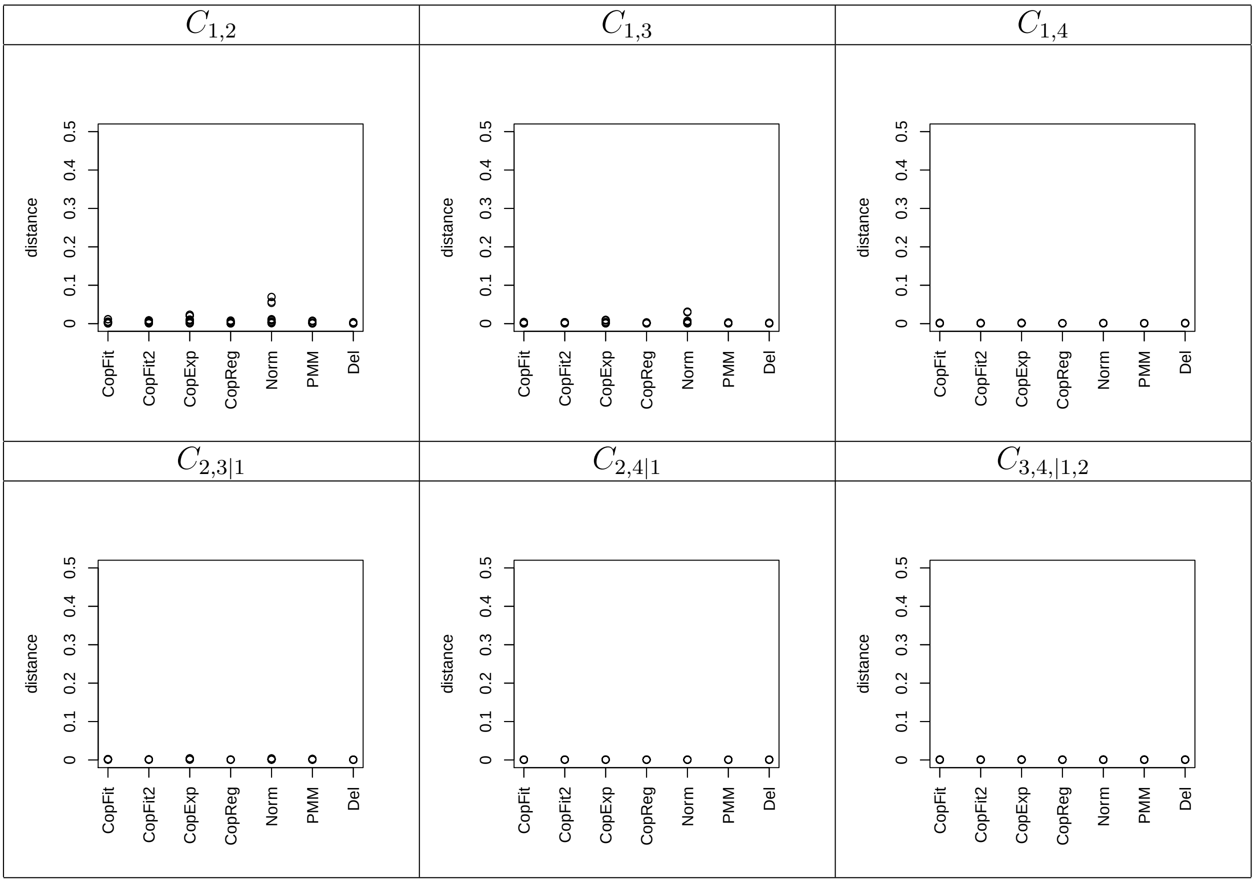
Figure 5.4: Plot of the Euclidean norm distance $d_j^{m,s}$ of the Kendall's tau values $\tau_j^{m,s}$ in block **MAR, High**. For every pair copula term j and each method m, 16 dots are plotted for the scenarios s in this block.

A little more challenging is a data set with **MAR** nonresponse, which is simulated here. An additional obstacle is the **High** dependence between the marginals. Four methods do quite well in our simulation, that are CopFit, CopReg, PMM, and Del. The two Copula Imputation methods have the advantage that they can look at dependence and marginal modeling separately, while the linear regression procedure, for example, faces real difficulties when it comes to the **MAR** scenarios. The PMM method does not simulate the imputation values. It uses values which are observed. These values are quite good for keeping the right dependence intensity, but marginally, this can be a shortfall if some values are missing that are not observed in the used data set. It is also interesting that the Kendall's tau value for the copula $C_{1,2}$ is the most challenging parameter. First, there are the most values missing in the first marginal and second, it has the highest dependence among all pair copulae. Additionally, $C_{2,3|1}$ is equipped with a high Kendall's tau. Therefor $C_{1,3}$ has higher fluctuation than $C_{1,4}$.

5.4.4 Comparing copula fit in block MAR, Low

sc	TauCopFit	TauCopFit2	TauCopExp	TauCopReg	TauNorm	**TauPMM**	TauDel
49)	4.5	3	7	1.5	6	4.5	1.5
50)	3	6	7	4	5	1	2
51)	3.5	5	7	2	6	1	3.5
52)	5	3.5	7	1	6	3.5	2
53)	4	6	7	2	5	1	3
54)	3	5	6	2	7	1	4
55)	4	5	7	1.5	6	3	1.5
56)	5	1	6	3.5	7	3.5	2
57)	6	4	7	1	5	2	3
58)	6	5	7	3	2	1	4
59)	6	3	7	4	5	1	2
60)	6	5	7	1	3.5	2	3.5
61)	5	6	7	3.5	2	1	3.5
62)	5.5	5.5	7	4	1.5	1.5	3
63)	7	5	6	2.5	2.5	4	1
64)	5	3.5	7	3.5	6	1	2
	total	total	total	total	total	total	total
	6	4	7	2	5	**1**	3

Table 5.6: Ranks $\mathbf{r}^s$ of the different methods over the sixteen scenarios in block **MAR, Low**.



$C_{1,2}$ $C_{1,3}$ $C_{1,4}$

$C_{2,3|1}$ $C_{2,4|1}$ $C_{3,4,|1,2}$

Figure 5.5: Plot of the Euclidean norm distance $d_j^{m,s}$ of the Kendall's tau values $\tau_j^{m,s}$ in block **MAR, Low**. For every pair copula term j and each method m, 16 dots are plotted for the scenarios s in this block.

Like in the **Low** dependence scenario before, imputation keeps the Kendall's tau values more or less the same for every method. Here it looks like just deleting incomplete data (Del) works best, but with our ranking system, the PMM method is closer to the real values. Again, the dependence parameter in $C_{1,2}$ is the highest one. This explains the slightly increased deviance from the true parameter.

5.4.5 Assessing marginal in block MAR, Exp&t-dist

sc	ParCopFit	ParCopFit2	ParCopExp	ParCopReg	**ParNorm**	ParPMM	ParDel
65)	7	2	5	1	4	3	6
66)	3.5	5	7	1	2	3.5	6
67)	3.5	3.5	6	2	1	7	5
68)	3	5	7	1	2	4	6
69)	5.5	2	4	3	1	5.5	7
70)	4.5	4.5	7	1	2	3	6
71)	5	3	6	1.5	1.5	4	7
72)	3.5	5	7	1	3.5	2	6
73)	5	6	7	1.5	1.5	3.5	3.5
74)	4	5	7	3	1.5	1.5	6
75)	4	6	7	3	1	2	5
76)	4	1.5	6.5	5	1.5	3	6.5
77)	7	4.5	4.5	1	3	2	6
78)	6	5	7	2	1	4	3
79)	5	4	3	2	1	7	6
80)	5	6	7	3.5	1	2	3.5
	total	total	total	total	total	total	total
	5	4	7	2	**1**	3	6

Table 5.7: Ranks $\mathbf{r}^s$ of the different methods over the sixteen scenarios in block **Exp&t-dist**.

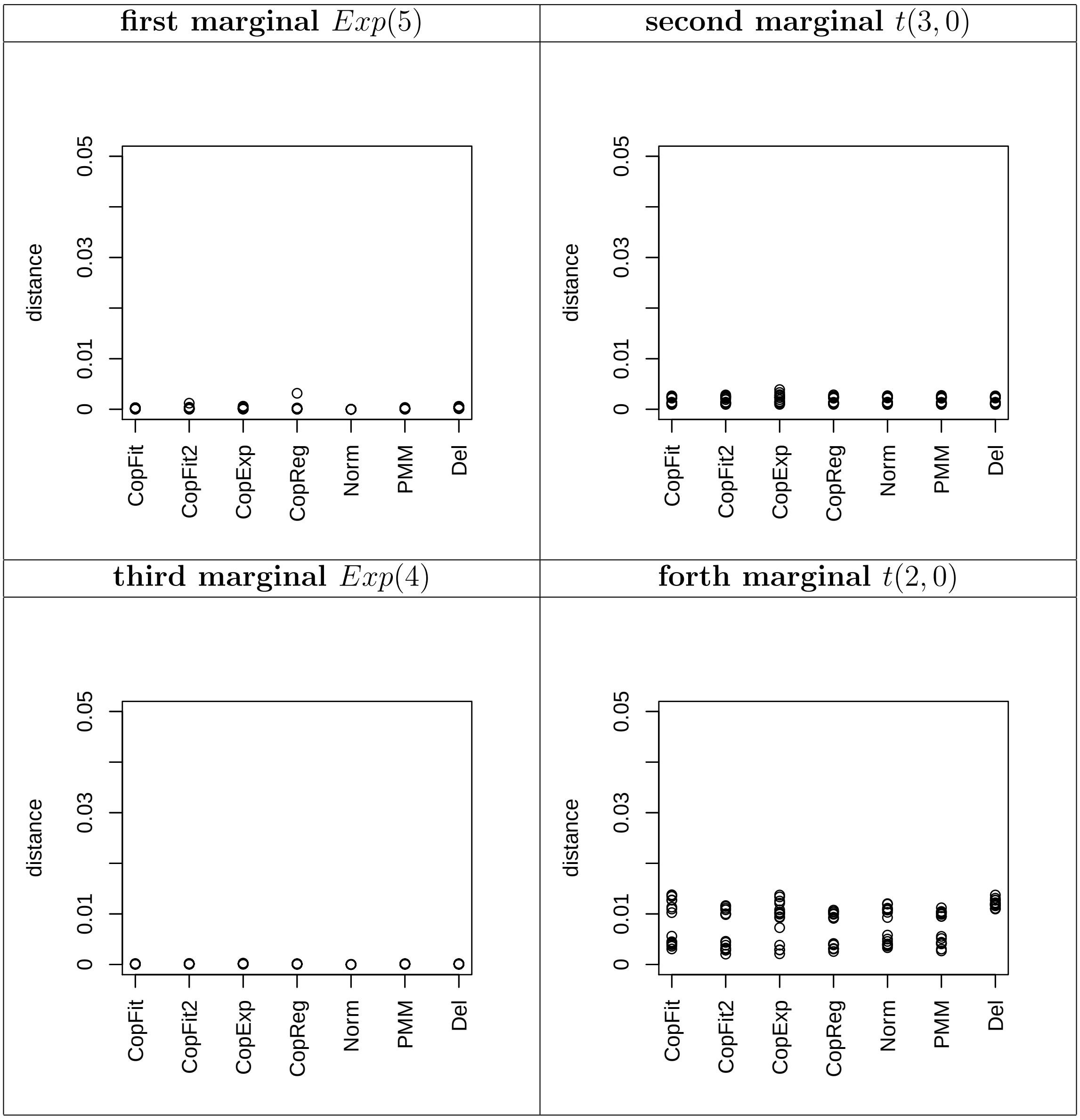


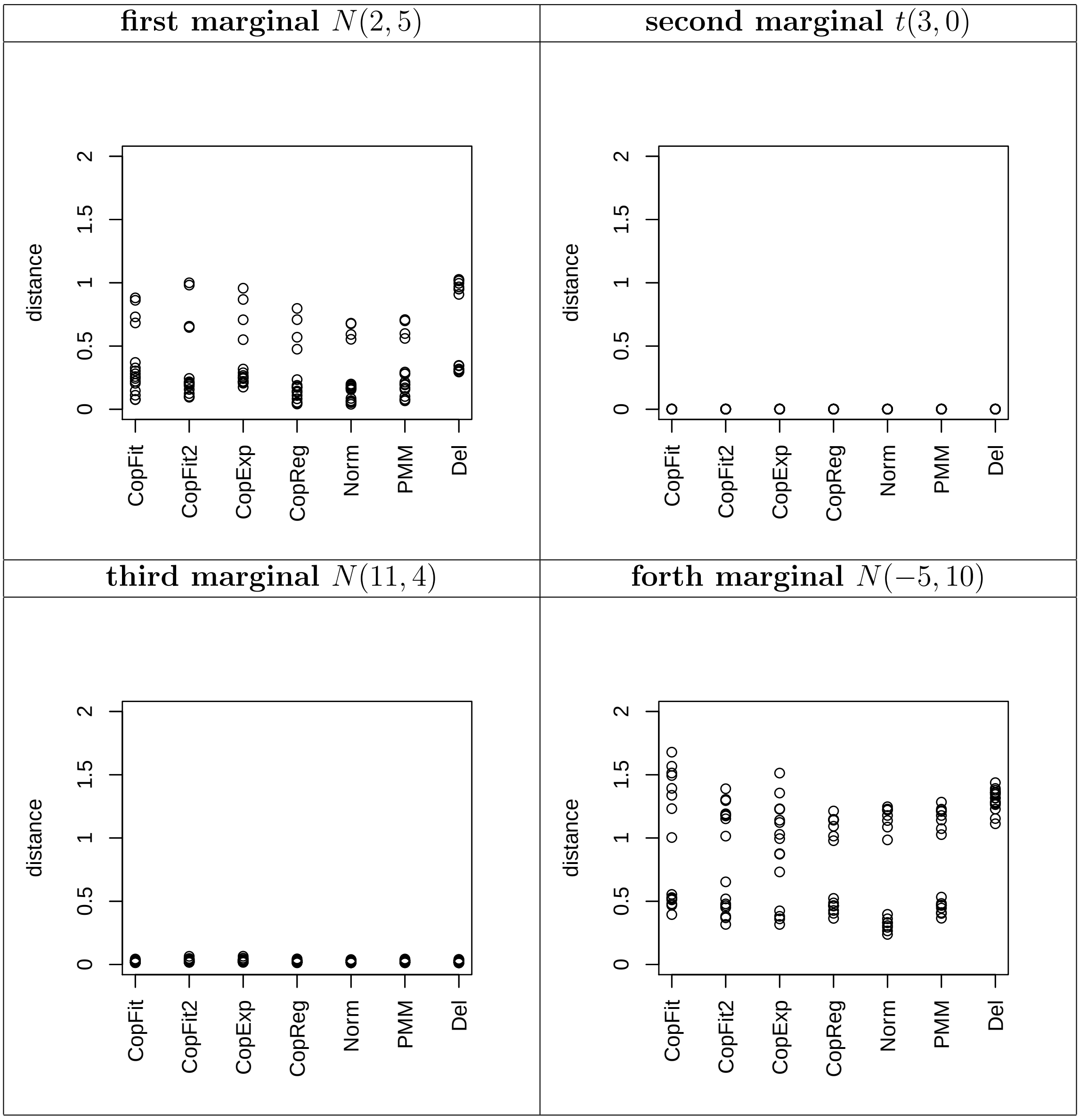
Figure 5.6: Plot of the Euclidean norm distance $d_j^{m,s}$ of the expected values $\eta_j^{m,s}$ in block **MAR, Exp&t-dist**. For every marginal j and each method m, 16 dots are plotted for the scenarios s in this block.

First, one can observe that Linear Regression Imputation (Norm) is closest to the true values. But also seen, Copula Regression Imputation (CopReg) is not far behind, and this procedure modeled the dependence structure much better. Second, observe that the last marginal is the most challenging for imputation. But the distances in this block are rather low. That is because the expectations and volatilities of the Exponential distribution are very low. Moreover, for the t-distributions, only the very high values are missing. The very low ones are still there. Together with the fat tails of this distribution it is helpful in finding good estimates.

5.4.6 Assessing marginal in block MAR, Norm&t-dist

sc	ParCopFit	ParCopFit2	ParCopExp	ParCopReg	**ParNorm**	ParPMM	ParDel
81)	5	4	6	2.5	1	2.5	7
82)	4	5.5	7	1	2	3	5.5
83)	7	3	6	1	2	4	5
84)	4	5	7	1.5	1.5	3	6
85)	5	3.5	6	2	1	3.5	7
86)	4	5	6.5	1.5	1.5	3	6.5
87)	7	6	2.5	2.5	1	4	5
88)	5	4	7	1	2	3	6
89)	4.5	6	7	2	1	4.5	3
90)	7	5.5	3	4	2	1	5.5
91)	6	4.5	7	1.5	3	1.5	4.5
92)	6.5	5	2.5	4	2.5	1	6.5
93)	2	4	6.5	3	1	5	6.5
94)	6	4.5	7	4.5	1	2	3
95)	5.5	7	2	1	3.5	5.5	3.5
96)	5	6	7	2	3	1	4
	total	total	total	total	total	total	total
	5	4	7	2	**1**	3	6

Table 5.8: Ranks $\mathbf{r}^s$ of the different methods over the sixteen scenarios in block **Norm&t-dist**.



first marginal $N(2,5)$

second marginal $t(3,0)$

third marginal $N(11,4)$

forth marginal $N(-5,10)$

Figure 5.7: Plot of the Euclidean norm distance $d_j^{m,s}$ of the expected values $\eta_j^{m,s}$ in block **MAR, Norm&t-dist**. For every marginal j and each method m, 16 dots are plotted for the scenarios s in this block.

The highest deviations are seen in the plot of the first and the last marginal. The first marginal is the one with the most nonresponse, and the last the one with the highest volatility. Here one can observe that imputation really can change parameter estimates, compared to deletion (Del). Normal distributions are very sensitive to missing values. For example in the first marginal, an absence of the highest 10% values changes the expectation significantly. Imputation reduces this miss-estimation. According to the rating criteria, the Linear Regression procedure (Norm) works best.

rtCopFit	rtCopFit2	rtCopExp	rtCopReg	rtNorm	rtPMM	**rtDel**
4.5	4.5	7	6	2.5	2.5	**1**

Table 5.9: Evaluation of the Runtime.

CopFit	CopFit2	CopExp	**CopReg**	Norm	PMM	Del
5	6	7	**1**	4	2	3

Table 5.10: Overall results **r** for a weighted sum of all results of all 7 tables above, with equal weights for Kendal's tau values and marginal parameters (= 0.16), and a smaller weight for runtime (= 0.04).

In the end the results are very close. Runtime is valued with a parameter of 0.25 according to the others. With a parameter of 0.4 the order of PMM and CopReg would have changed. To observe each method in every situation graphically, have a look at the appendix. There are box plot diagrams, which show each simulation category separately.

Chapter 6

Case Study (4 dimensions)

To see strengths and weaknesses under real conditions, it is crucial to apply the developed methods to real data. In a simulation test, one simply produces what is needed to try out theoretical methods straightforward and under a relatively simple setup, but in reality there are often obstacles not seen if one makes life too elementary. For this purpose a case study is presented in the following, in which a closer look is taken at the candidates in the simulation test ranking. In this case it was decided to analyze a medical survey because this is one of the most common areas of application in the field of imputation. As mentioned before, these studies are often very expensive and time consuming, so throwing collected data away just because one value is missing is not an option. Another reason why a medical study was chosen is the mostly not trivial dependence structure existing between the variables and therefore is very interesting in the topic of missing values. Last but not least, there is nearly always missing data. So the conditions in such an environment are very suitable for applying vine copula imputation techniques.
In a first try, the study is restricted to a 4-*dimensional* environment with continuous marginal distributions, like in the simulation test, to make comparison to earlier chapters possible. Then, if after the simulation study and the real data test one or more vine copula imputation methods prove to be successful, they will be chosen for higher dimensional (and thus much more time consuming) practice. So the first aim is to select four continuous dependent random variables from a medical survey, done by an exploratory data analysis.

6.1 Exploratory Data Analysis

Data was collected for the study of ”Prenatal Lead Exposure and Weight of 0- to 5-Year-Old Children in Mexico City” (see Afeiche A., et al, (2011)), with the Background that ”[c]umulative prenatal lead exposure, as measured by maternal bone lead burden, has been associated with smaller weight of offspring at birth and 1 month of age”, and the objective of ”investigating the association of perinatal maternal bone lead, a biomarker of cumulative prenatal lead exposure, with children's attained weight over time from birth to 5 years of age”. The measurements were done within two groups (A with n=327 and B with n=463) that differ in total number of children and year of birth. In the following only the dataset of group B with (considering only different mothers and children with known sex) a total number of n=363 children ($\#female = 179$ $(49,3\%)$ and $\#male = 184$ (50,7%)) was analyzed.

The data considered has the following variables:

- **Visit**: follow-up visit in months. 0 corresponds to the visit of child at birth. Integer variable with values in the set $\{0, 3, 6, 12, 18, 24, 30, 36, 48, 60\}$.
- **Weight**: longitudinal measure of weight for a child. Continuous variable.
- **Birth-Sex**: sex of a child. Integer variable with values in the set $\{0, 1\} := \{female, male\}$, with $\#female = 179$ $(49,3\%)$ and $\#male = 184$ (50,7%).
- **Child-PB**: lead concentration in child's cord blood at birth. Continuous variable.
- **Rotula**: lead concentration in child's rotula (patella) bone with respect to a benchmark. Continuous variable.
- **Tibia**: lead concentration in child's tibia bone with respect to a benchmark. Continuous variable.
- **Mother-PB**: lead concentration in mother's blood. Continuous variable.
- **Mother-Age**: age of the mother at birth. Continuous variable measured in years.
- **Birth-Gestage**: number of gestational months. Continuous variable measured in month.

Note that only ”Weight” differs with varying variable ”Visit”. The missing values are listed in the table below:

Varying Variable	Missing Values
Weight (0 Month)	0 (0%)
Weight (3 Month)	17 (4,7%)
Weight (6 Month)	12 (3,3%)
Weight (12 Month)	19 (5,2%)
Weight (18 Month)	27 (7,4%)
Weight (24 Month)	28 (7,7%)
Weight (30 Month)	40 (11%)
Weight (36 Month)	52 (14,3%)
Weight (48 Month)	60 (16,5%)
Weight (60 Month)	102 (28%)
Weight (over all)	357 (9,8%)

Table 6.1: Percentage of missing values for the longitudinally measured variable Weight.

Non-varying Variables	Missing Values
Birth-SEX	0 (0%)
Child-PB	92 (25,3%)
Rotula	2 (0,6%)
Tibia	2 (0,6%)
Mother-PB	13 (3,6%)
Mother-Age	0 (0%)
Birth-Gestage	0 (0%)

Table 6.2: Percentage of missing values for all variables which are only measured at birth.

Because only four candidates are needed for simultaneous imputation, it is more interesting to look for some different dependence structures between these variables with missing data. It is easy to recognize that all lead measurements on different parts of the body (mother or child) are highly correlated. Further, by results of the study, ”a 1 standard deviation-increase in maternal patella lead was associated with a 130.9 g decrease in weight (95% CI= -227.4 to -34.4) among females and a 13.0 g non-significant increase in weight among males (95% CI= -73.7 to 99.9) at 5 years of age”. This leads to the conclusion that ”the association was evident for patellar but not tibial lead levels, and limited to females”. So it is highly relevant to investigate the variables ”Child-PB”, ”Mother-PB”, ”Rotula” and ”Weight at 6 month”. The first candidate has a significantly high percentage of missing data and is highly correlated with the second and the third. ”Rotula” and ”Weight”, as mentioned before, have an interesting relationship because of the dependence differences in sex, which ”Tibia” and ”Weight” have not. One also could choose the variable ”Weight” at every other measurement ”Visit”, but the study also mentions ”[p]renatal lead exposure measured by maternal blood lead has been associated with decreases in childrens anthropometry at 6 and 15 months respectively (Schell et al. 2009; Shukla et al. 1989; Shukla et al. 1991)”, and there are just some values missing. That is why those four variables are chosen first for testing imputation methods under real data conditions. The immediately preceding considerations are now displayed in images, in order to convince the reader of their usefulness.

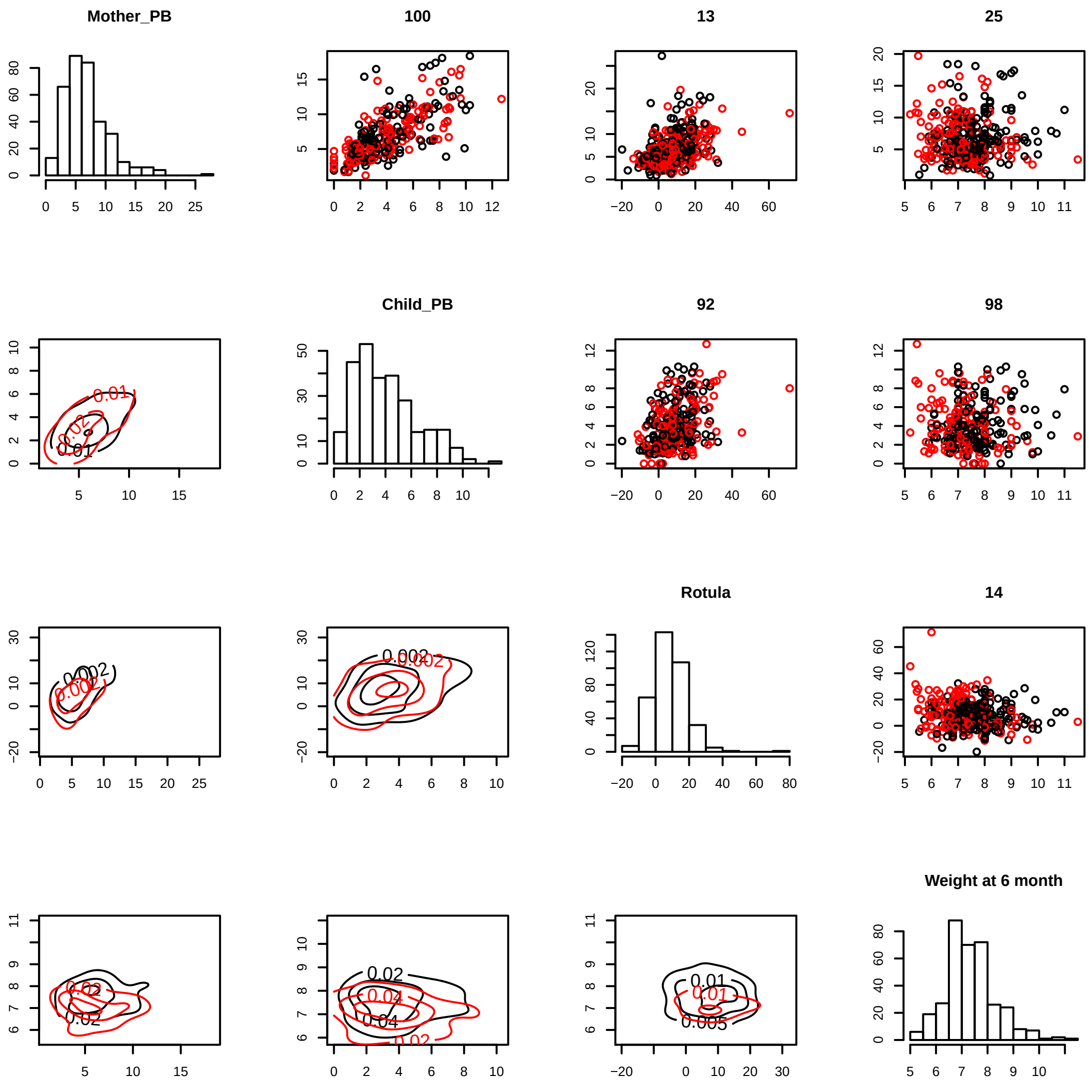


Figure 6.1: The chosen variables, with upper diagonal scatter plots (male=black, female=red dots) and # pairwise missing values, lower diagonal contour plots (male=black, female=red), on the diagonal histograms of the data, with data on X-scale (original).

From the scatter plots and the contours, it can already be guessed that there are interesting dependence structures between the variables varying over sex. The contours on Z-scale might offer further opportunities for interpretation, but more work is needed to get them.

To apply two step vine copula methods, it is crucial to fit appropriate marginal models first to get to uniform distributions. It also helps to understand the dependence structure between the four variables better if there is no marginal effect distorting the pictures.

6.2 Fitting Marginals for Mother-PB, Child-PB, Rotula, Weight at 6 month

Here, this is done by eliminating some other effects (also observed in the study) with a linear regression model. Then the residuals are fitted with a normal, or skewed normal error distribution respectively. In the following, possible effects will be examined to have some linear relationship on the chosen variables. Then the significance of these effects and their interactions will be computed. The goodness of fit will be decided with the help of studentized residuals and qq-plots.

6.2.1 Using complete data for male and female jointly

First, the marginal fit is done using the complete data for male and female. One can observe that marginally there is no difference between genders for the four variables. For simplicity, we introduce

- **BSEX** for Birth-Sex,
- **MAGE** for Mother-Age,
- **BGES** for Birth-Gestege,
- and **T** for Tibia.

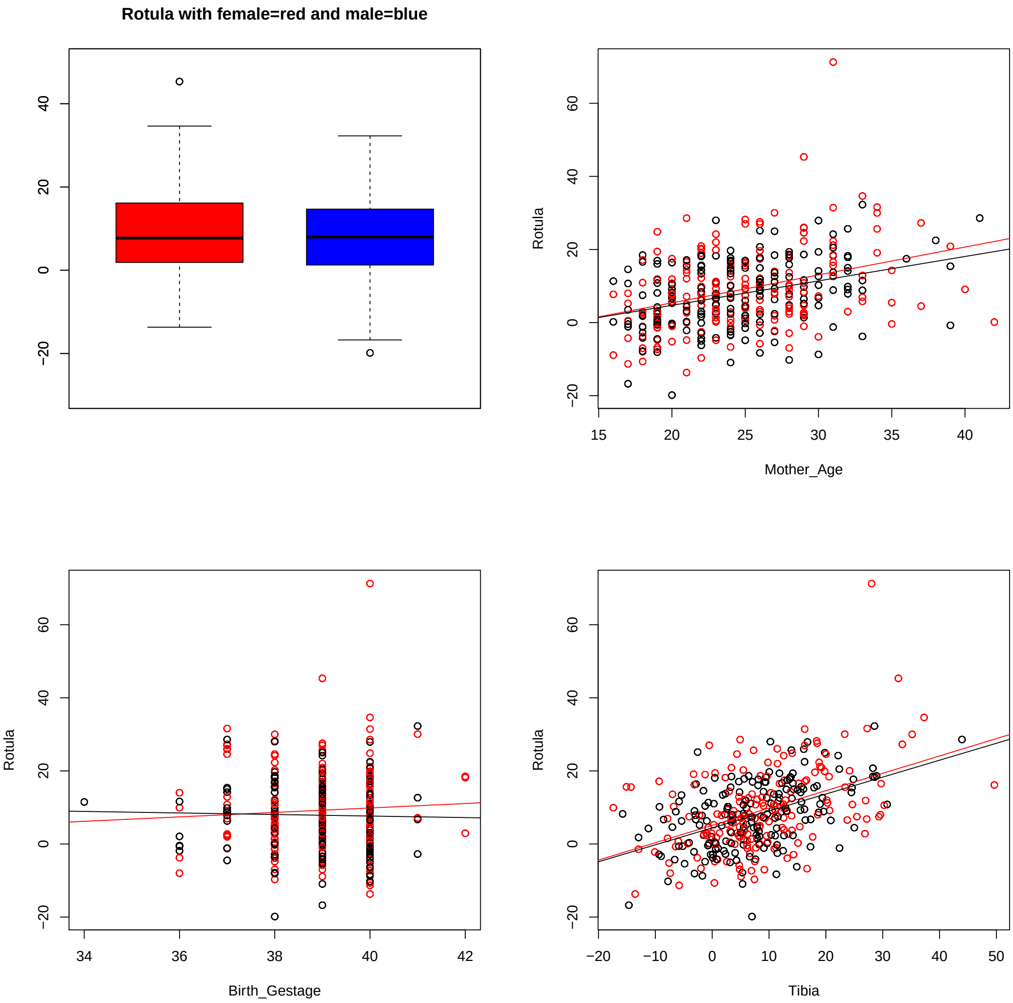


Figure 6.2: Linear "Rotula" effects. Rotula is plotted against Sex (box plot upper left), Mother-Age (upper right), Birth-Gestage (lower left) and Tibia (lower right). The red regression line is done by only considering female (red dots), and the black only considering male (black dots) children. One can see that marginally, "Rotula" has no noticeable effects or interaction effects on the child's gender. However, there seem to be relationships with "Mother-Age" and Tibia.

		Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
H]	1	350	26174.11				
	2	355	26687.83	-5	-513.73	1.37	0.2334

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	5.6882	82.8331	0.07	0.9453
BSEX	46.4045	32.6361	1.42	0.1560
MAGE	-0.7301	3.3516	-0.22	0.8277
BGES	-0.3455	2.1195	-0.16	0.8706
T	-1.6442	1.8653	-0.88	0.3787
BSEX:MAGE	-0.3295	0.1909	-1.73	0.0853
BSEX:BGES	-0.9926	0.8287	-1.20	0.2318
BSEX:T	-0.0659	0.0950	-0.69	0.4887
MAGE:BGES	0.0322	0.0856	0.38	0.7067
MAGE:T	0.0221	0.0081	2.73	0.0066
BGES:T	0.0391	0.0465	0.84	0.4013

Big model for "Rotula" with all bivariate interactions.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-12.9538	16.2918	-0.80	0.4271
MAGE	0.3659	0.1161	3.15	0.0018
T	-0.0661	0.2089	-0.32	0.7517
BGES	0.2441	0.4192	0.58	0.5607
BSEX	-0.9648	0.9216	-1.05	0.2959
MAGE:T	0.0184	0.0079	2.33	0.0201

Smaller model with only the single variables and the interaction of "Mother-Age" and "Tibia" included.
Anova with the result that the smaller model is preferred.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-4.3615	2.8306	-1.54	0.1242
MAGE	0.3824	0.1150	3.32	0.0010
T	-0.0190	0.2041	-0.09	0.9260
MAGE:T	0.0167	0.0077	2.17	0.0310

Smaller model with only "Mother-Age" and "Tibia" and the intersection included.

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	355	26687.83				
2	357	26792.25	-2	-104.42	0.69	0.5000

Anova with the result that the smaller model is preferred.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-7.9142	2.3185	-3.41	0.0007
MAGE	0.5281	0.0938	5.63	0.0000
T	0.4116	0.0464	8.87	0.0000

Smaller model with only "Mother-Age" and "Tibia" included.

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	357	26792.25				
2	358	27144.15	-1	-351.90	4.69	0.0310

Anova with the result that the smaller model is preferred.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	4.7228	0.6097	7.75	0.0000
T	0.4719	0.0470	10.03	0.0000

Model with only "Tibia" included.

The tables show the significance of the effects with respect to the response variable "**Rotula**". The calculated *p-values* for a 99% significance level can be observed, with an adj. R^2 of $0,217$ for only "Tibia" and $0,279$ for both significant variables in the smallest model included.

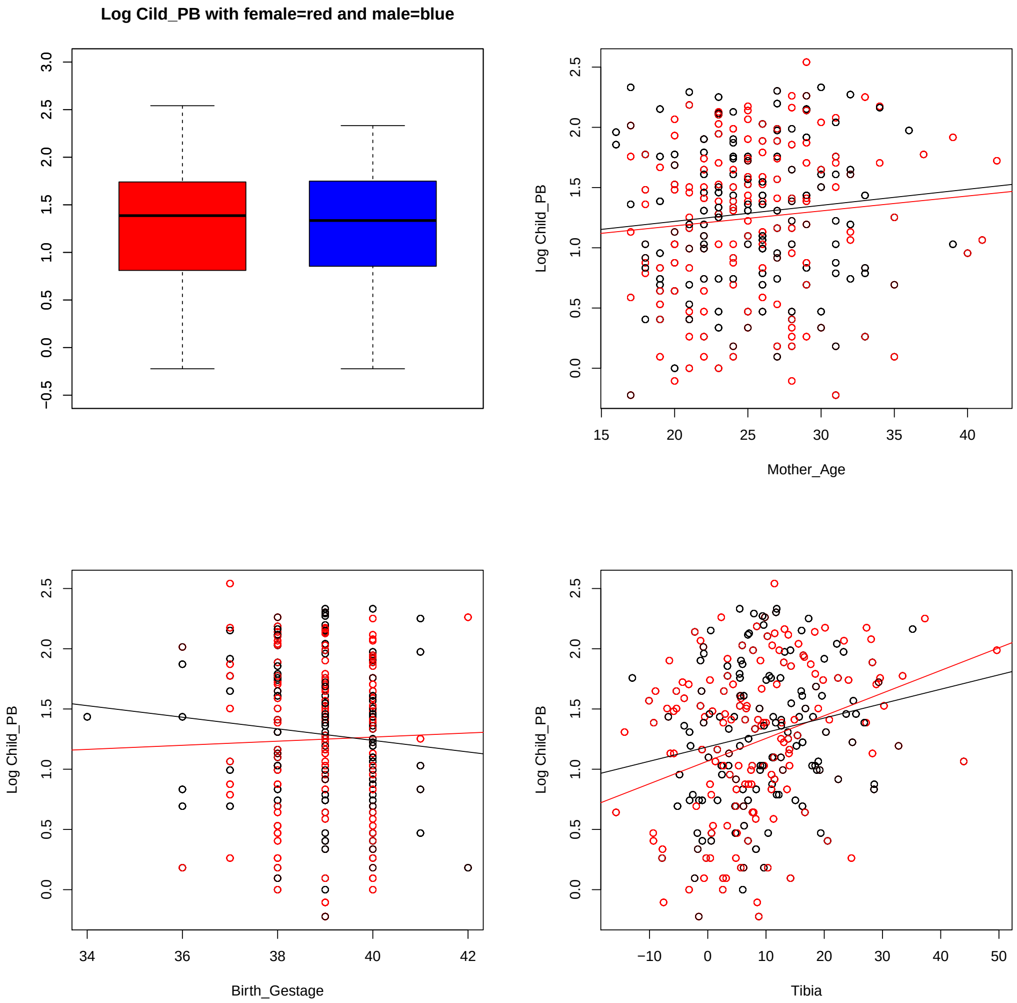


Figure 6.3: Linear Log "Child-PB" effects. Log Child-PB is plotted against Sex (box plot upper left), Mother-Age (upper right), Birth-Gestage (lower left) and Tibia (lower right). The red regression line is done by only considering female (red dots), and the black only considering male (black dots) children. For logarithmically transformed "Child-PB" there should be no single sex effect, either. Interaction effects with the "Birth-Gestage" or the "Tibia" variable are possible. But they turn out not to be significant as one can see in the *p-value* table.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-4.2296	5.9190	-0.71	0.4755
BSEX	3.8911	2.2188	1.75	0.0807
MAGE	0.1054	0.2481	0.42	0.6713
BGES	0.1316	0.1515	0.87	0.3856
T	0.0748	0.1399	0.54	0.5930
BSEX:MAGE	-0.0127	0.0139	-0.91	0.3618
BSEX:BGES	-0.0879	0.0563	-1.56	0.1194
BSEX:T	-0.0081	0.0068	-1.19	0.2365
MAGE:BGES	-0.0023	0.0063	-0.37	0.7151
MAGE:T	-0.0003	0.0006	-0.47	0.6423
BGES:T	-0.0013	0.0035	-0.37	0.7100

Big model for "Child-PB" with all bivariate interactions.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.3476	1.1063	0.31	0.7536
BSEX	0.0723	0.0638	1.13	0.2587
MAGE	0.0063	0.0068	0.93	0.3538
BGES	0.0206	0.0281	0.73	0.4636
T	0.0139	0.0032	4.30	0.0000

Smaller model without interactions.

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	260	70.88				
2	266	72.86	-6	-1.98	1.21	0.3003

Anova with the result that the smaller model is preferred.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.3406	0.0418	32.09	0.0000
T	0.0146	0.0031	4.72	0.0000

Smaller model just with "Tibia" included.

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	266	72.86				
2	269	73.62	-3	-0.76	0.92	0.4315

Anova with the result that the smallest model is preferred.
The tables show the significance of the effects with respect to the response variable Log "**Child-PB**". In the end it is a linear relationship with "Tibia" only, with an adj. R^2 of $0,073$.

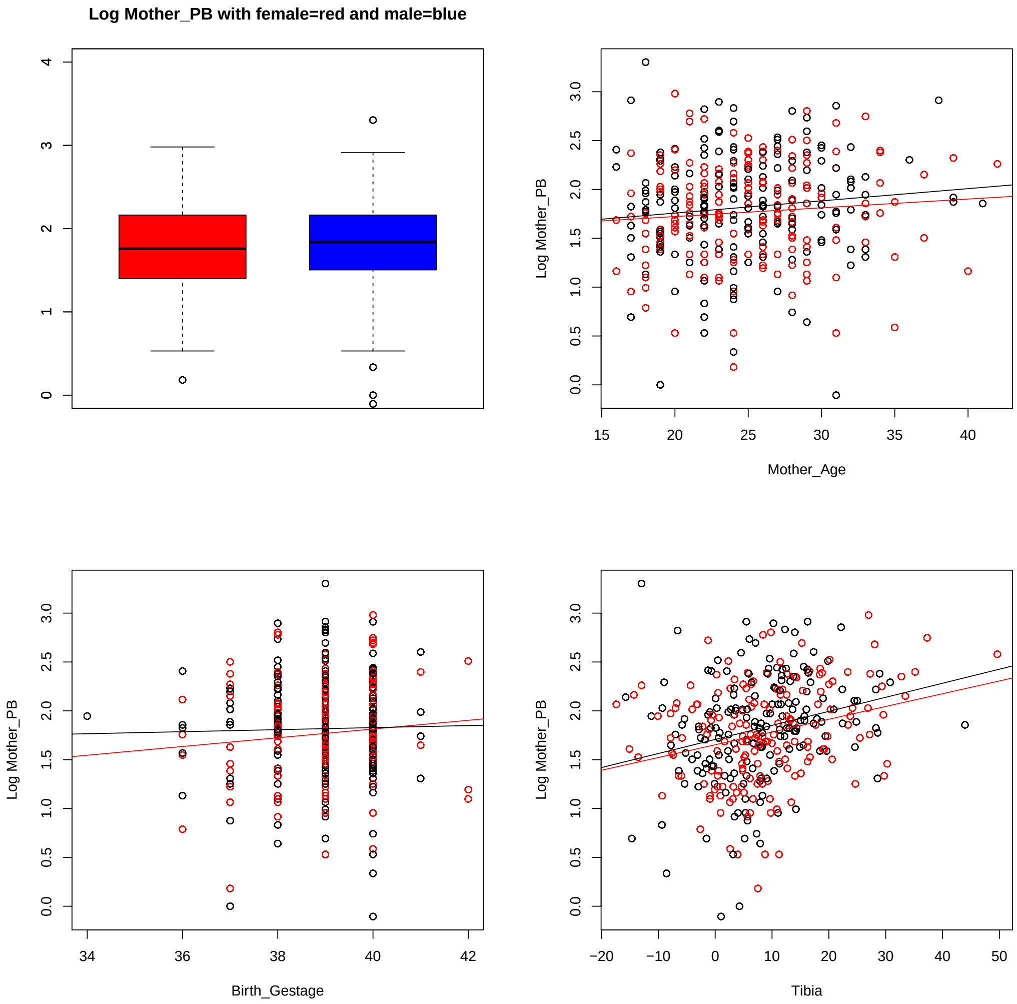


Figure 6.4: Linear Log "Mother-PB" effects. Log Mother-PB is plotted against Sex (box plot upper left), Mother-Age (upper right), Birth-Gestage (lower left) and Tibia (lower right). The red regression line is done by only considering female (red dots), and the black only considering male (black dots) children. The plots of "Mother-PB" are very similar to those of the "Child-PB" variable. That is conditioned on the strong dependency between these two measurements. Intuitively, the fit of the linear model should not be very different. This suggestion is underlined in the pictures.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-6.4863	4.9883	-1.30	0.1944
BSEX	1.6768	1.9717	0.85	0.3957
MAGE	0.3064	0.2018	1.52	0.1298
BGES	0.2054	0.1276	1.61	0.1086
T	-0.1681	0.1122	-1.50	0.1351
BSEX:MAGE	-0.0070	0.0115	-0.61	0.5437
BSEX:BGES	-0.0372	0.0501	-0.74	0.4588
BSEX:T	0.0013	0.0057	0.23	0.8204
MAGE:BGES	-0.0077	0.0052	-1.50	0.1353
MAGE:T	0.0004	0.0005	0.90	0.3691
BGES:T	0.0043	0.0028	1.55	0.1232

Big model for "Mother-PB" with all bivariate interactions.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.6495	0.9778	0.66	0.5070
BSEX	0.0647	0.0554	1.17	0.2438
MAGE	0.0047	0.0056	0.83	0.4054
BGES	0.0228	0.0249	0.92	0.3606
T	0.0130	0.0028	4.68	0.0000

Smaller model without interactions.

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	339	91.29				
2	345	92.43	-6	-1.15	0.71	0.6414

Anova with the result that the smaller model is preferred.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.6821	0.0355	47.43	0.0000
T	0.0135	0.0027	4.98	0.0000

Smaller model just with "Tibia" included.

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	345	92.43				
2	348	93.22	-3	-0.78	0.97	0.4062

Anova with the result that the smallest model is preferred.
The tables show the significance of the effects with respect to the response variable Log "**Mother-PB**", with an adj. R^2 of $0,064$ for the smallest model.

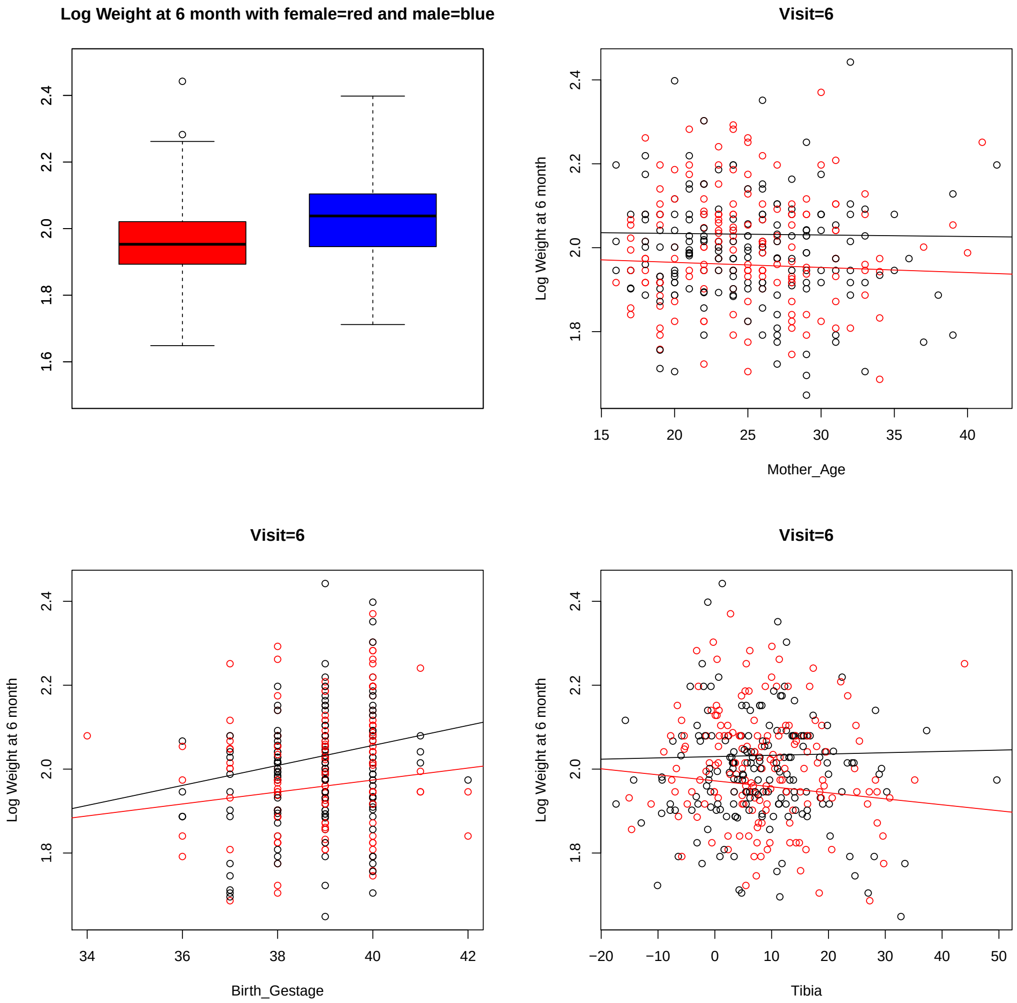


Figure 6.5: Linear Log "Weight at 6 month" effects. Log Weight at 6 month is plotted against Sex (box plot upper left), Mother-Age (upper right), Birth-Gestage (lower left) and Tibia (lower right). The red regression line is done by only considering female (red dots), and the black only considering male (black dots) children. The only optical relationship could be found between "Weight at 6 month" and "Birth-Gestage".

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.5876	1.1964	0.49	0.6236
BSEX	-0.2628	0.4725	-0.56	0.5784
MAGE	0.0254	0.0484	0.52	0.6001
BGES	0.0364	0.0306	1.19	0.2353
T	0.0094	0.0270	0.35	0.7267
BSEX:MAGE	-0.0006	0.0028	-0.22	0.8296
BSEX:BGES	0.0085	0.0120	0.71	0.4794
BSEX:T	0.0021	0.0014	1.49	0.1374
MAGE:BGES	-0.0007	0.0012	-0.55	0.5802
MAGE:T	0.0001	0.0001	0.67	0.5005
BGES:T	-0.0003	0.0007	-0.50	0.6177

Big model for "Weight at 6 month" with all bivariate interactions.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.2168	0.2340	5.20	0.0000
BSEX	0.0708	0.0132	5.35	0.0000
MAGE	-0.0003	0.0014	-0.24	0.8131
BGES	0.0194	0.0059	3.27	0.0012
T	-0.0007	0.0007	-1.04	0.3014

Smaller model without interactions.

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	338	5.17				
2	344	5.24	-6	-0.07	0.74	0.6160

Anova with the result that the smaller model is preferred.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.2108	0.2311	5.24	0.0000
BSEX	0.0717	0.0132	5.43	0.0000
BGES	0.0192	0.0059	3.24	0.0013

Smaller model just with "Birth-Sex" and "Birth-Gestage" included.

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	344	5.24				
2	346	5.26	-2	-0.02	0.66	0.5193

Anova with the result that the smaller model is preferred.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.9589	0.0095	205.25	0.0000
BSEX	0.0733	0.0133	5.50	0.0000

Smaller model just with "Birth-Sex" included.
The tables show the significance of the effects with respect to the response variable Log "**Weight at 6 month**". The gender effect plays a role in the linear regression model fitting of the weight of 6 month old children. The model with just "Birth-Sex" has an adj. R^2 of 0.080 and the model with "Birth-Sex" and "Birth-Gestage" included an adj. R^2 of 0.107.
That results in four different linear models. Note that the significant "Mother-Age" is omitted in the "Rotula" linear regression model, because of two reasons: First, the R^2 coefficient in the bigger model, eliminating this effect, was only a little higher. Therefore the smaller model is preferred. Second, the transformation on uniform $[0, 1]$ scale works nearly the same and looks a little better on the U-graph later on. Because of the same reasons, the significant variable "Birth-Gestage" in the "Weight at 6 month" linear regression model is omitted, too.

- $\log(Weight) = \beta_0 + \beta_1 * Birth\text{-}Sex$,
- $Rotula = \beta_0 + \beta_1 * Tibia$,
- $\log(Mother\text{-}PB) = \beta_0 + \beta_1 * Tibia$,
- $\log(Child\text{-}PB) = \beta_0 + \beta_1 * Tibia$,

with skewed, centered normally distributed "Rotula" residuals and centered normally distributed residuals for the other three marginals. The small adj. R^2 in the models is only an indicator for low relationships between the predictor and the response variables. Always keep in mind, the goal is to fit marginal distributions for uniform transformations here.

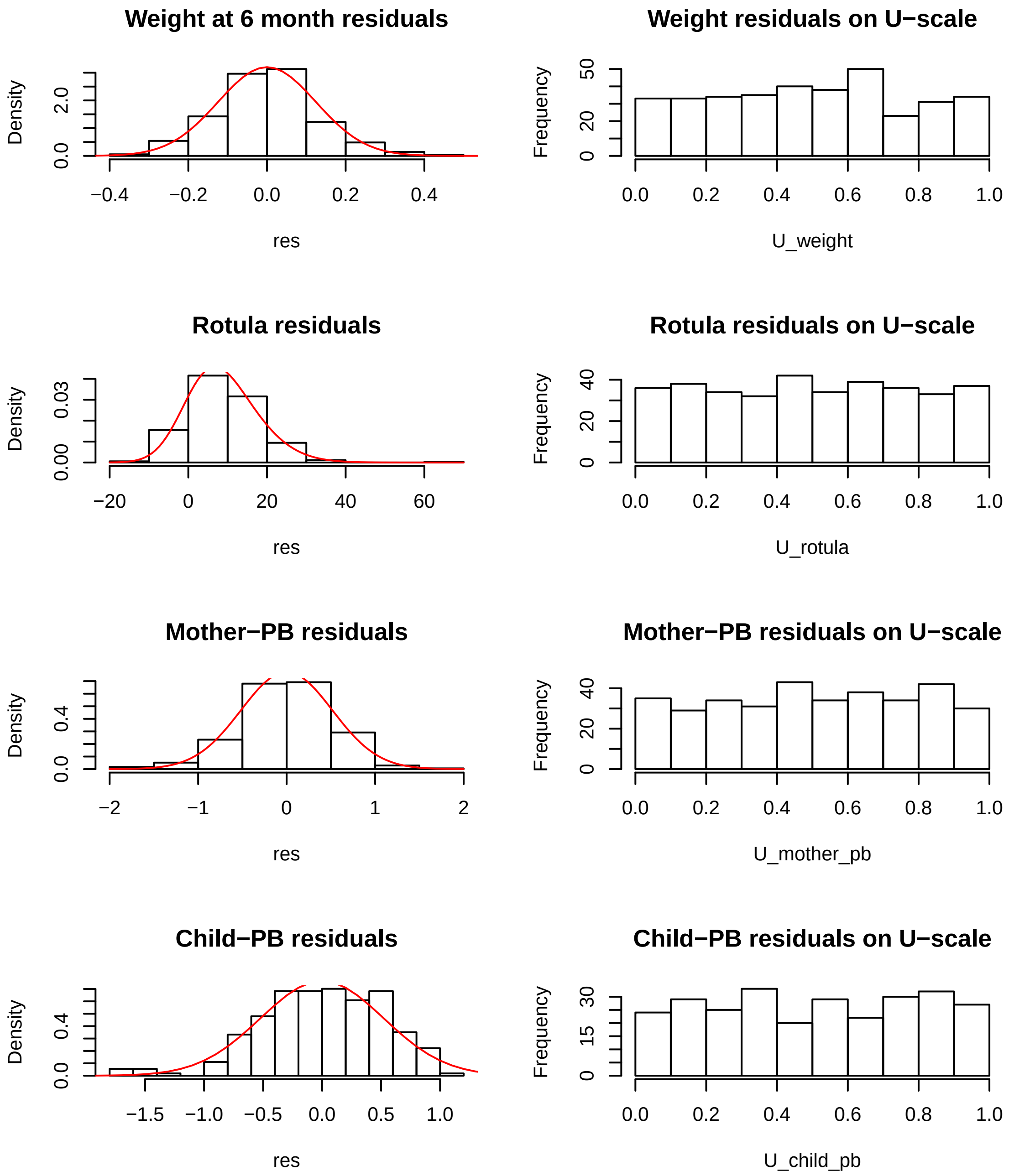


Figure 6.6: Residuals of the four linear models transformed to uniform distributions on $[0, 1]$ using the complete data for male and females.

If both sex is taken in common to fit the four different marginal distributions, the data looks uniform enough after transformation. The quality of the marginal model fit is also seen in linearity of qq-plots,

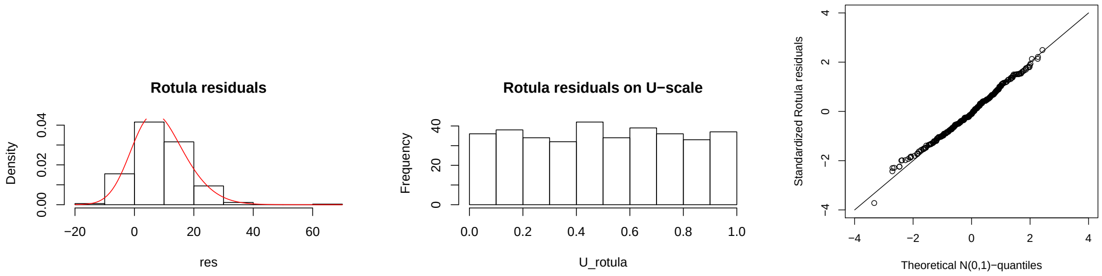


Figure 6.7: The smaller model for ”Rotula” with Mother-Age effect, i.e. Rotula∼Tibia.

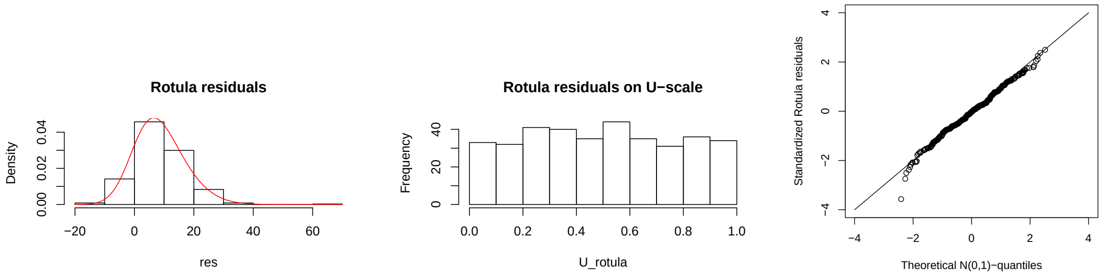


Figure 6.8: To compare, the bigger model for ”Rotula” without Mother-Age effect, i.e. Rotula∼Mother-Age+Tibia.

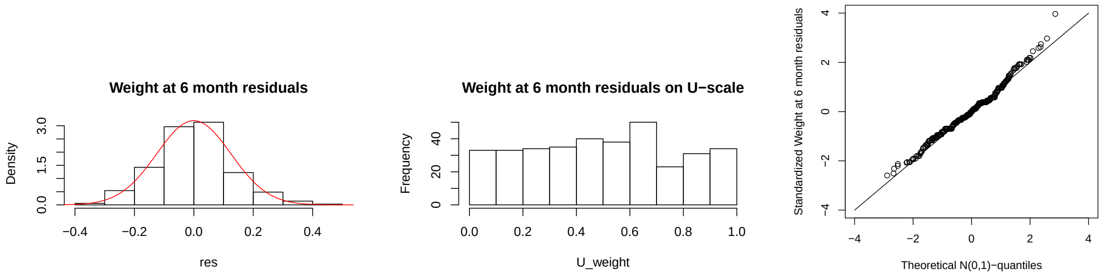


Figure 6.9: The smaller model for ”Weight at 6 month” with Birth-Gestage effect, i.e.Weight∼Birth-Sex.

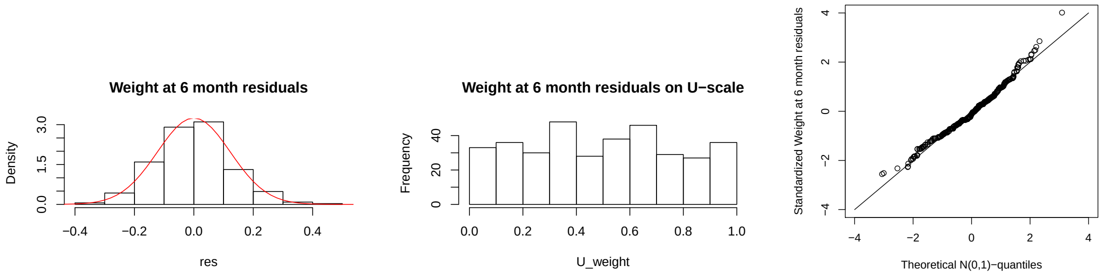


Figure 6.10: To compare, the bigger model for ”Weight at 6 month” without Birth-Gestage effect, i.e. Weight∼Birth-Sex+Birth-Gestage.

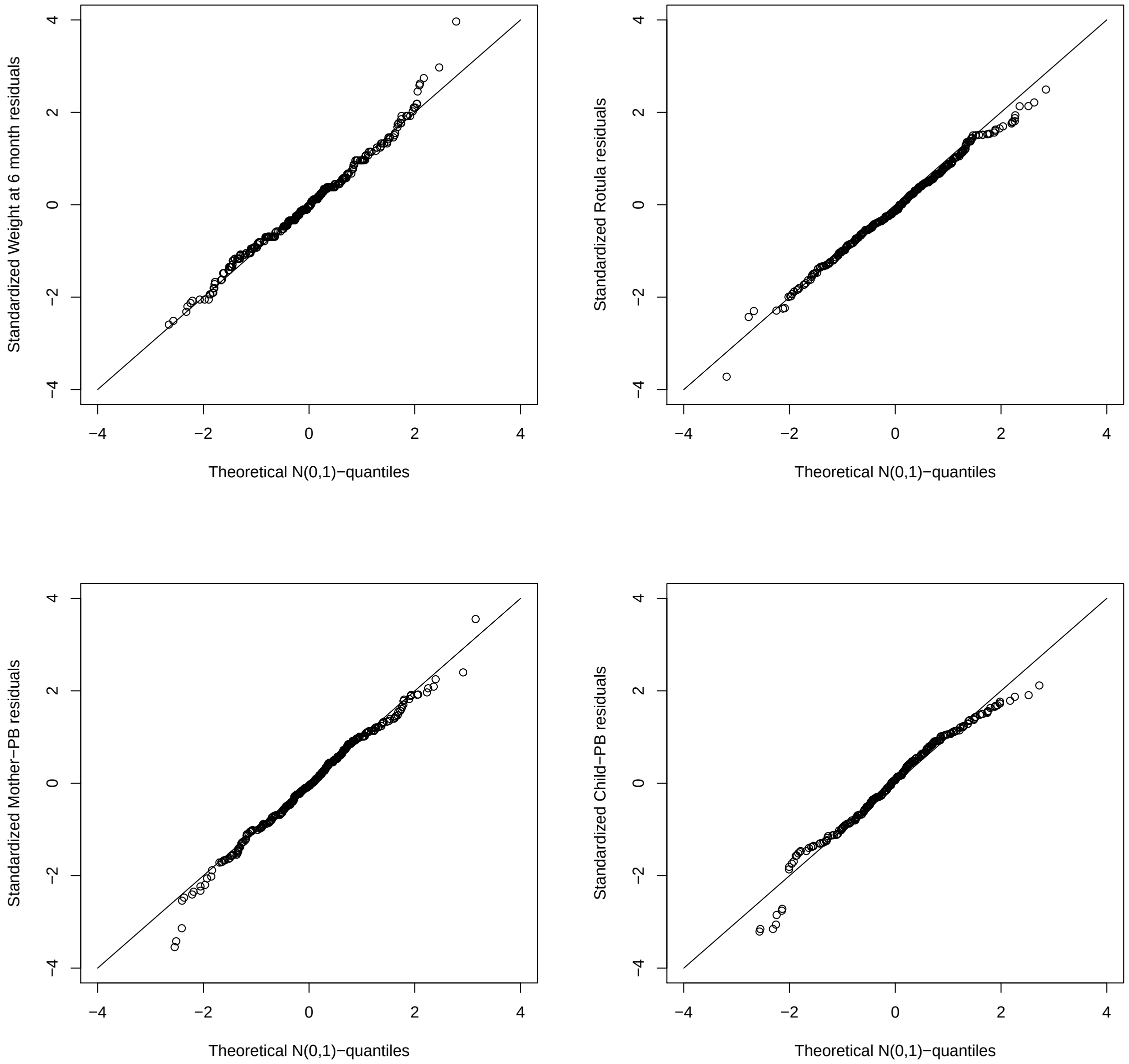


Figure 6.11: QQ-plots for the chosen models using the complete data for male and female.

and the randomness in the studentized residuals without many outliers.

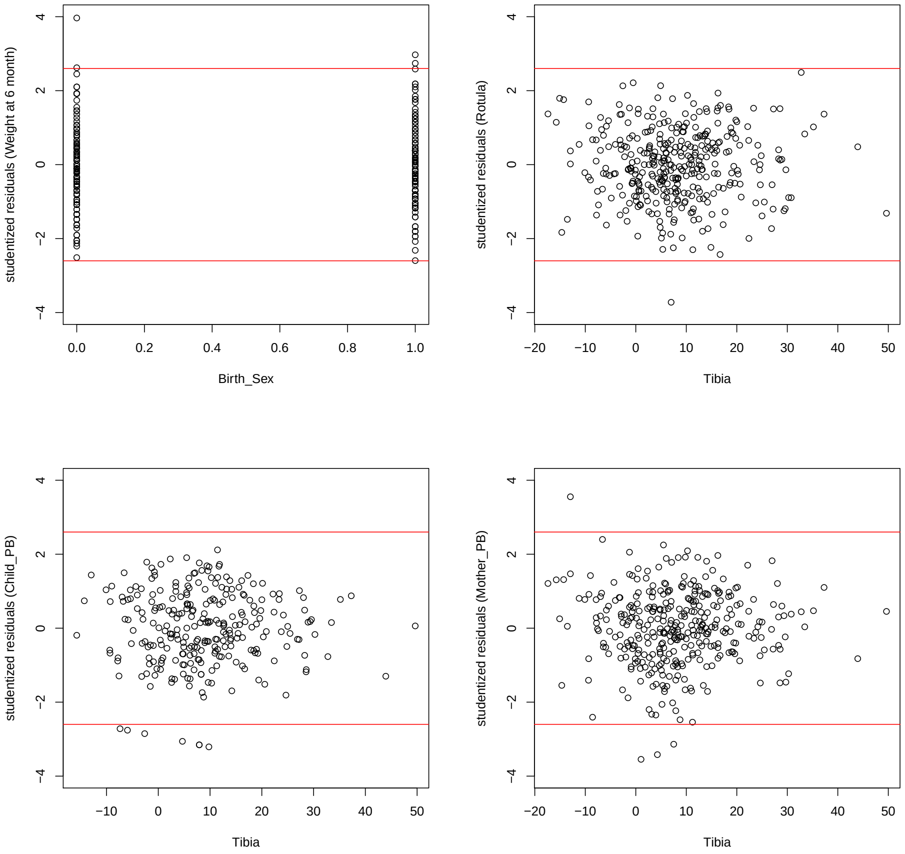


Figure 6.12: Studentized residual-plots for the chosen variables in the linear regression models, using the complete data for male and female. Note that in the linear regression model for the marginal "Weight at 6 month", only Birth-Gestage was decided to be a predictor variable, while in the other three linear regression models, only Tibia was decided to have a linear effect (upper left Weight at 6 month, upper right Rotula, lower left Child-PB and lower right Mother-PB).

6.2.2 Using complete data for male and female separately

Next, the marginal transformations for both sexes are added separately. After finding out that the dependence structure distinguishes between female and male, but marginally there are only differences for one single variable (Weight at 6 month), it should be enough to take only one marginal linear regression model for both sexes for "Rotula", "Mother-PB" and "Child-PB", which does not separate the gender. However, it does not make a big difference separating marginally between genders, since the parameters are nearly the same except for "Rotula". The residuals are fitted via a skewed normal distribution and those parameters differ a lot between male and female children. For this variable there could be parameters used for each sex, but in the Studentized residuals plot, one can observe that the fit is also well chosen when there is no differentiation between sexes. Here, the separation is done in every marginal because there is enough data to do so, it makes programming life a little easier, and the changes are insignificant.

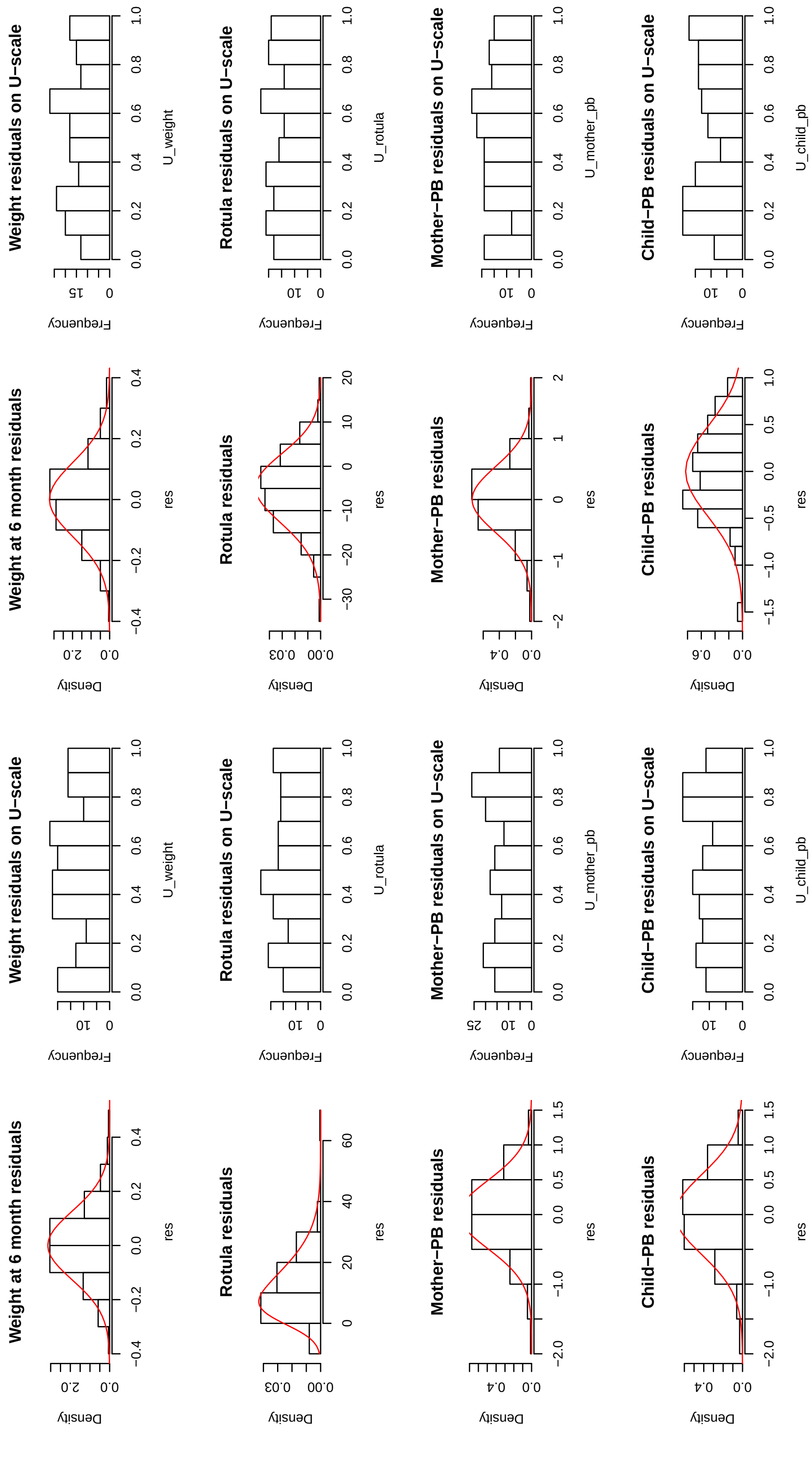


Figure 6.13: Residuals of the four linear models transformed to uniform distributions on [0, 1] left for female only and right for male only.

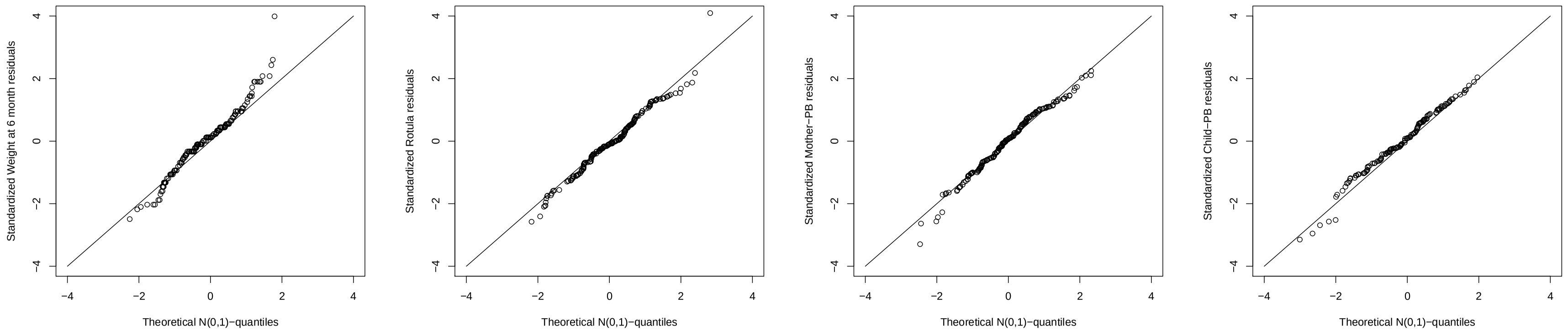


Table 6.3: QQ-plots for the chosen models for female only.

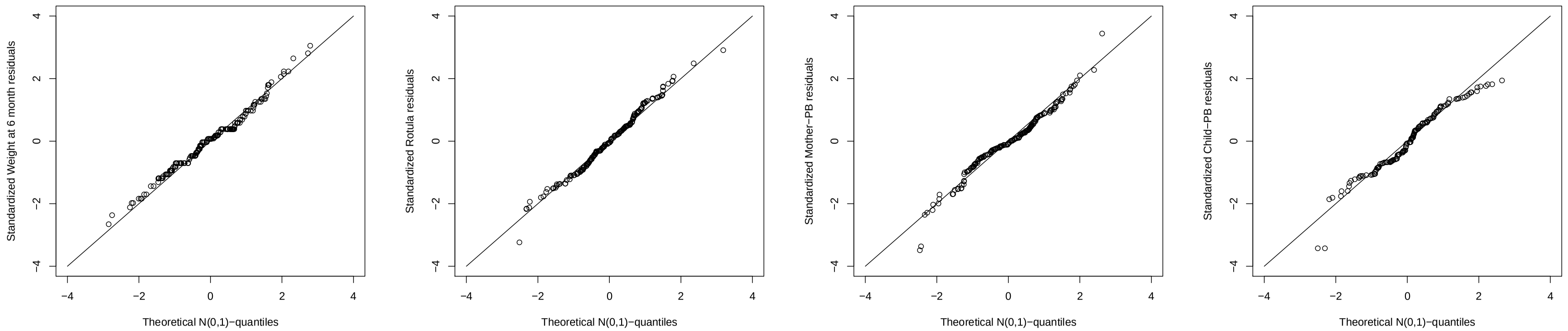


Table 6.4: QQ-plots for the chosen models for male only.

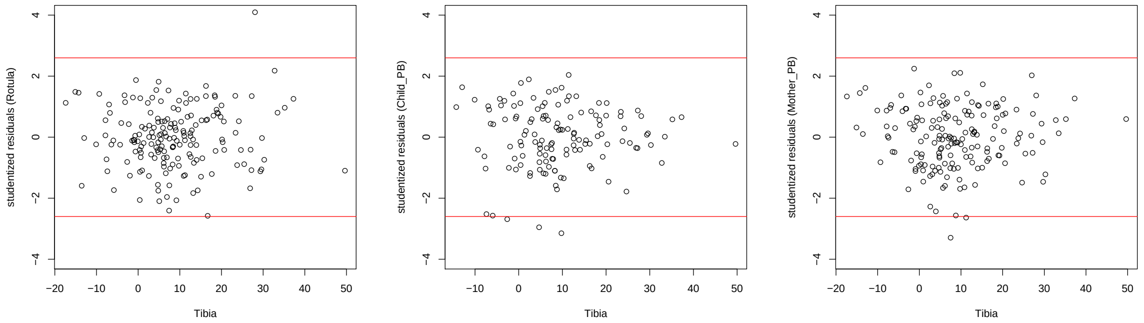


Figure 6.14: Studentized residual-plots for the chosen random effects in the models with female children only.

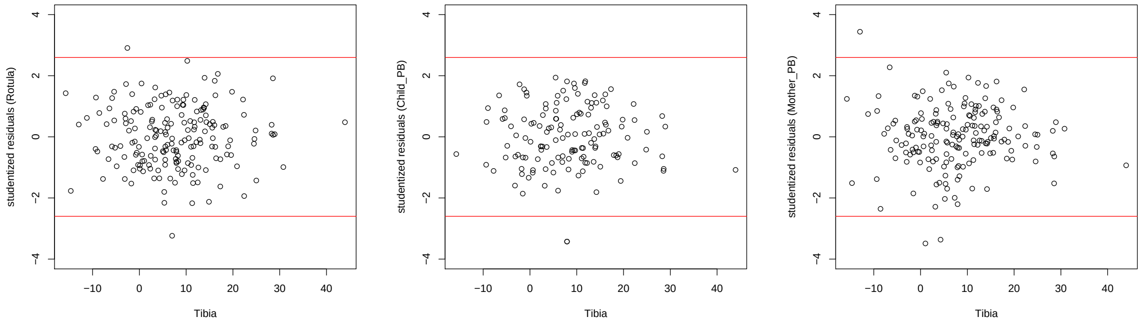


Figure 6.15: Studentized residual-plots for the chosen random effects in the models with male children only.

After changing to univariate uniform distribution, it is practical to look once again at the depencence structure without any marginal effects that might distort some of the structures displayed some pages before.

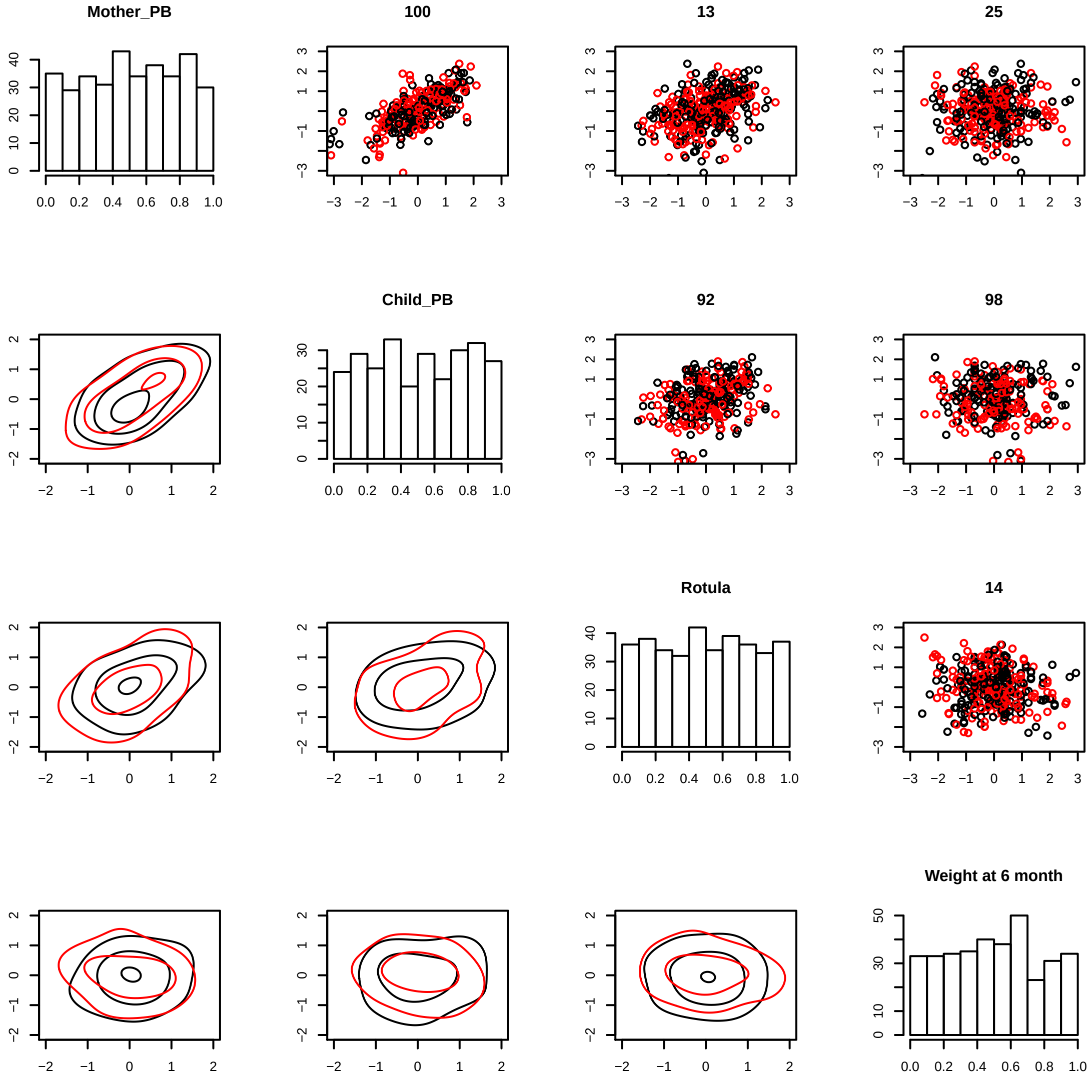


Figure 6.16: The chosen variables, with upper diagonal scatter plots on Z-scale (male=full dots, female=empty dots) and # pairwise missing values, lower diagonal contour plots on Z-scale (male=black, female=red), on the diagonal histograms of the data on U-scale.

	Female	Male	Difference
Weight at 6 month, Mother-PB	−0.13	0.11	0.24
Weight at 6 month, Child-PB	−0.13	0.07	0.2
Weight at 6 month, Rotula	−0.12	−0.01	0.11
Mother-PB, Child-PB	0.54	0.47	0.07
Mother-PB, Rotula	0.3	0.26	0.04
Child-PB, Rotula	0.23	0.19	0.04

Table 6.5: Kendall's tau values for every bivariate combination of the four variables for female and male separately, using the complete data.

Like the Kendall's tau values for the variable ”Weight at 6 month”, the gender difference is noticeable in the Z-scaled contours. There is a twist between male and female contours for this variable.

6.3 Imputation

After analyzing the data, it makes sense to impute nonresponse for female and male children separately. For comparison, the case of just one model for both sexes is added. To evaluate the level of success, there will be two measurements. First, the Kendall's tau of all (in total 6) bivariate cases will be computed before and after imputation. The same will be done for the Kendall‘s tau of the families of the best fitting C-vine structure (with highest AIC). A box plot diagram with 20 tries of each imputation method will show the discrepancy between the measurements with and without imputation values. The smaller the change, the better the method. But Kendall' tau is only a measure of the strength and of whether there is positive or negative dependence, not of what it looks like. So second, non varying contour plots (or mean contour plots) are a good sign for not changing the dependence structure in sense of the copula family. The contours will be plotted for the bivariate cases only. Note that for the linear regression method, the marginals were transformed to standard normals and there was no transformation for the PMM. After Imputation every marginal was standard normal transformed, using the linear models, to get comparable results.

6.3.1 Both Sexes

First, there are the best fitting C-vine trees, estimated for the vine copula imputation methods. Note that only Copula Regression Imputation needs more than one tree estimation. The edges are labeled with the estimated bivariate pair copula family and the corresponding empirical Kendall's tau value.

C-vine	Order	AIC	BIC	# Parameter
1)	(4,1,2,3)	216.7	191.7	7
2)	(1,3,2,4)	214.3	189.5	7
3)	(1,2,3,4)	214.3	189.5	7
4)	(1,4,2,3)	214.1	189.3	7
5)	(4,2,3,1)	209.3	188.1	6
6)	(4,3,1,2)	206.7	181.7	7
7)	(2,4,1,3)	206.7	185.4	6
8)	(2,3,1,4)	206.2	184.9	6
9)	(2,1,3,4)	206.2	184.9	6
10)	(3,2,1,4)	203.5	178.7	7
11)	(3,4,1,2)	203.4	178.6	7
12)	(3,1,2,4)	203.4	178.6	7

Table 6.6: AIC, BIC and number of parameters for the C-vine tree structures with (1,2,3,4) denotes the vector $(U_{Mother-PB}, U_{Child-PB}, U_{Rotula}, U_{Weight})$ and e.g. (2,1,3,4) the vector $(U_{Child-PB}, U_{Mother-PB}, U_{Rotula}, U_{Weight})$.

Using data for both sexes, C-vine model for CopFit and CopExp. It is also used as one of the models in CopReg.

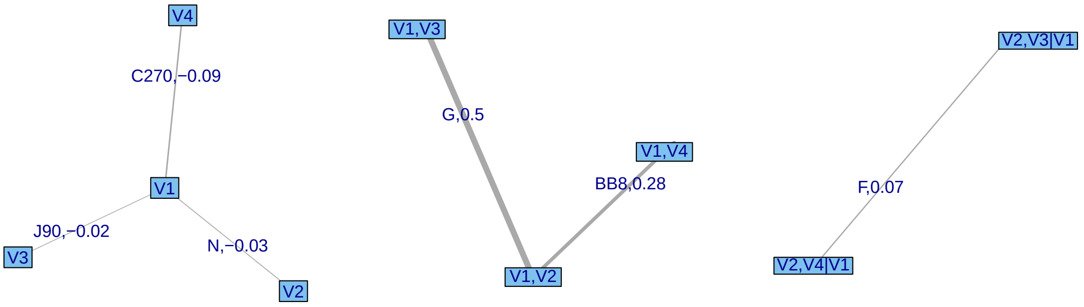


Figure 6.17: 1) Best fitting C-vine structure (with highest AIC under all C-vines) estimated for Copula Fitting Imputation and Copula Expectation Imputation. Also estimated to impute ”**Rotula**” and/or ”**Child-PB**” in method Copula Regression. (Weight=V1, Mother-PB=V2, Child-PB=V3 and Rotula=V4).

Using data for both sexes, C-vine model for CopReg for the remaining single variables missing.

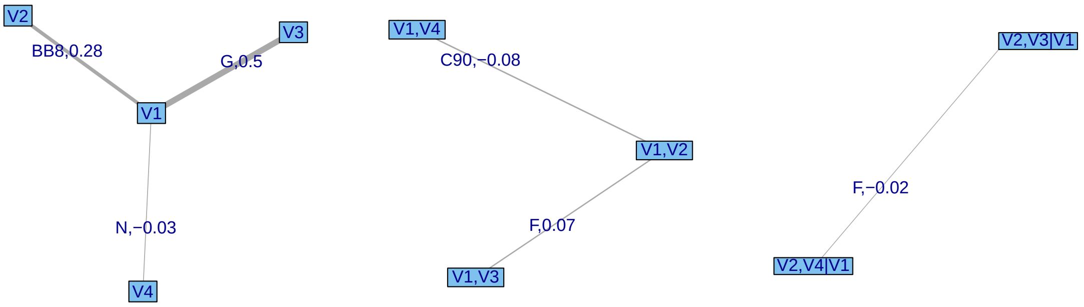


Figure 6.18: 2) Best fitting C-vine structure (with highest AIC under all admissible C-vines) estimated to impute ”**Weight**” in method Copula Regression. (Mother-PB=V1, Rotula=V2, Child-PB=V3 and Weight=V4).

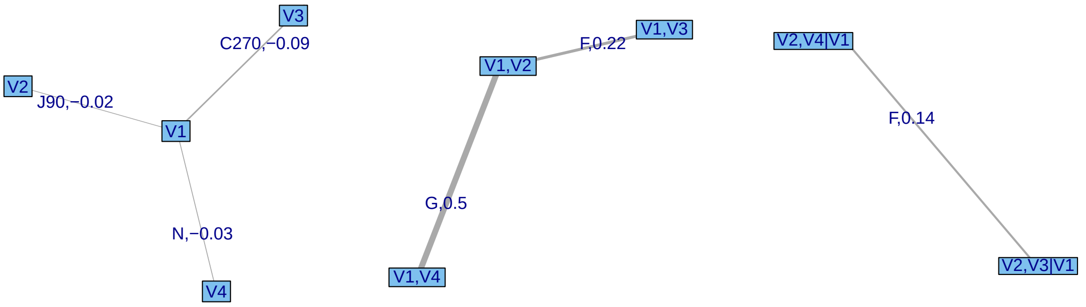


Figure 6.19: 5) Best fitting C-vine structure (with highest AIC under all admissible C-vines) estimated to impute "**Mother-PB**" in method Copula Regression. (Weight=V1, Child-PB=V2, Rotula=V3 and Mother-PB=V4).

Using data for both sexes, C-vine model for CopReg for the remaining two missing variables.

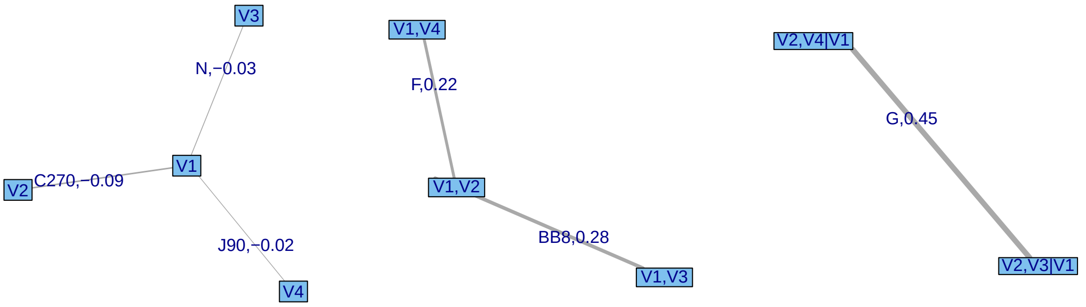


Figure 6.20: Best fitting C-vine structure (with highest AIC under all admissible C-vines) estimated to impute "**Mother-PB**" and "**6) Child-PB**" in method Copula Regression. (Weight=V1, Rotula=V2, Mother-PB=V3 and Child-PB=V4).

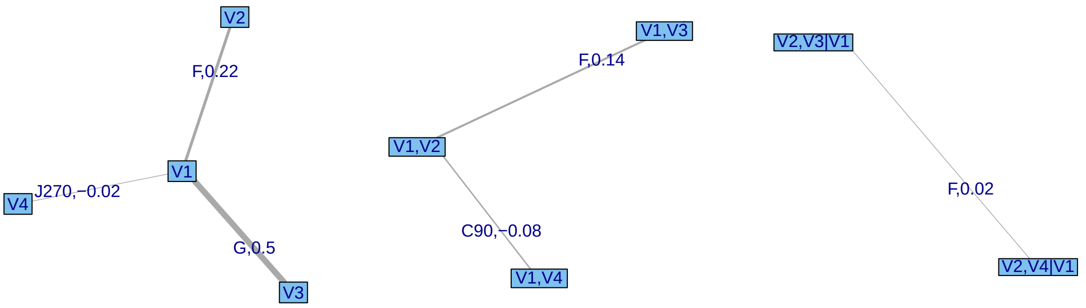


Figure 6.21: Best fitting C-vine structure (with highest AIC under all admissible C-vines) estimated to impute "**Mother-PB**" and "**8) Weight**" in method Copula Regression. (Child-PB=V1, Rotula=V2, Mother-PB=V3 and Weight=V4).

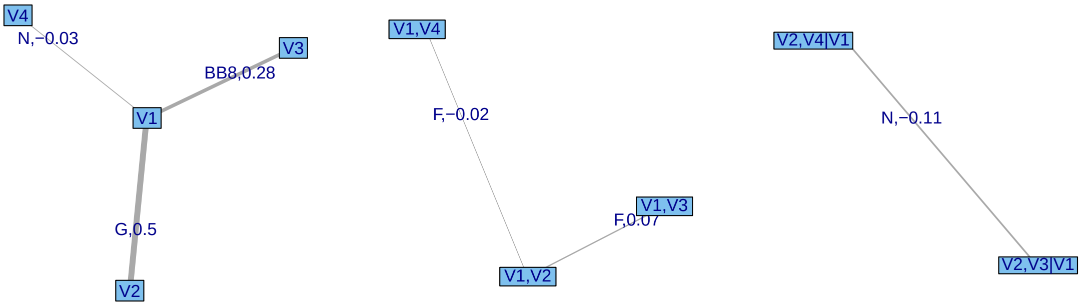


Figure 6.22: 3) Best fitting C-vine structure (with highest AIC under all admissible C-vines) estimated to impute "**Weight**" and "**Rotula**" in method Copula Regression. (Mother-PB=V1, Child-PB=V2, Rotula=V3 and Weight=V4).

As seen, one really has to fit several models in the Vine Copula Regression approach. They differ in the tree structure as well as in the bivariate copulae. It is important to have a range of bivariate copula families available which have some different variability features and different dependence structures. Otherwise, the fit is very restricted, because this method has less options in choosing the tree structures for the different imputation combinations.

Evaluation of the different imputation methods, using Kendall’s tau values.

For every box plot diagram a range of $0,4$ is chosen. The red line is the value of complete cases only. It does not change when repeating the imputation methods. As mentioned, the box plots are of 20 tries each. At first there will be the ”both sexes in common” scenario, where there will not be a differentiation between male and female. Then the separation case will be presented.

Data	Kendall’s tau in C-vine	Figure number
female&male	female	6.23
female&male	male	6.24
female	female	6.41
male	male	6.42

Data	pairwise Kendall’s tau	Figure number
female&male	female	6.25
female&male	male	6.26
female	female	6.43
male	male	6.44

Table 6.7: Overview of the following figures with box plot diagrams of Kendall’s tau values.

Because of the C-vine structure, the first plot in Figure 6.23 and 6.24 captures most of the dependence between ”lead measurements” and ”weight”. Therefore this plot shows best how important it is to separate between genders before imputation is applied. In the female plot (see Figure 6.23), there is an average mismatch of almost $0,1$ between with and without imputation independent of the method. And even worse, the sign has changed from negative to positive dependence exept for one method (). That means the nonresponse is filled out with clearly wrong data. Copula regression Imputation captures the misspecified model in the first plot of Figure 6.23 and 6.24 best, but that leads in more errors in the last (conditioned) copula family parameter that is connected to the same ”lead measurements-weight” dependence. In the male case (see Figure 6.24), nearly the same problem occurs, but the other way around. Not surprising, because on average the models are right.

A more intuitive and interpretable way is to look at the bivariate Kendell’s tau, without conditioning, only (see Figure 6.25 and 6.26). In the female case (see Figure 6.25), the PMM method does better than the others, but therefore gets worse with the male imputation (see Figure 6.26). Over all, high deviations from the complete cases can be observed.

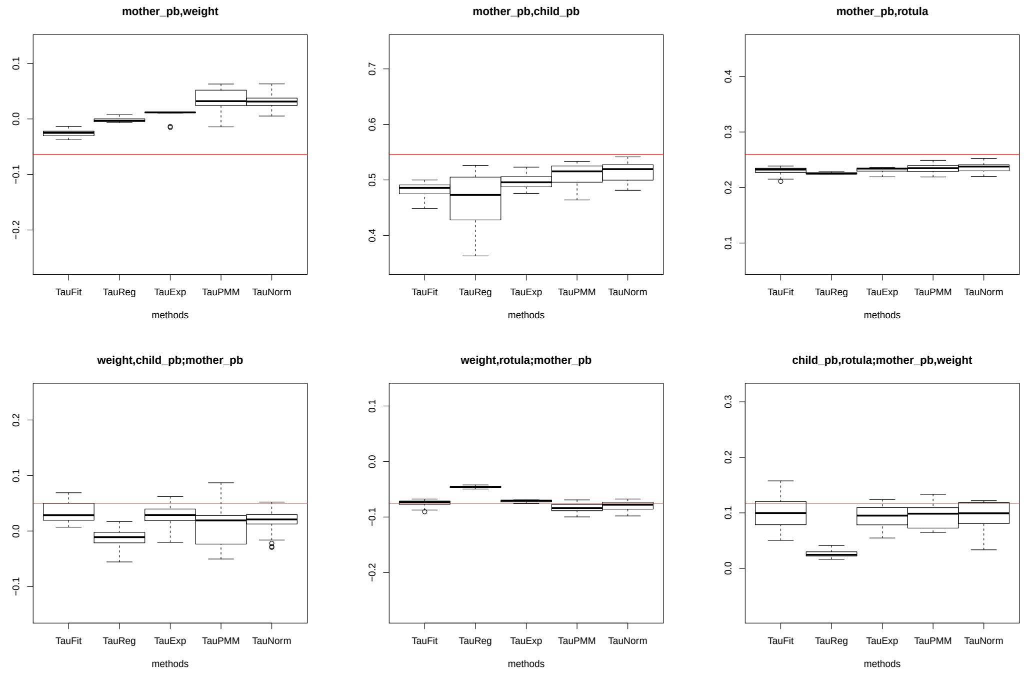


Figure 6.23: Female Kendall's Tau for the families in the best fitting C-vine structure, according to the AIC criteria, using female & male complete data after 20 imputations using 6 imputation methods (the red line is deletion, i.e. complete case). Range of Kendall's tau values = 0.4.

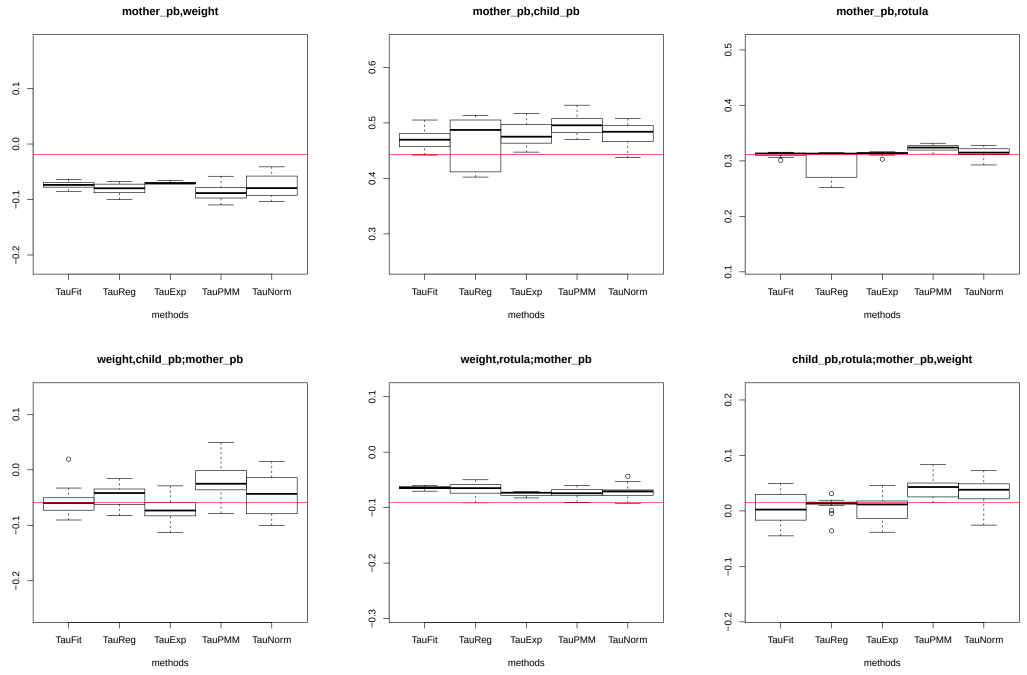


Figure 6.24: Male Kendall's Tau for the families in the best fitting C-vine structure, according to the AIC criteria, using female & male complete data after 20 imputations using 6 imputation methods (the red line is deletion, i.e. complete case). Range of Kendall's tau values = 0.4.

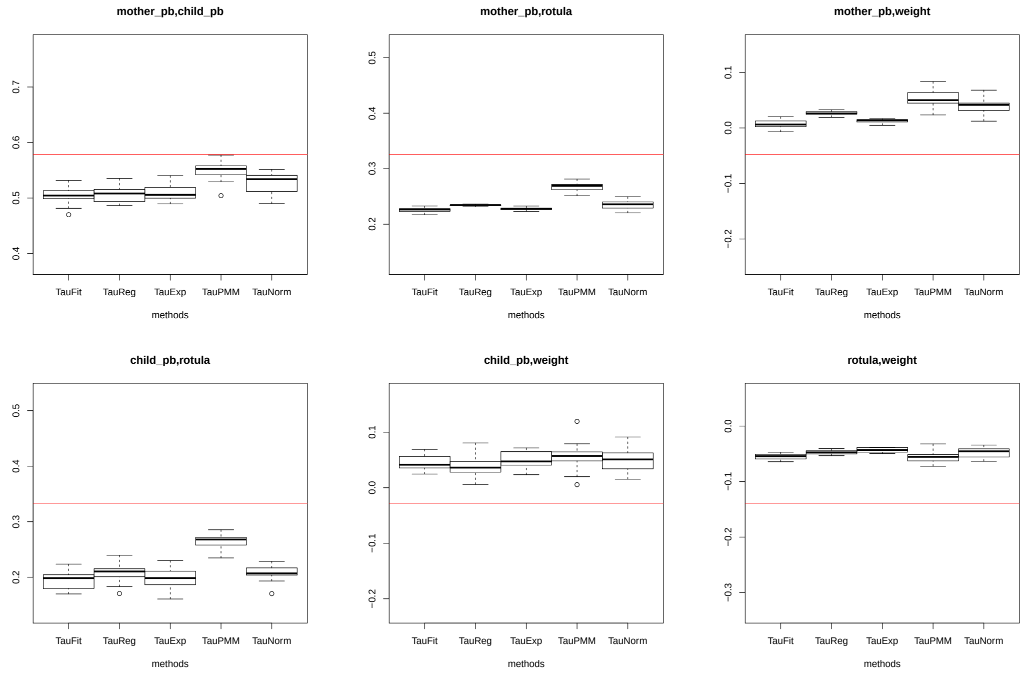


Figure 6.25: Female empirical Kendall's Tau for all possible bivariate combinations without conditioning, using female & male complete data after 20 imputations using 6 imputation methods (the red line is deletion, i.e. complete case). Range of Kendall's tau values = 0.4.

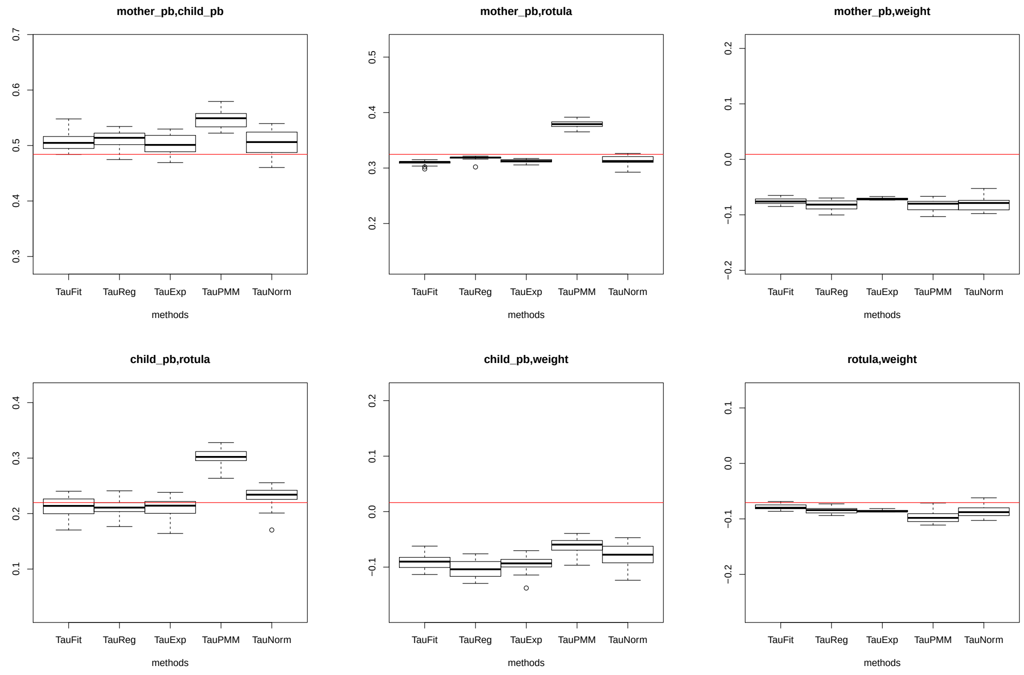


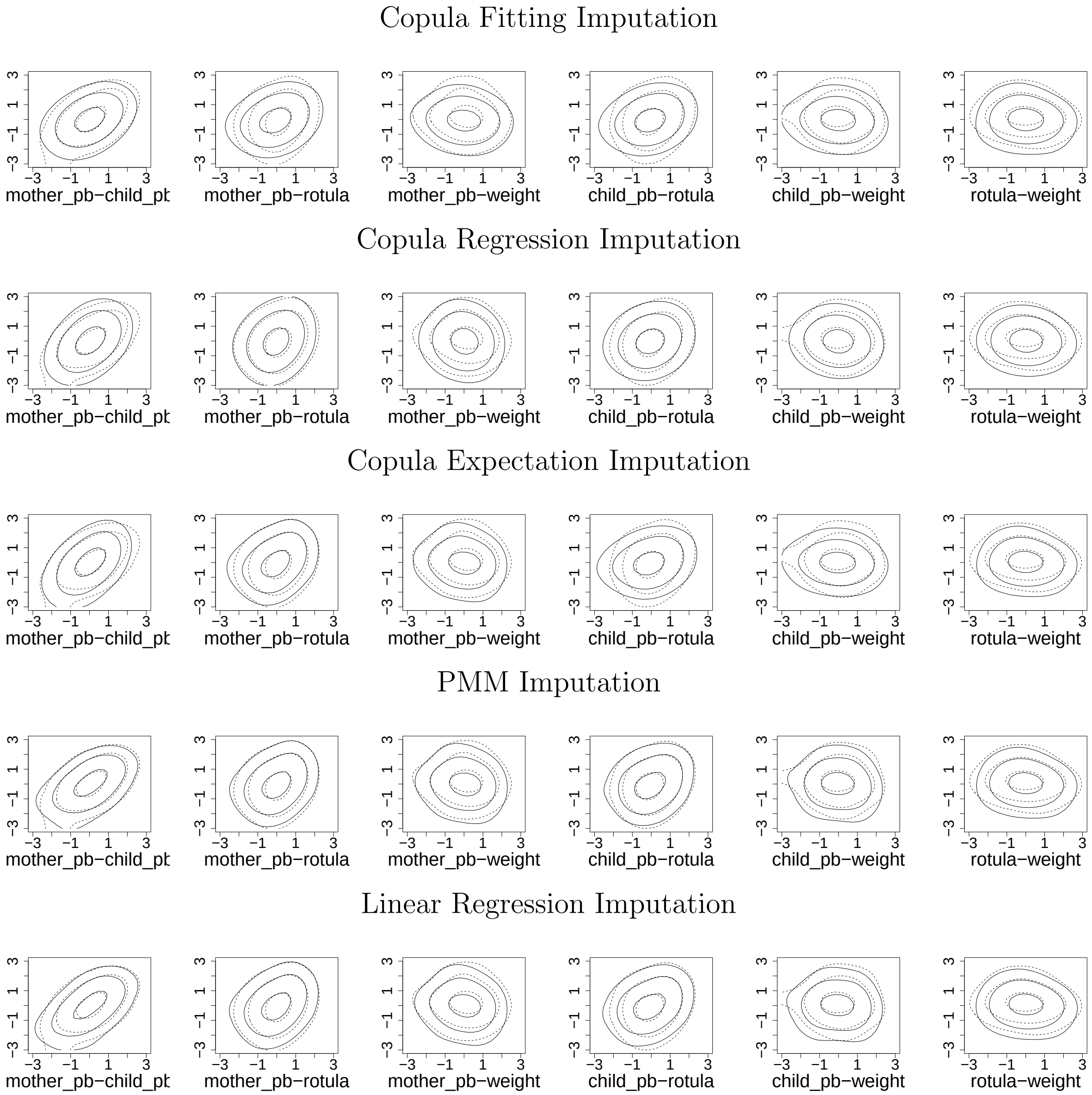
Figure 6.26: Male empirical Kendall's Tau for all possible bivariate combinations without conditioning, using female & male complete data after 20 imputations using 6 imputation methods (the red line is deletion, i.e. complete case). Range of Kendall's tau values = 0.4.

Evaluation of the different imputation methods, using mean contour plots.

Data	Contour	Figure number
female&male	female	6.27
female&male	male	6.28
female	female	6.45
male	male	6.46

Table 6.8: Overview of the following figures with mean contour plots after imputation.

Now the mean contour plots (on the levels 50%, 75%, 95%) for the six bivariate combination possibilities. The dashed lines are the complete cases. They are added for simpler comparison. The procedure for mean contour plots is presented in the appendix.



Copula Fitting Imputation

Copula Regression Imputation

Copula Expectation Imputation

PMM Imputation

Linear Regression Imputation

Figure 6.27: Pairwise female empirical mean contours, using imputed data (20 tries). The dashed contours are the complete cases.

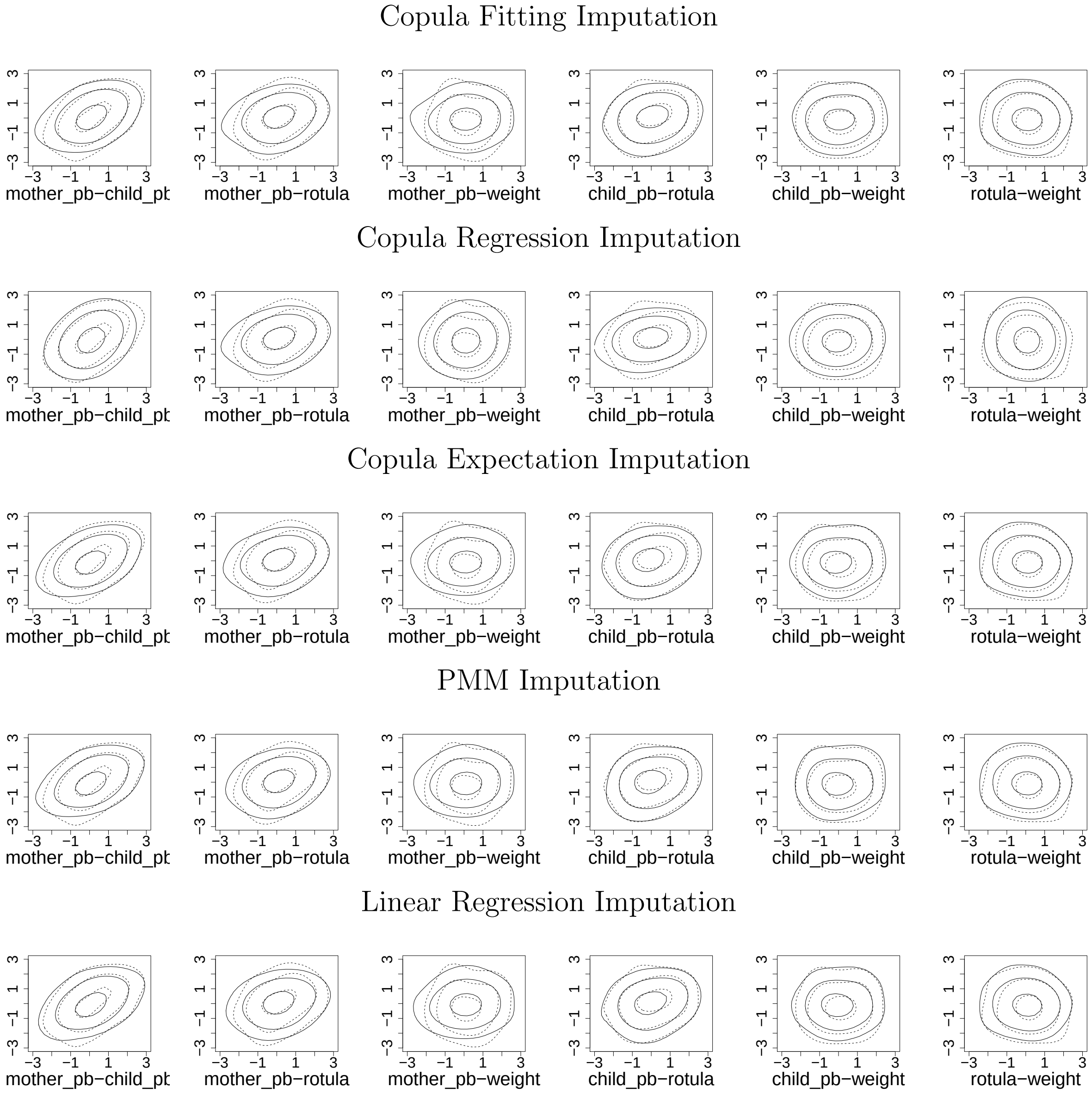


Figure 6.28: Pairwise male empirical mean contours, using imputed data (20 tries). The dashed contours are the complete cases.

The pairwise contours with ”Mother-PB” (first three) seem to be difficult, especially ”Mother-PB” and ”Weight”. In the contours of female children (see Figure 6.27), it looks like the last two methods can compensate the gender problem more effectively. In the male contours (see Figure 6.28), however, CopReg fits better than the rest. As guessed, one has to separate between sexes before imputation.

6.3.2 Separated Sex

Next there are the results with separating gender in two different models and applying the methods for each. The results should be much closer to reality and maybe some methods do better than before. Interestingly, but foreseeably, the highest AIC C-vine structures do not change, but the best fitting bivariate copulae and the according parameter are different according to the model without the separation.

Using female data, C-vine model for CopFit and CopExp. It is also used as one of the models in CopReg.

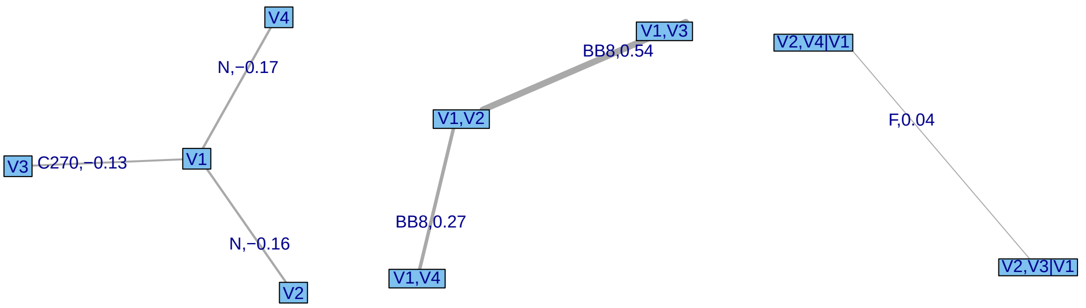


Figure 6.29: Female C-vine structure (with highest AIC under all C-vines) estimated for Copula Fitting Imputation (CopFit) and Copula Expectation Imputation (CopExp). Also estimated to impute "**Rotula**" and "**Child-PB**" in method Copula Regression Imputation (CopReg). (Weight=V1, Mother-PB=V2, Child-PB=V3 and Rotula=V4)

Using female data, C-vine model for CopReg for the remaining single variables missing.

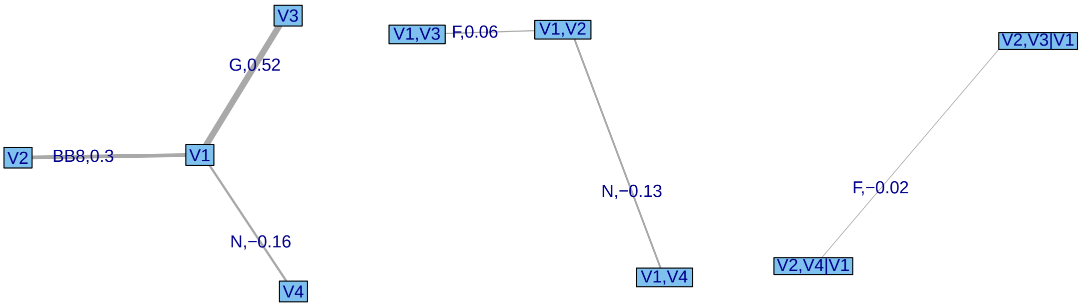


Figure 6.30: Female C-vine structure (with highest AIC under all admissible C-vines) estimated to impute "**Weight**" in method Copula Regression. (Mother-PB=V1, Rotula=V2, Child-PB=V3 and Weight=V4)

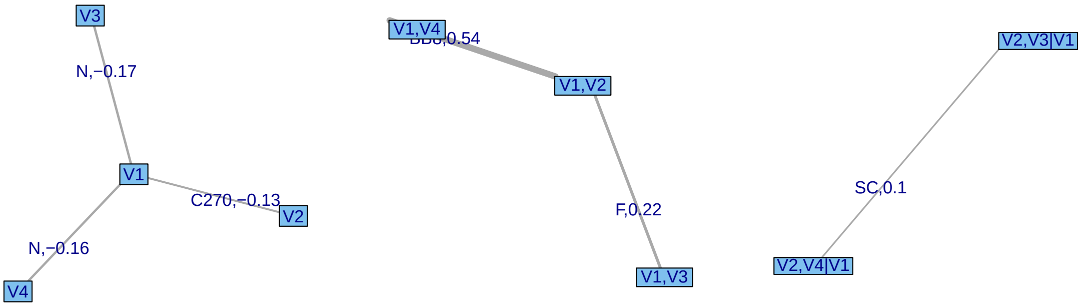


Figure 6.31: Female C-vine structure (with highest AIC under all admissible C-vines) estimated to impute "**Mother-PB**" in method Copula Regression. (Weight=V1, Child-PB=V2, Rotula=V3 and Mother-PB=V4)

Using female data, C-vine model for CopReg for the remaining two missing variables.

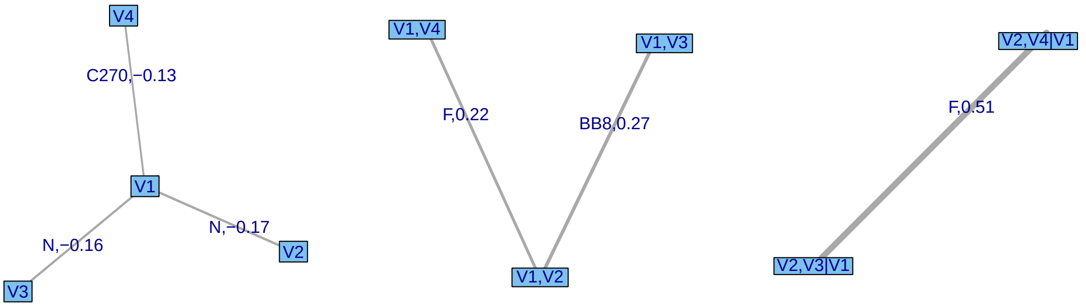


Figure 6.32: Female C-vine structure (with highest AIC under all admissible C-vines) estimated to impute "**Mother-PB**" and "**Child-PB**" in method Copula Regression. (Weight=V1, Rotula=V2, Mother-PB=V3 and Child-PB=V4)

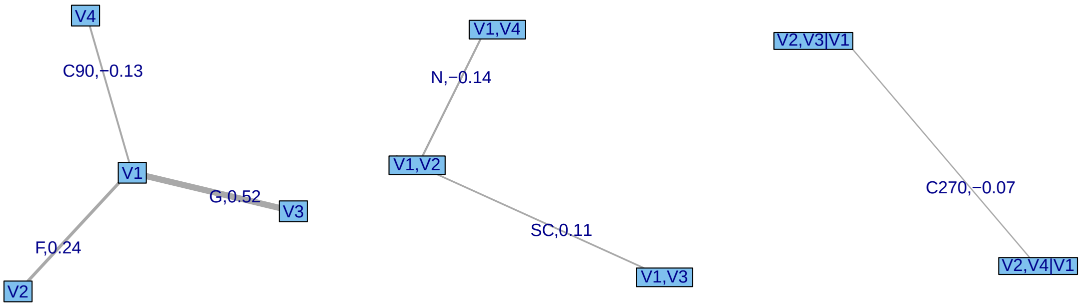


Figure 6.33: Female C-vine structure (with highest AIC under all admissible C-vines) estimated to impute "**Mother-PB**" and "**Weight**" in method Copula Regression. (Weight=V1, Child-PB=V2, Rotula=V3 and Mother-PB=V4)

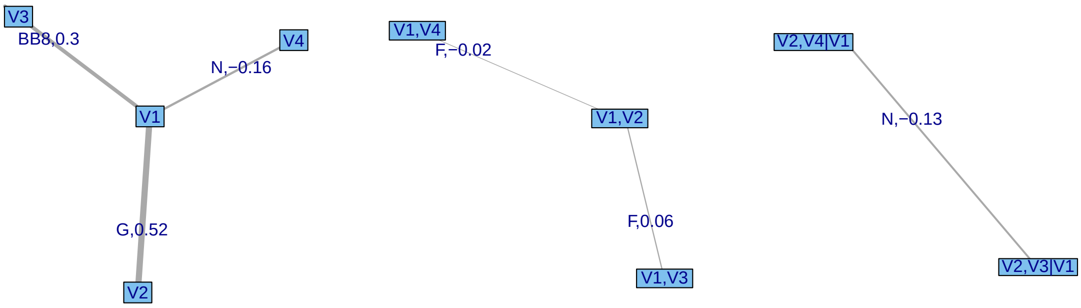


Figure 6.34: Female C-vine structure (with highest AIC under all admissible C-vines) estimated to impute "**Weight**" and "**Rotula**" in method Copula Regression. (Mother-PB=V1, Child-PB=V2, Rotula=V3 and Weight=V4)

Using male data, C-vine model for CopFit and CopExp. It is also used as one of the models in CopReg.

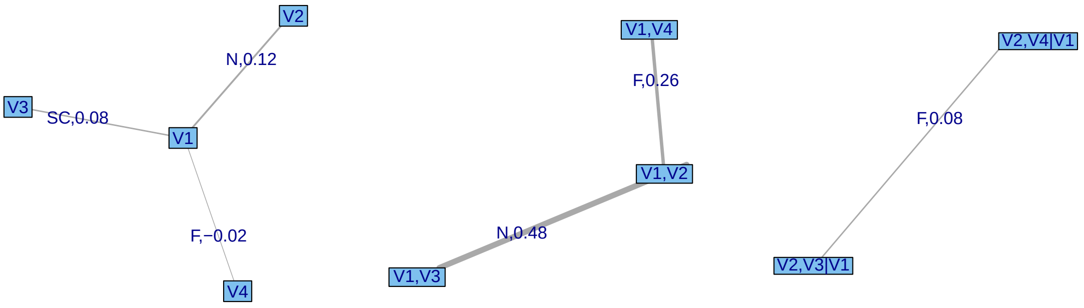


Figure 6.35: Male C-vine structure (with highest AIC under all C-vines) estimated for Copula Fitting Imputation (CopFit) and Copula Expectation Imputation (CopExp). Also estimated to impute "**Rotula**" and "**Child-PB**" in method Copula Regression Imputation (CopReg). (Weight=V1, Mother-PB=V2, Child-PB=V3 and Rotula=V4)

Using male data, C-vine model for CopReg for the remaining single variables missing.

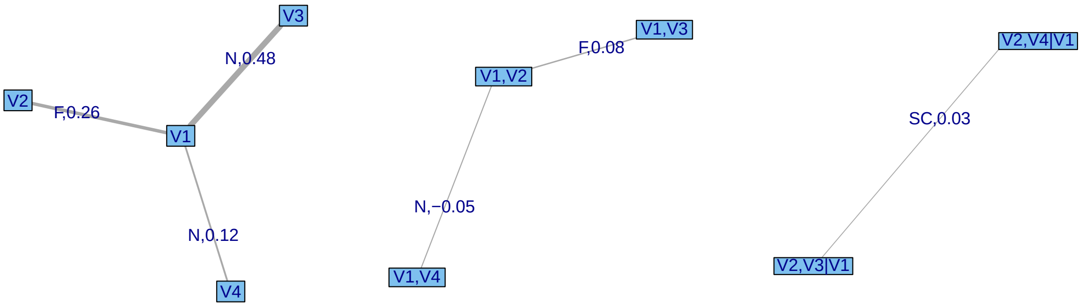


Figure 6.36: Male C-vine structure (with highest AIC under all admissible C-vines) estimated to impute "**Weight**" in method Copula Regression. (Mother-PB=V1, Rotula=V2, Child-PB=V3 and Weight=V4)

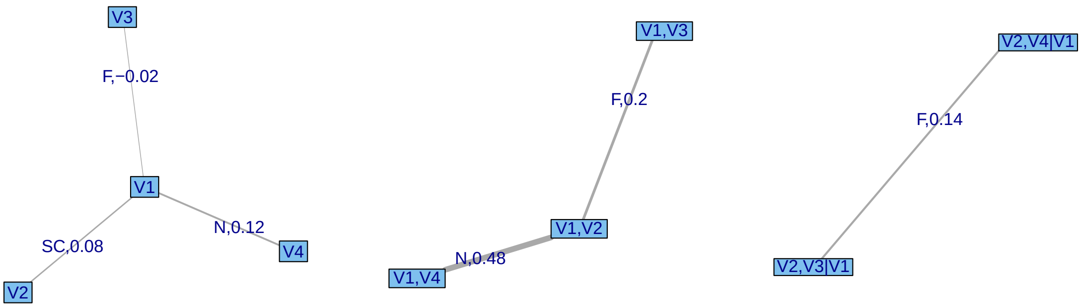


Figure 6.37: Male C-vine structure (with highest AIC under all admissible C-vines) estimated to impute "**Mother-PB**" in method Copula Regression. (Weight=V1, Child-PB=V2, Rotula=V3 and Mother-PB=V4)

Using male data, C-vine model for CopReg for the remaining two variables missing.

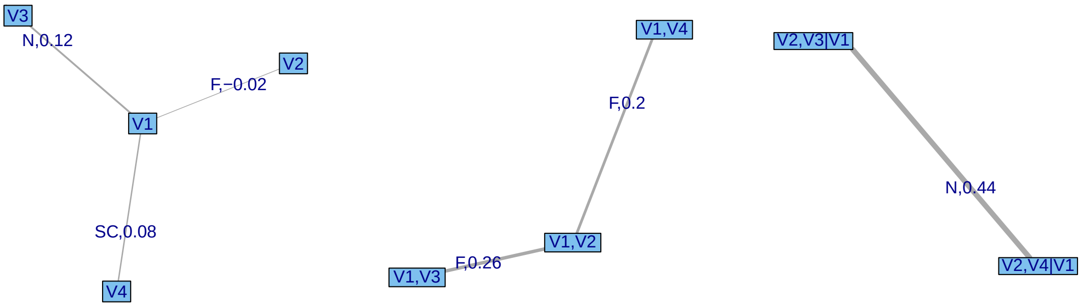


Figure 6.38: Male C-vine structure (with highest AIC under all admissible C-vines) estimated to impute "**Mother-PB**" and "**Child-PB**" in method Copula Regression. (Weight=V1, Rotula=V2, Mother-PB=V3 and Child-PB=V4)

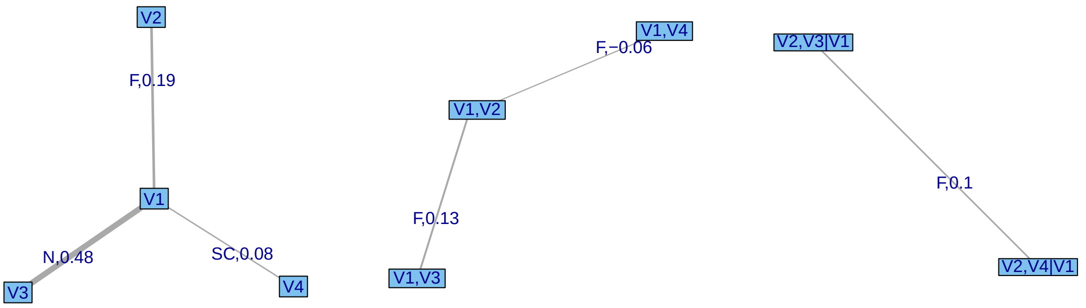


Figure 6.39: Male C-vine structure (with highest AIC under all admissible C-vines) estimated to impute "**Mother-PB**" and "**Weight**" in method Copula Regression. (Child-PB=V1, Rotula=V2, Mother-PB=V3 and Weight=V4)

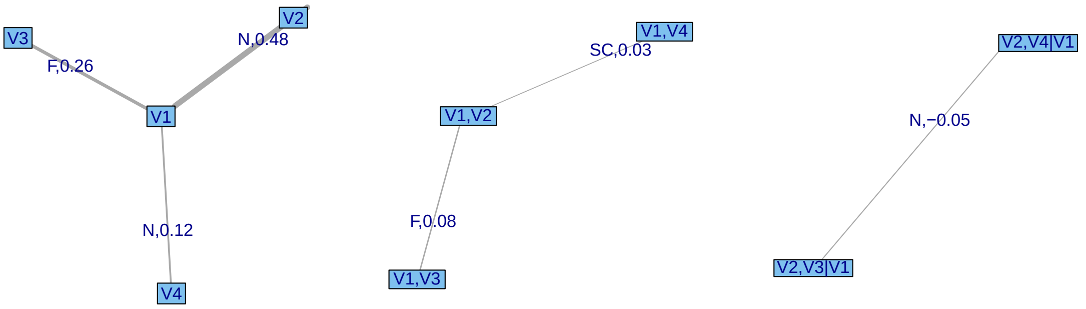


Figure 6.40: Male C-vine structure (with highest AIC under all admissible C-vines) estimated to impute "**Weight**" and "**Rotula**" in method Copula Regression. (Mother-PB=V1, Child-PB=V2, Rotula=V3 and Weight=V4)

Evaluation of the different imputation methods, using Kendall's tau values.

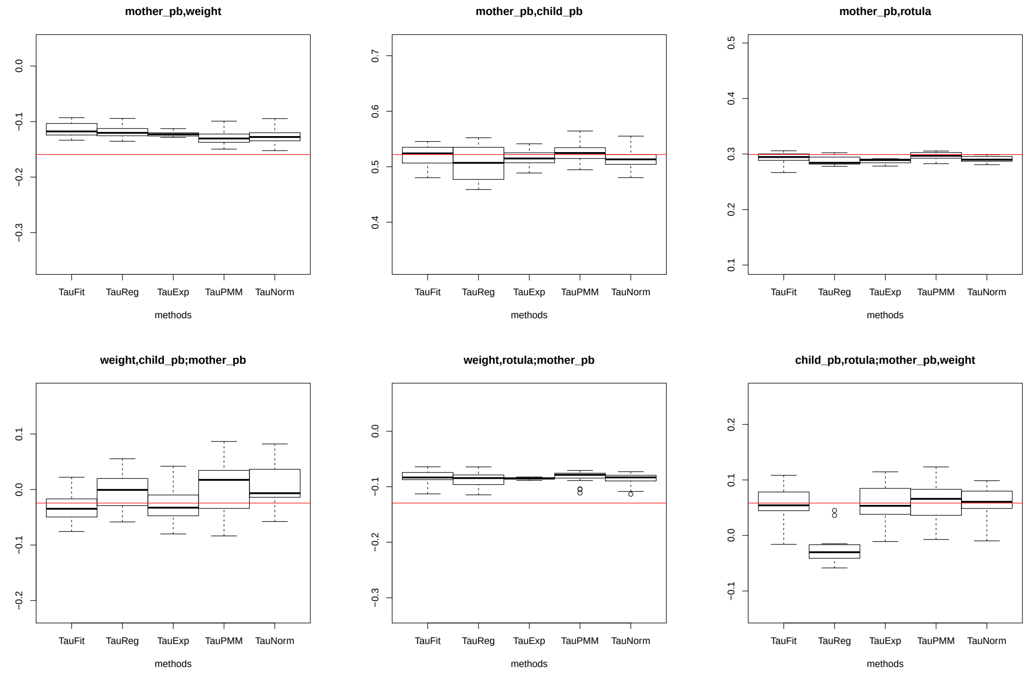


Figure 6.41: Female Kendall's Tau for the families in the best fitting C-vine structure, according to the AIC criteria, using only female complete data after 20 imputations using 6 imputation methods (the red line is deletion, i.e. complete case). Range of Kendall's tau values = 0.4.

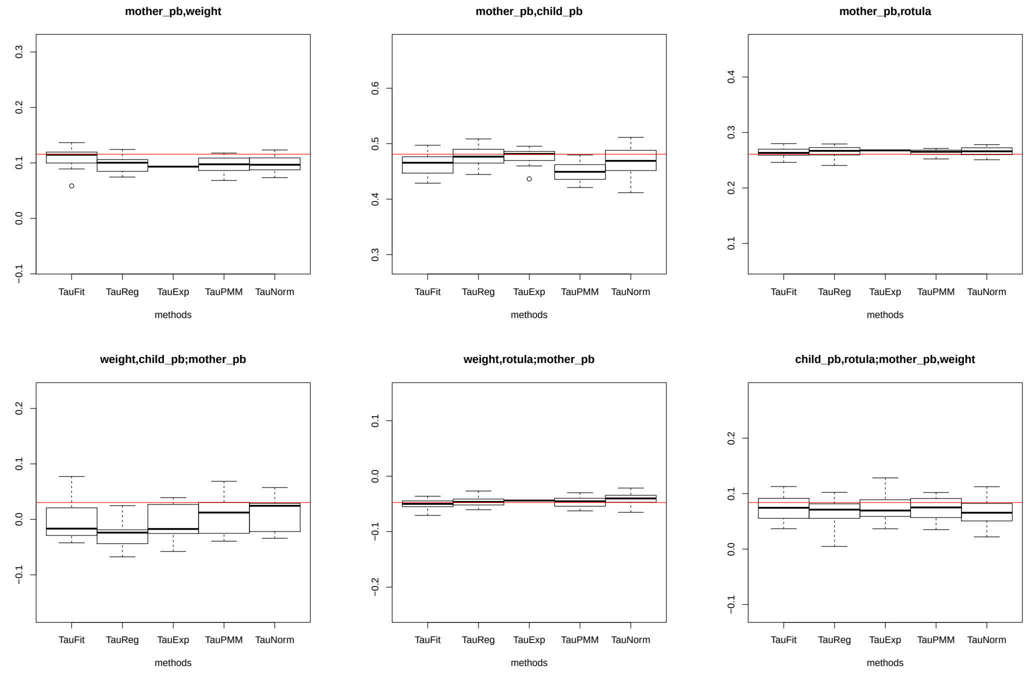


Figure 6.42: Male Kendall's Tau for the families in the best fitting C-vine structure, according to the AIC criteria, using only male complete data after 20 imputations using 6 imputation methods (the red line is deletion, i.e. complete case) Range of Kendall's tau values = 0.4.

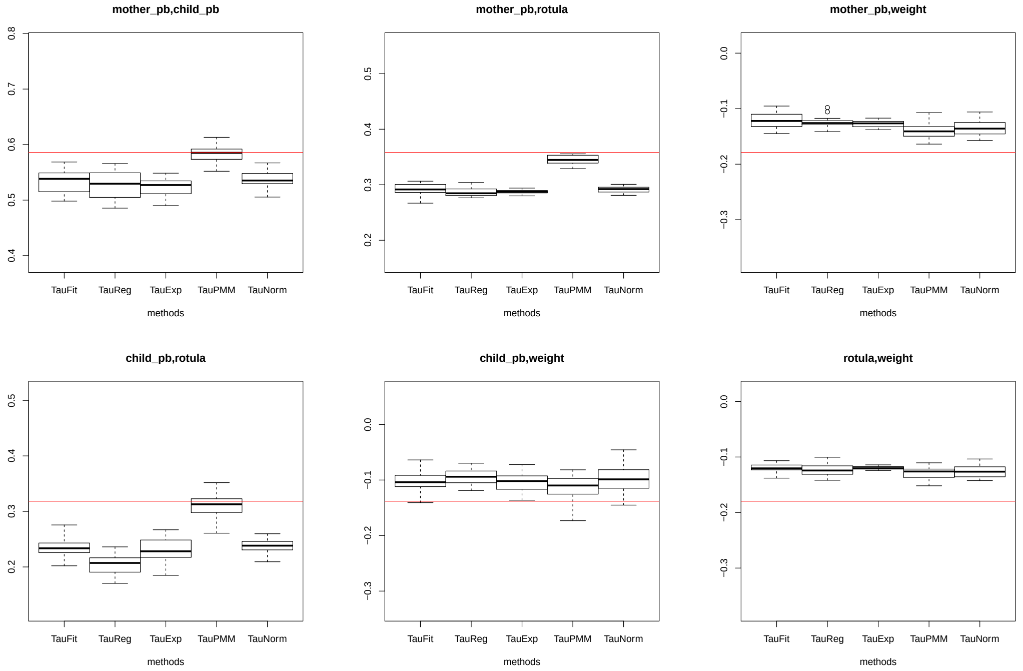


Figure 6.43: Female empirical Kendall's Tau for all possible bivariate combinations without conditioning, using only female complete data after 20 imputations using 6 imputation methods (the red line is deletion, i.e. complete case). Range of Kendall's tau values = 0.4.

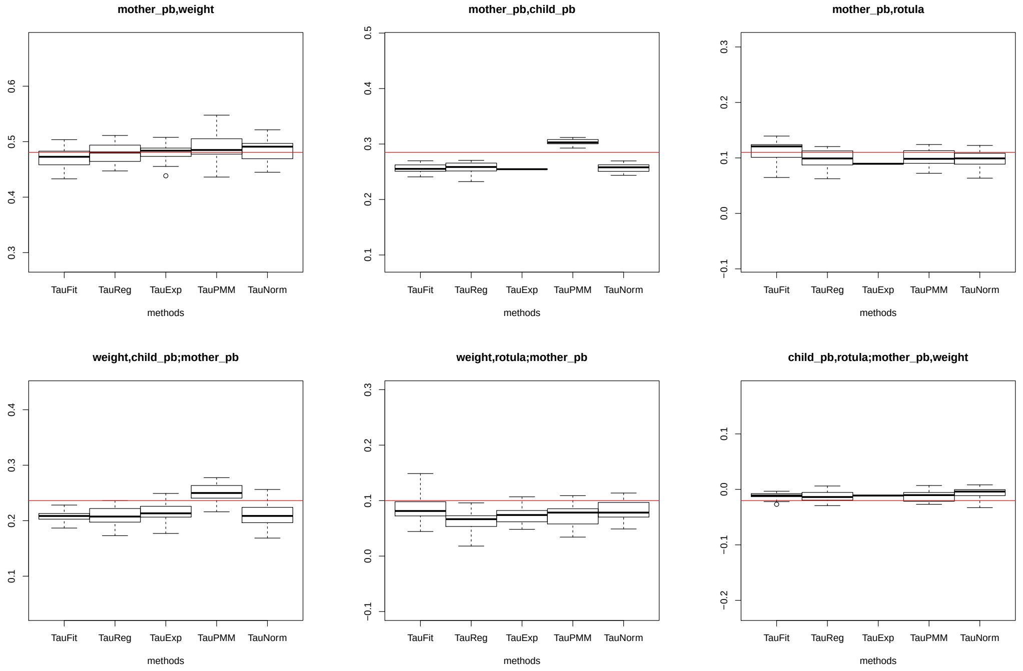


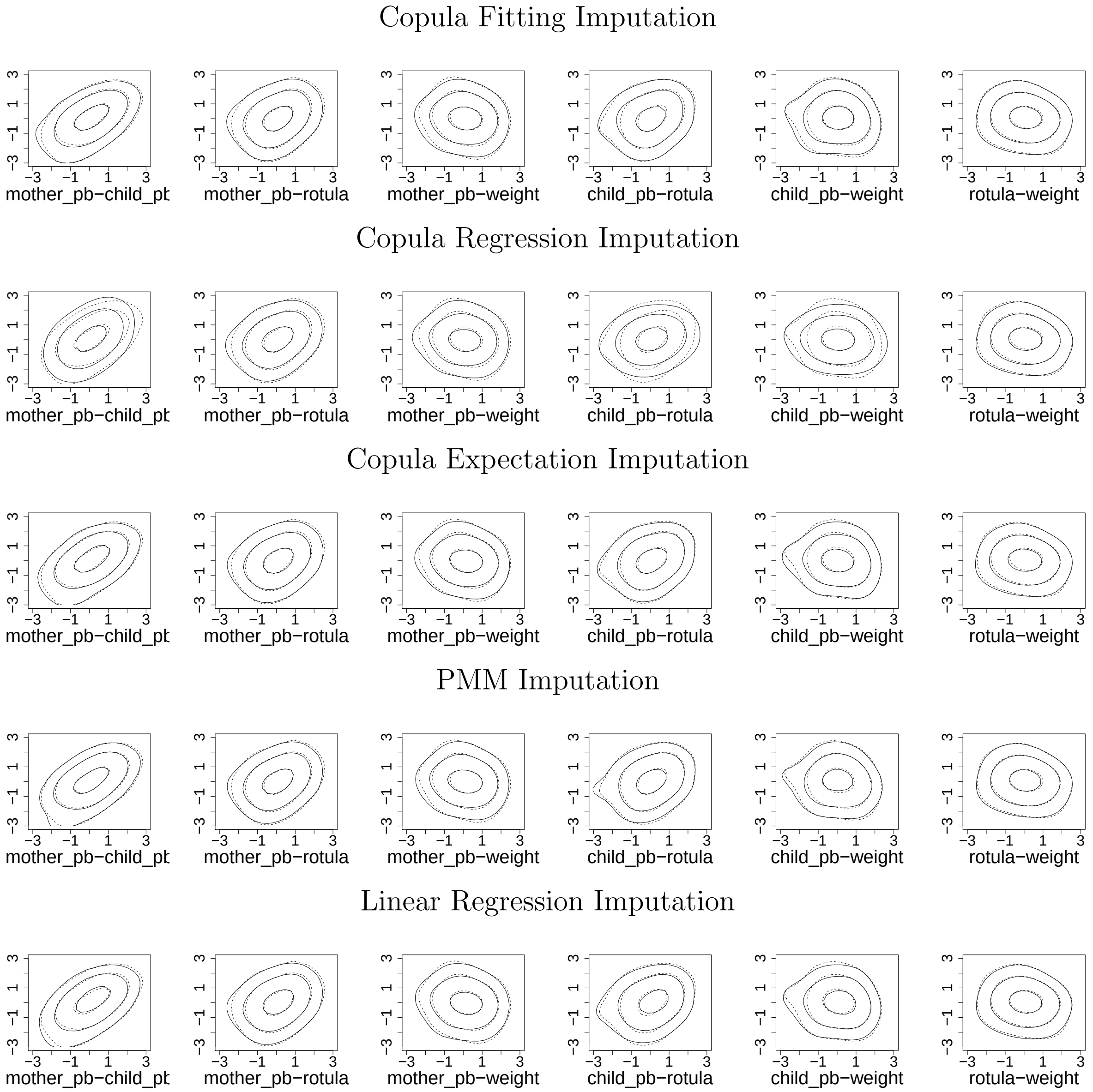
Figure 6.44: Male empirical Kendall's Tau for all possible bivariate combinations without conditioning, using only male complete data after 20 imputations using 6 imputation methods (the red line is deletion, i.e. complete case). Range of Kendall's tau values = 0.4.

A much better fit is seen here for every method. Only Copula Regression has the wrong sign in the last female plot (see Figure 6.41). One reason could be a different copula fit for imputing a specific variable, because this is the only method which uses more than one C-vine for modeling the data. It is not clear if the structure given above is the true structure for the lead measures and the weight of children at 6 month. It is chosen according to the highest *AIC* under all C-vine possibilities. But even if the structure is not well modeled, it should not change with filling out nonresponse.
For female children, the PMM procedure works best in looking at bivariate Kendall's tau values. In the case of the male children, all different imputation methods were nearly equally successful (see Figure 6.44).

Evaluation of the different imputation methods, using mean contour plots.

Moving on to the mean contours with separated sex (again on the levels 50%, 75%, 95%).

Again there is not a huge gap between the different imputations. It is possible that the mean of the contours is smoothing the result a little too much to interpret some success or failure, but as seen in the Kendall's tau tries, in total there is not a big mismatch. The most noticeable difference while working with every single procedure is the time consuming model fitting in the Copula Regression method (which can be improved with efficient algorithms) and the even longer integral evaluations for the computations in the Copula Expectation procedure. It is only possible to improve efficiency here by avoiding the numerical integration part, but some of the bivariate copulae can not be expressed in closed form. So it is either possible to use only copula families that allow for closed form evaluations (like only using Gaussian copulae), which is very restrictive, or to lose accuracy within the imputation values.


Copula Fitting Imputation

Copula Regression Imputation

Copula Expectation Imputation

PMM Imputation

Linear Regression Imputation

Figure 6.45: Pairwise female empirical mean contours, using imputed data (20 tries) for female and male separately. The dashed contours are the complete cases.

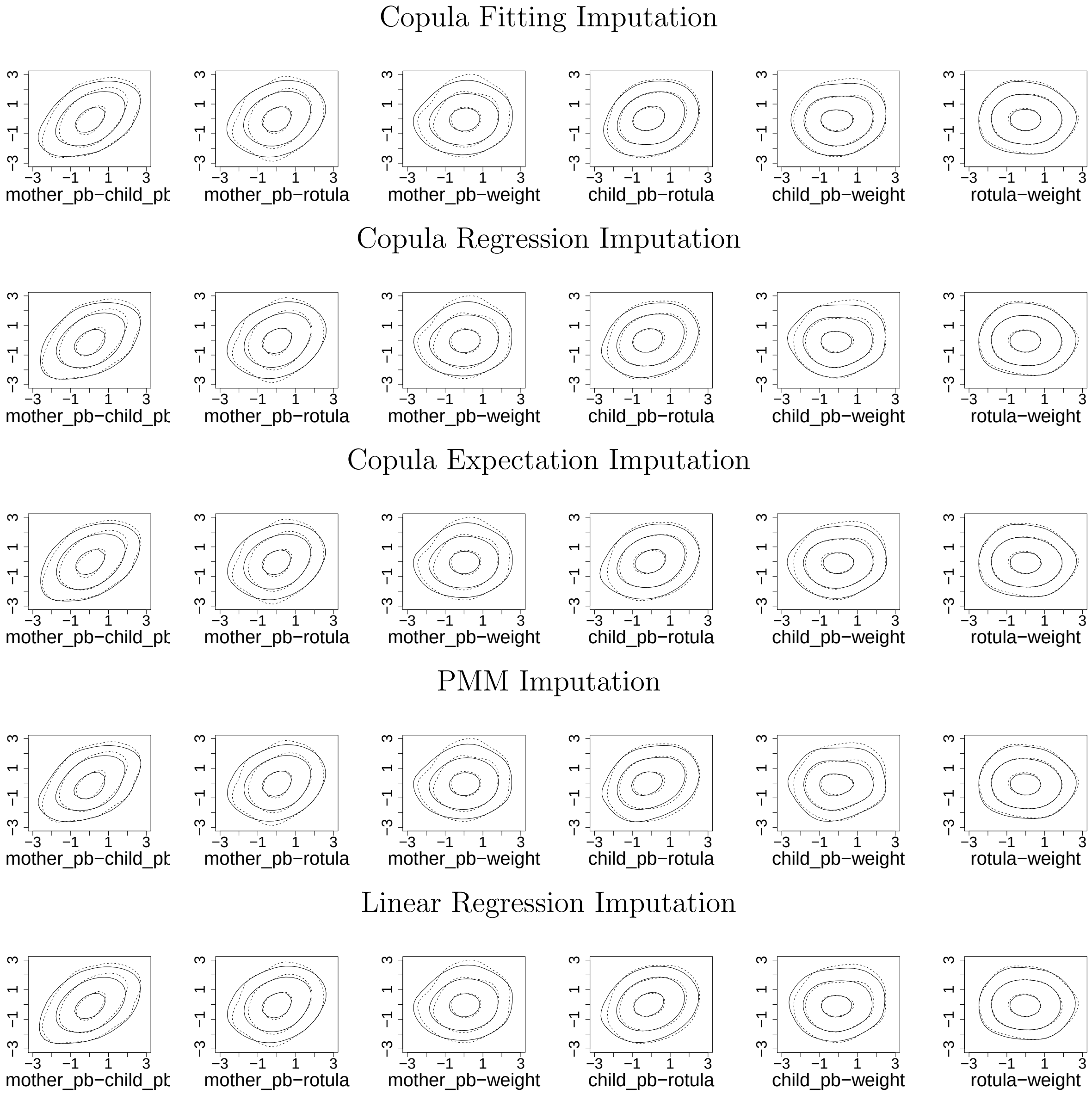


Figure 6.46: Pairwise male empirical mean contours, using imputed data (20 tries) for female and male separately. The dashed contours are the complete cases.

Chapter 7

Case Study (6 dimensions)

Up to 4 dimensions it is not possible to construct a "real" R-vine tree structure. There are only C-vines or D-vines in these low-dimensional cases. For testing copula imputation methods under general conditions (with possible R-vine dependence structures), one needs a data set with more than 4 dimensions.

We decided to take a look at the same study as in the 4-*dimensional* case study presented, and to include two more "weight" variables on two more time measurements. Weight at different points in time is highly correlated and has a very interesting and non-trivial dependence structure with respect to the lead measurements. So it makes sense to add these variables for imputation. The reason why not all weight measurements in time are considered is the amount of data we have. When considering the complete cases without missing data, there are only about 100 observations left for each, female and male children. Because we distinguish between genders, it is enough to use only a 6-*dimensional* data set. So the different variables are:

- Weight6: measure of weight for a child at 6 month. Continuous variable.
- Weight12: measure of weight for a child at 12 month. Continuous variable.
- Weight24: measure of weight for a child at 24 month. Continuous variable.
- Child-PB: lead concentration in child's cord blood at birth. Continuous variable.
- Rotula: lead concentration in child's rotula (patella) bone with respect to a benchmark. Continuous variable.
- Mother-PB: lead concentration in mother's blood. Continuous variable.

And we distinguish between Birth-Sex: sex of a child. Integer variable with values in the set $\{0, 1\} := \{female, male\}$, with $\#female = 179$ $(49,3\%)$ and $\#male = 184$ (50,7%).

Varying Variable	Missing Values
Weight (6 Month)	12 (3,3%)
Weight (12 Month)	19 (5,2%)
Weight (24 Month)	28 (7,7%)

Table 7.1: Percentage of missing values for the longitudinal measured variable "Weight".

Non-varying Variables	Missing Values
Birth-SEX	0 (0%)
Child-PB	92 (25,3%)
Rotula	2 (0,6%)
Mother-PB	13 (3,6%)

Table 7.2: Percentage of missing values for the variables measured at birth.

The amount of complete cases for male children is 120 and for female children 114. So there are 34.8% missing for male sex, and 36.3% for female sex. This is a setting where imputation is needed, because nobody throws away 35% of data that has been collected with a lot of money and effort within more than 2 years.

7.1 Imputation

Here, only Vine Copula Regression Imputation is tested, because numerical integration over 6 dimensions is too time consuming for Vine Copula Expectation Imputation. The Vine Copula Fitting Imputation method is omitted ,too, because this procedure seems to not work well in the simulation study. For comparison, again the PMM and the Linear Regression approaches are added for the reasons explained in Chapter 2 (Commonly used Imputation Methods).
For simplicity, we introduce

- **CPB** for Child-PB,
- **MPB** for Mother-PB,
- **R** for Rotula,
- **W6** for Weight6,
- **W12** for Weight12,
- **W24** for Weight24

while plotting the tree structures.

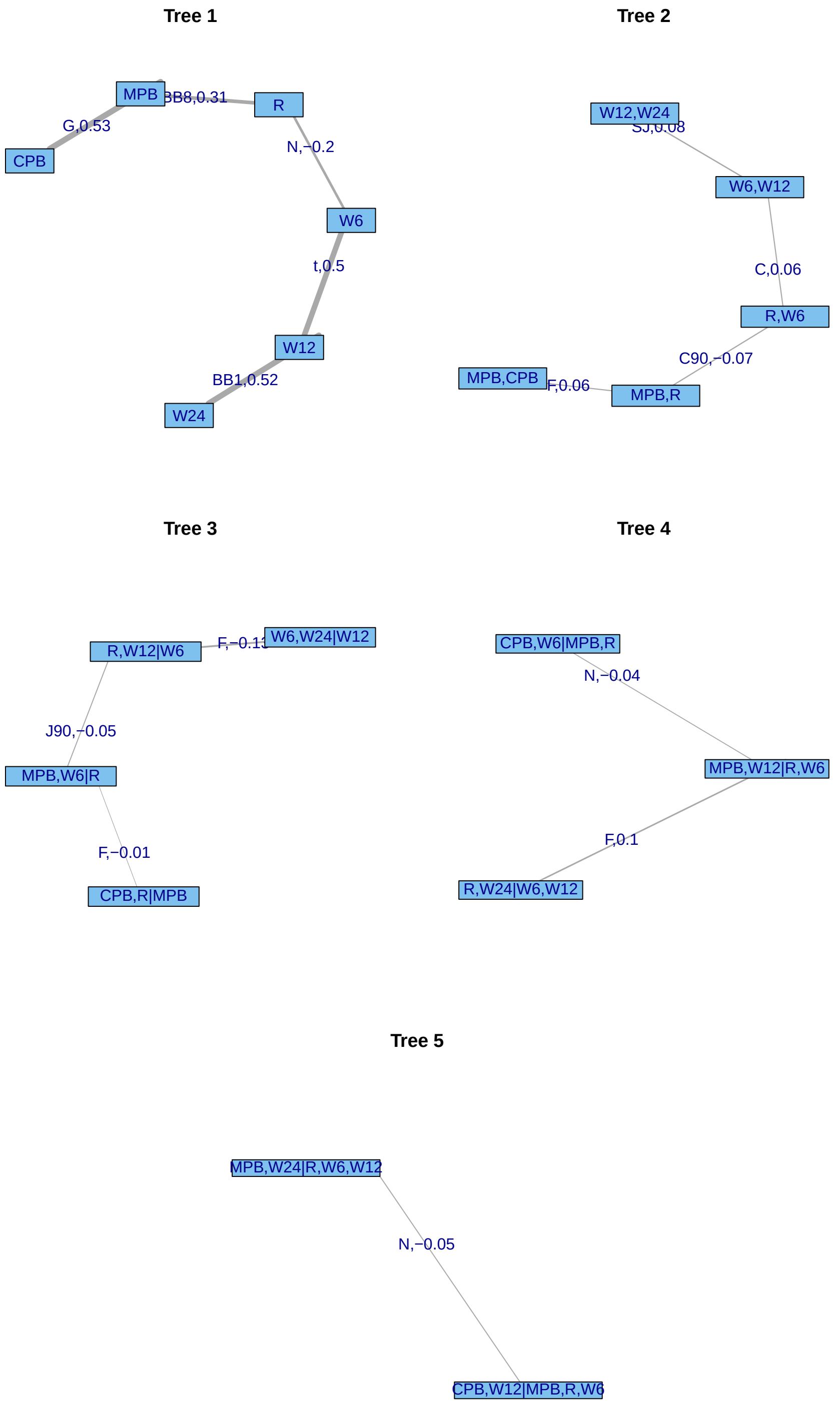


Figure 7.1: Best fitting R-vine structure, only using complete data for female children.

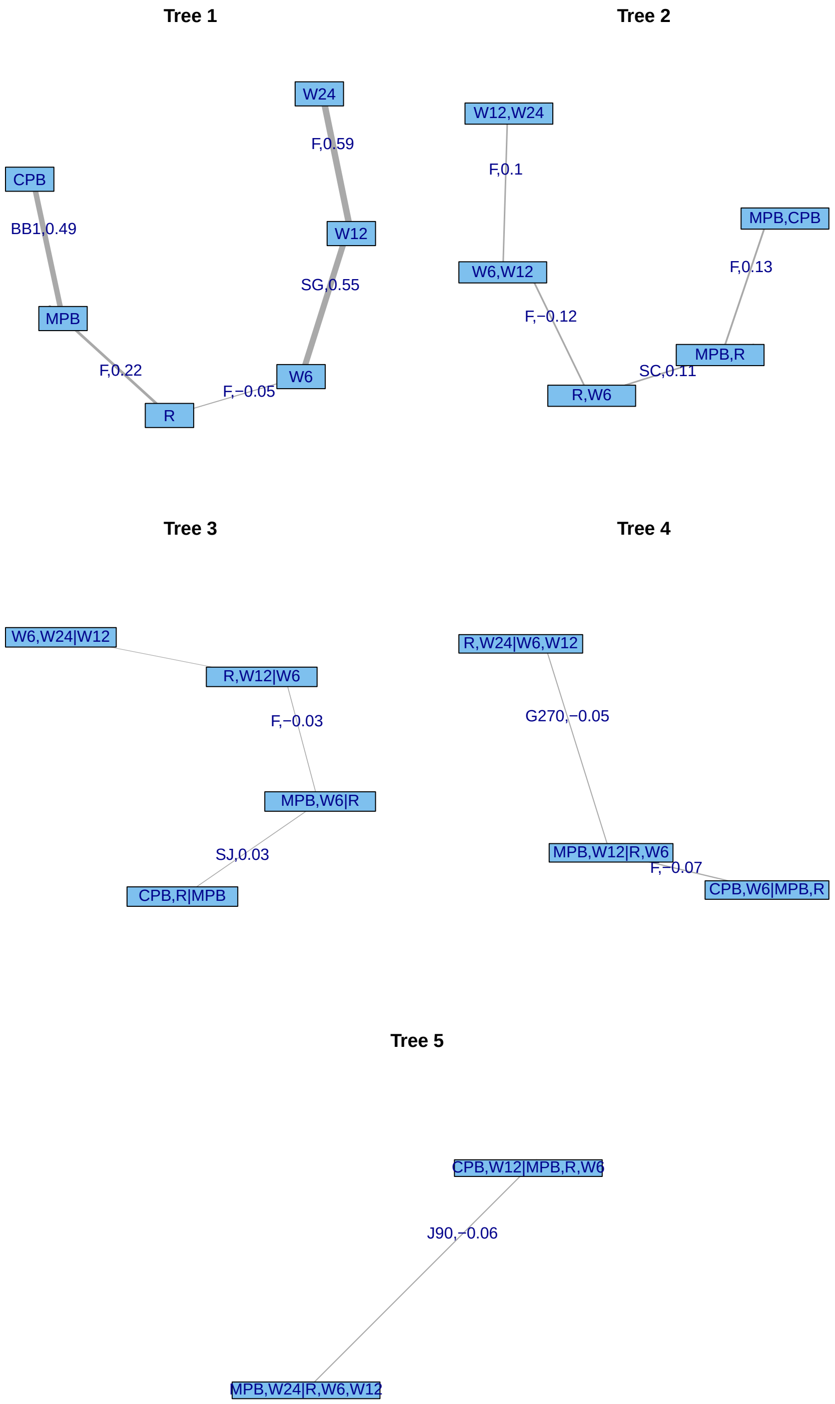


Figure 7.2: Best fitting D-vine structure, only using complete data for male children.

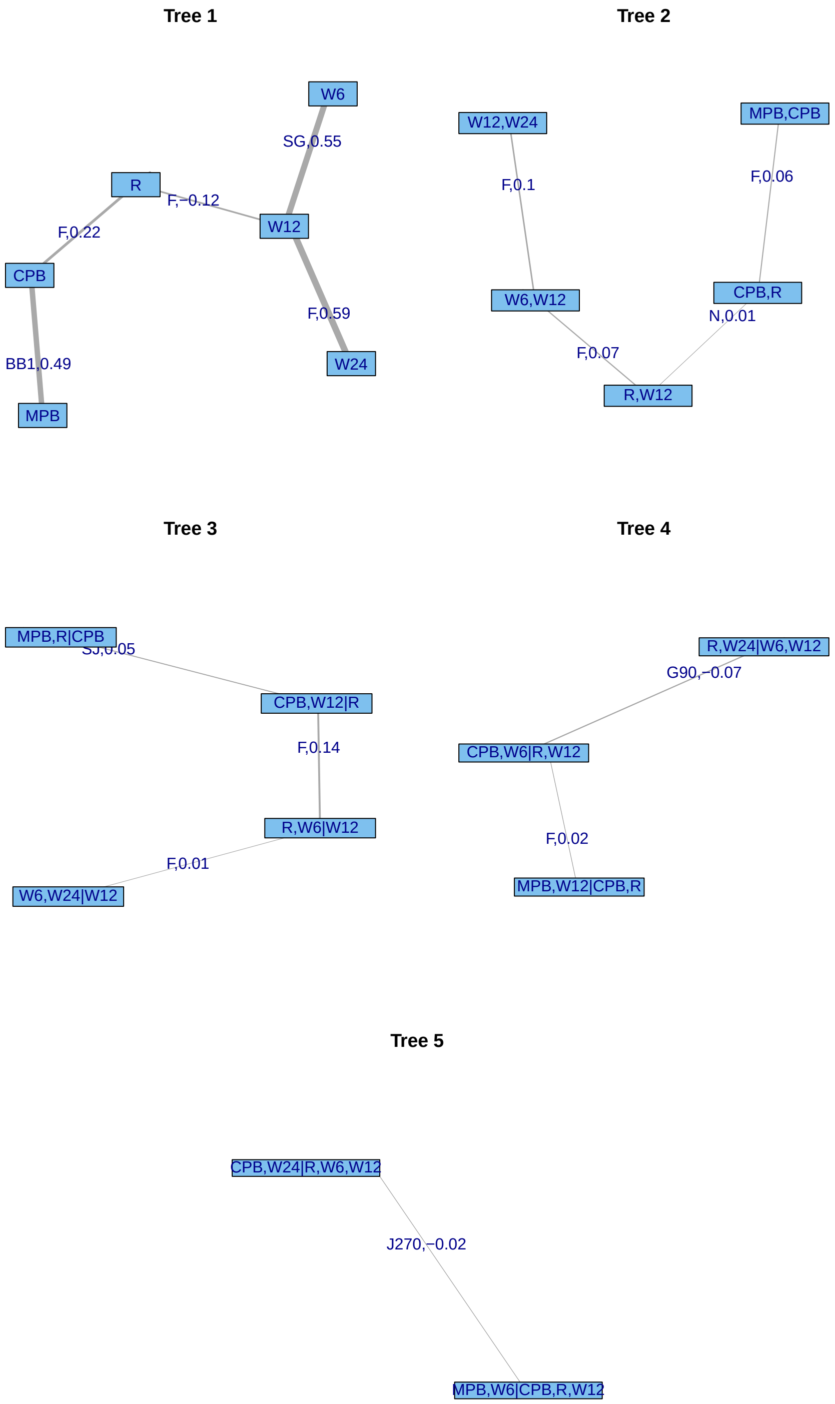


Figure 7.3: Best fitting R-vine structure, only using complete data for male children.

gender	vine	AIC	BIC	# parameter
female	R-vine (=D-vine)	285.2	236.0	18
male	R-vine	271.9	227.3	16
male	D-Vine	259.6	215.0	16

Table 7.3: AIC and BIC for the best fitting R-vine structures, using complete data for female and male separately. For comparison, the D-vine structure for male is added.

7.1.1 Evaluation via Mean Quantiles

In the appendix, there are the 95% quantile contour plots added, where one can see that the PMM method differs in some bivariate cases from the deletion dependence structure for female children, while Vine Copula Regression Imputation keeps the structure more precisely. But for 20 tries, the 95% quantiles of data, are more or less just the highest values. Therefore significance is less than for the mean quantiles. So it was decided to put the additional pictures in the appendix. For the mean quantiles (see Table 7.4, 7.5, 7.6, 7.7, 7.8, 7.9), all three methods do not have high deviations from the complete case structure. Only the Predictive Mean Matching (PMM) has some difference in the dependence structure for "Mother-PB, Child-PB".

Copula Regression Imputation

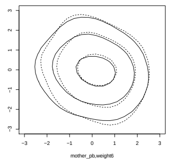
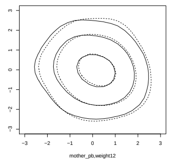
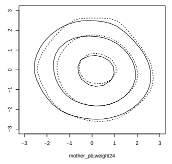
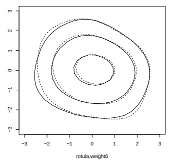
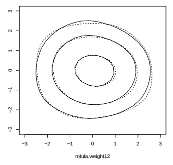
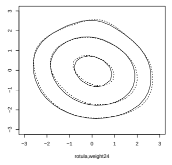
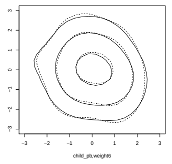
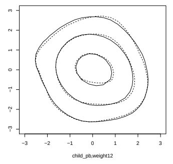
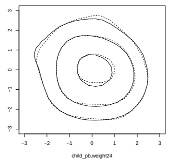
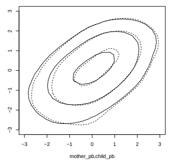
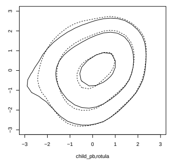
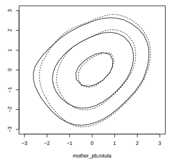
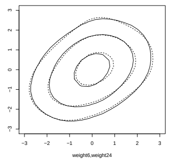
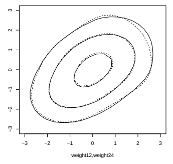
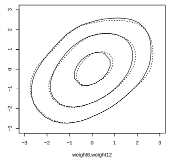
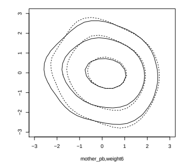
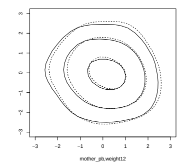
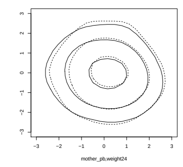
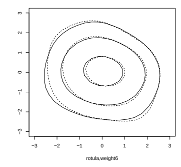
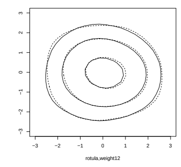
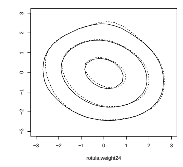
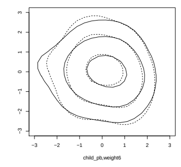
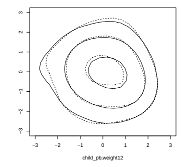
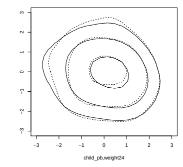
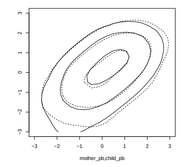
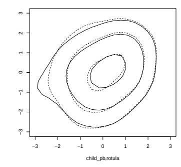
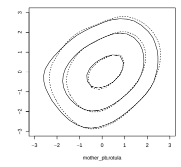
Female	Weight6	Weight12	Weight24
Mother-PB			
Rotula			
	Mother-PB	**Rotula**	
Child-PB			
Child-PB			
Rotula			
	Weight6	**Weight12**	
Weight24			
Weight12			

Table 7.4: Female empirical mean contours (20 tries) for all possible bivariate combinations without conditioning. The dashed contours are the complete cases.

PMM Imputation

Female	Weight6	Weight12	Weight24
Mother-PB			
Rotula			
Child-PB			

	Mother-PB	Rotula	
Child-PB			
Rotula			

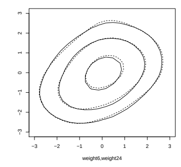
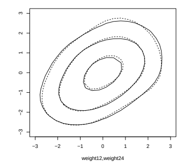
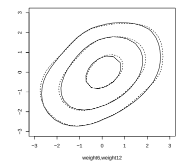
	Weight6	Weight12	
Weight24			
Weight12			

Table 7.5: Female empirical mean contours (20 tries) for all possible bivariate combinations without conditioning. The dashed contours are the complete cases.

Linear Regression Imputation

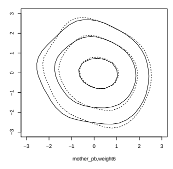
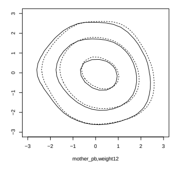
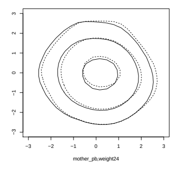
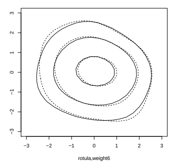
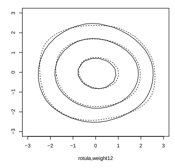
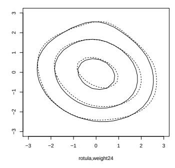
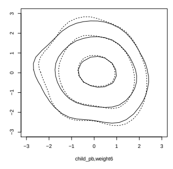
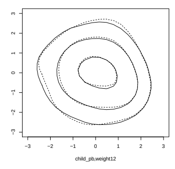
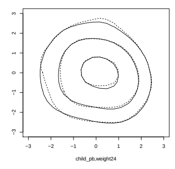
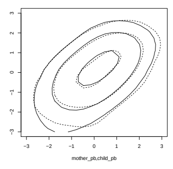
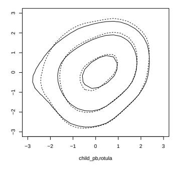
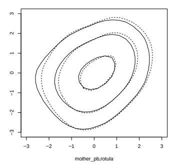
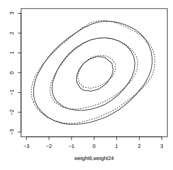
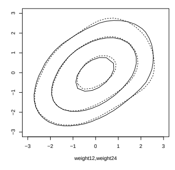
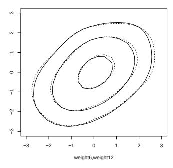
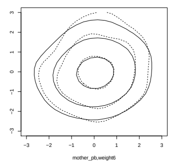
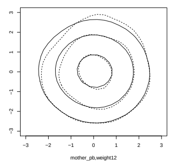
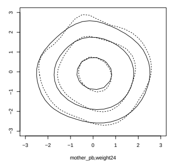
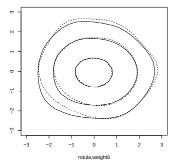
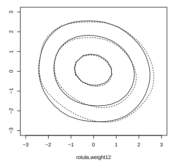
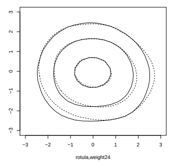
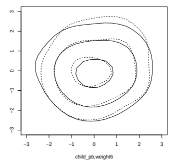
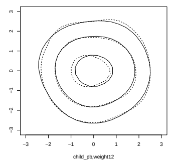
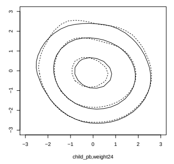
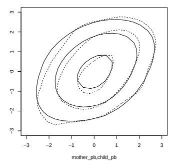
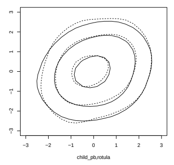
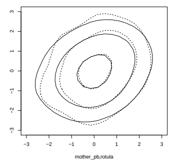
Female	Weight6	Weight12	Weight24
Mother-PB			
Rotula			
Child-PB			
	Mother-PB	Rotula	
Child-PB			
Rotula			
	Weight6	Weight12	
Weight24			
Weight12			

Table 7.6: Female empirical mean contours (20 tries) for all possible bivariate combinations without conditioning. The dashed contours are the complete cases.

Copula Regression Imputation

Male	Weight6	Weight12	Weight24
Mother-PB			
Rotula			
Child-PB			

	Mother-PB	Rotula	
Child-PB			
Rotula			

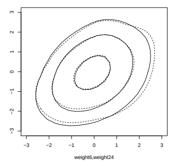
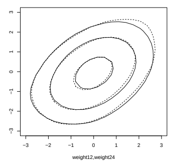
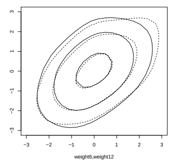
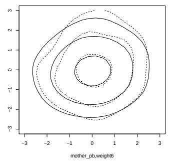
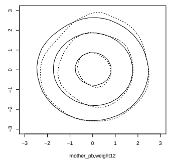
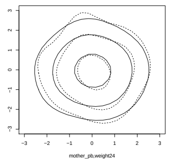
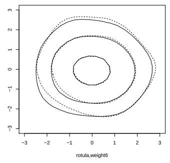
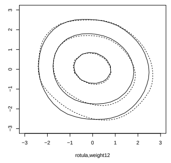
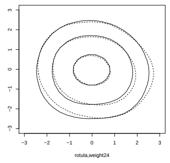
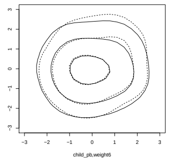
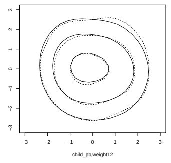
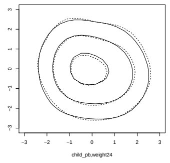
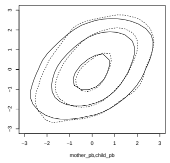
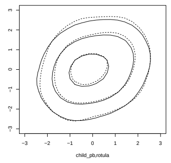
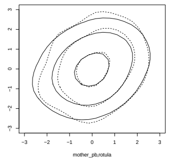
	Weight6	Weight12	
Weight24			
Weight12			

Table 7.7: Male empirical mean contours (20 tries) for all possible bivariate combinations without conditioning. The dashed contours are the complete cases.

PMM Imputation

Male	Weight6	Weight12	Weight24
Mother-PB			
Rotula			
Child-PB			

	Mother-PB	Rotula	
Child-PB			
Rotula			

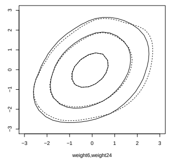
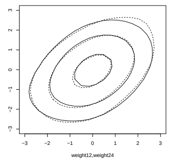
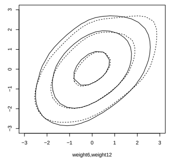
	Weight6	Weight12	
Weight24			
Weight12			

Table 7.8: Male empirical mean contours (20 tries) for all possible bivariate combinations without conditioning. The dashed contours are the complete cases.

Linear Regression Imputation

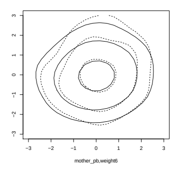
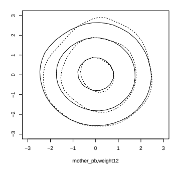
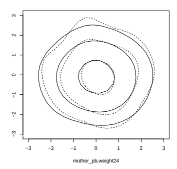
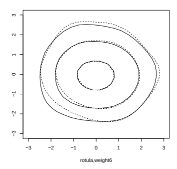
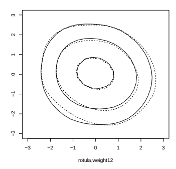
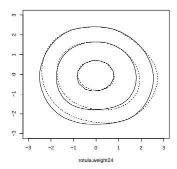
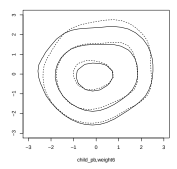
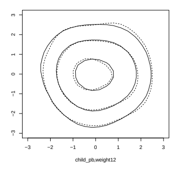
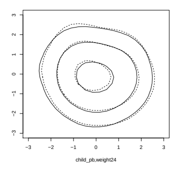
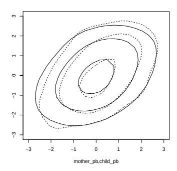
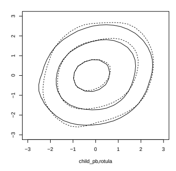
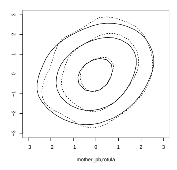
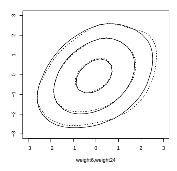
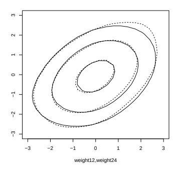
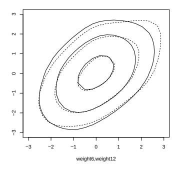
Male	Weight6	Weight12	Weight24
Mother-PB			
Rotula			
Child-PB			
	Mother-PB	Rotula	
Child-PB			
Rotula			
	Weight6	Weight12	
Weight24			
Weight12			

Table 7.9: Male empirical mean contours (20 tries) for all possible bivariate combinations without conditioning. The dashed contours are the complete cases.

7.1.2 Evaluation via Kendall's Tau Values

In this case, it is quite difficult to evaluate whether a method is better or worse than other procedures. What one can observe is that the range of the Kendall's tau value box plot diagrams for Copula Regression Imputation (TauReg) is mostly the smallest among all three methods (see Figure 7.4,7.5). This is an indicator for being stable while repeating the procedure. For a stochastic single imputation method this is an attainable property. It is also noticeable that, in most situations, the Predictive Mean Matching (TauPMM) has the least deviance from the complete case parameters.

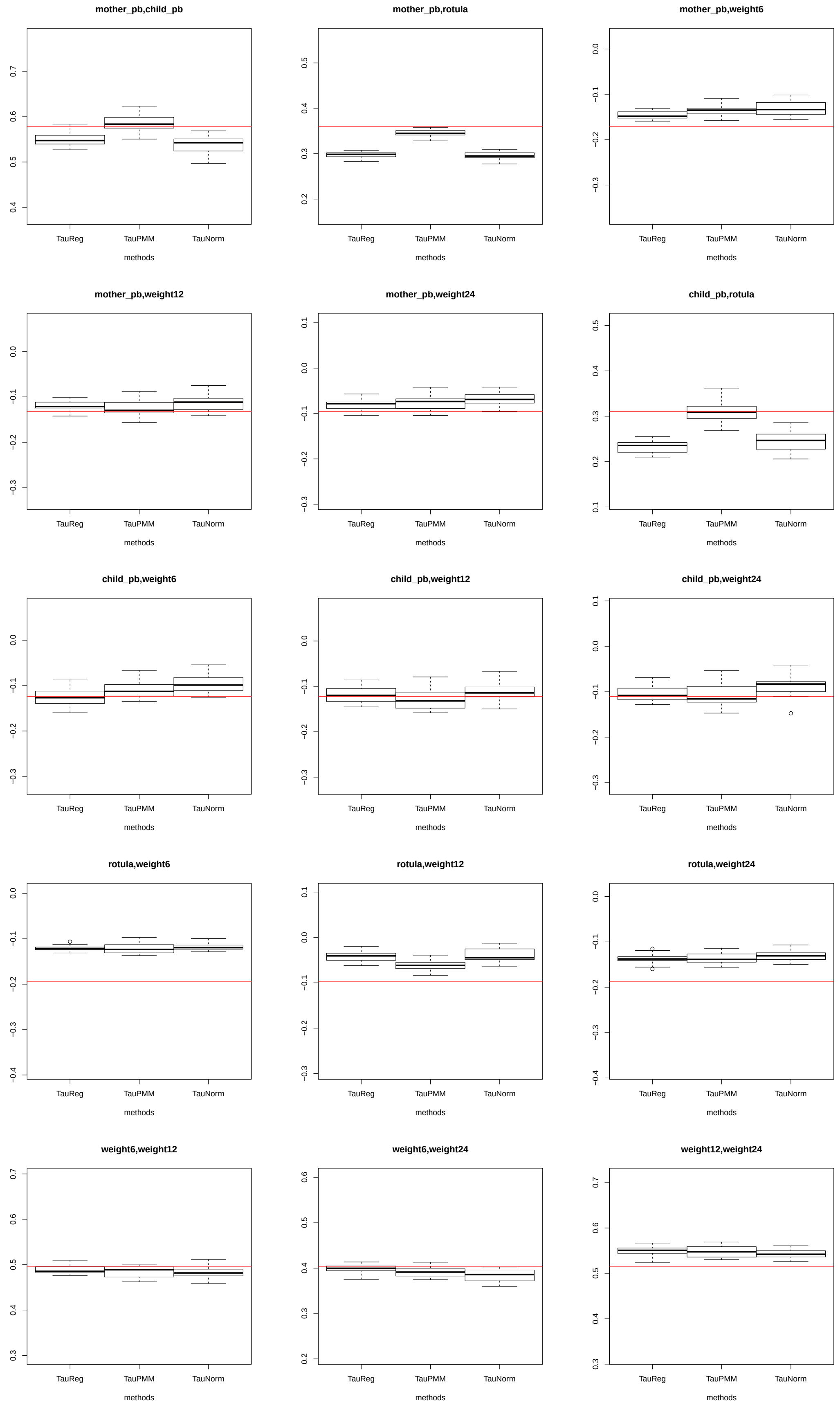


Figure 7.4: Female empirical Kendall's Tau for all possible bivariate combinations without conditioning, using only female complete data after 20 imputations using 6 imputation methods (the red line is deletion, i.e. complete case). Range of Kendall's tau values = 0.4.

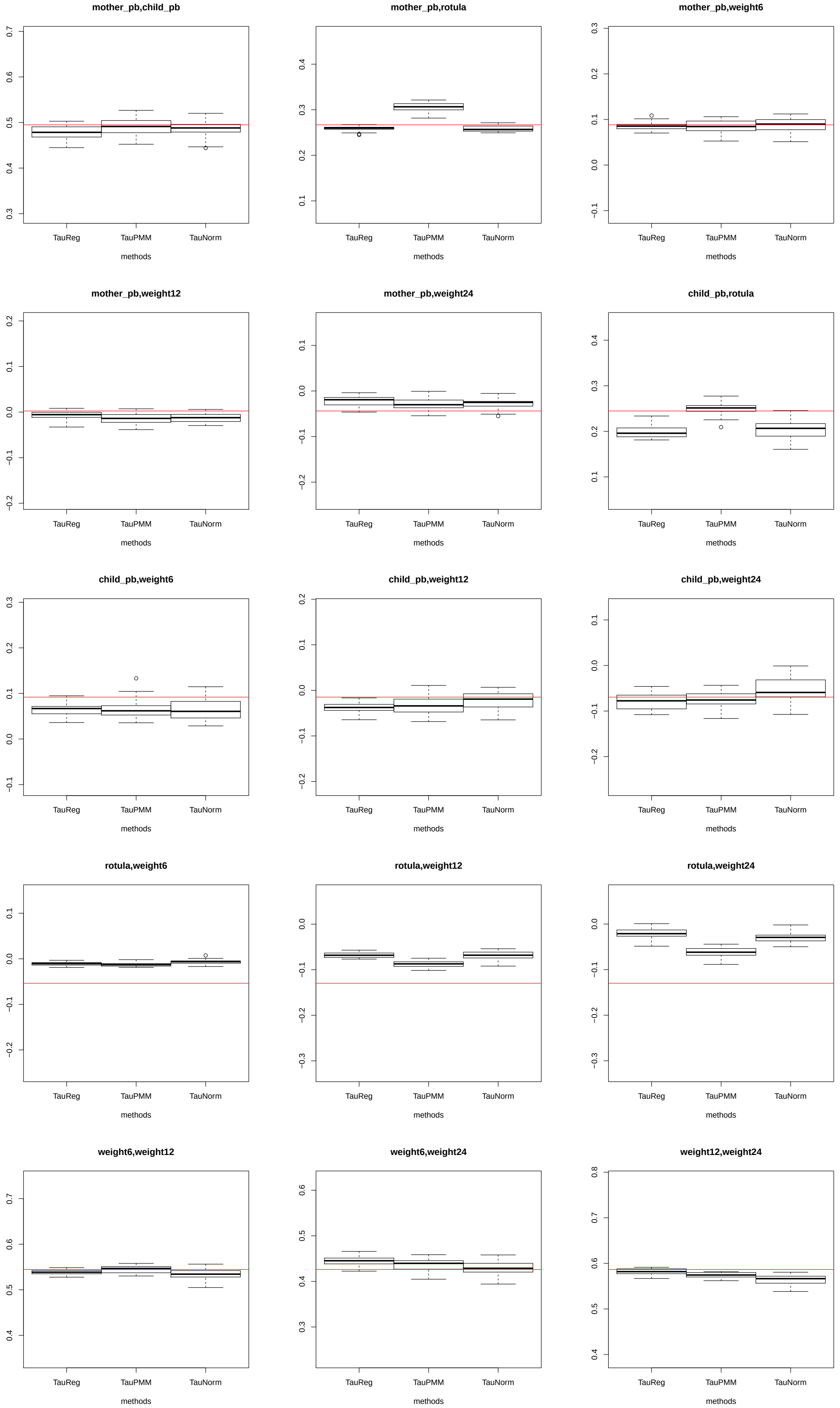


Figure 7.5: Male empirical Kendall's Tau for all possible bivariate combinations without conditioning, using only male complete data after 20 imputations using 6 imputation methods (the red line is deletion, i.e. complete case). Range of Kendall's tau values = 0.4.

7.1.3 Evaluation via AIC & BIC

The last evaluation is done via AIC and BIC. An R-vine model was fitted with all available data, once for female children only, and once for male children only. Then the AIC and the BIC were computed for this model with data for complete cases (female and male separately) only. Then, after imputation, the AIC and BIC were computed again for all three methods separately using the R-vine model with the same bivariate copula families and the corresponding parameters from the complete cases, but with imputed data. The results are plotted (see Figure 7.6, 7.7) and tabled (see Table 7.10, 7.11). One can observe that clearly Vine Copula Regression Imputation has the best fit for both female and male children.

Female	AIC	BIC
Del	285.2	236
CopReg	455.4	398.1
PMM	440.5	382.9
Norm	432.3	375

Table 7.10: Mean (20 tries) AIC, BIC before and after imputation for the best fitting R-vine, only using complete data for female children.

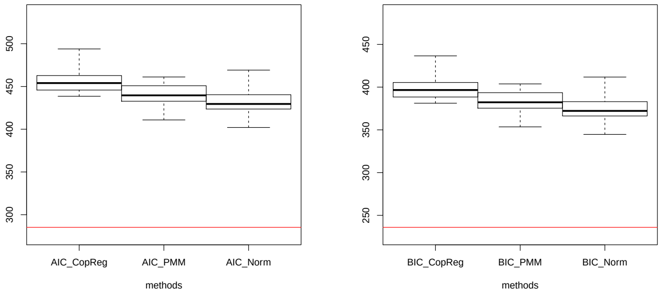


Figure 7.6: Box plot of AIC (left) and BIC (right) for 20 different imputations, only using complete data for female children. The red line corresponds to complete case.

Male	AIC	BIC
Del	271.9	227.3
CopReg	414.8	363.3
PMM	373	321.5
Norm	387.1	335.6

Table 7.11: Mean (20 tries) AIC, BIC before and after imputation for the best fitting R-vine, only using complete data for male children.

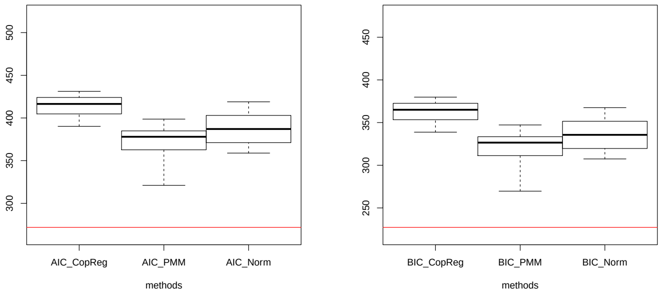


Figure 7.7: Box plot of AIC (left) and BIC (right) for 20 different imputations, only using complete data for male children. The red line corresponds to complete case.

Chapter 8

Conclusion

Missing Data is an issue affecting the data analyzing process in its way from the beginning of collecting observations up to the end. In the course of this work, three applicable methods were proposed and discussed, helping to overcome the absence of values in a data set. Each of these methods models the dependence structure using an R-vine copula based on the complete observations. While two of the new developed procedures (CopFit and CopExp) turned out to have disadvantages compared to already existing imputation methods, one (CopReg) was found to be competitive (and in some situations even better) and applicable in general.

For application of the three R-vine copula based imputation methods, three algorithms were proposed that can be used apart from filling out data sets. Two of them are simulation schemes from a given R-vine structure with potentially given values, and the remaining is a procedure to find sub R-vine structures where simulation with given values is possible.

In the simulation study presented, the R-vine copula based imputation method CopReg did not outperform all other procedures included in every single scenario, but over all it was the best imputation method according to different evaluation criteria. CopFit and CopExp, the remaining R-vine copula based imputation methods, turned out not to perform as expected in some scenarios, which does not lead to an improvement for statistical analysis only based on complete cases. The simulation study also showed that it is worth using imputation to enlarge the dataset for the analyzing process. Particularly the marginal parameters significantly changed when data is not missing completely at random, while the dependence structure stayed the same.

Finally, a medical study with missing data was proposed to test the imputation methods under real conditions, first in 4 and later in 6 dimensions. Therefor a lot of different aspects were discussed that statisticians have to deal with in the absence of values in a data set, while using an R-vine copula based imputation method. One challenge is to find well fitting marginal models for transforming the data on $[0, 1]$ (U-level). In most situations the marginal transformation is not a huge obstacle, but a necessary step for modeling the dependence structure this way. Having data on the right scale, the proposed theory can be applied to fill out the incomplete data set. The next challenge is to find criteria for measuring the quality of imputed values.

Because of the observation that the dependence structure did not change when looking at data with and without missing values in the simulation study, the evaluation criteria if an imputation method performed better or worse were based on the dependence structure measured before and after imputation. According to these criteria, the proposed method CopReg managed the imputation quite well, but there was no great difference to the other imputation procedures. Unlike in the simulation study, one could not conclude that one method completely failed imputing the dataset.

The study was interesting and challenging in itself, because of the highly complex dependence structure between the different measurements. Prenatal lead measurements were found to have an influence on female children's weight in early childhood, while there is no significance for male. This influence only was measurable through the dependence structure, because marginally there was no difference in lead measurements for gender. This example justified the assumption of flexible (e.g. nonlinear) dependence structures that was done in the R-vine copula models.

In the end it depends on the data if it is worth performing imputation with an R-vine copula dependence model. It is a very flexible way of modeling and it was shown to lead to good results, but also requires more time and knowledge than some simpler, already existing imputation methods.

Bibliography

[1] Aas K., Czado C., Frigessi A. and Bakken H., (2009): *Pair-copula constructions of multiple dependence. Insurance: Mathematics and Economics 44 (2), 182-198.*

[2] Afeiche A., Peterson E. K., Sánchez N. B., Cantonwine D., Lamadrid-Figueroa H., Schnaas L., Ettinger S. A., Hernández-Avila M., Hu H., Téllez-Rojo M. M., (2011): *Prenatal Lead Exposure and Weight of 0- to 5-Year-Old Children in Mexico City. Environ Health Perspect 119, 1436-1441.*

[3] Bauer A. X., (2013): *Pair-copula constructions for non-Gaussian Bayesian networks. Submitted for publication.*

[4] Death Penalty Information Center, (2013): *www.deathpenaltyinfo.org/execution-list-2013.*

[5] Dissmann J., Brechmann E. C., Czado C., Kurowicka D., (2013): *Selecting and estimating regular vine copulae and application to financial returns. Computational Statistics and Data Analysis 59, 5269.*

[6] Joe H., (1996): *Families of m-variate distributions with given margins and m(m-1)/2 bivariate dependence parameters. In L. Rüschendorf and B. Schweizer and M. D. Taylor (Ed.), Distributions with Fixed Marginals and Related Topics..*

[7] Kurowicka D. and Cooke R., (2006): *Uncertainty Analysis with High Dimensional Dependence Modelling. Chichester: John Wiley & Sons.*

[8] Little R. J. A., (1987): *Statistical Analysis With Missing Data. New York: Wiley series in probability and mathematical statistics.*

[9] Morales-Napoles O., Cooke R., Kurowicka D., (2010): *About the number of vines and regular vines on n nodes. Submitted for publication.*

[10] Rao J. N. K. and Shao J., (1992): *Jackknife variance estimation with survey data under hot deck imputation. Biometrika, 79, 811-822.*

[11] Rubin D. B., (1976): *Inference and missing data. Biometrika, 63, 581-592.*

[12] Rubin D. B., (1987): *Multiple Imputation for Nonresponse in Surveys. New York: Wiley series in probability and mathematical statistics.*

Appendix A

Test via Simulation

A.1 Box Plot

In the following, the box plot diagrams according to the simulation test are presented. For each situation, each parameter measured is observable for every iteration. So the performance of the different methods is visually comparable.

A.1.1 MCAR, High (Kendall's tau values)

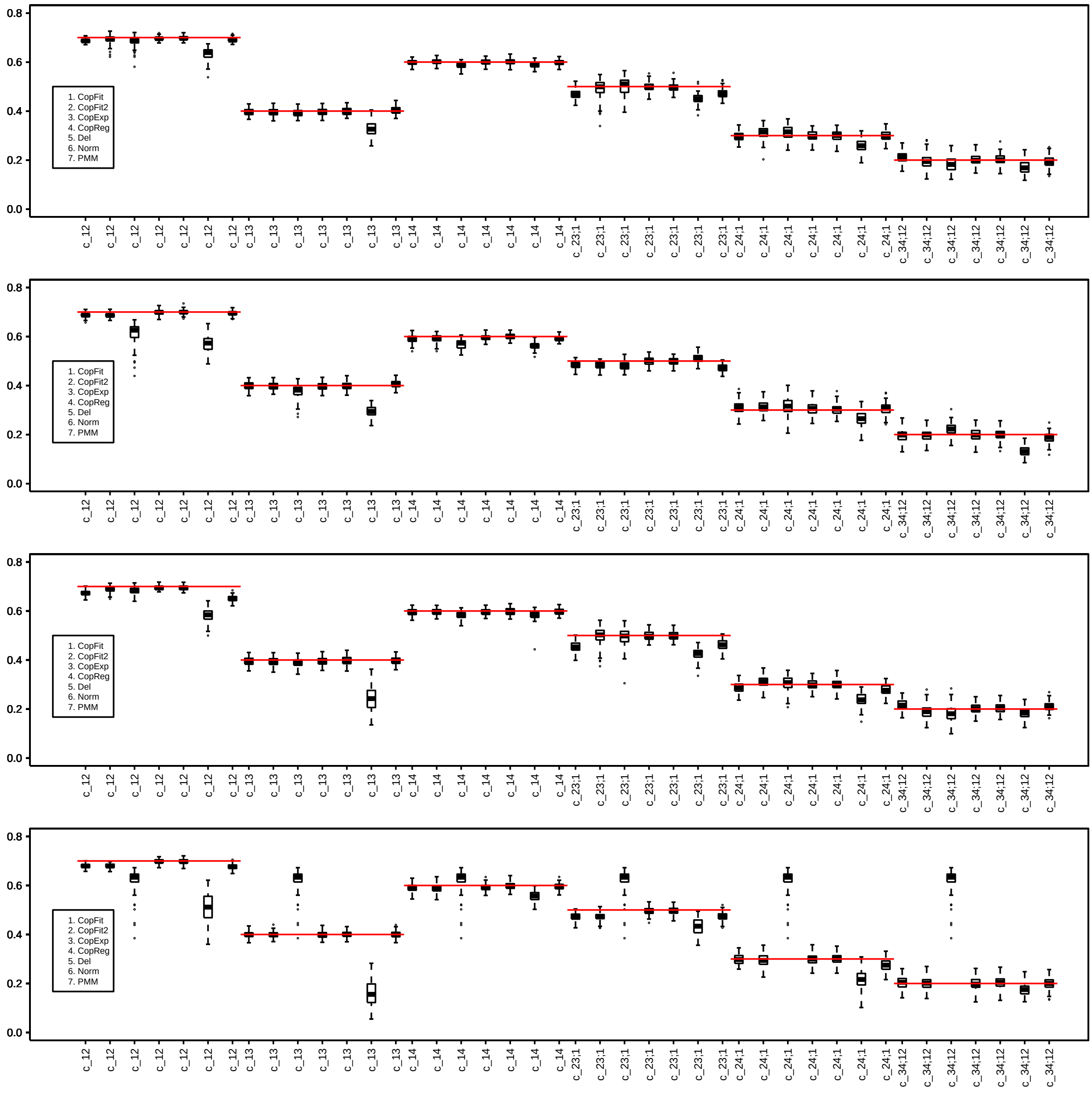


Table A.1: Fix scenarios: 1b, 3a, 4a, 6a. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

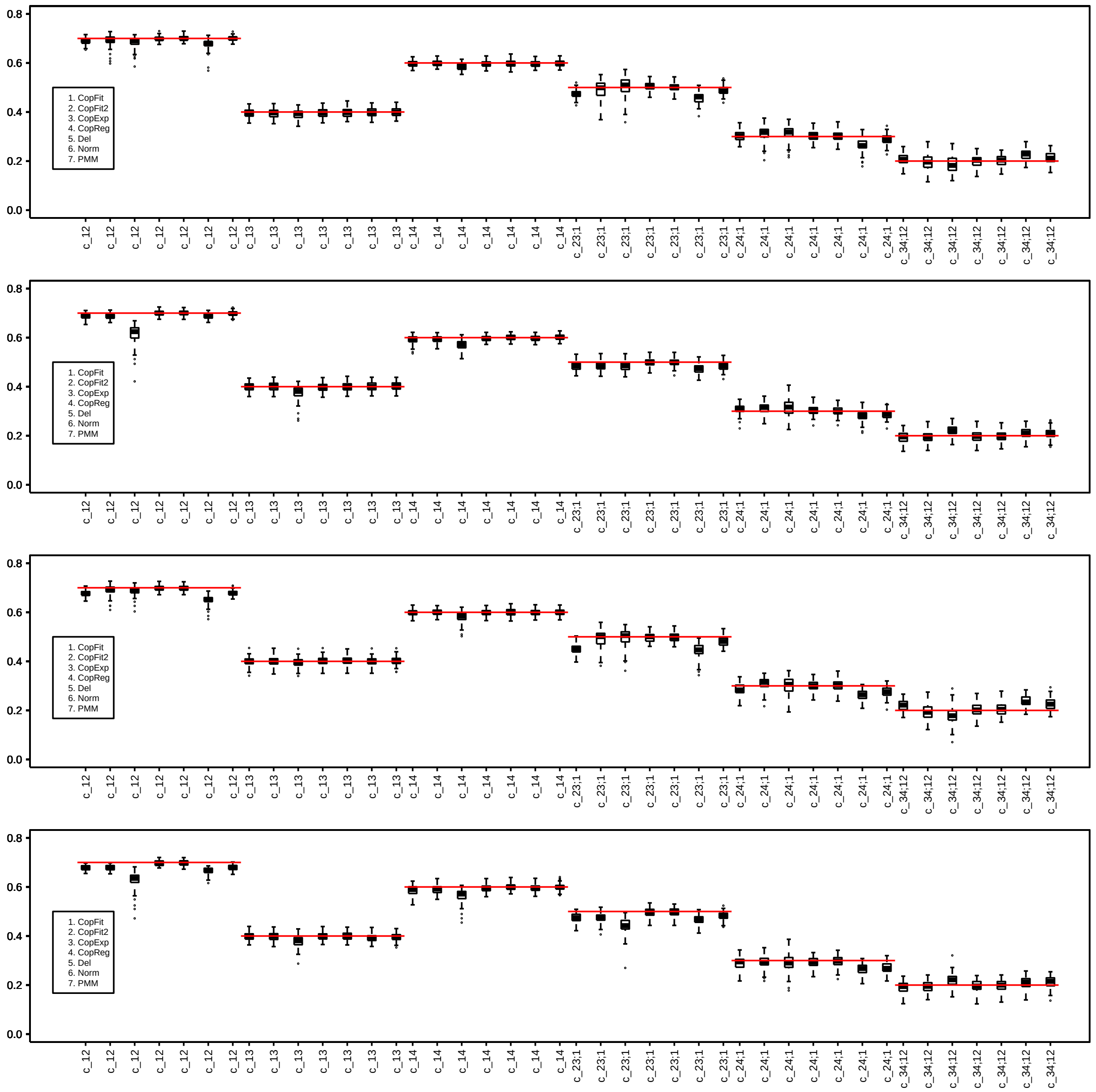


Table A.2: Fix scenarios: 1b, 3a, 4b, 6a. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

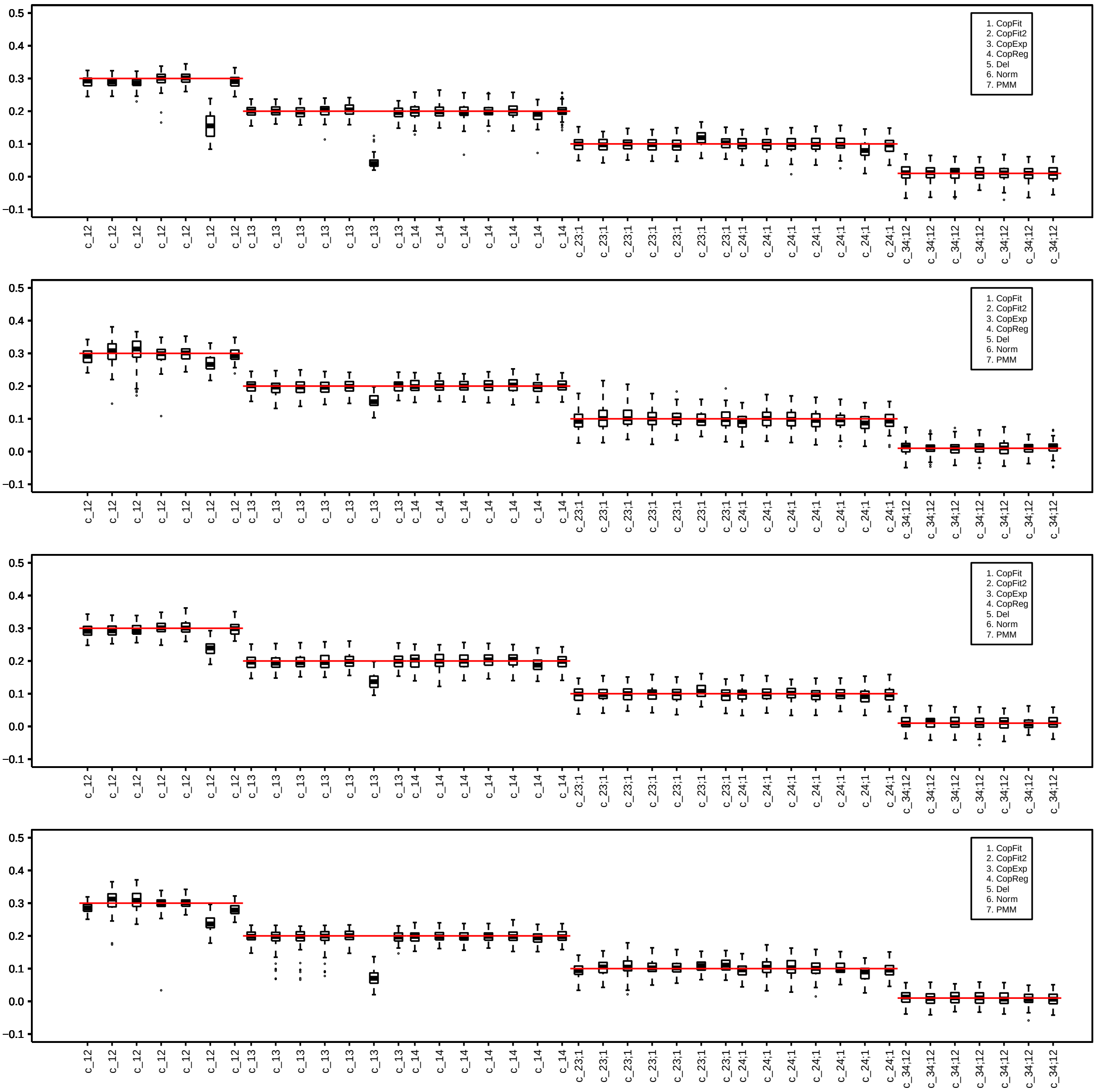


Table A.3: Fix scenarios: 1b, 3a, 4a, 6b. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

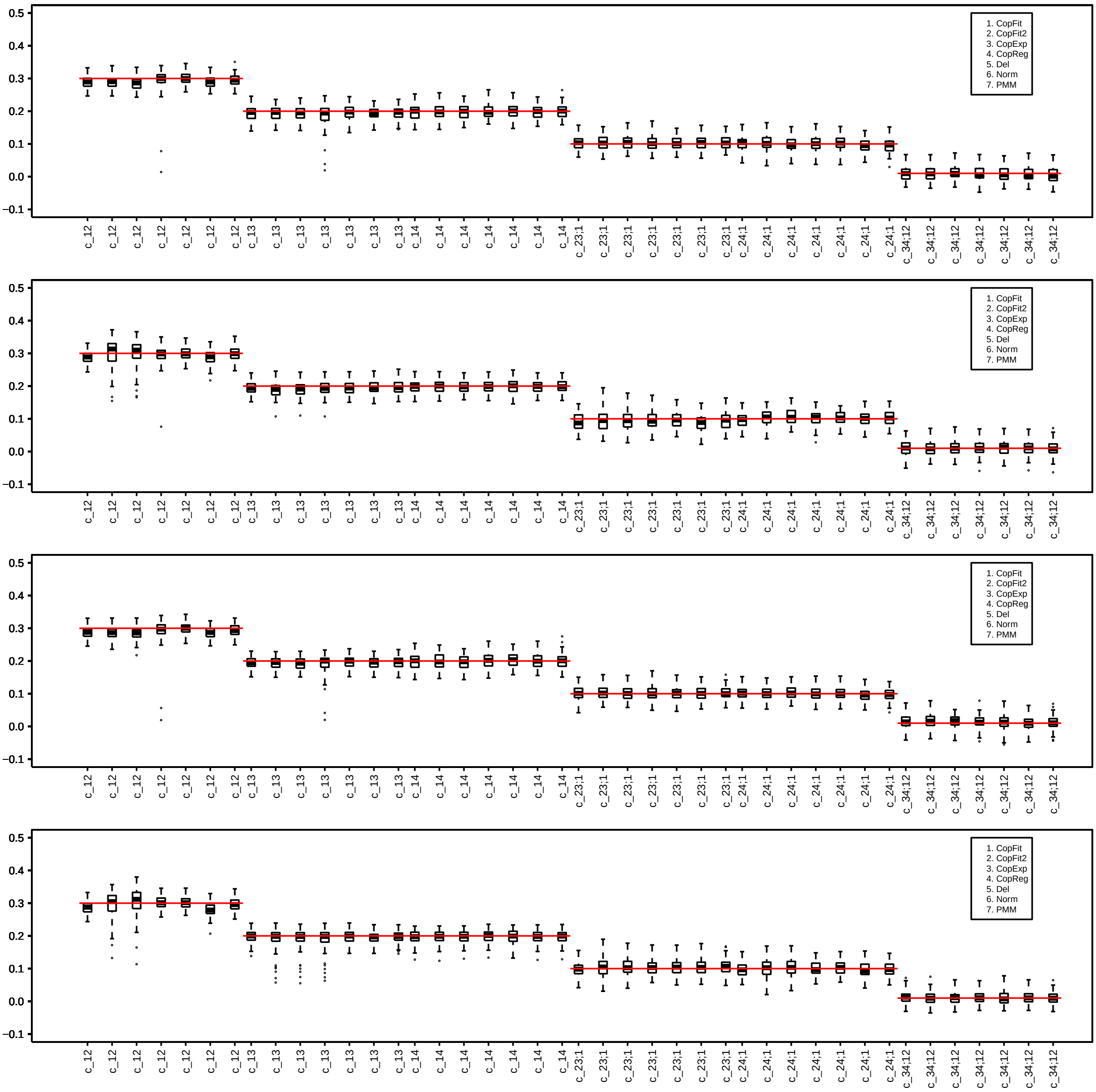


Table A.4: Fix scenarios: 1b, 3a, 4b, 6b. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

A.1.2 MCAR, Low (Kendall's tau values)

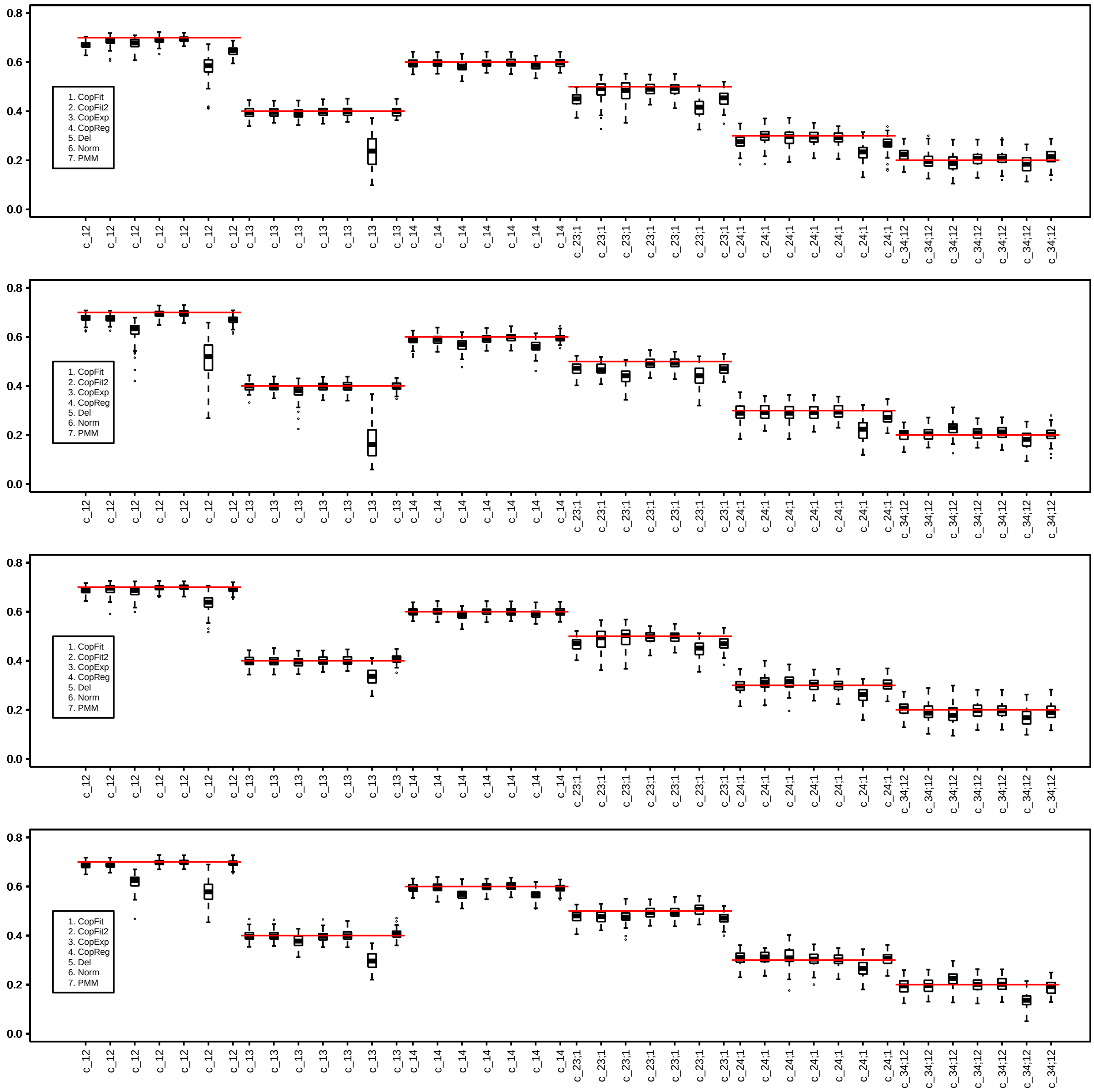


Table A.5: Fix scenarios: 1a, 3a, 4a, 6a. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

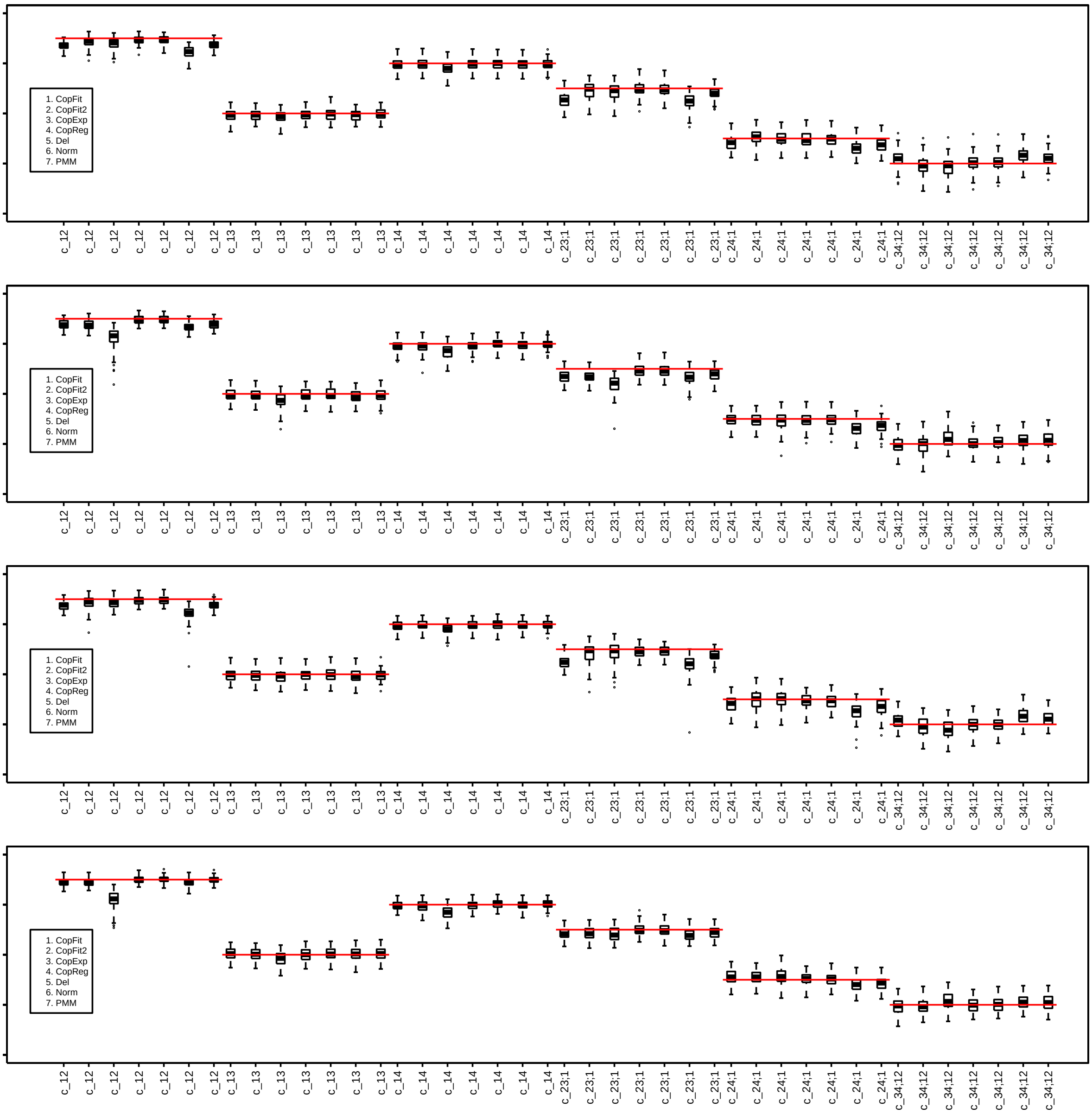


Table A.6: Fix scenarios: 1a, 3a, 4b, 6a. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

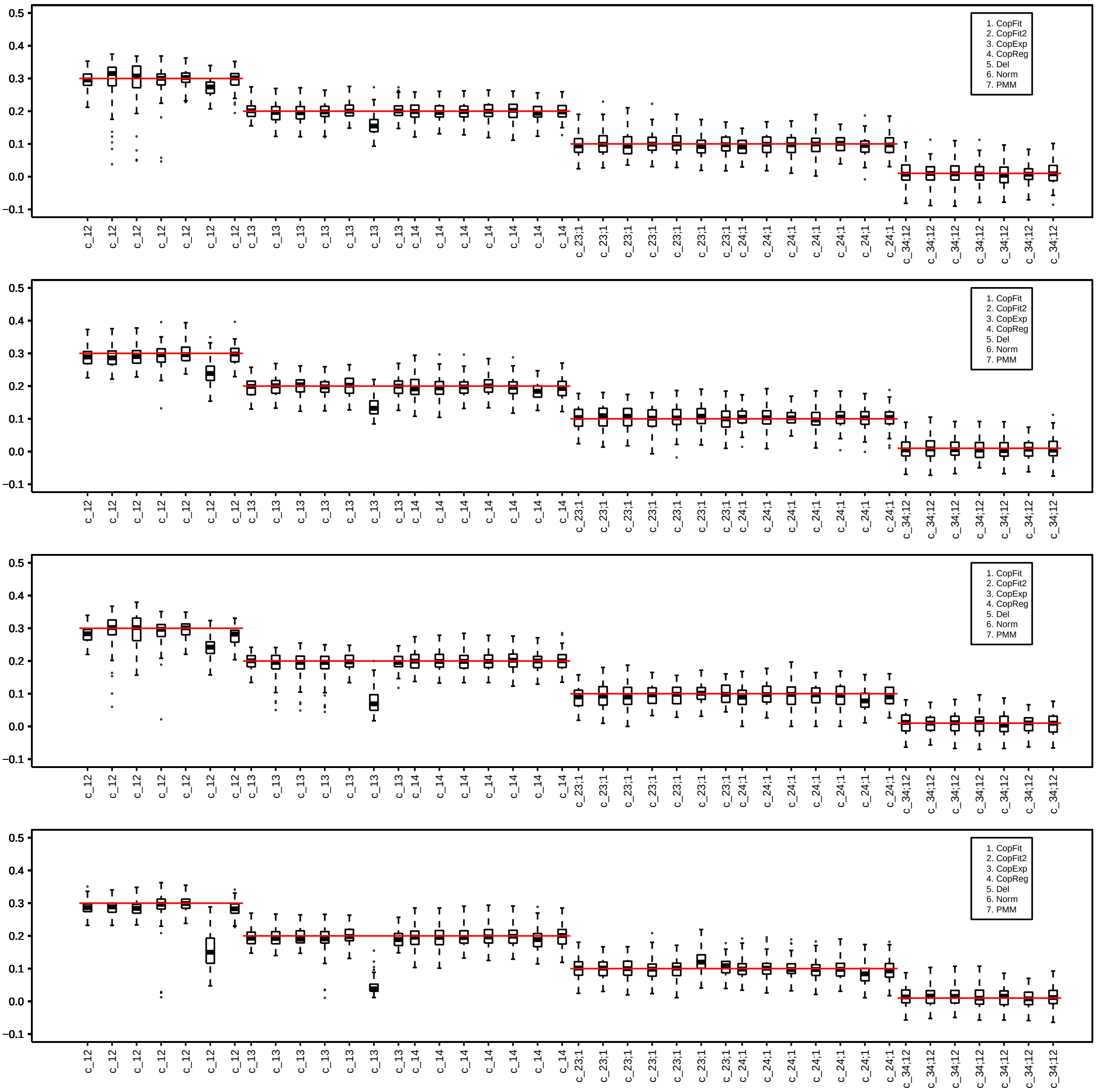


Table A.7: Fix scenarios: 1b, 3a, 4a, 6b. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

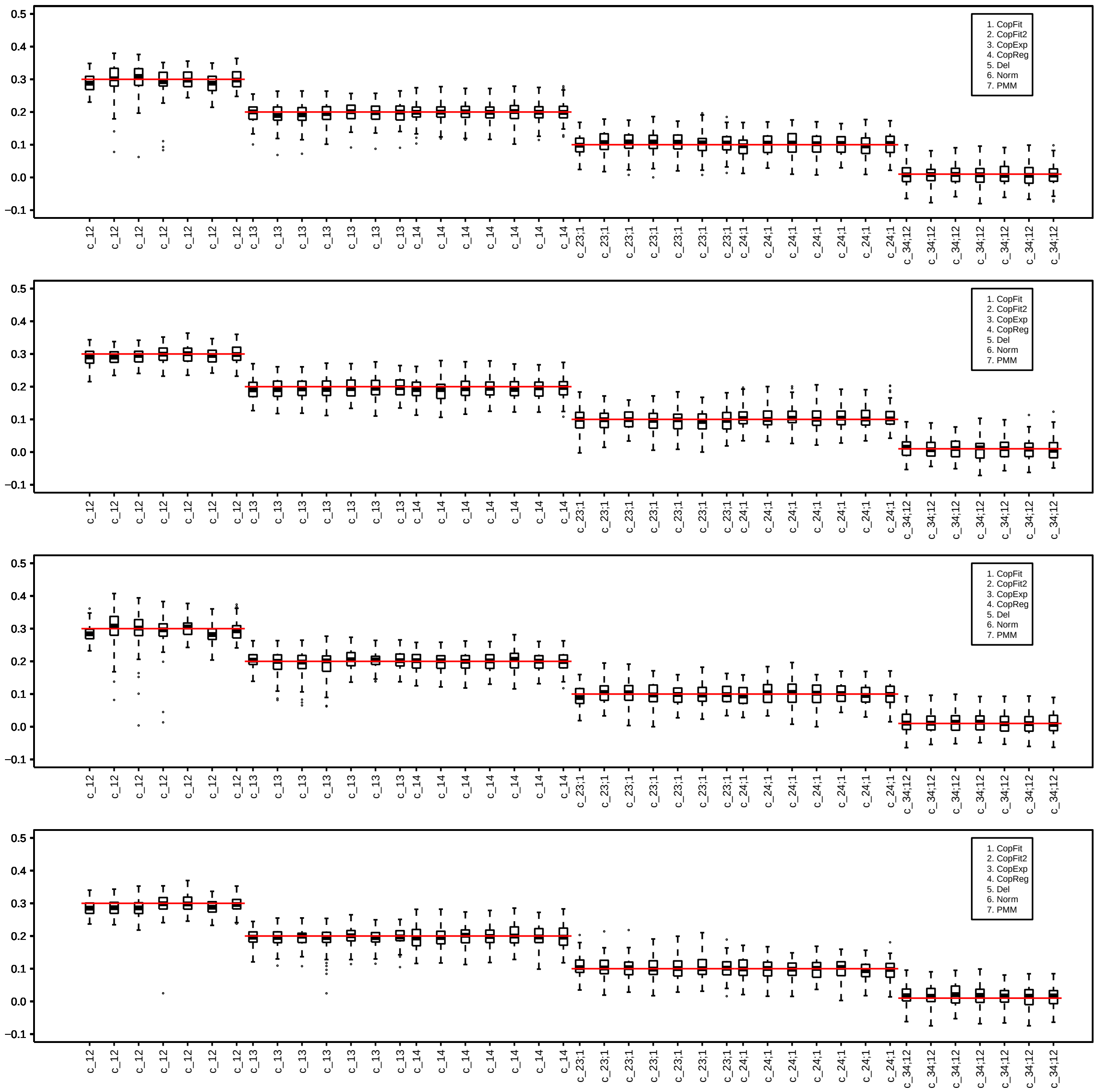


Table A.8: Fix scenarios: 1b, 3a, 4b, 6b. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

A.1.3 MAR, High (Kendall's tau values)

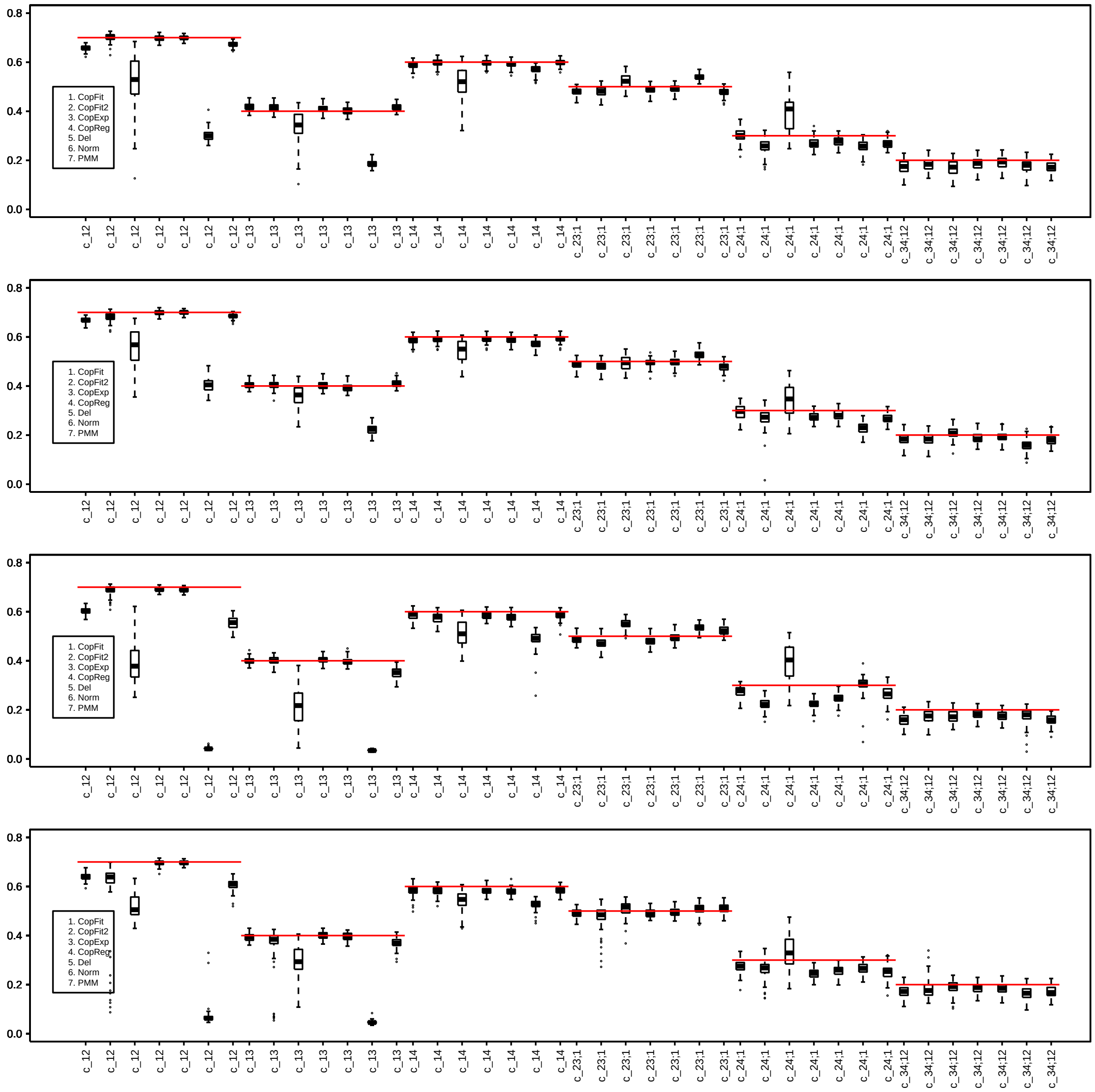


Table A.9: Fix scenarios: 1b, 3a, 4a, 6a. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

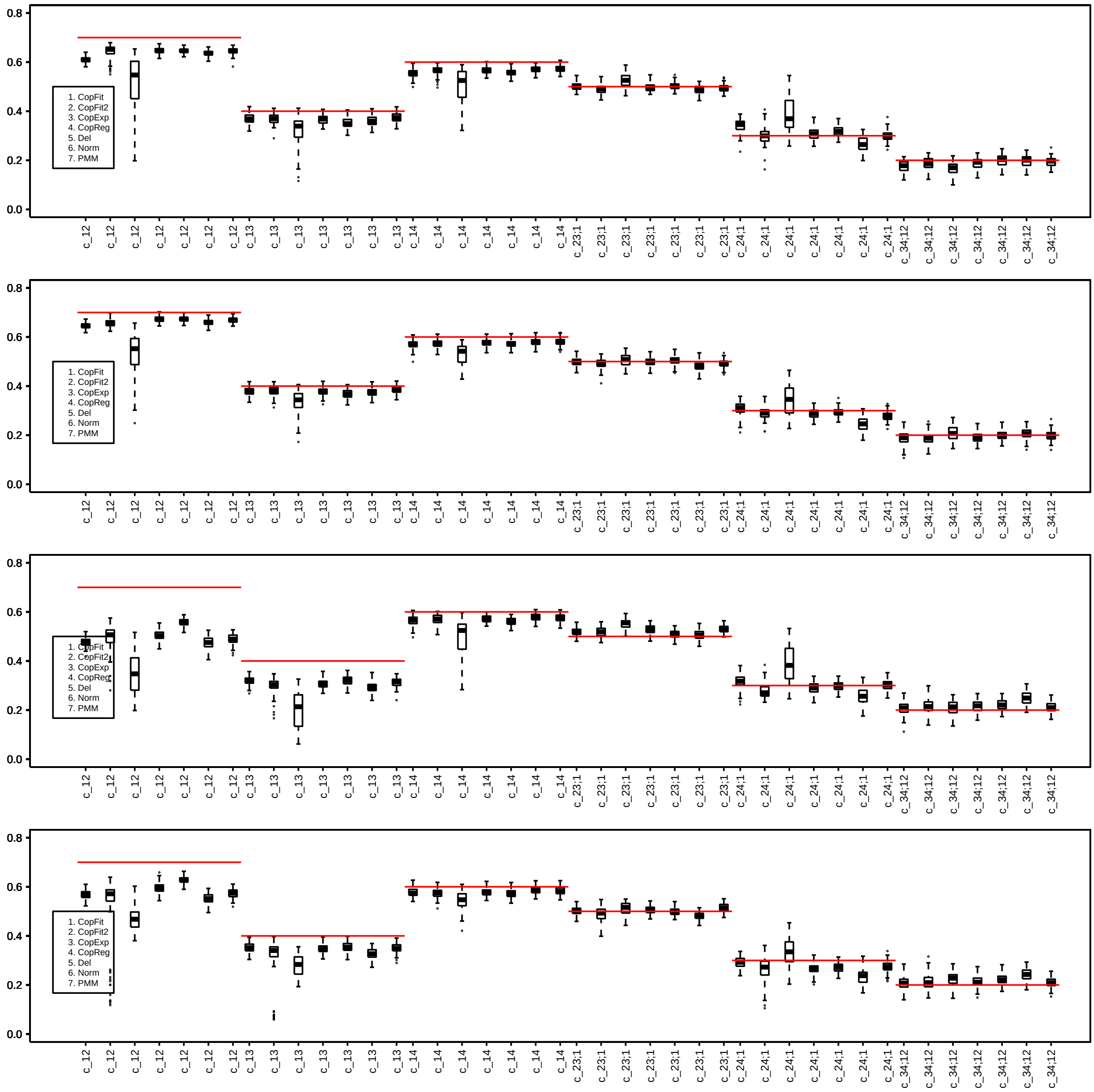


Table A.10: Fix scenarios: 1b, 3b, 4b, 6a. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

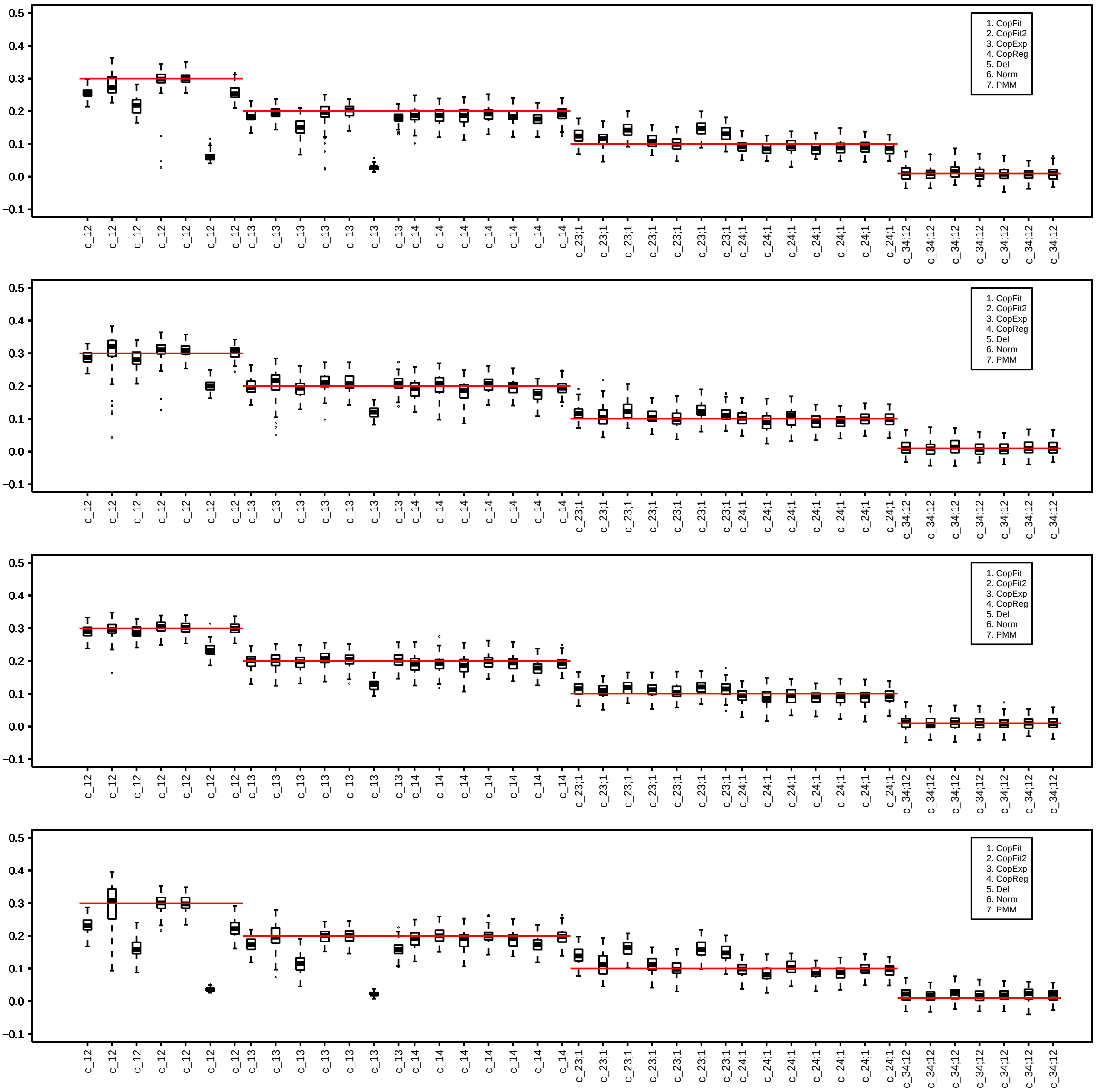


Table A.11: Fix scenarios: 1b, 3b, 4a, 6b. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

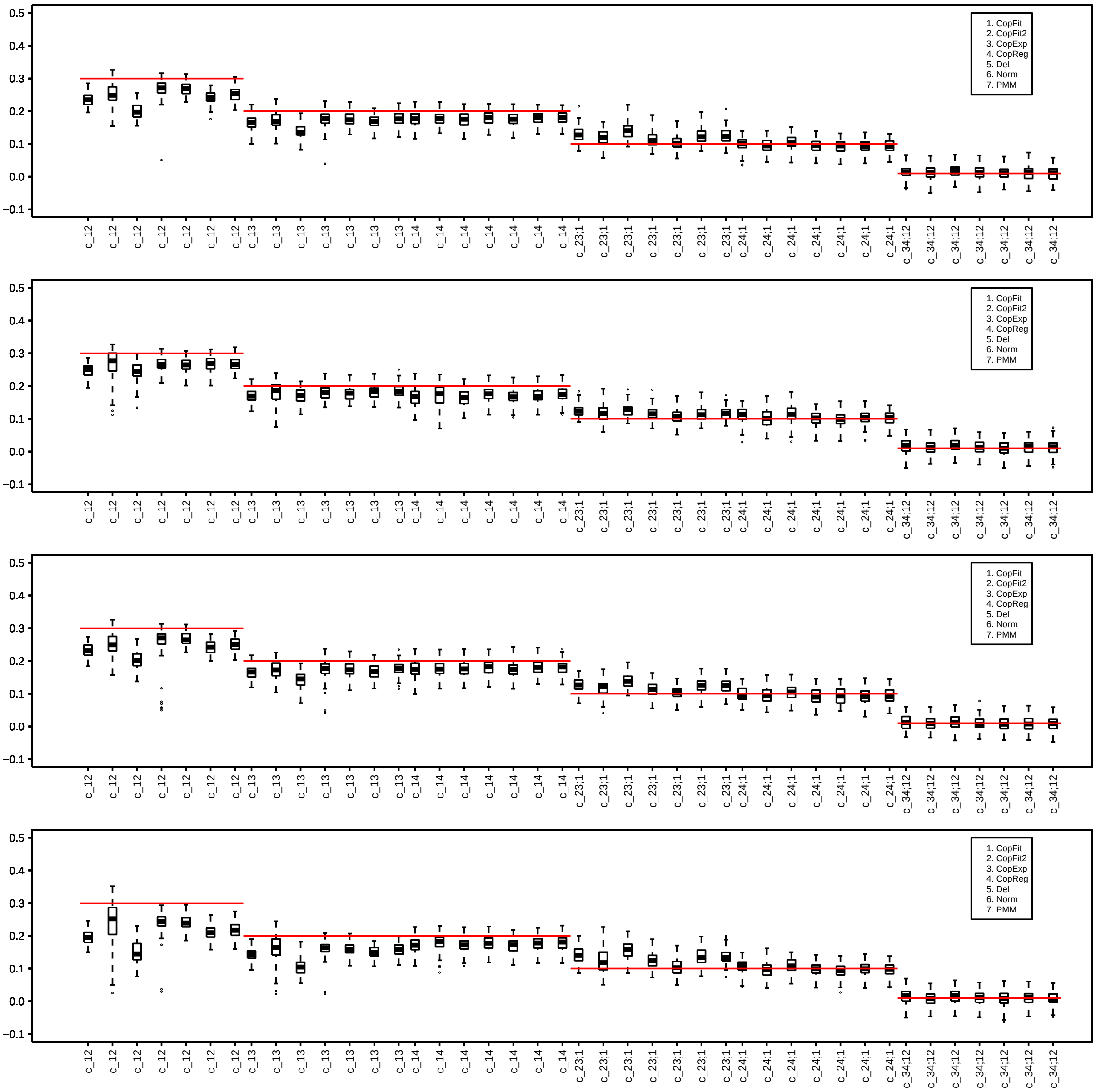


Table A.12: Fix scenarios: 1b, 3b, 4b, 6b. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

A.1.4 MAR, Low (Kendall's tau values)

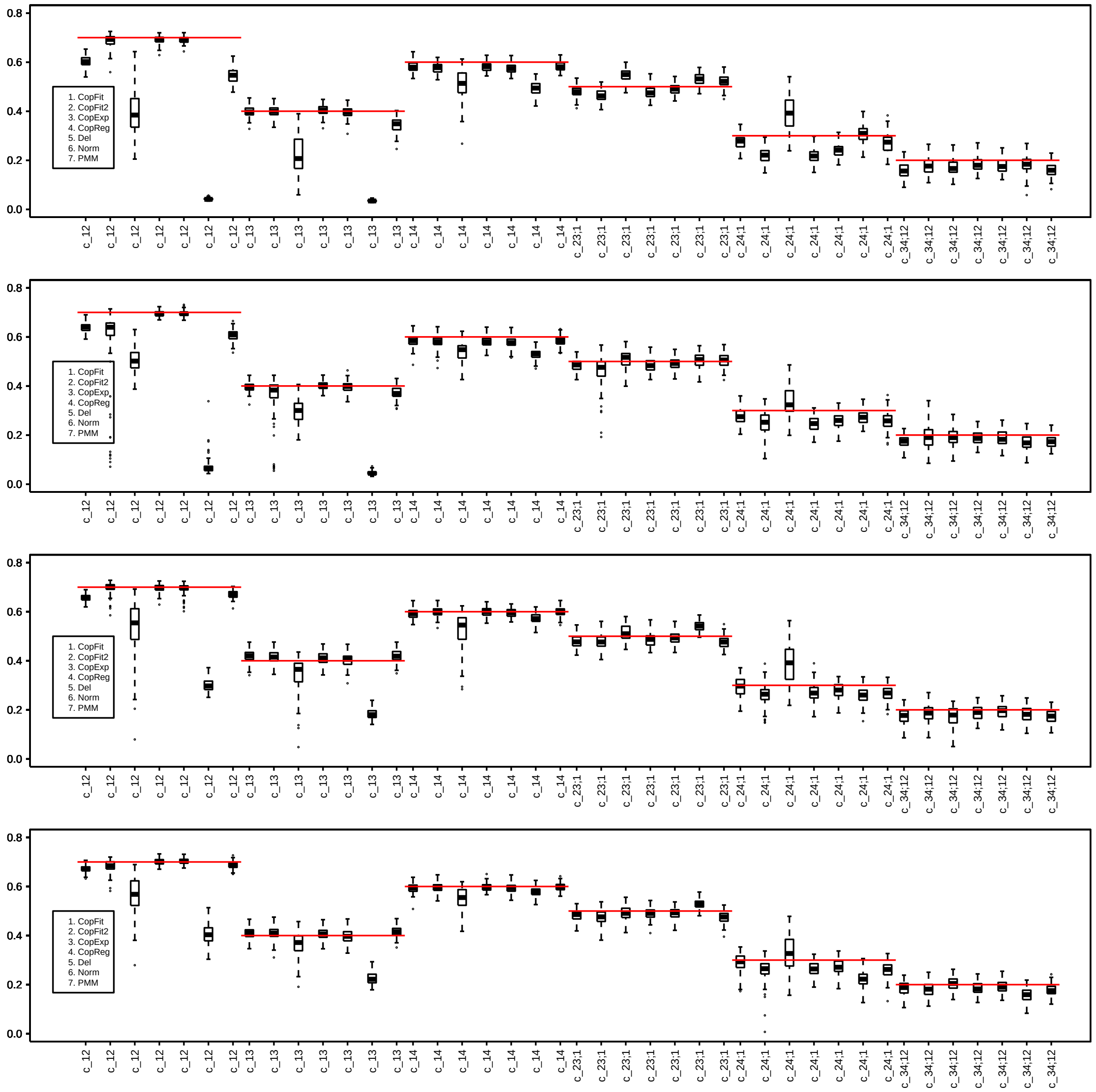


Table A.13: Fix scenarios: 1a, 3b, 4a, 6a. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

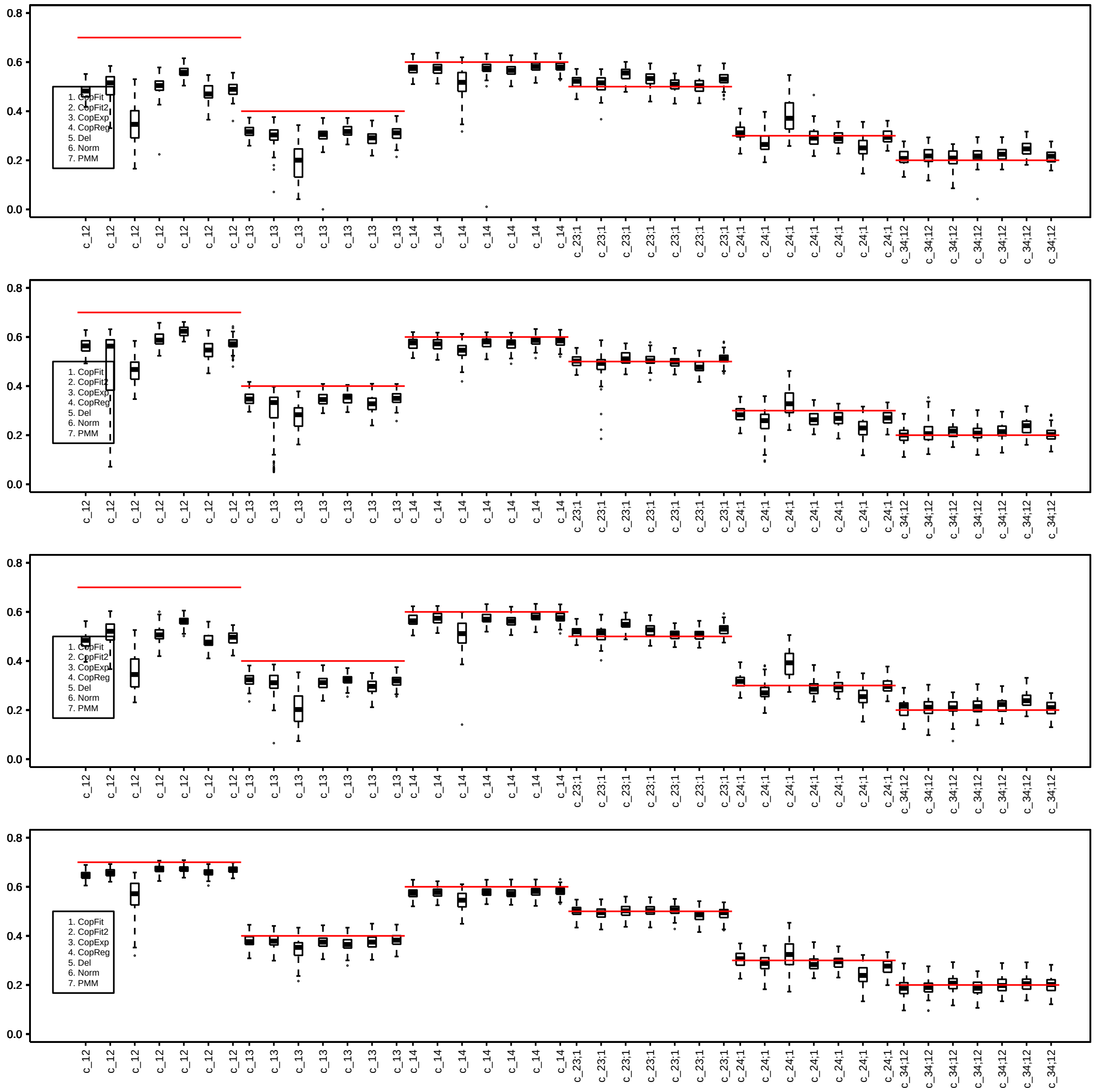


Table A.14: Fix scenarios: 1a, 3b, 4b, 6a. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

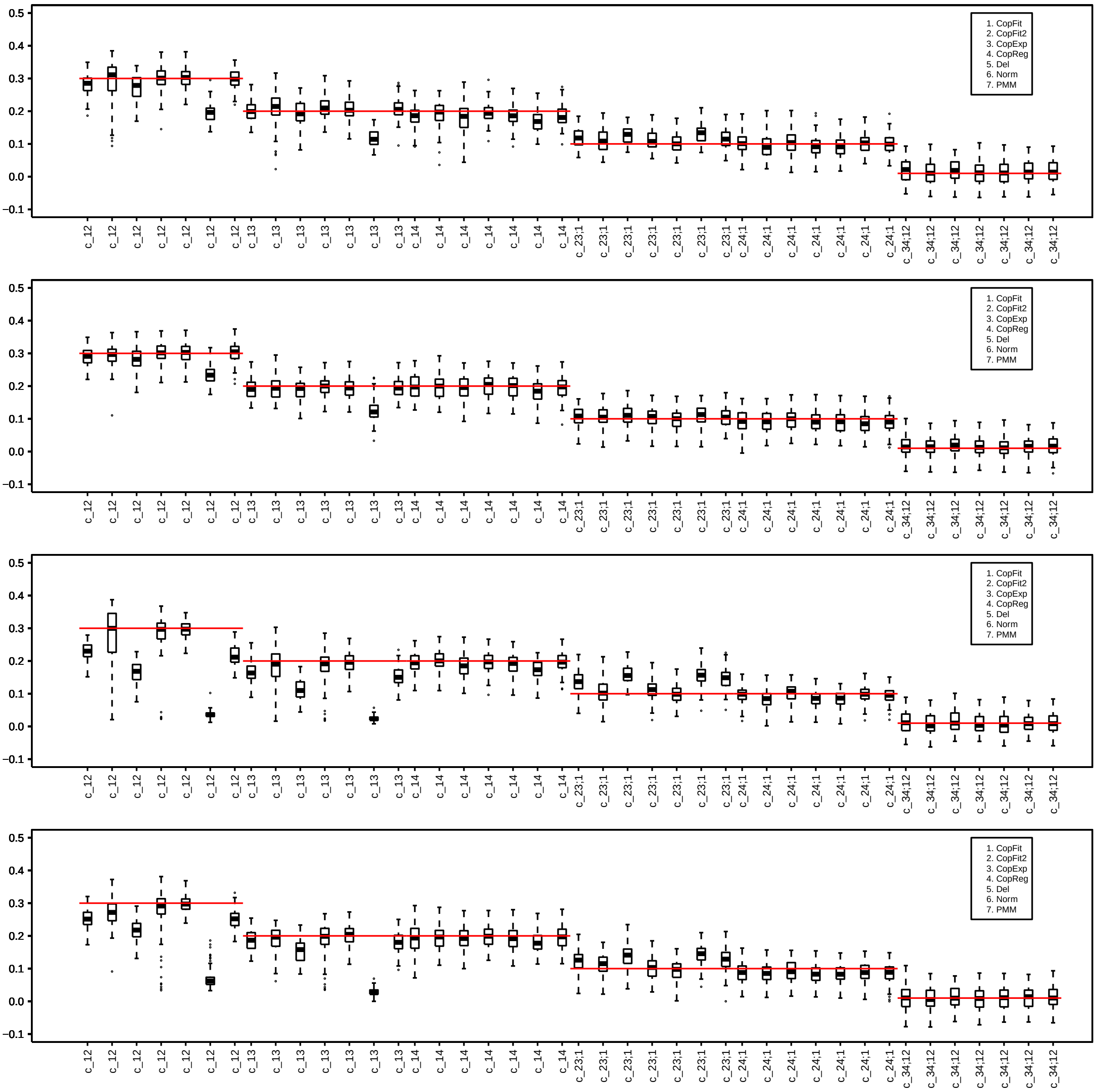


Table A.15: Fix scenarios: 1b, 3b, 4a, 6b. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

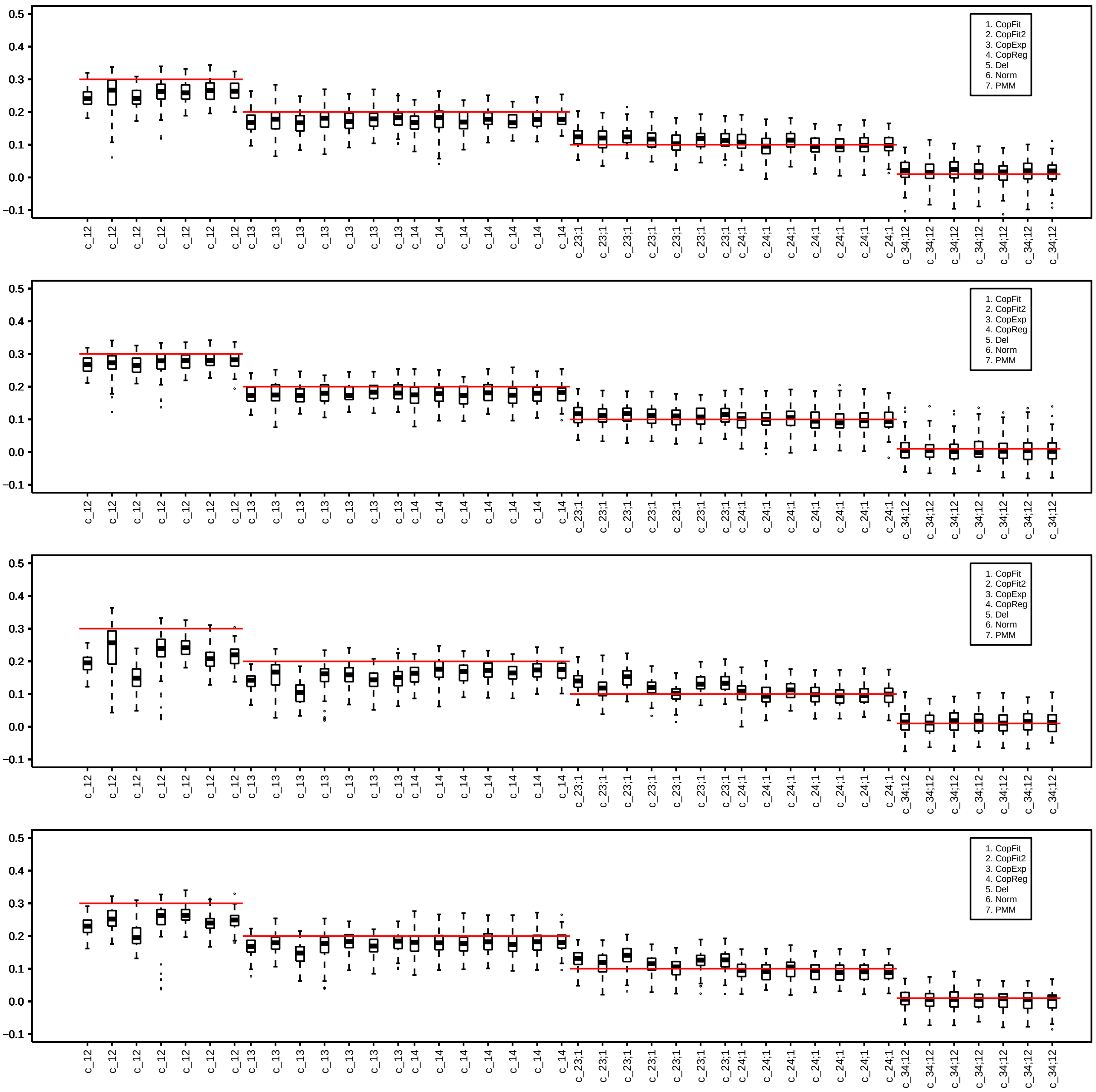


Table A.16: Fix scenarios: 1b, 3b, 4b, 6b. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

A.1.5 MAR, Exp&t-dist (Marginal para)

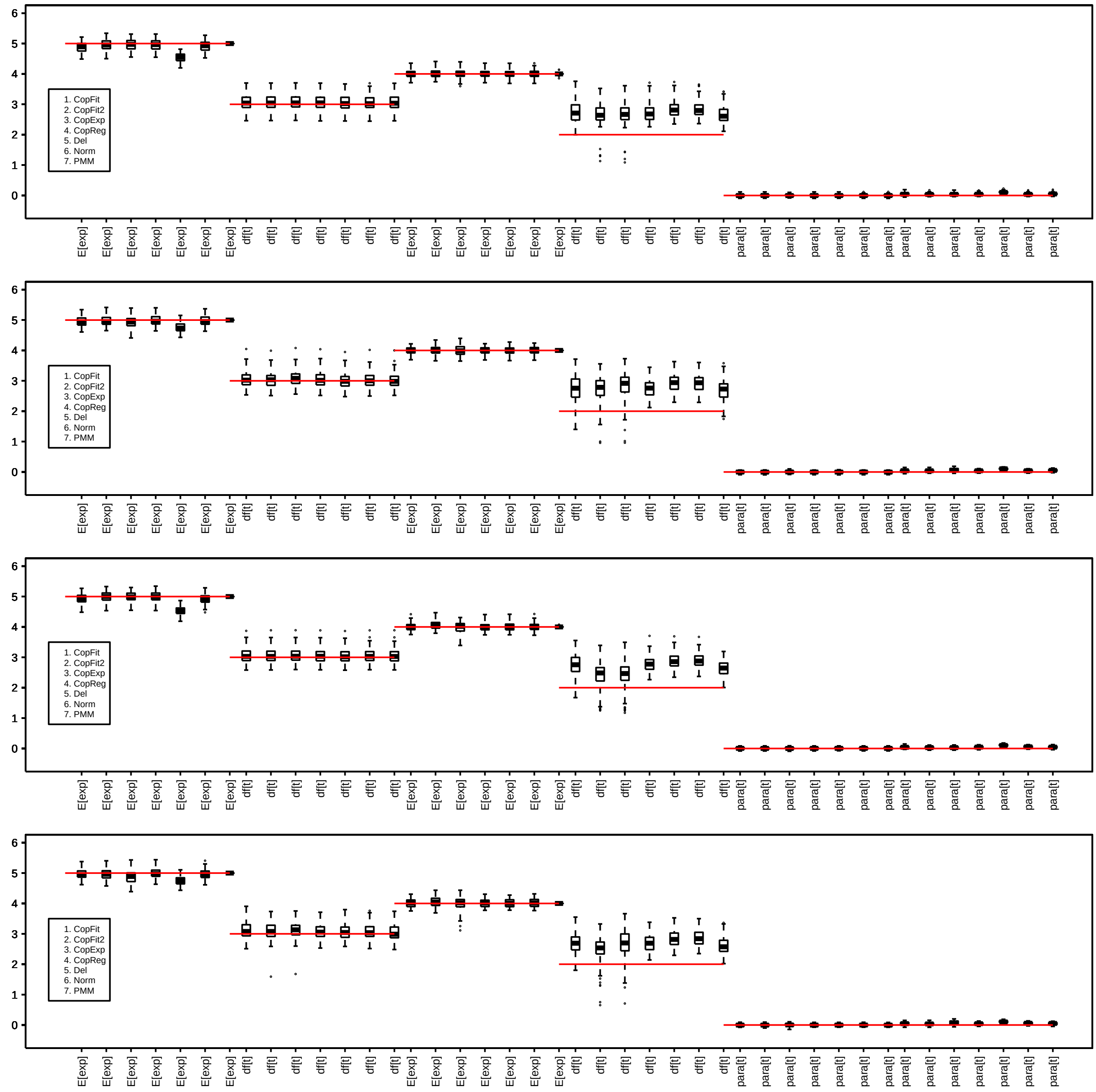


Table A.17: Fix scenarios: 1b, 3a, 4a, 6a. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

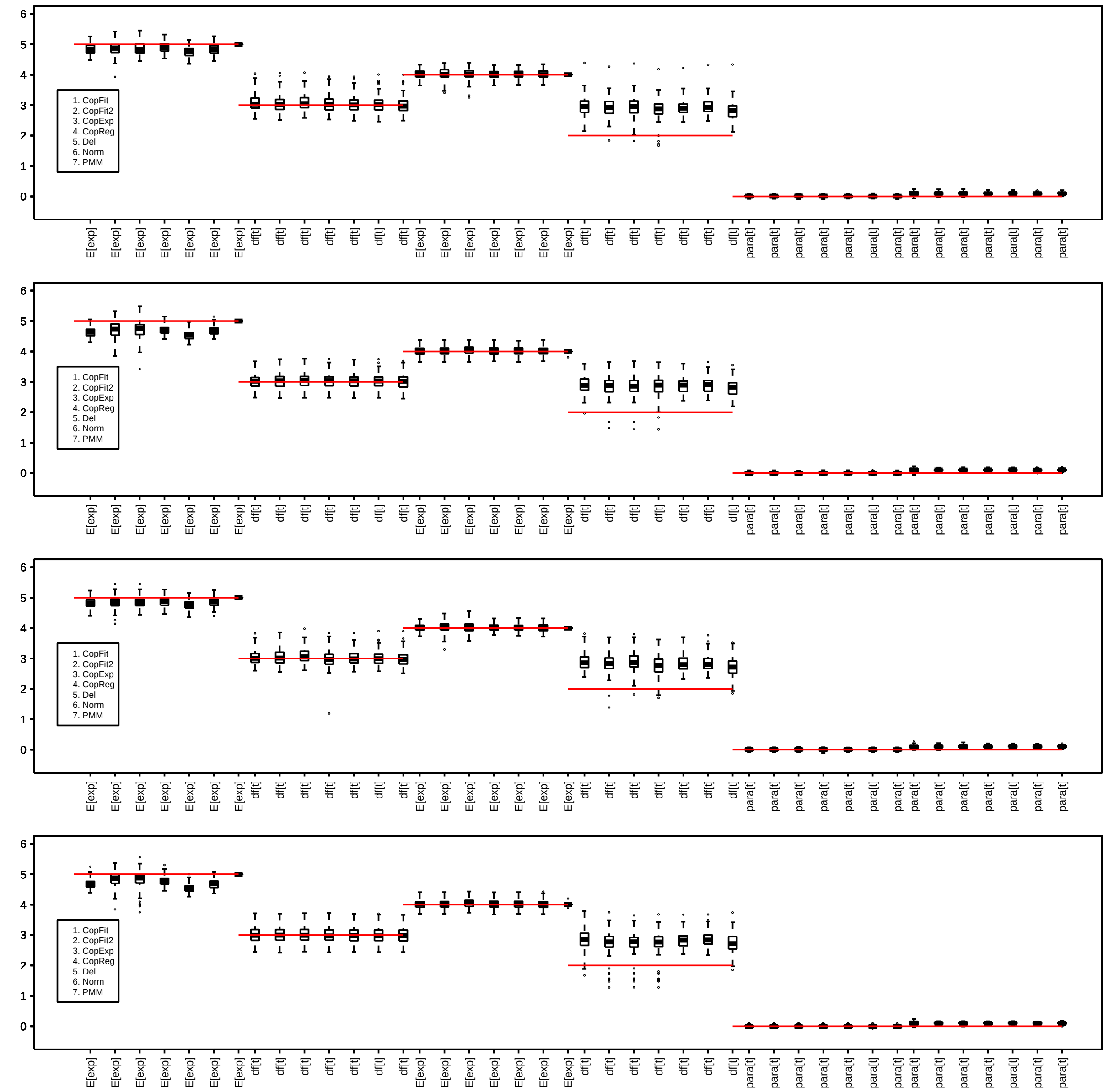


Table A.18: Fix scenarios: 1b, 3b, 4b, 6a. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

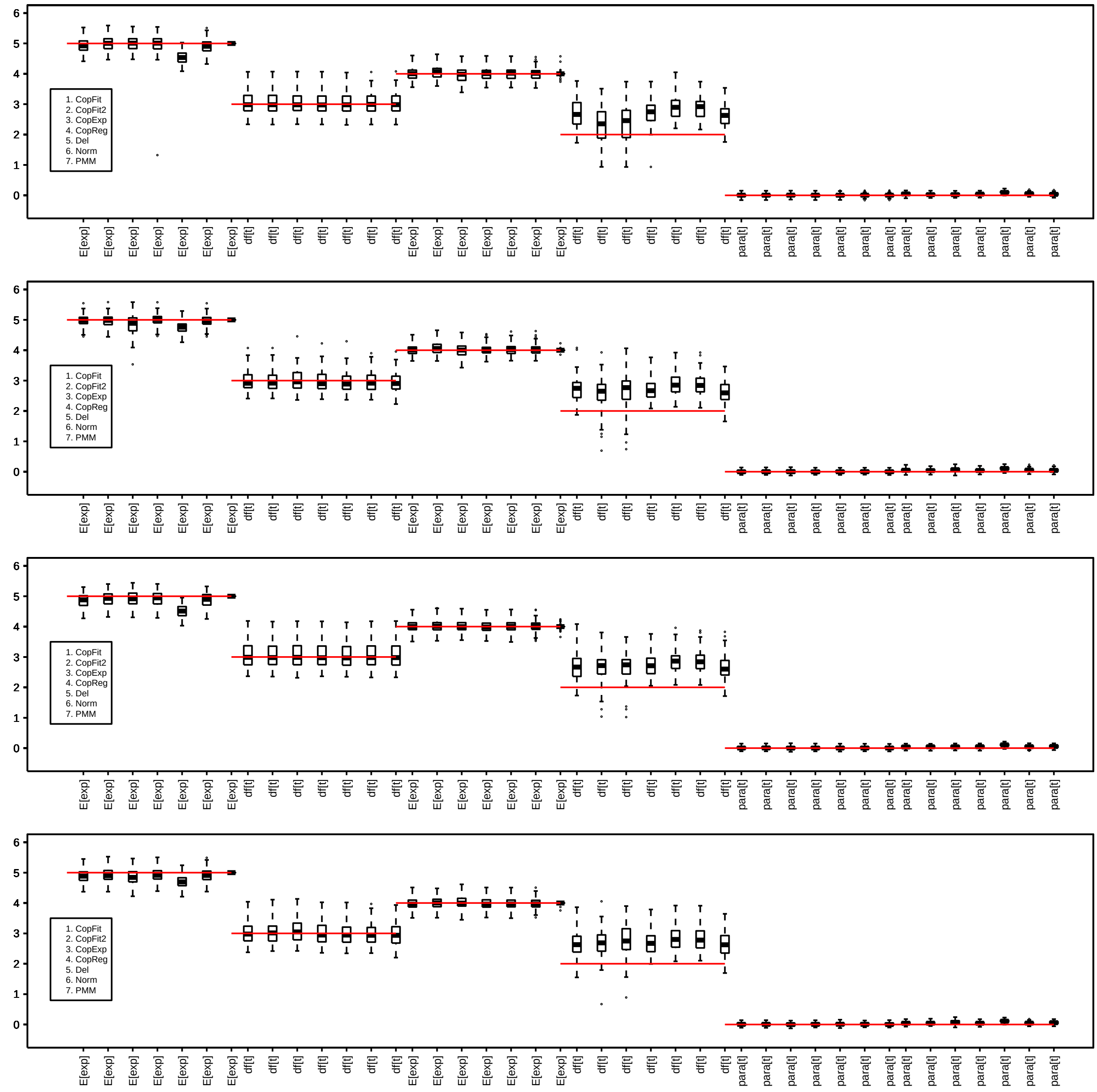


Table A.19: Fix scenarios: 1b, 3b, 4a, 6b. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

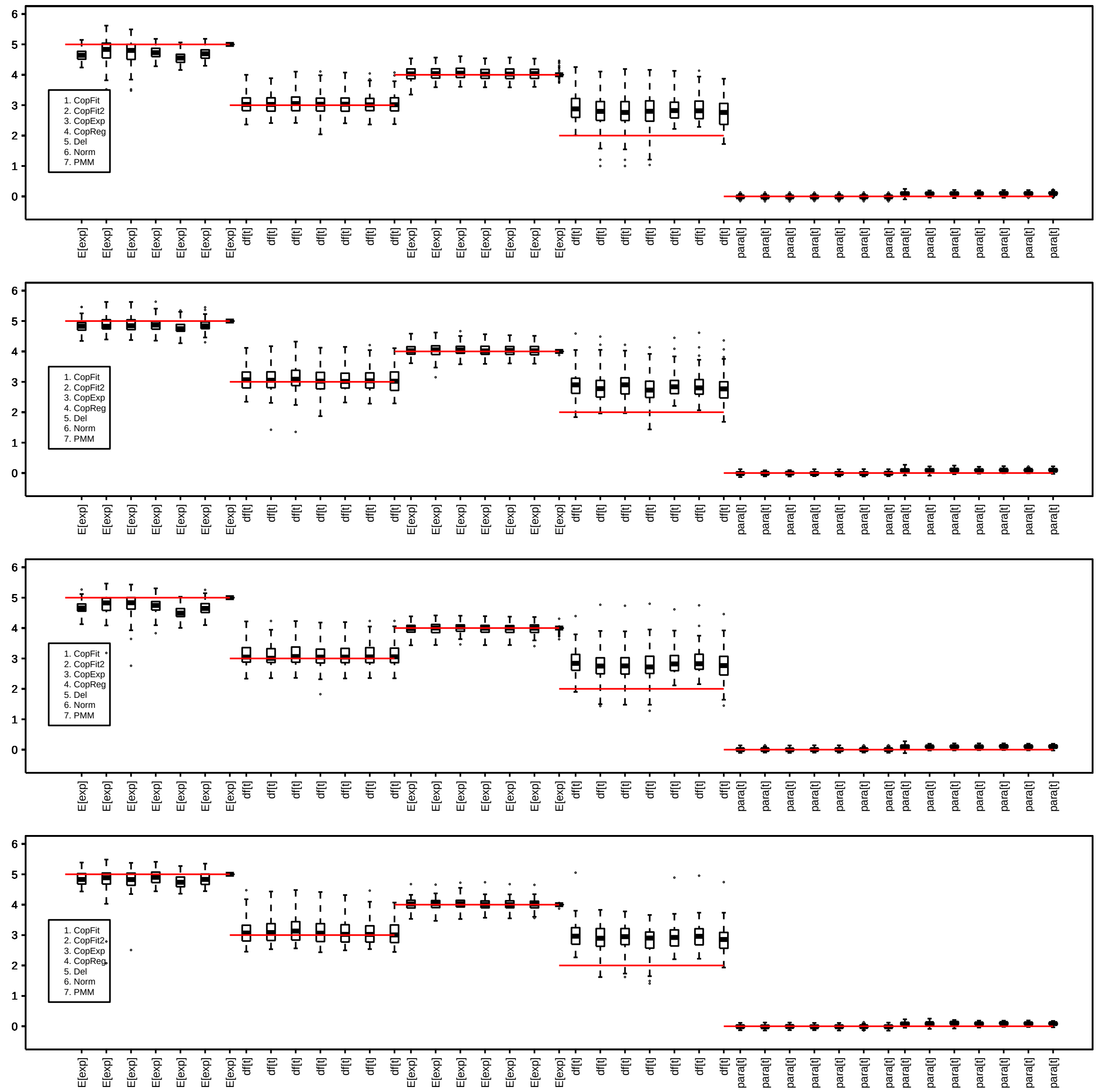


Table A.20: Fix scenarios: 1b, 3b, 4b, 6b. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

A.1.6 MAR, Norm&t-dist (Marginal para)

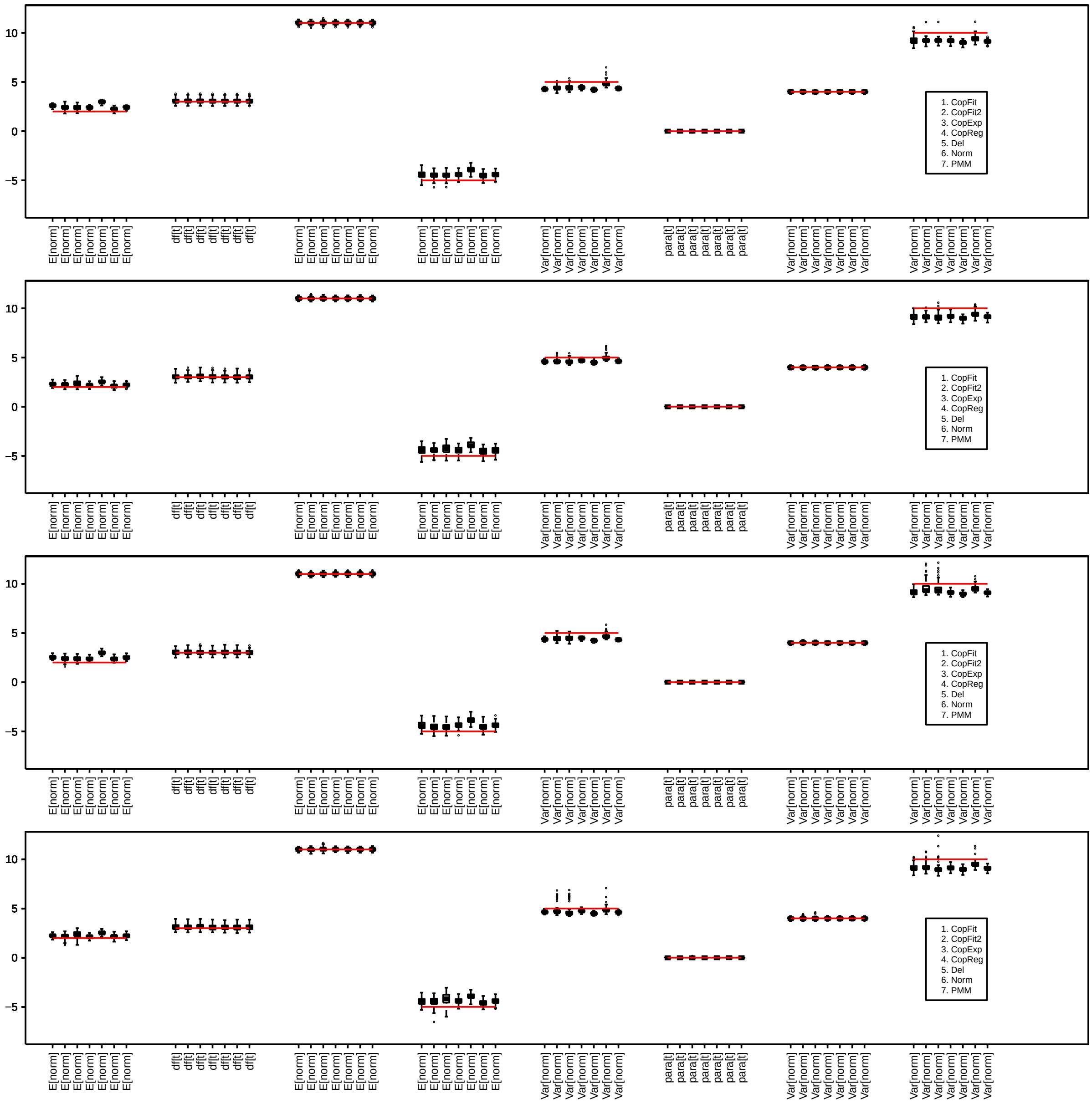


Table A.21: Fix scenarios: 1a, 3b, 4a, 6a. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

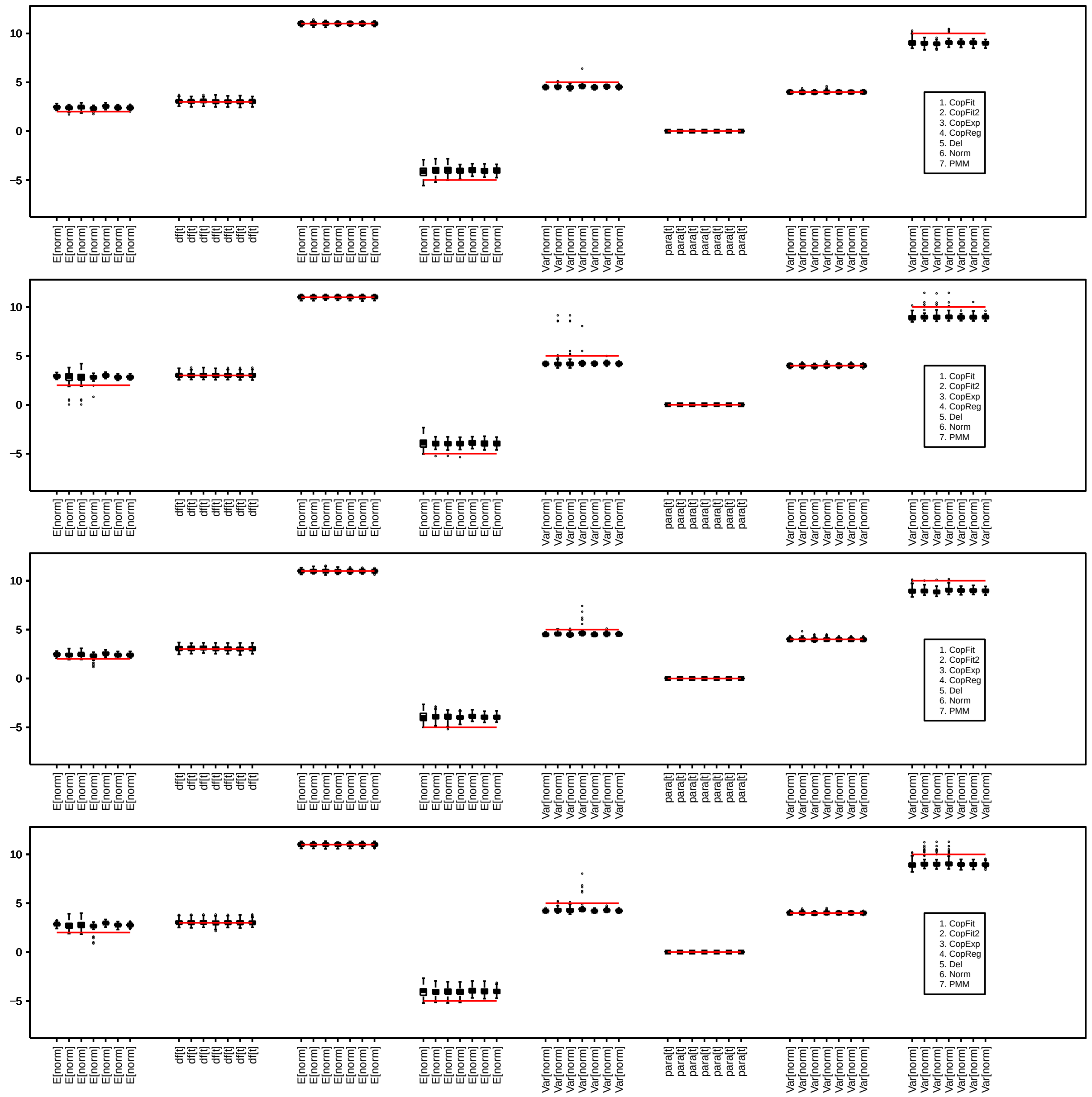


Table A.22: Fix scenarios: 1a, 3b, 4b, 6a. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

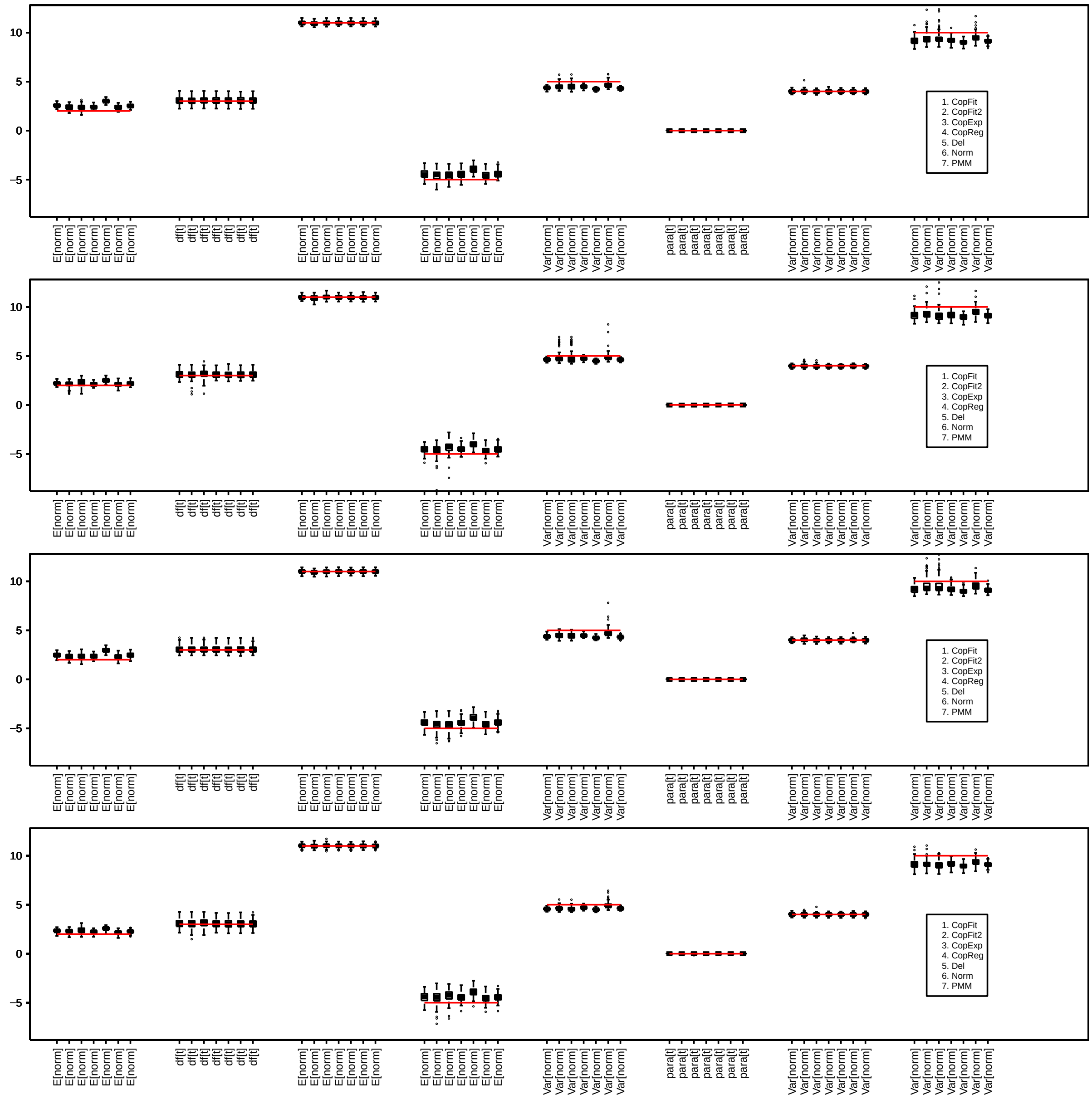


Table A.23: Fix scenarios: 1b, 3b, 4a, 6b. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

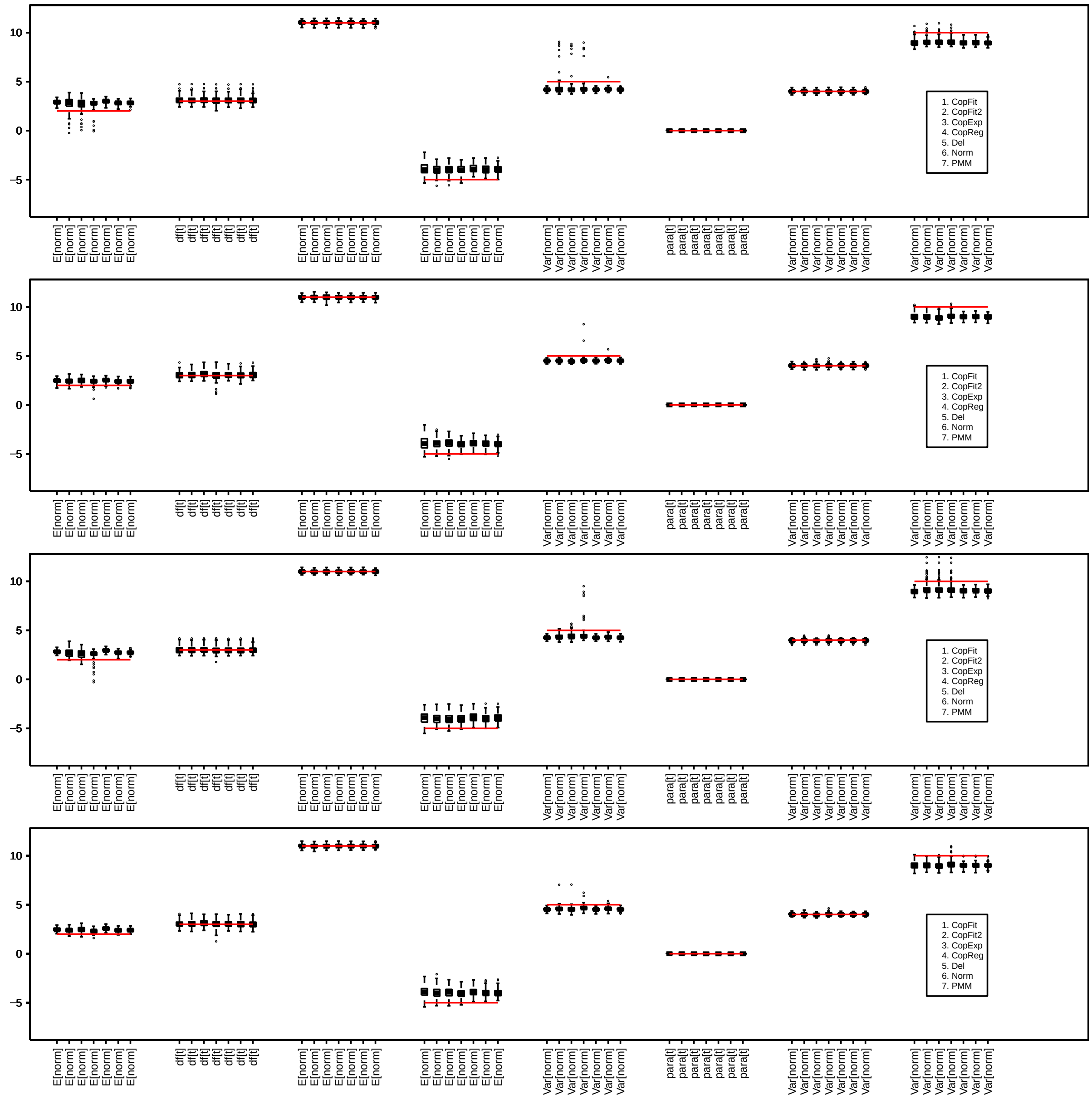


Table A.24: Fix scenarios: 1b, 3b, 4b, 6b. From the top to the bottom: 1) 2b,5b; 2) 2a,5b; 3) 2b,5a; 4) 2a,5a

A.2 Contours

Further we tried to compare the success rate for each method with the help of level plots. So for each method in every simulation situation, the empirical mean, the empirical 5%, and the empirical 95% quantile of the level curves were plotted. So the procedure is the following:

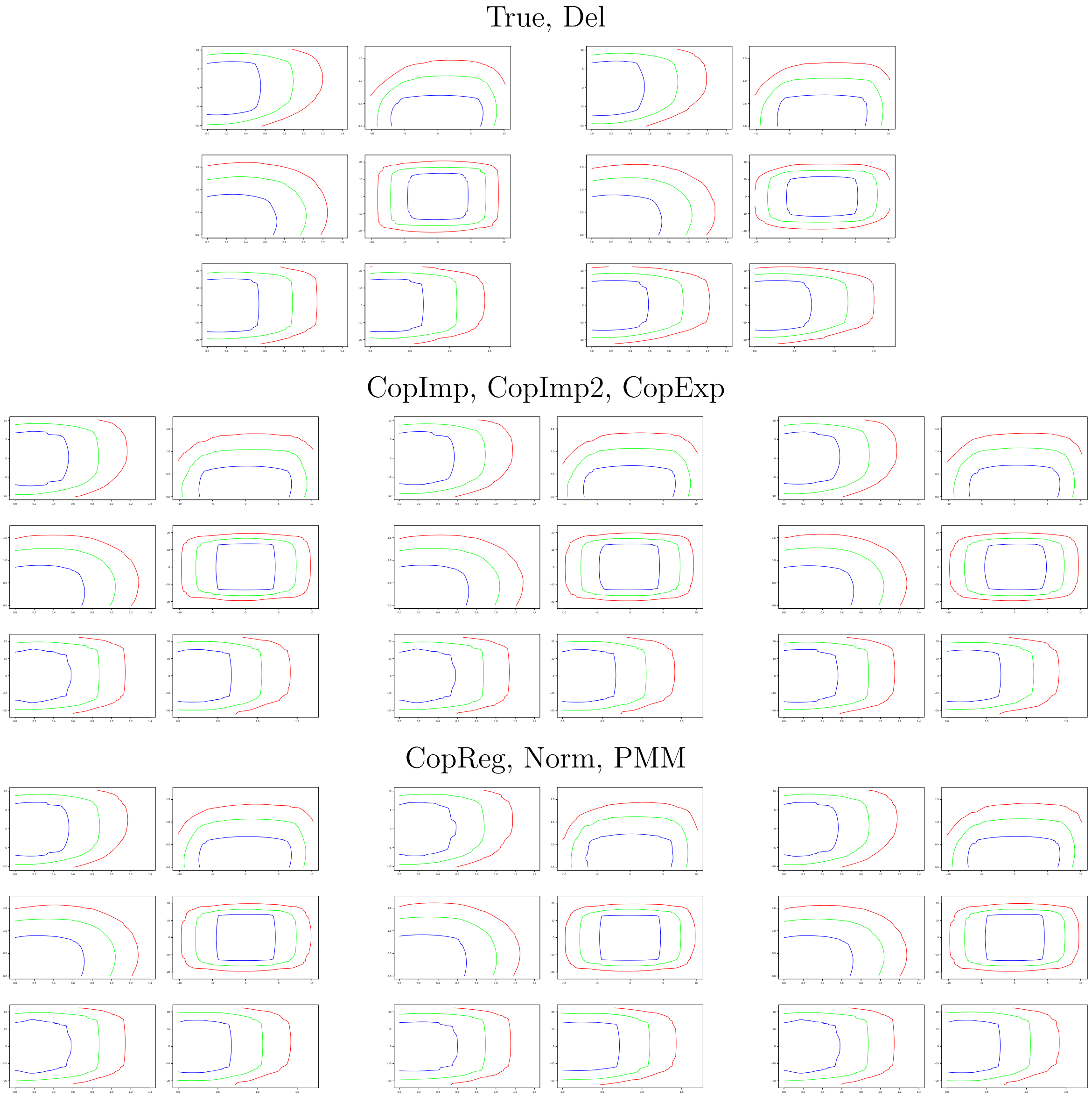
1. Fix a grid for the points of the level curves (100×100 between the theoretical 99.9% and 0.1% quantiles of the marginal distribution).

2. For each data set $(1, \ldots, 100)$ compute the points in the grid for the level curves.

3. For each value in the grid compute the empirical mean or the empirical 5% or the empirical 95% quantile. This results in a new grid.

4. Plot the level curves (25%, 50%, 75%) of the empirical mean or the empirical 5% or the empirical 95% quantile grid.

For comparison, the case without deleted values was added. If the deviation of the picture after imputation compared to the case without deleting values is very low, this is an indicator for a well performing method. Unfortunately the pictures look almost the same for each method, and no big difference is observable. For illustration one case is added.

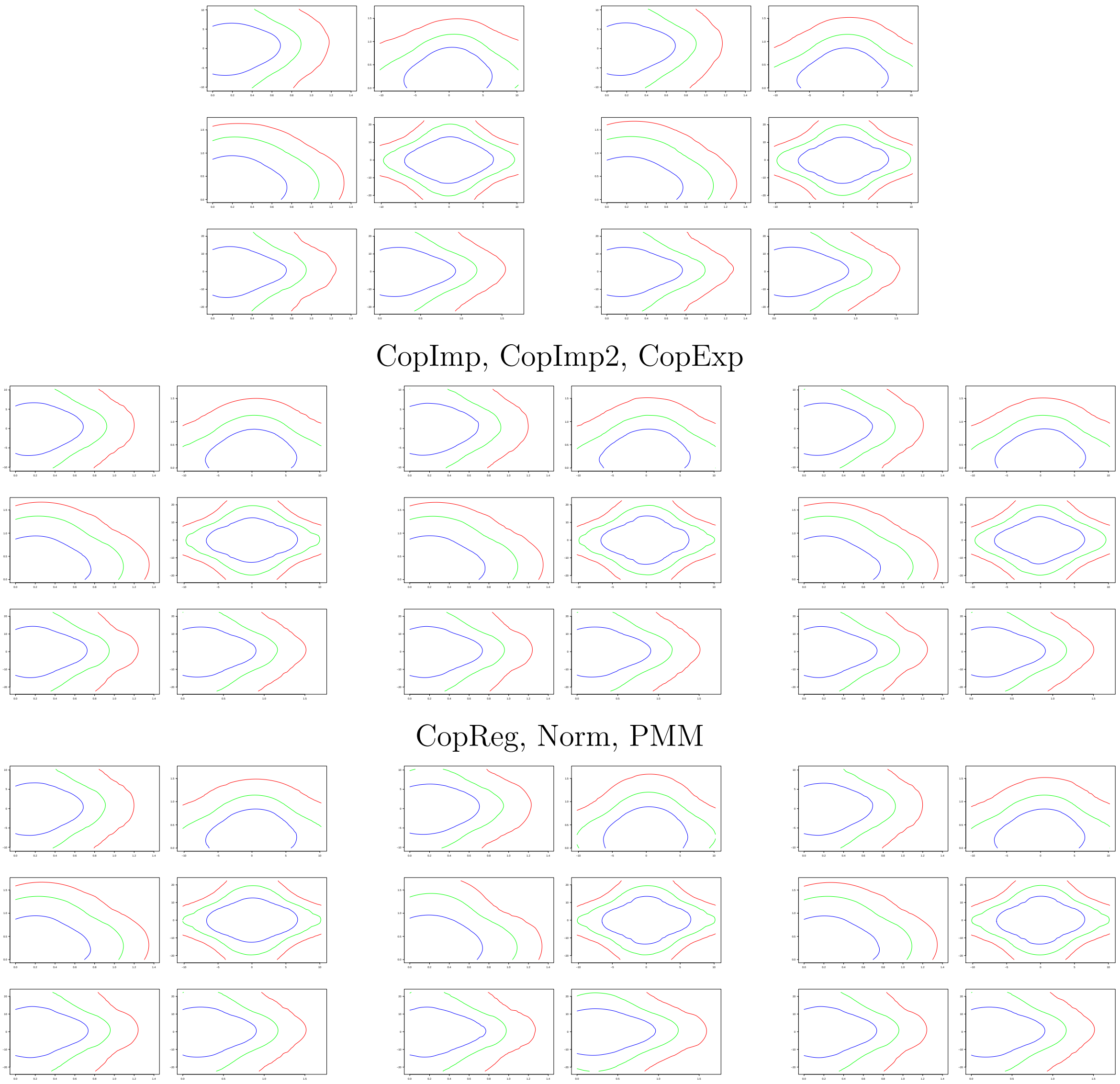


True, Del

CopImp, CopImp2, CopExp

CopReg, Norm, PMM

Table A.25: 5% Quantile Contours 1b,2b,3a,4a,5b,6a

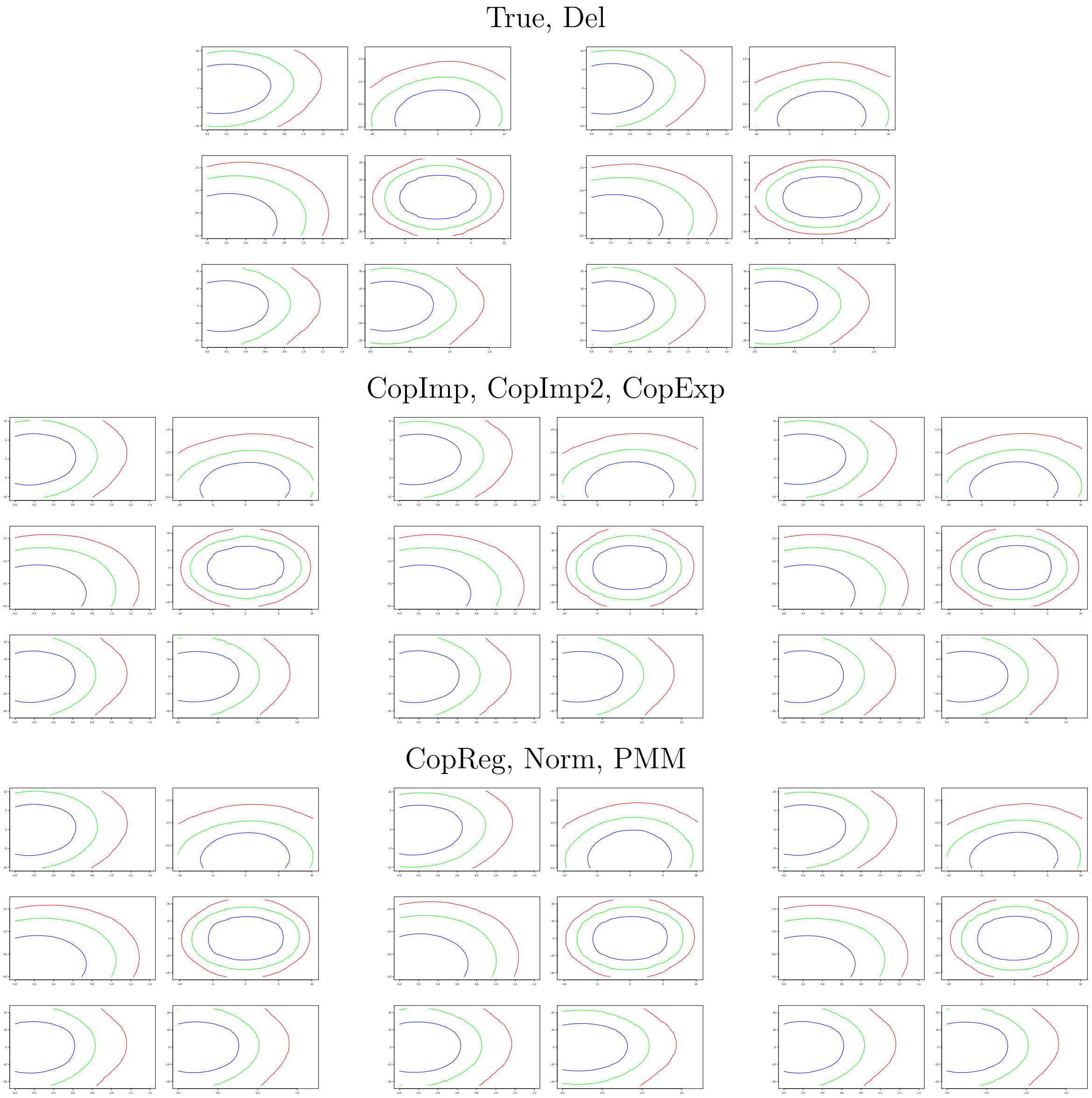


True, Del

CopImp, CopImp2, CopExp

CopReg, Norm, PMM

Table A.26: 95% Quantile Contours 1b,2b,3a,4a,5b,6a



True, Del

CopImp, CopImp2, CopExp

CopReg, Norm, PMM

Table A.27: Mean Contours 1b,2b,3a,4a,5b,6a

Appendix B

Case Study (6 dimensions)

The 95% contours are added to compare the different imputation methods.

Copula Regression Imputation

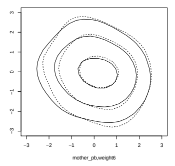
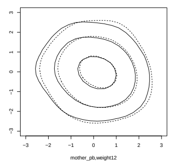
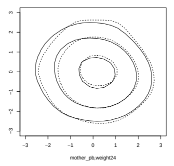
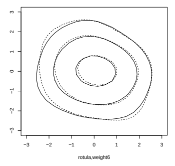
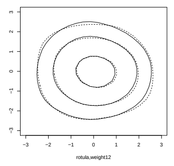
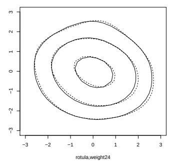
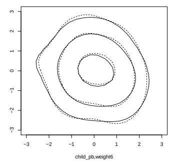
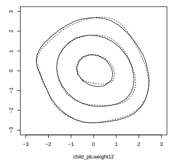
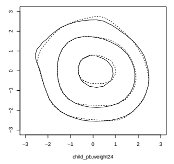
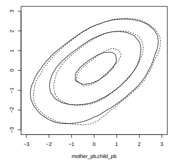
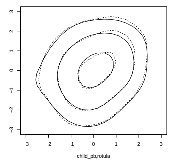
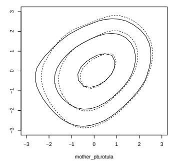
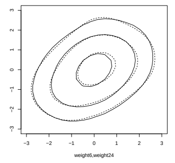
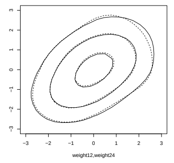
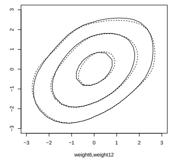
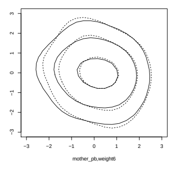
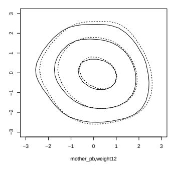
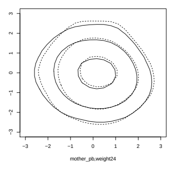
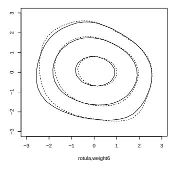
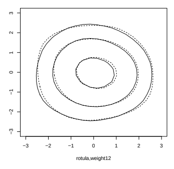
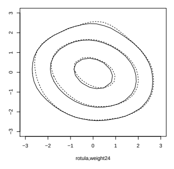
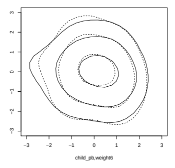
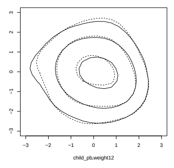
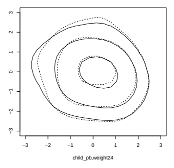
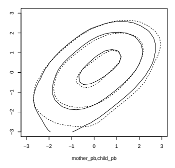
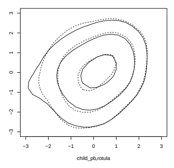
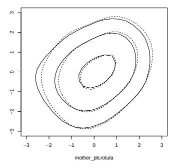
Female	Weight6	Weight12	Weight24
Mother-PB			
Rotula			
Child-PB			
	Mother-PB	Rotula	
Child-PB			
Rotula			
	Weight6	Weight12	
Weight24			
Weight12			

Table B.1: Female empirical 95% contours (20 tries) for all possible bivariate combinations without conditioning. The dashed contours are the complete cases.

PMM Imputation

Female	Weight6	Weight12	Weight24
Mother-PB			
Rotula			
Child-PB			

	Mother-PB	Rotula	
Child-PB			
Rotula			

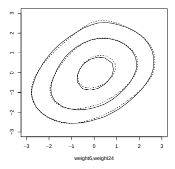
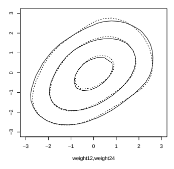
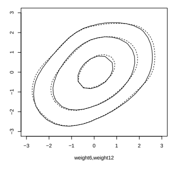
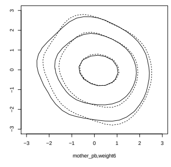
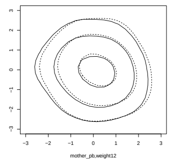
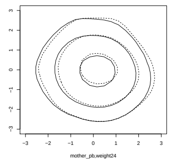
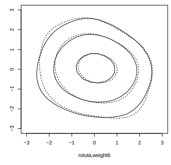
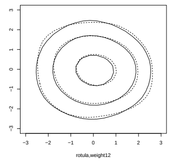
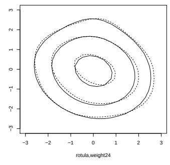
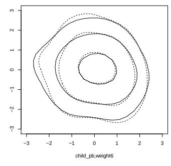
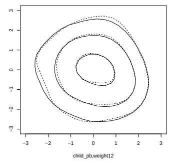
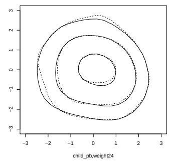
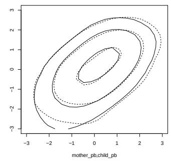
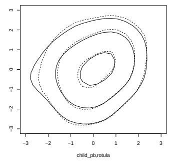
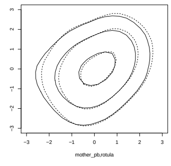
	Weight6	Weight12	
Weight24			
Weight12			

Table B.2: Female empirical 95% contours (20 tries) for all possible bivariate combinations without conditioning. The dashed contours are the complete cases.

Linear Regression Imputation

Female	Weight6	Weight12	Weight24
Mother-PB			
Rotula			
Child-PB			

	Mother-PB	Rotula	
Child-PB			
Rotula			

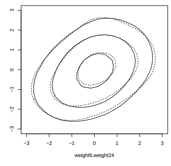
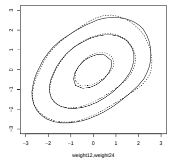
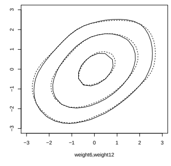
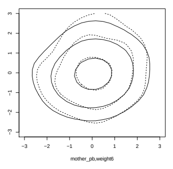
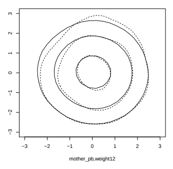
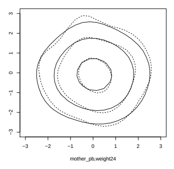
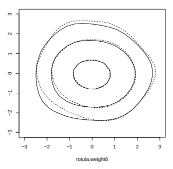
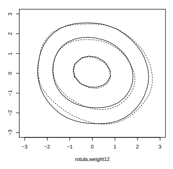
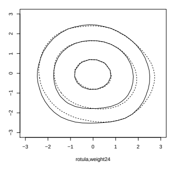
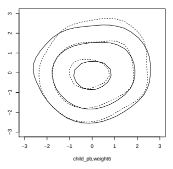
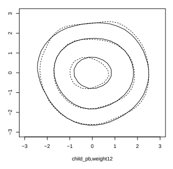
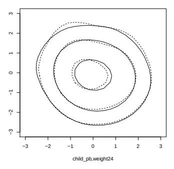
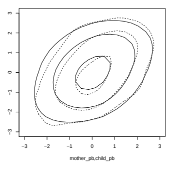
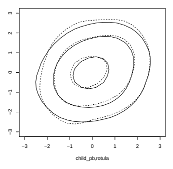
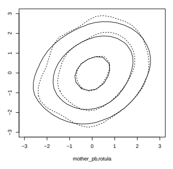
	Weight6	Weight12	
Weight24			
Weight12			

Table B.3: Female empirical 95% contours (20 tries) for all possible bivariate combinations without conditioning. The dashed contours are the complete cases.

Copula Regression Imputation

Male	Weight6	Weight12	Weight24
Mother-PB			
Rotula			
Child-PB			

	Mother-PB	Rotula	
Child-PB			
Rotula			

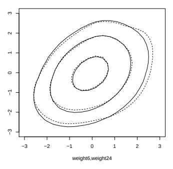
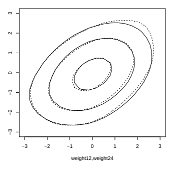
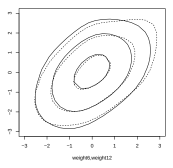
	Weight6	Weight12	
Weight24			
Weight12			

Table B.4: Male empirical 95% contours (20 tries) for all possible bivariate combinations without conditioning. The dashed contours are the complete cases.

PMM Imputation

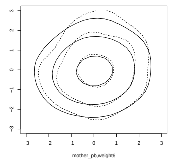
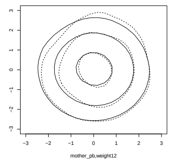
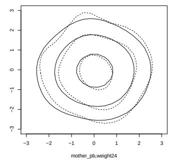
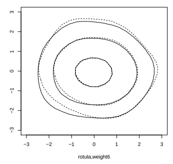
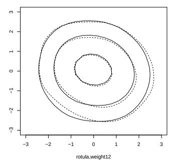
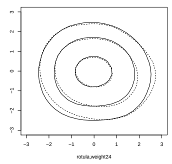
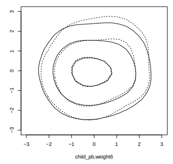
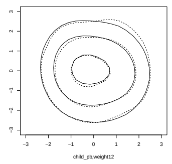
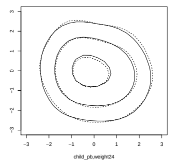
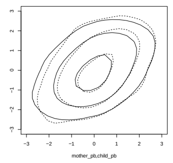
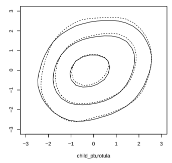
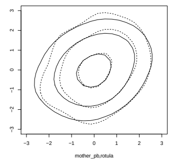
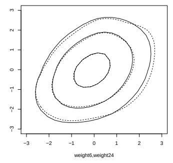
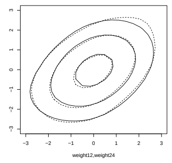
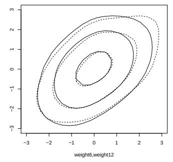
Male	Weight6	Weight12	Weight24
Mother-PB			
Rotula			
Child-PB			
	Mother-PB	Rotula	
Child-PB			
Rotula			
	Weight6	Weight12	
Weight24			
Weight12			

Table B.5: Male empirical 95% contours (20 tries) for all possible bivariate combinations without conditioning. The dashed contours are the complete cases.

Linear Regression Imputation

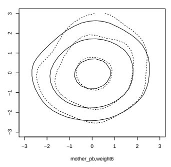
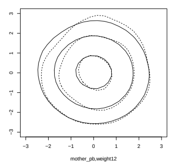
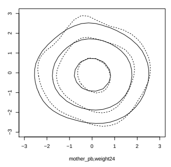
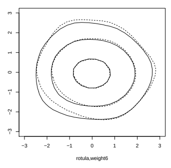
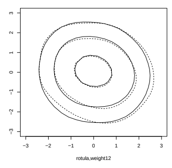
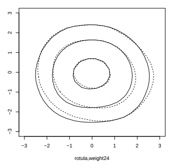
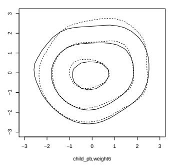
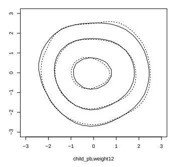
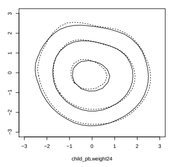
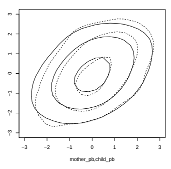
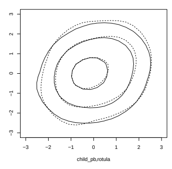
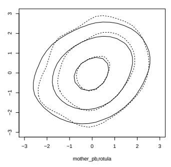
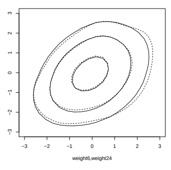
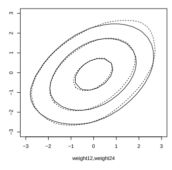
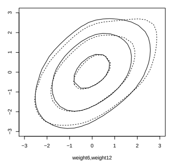
Male	Weight6	Weight12	Weight24
Mother-PB			
Rotula			
Child-PB			
	Mother-PB	Rotula	
Child-PB			
Rotula			
	Weight6	Weight12	
Weight24			
Weight12			

Table B.6: Male empirical 95% contours (20 tries) for all possible bivariate combinations without conditioning. The dashed contours are the complete cases.