Multiple **I**mputation via **L**ocal Regr**es**sion *(Miles)*

Dissertation

zur Erlangung des akademischen Grades

eines Doktors der Sozial- und Wirtschaftswissenschaften
(Dr. rer. pol.)

an der Fakultät Sozial- und Wirtschaftswissenschaften
der Otto-Friedrich-Universität Bamberg

vorgelegt von
Diplom Volkswirt Philipp Gaffert
geboren am 22. Juli 1984
in Lutherstadt Eisleben

Bamberg, März 2017

Datum der Disputation: 17. Juli 2017

URN: urn:nbn:de:bvb:473-opus4-498847
DOI: http://dx.doi.org/10.20378/irbo-49884

Abstract of the Dissertation

Multiple **I**mputation via **L**ocal Regr**es**sion *(Miles)*

by

Philipp Gaffert

Otto-Friedrich-Universität Bamberg, Germany, 2017

Committee:

Prof. Dr. Susanne Rässler

Prof. Trivellore E. Raghunathan, Ph.D.

Prof. Dr. Björn Ivens

Methods for statistical analyses generally rely upon complete rectangular data sets. When the data are incomplete due to, e.g. nonresponse in surveys, the researcher must choose between three alternatives:

1. *The analysis rests on the complete cases only:* This is almost always the worst option. In, e.g. market research, missing values occur more often among younger respondents. Because relevant behavior such as media consumption or past purchases often correlates with age, a complete case analysis provides the researcher with misleading answers.

2. *The missing data are imputed (i.e., filled in) by the application of an ad-hoc method:* Ad-hoc methods range from filling in mean values to applying nearest neighbor techniques. Whereas filling in mean values performs poorly, nearest neighbor approaches bear the advantage of imputing plausible values and work well in some applications. Yet, ad-hoc approaches generally suffer from two limitations: they do not apply to complex missing data patterns, and they distort statistical inference, such as t-tests, on the completed data sets.

3. *The missing data are imputed by the application of a method that is based on an explicit model:* Such model-based methods can cope with the broadest range of missing data problems. However, they depend on a considerable set of assumptions and are susceptible to their violations.

This dissertation proposes the two new methods *midastouch* and *Miles* that build on ideas by Cleveland & Devlin (1988) and Siddique & Belin (2008). Both these methods combine model-based imputation with nearest neighbor techniques. Compared to default model-based imputation, these methods are as broadly applicable but require fewer assumptions and thus hopefully appeal to practitioners. In this text, the proposed methods' theoretical derivations in the multiple imputation framework (Rubin, 1987) precede their performance assessments using both artificial data and a natural TV consumption data set from the GfK SE company. In highly nonlinear data, we observe *Miles* outperform alternative methods and thus recommend its use in applications.

Keywords: Multiple Imputation, Predictive Mean Matching, Sequential Regressions, Local Regression, Distance-Aided Donor Selection

Contents

List of Figures

List of Tables

List of Symbols and Abbreviations

Mathematics: General symbols	
ϵ	Small positive scalar
I_a	Identity matrix with dimensions $a \times a$
$f(a)$	Arbitrary function of a
$\partial\{f(a)\}/\partial(a)$	First derivative of $f(a)$ by a
$ln(a)$	Natural logarithm of a

Random variables and their realizations	
Q, Y, Z	Random variables
$pr(Z)$	Density function of Z
y_h	Incomplete $n \times 1$ vector of realizations of the Y variable
Y_h	Realizations of the multivariate Y variable (chapter 6 only)
$i = 1, \ldots, n_{obs}$	Index for the observed elements of y
$j = 1, \ldots, n_{mis}$	Index for the missing elements of y
R	The random variable indicating the response to the variable Y
r_h	Realizations of R
Q_h	Matrix of the completely observed realizations of the Q variables
z_h	Vector of the completely observed realizations of the Z variable
X	$= (Z, Q)$
X_h	Matrix of size $n \times p$ comprising a leading constant and the realized (z_h, Q_h)
x_1	The first nonconstant column of X
$x_{\underline{0}}$	The row of X corresponding to point P_0

Statistics: General symbols and distributions	
$E(A)$	Expectation of the random variable A
$var(A)$	Variance of the random variable A
α	Significance level for statistical tests
i.i.d.	Independently identically distributed
$A \sim N(\mu, \sigma_A^2)$	A follows a normal distribution with mean μ and variance σ^2
Φ	Normal distribution function
$A \sim \Gamma^{-1}(a_1, a_2)$	A follows an inverse-Gamma distribution with parameter a_1 and a_2
$A \sim t(\mu, \sigma^2, \iota)$	A follows a t distribution with ι degrees of freedom
$A \sim \chi^2(\iota)$	A follows a χ^2 distribution with ι degrees of freedom
N	Number of elements in the population

$h = 1, \ldots, n$	Index for the sample elements
$\hat{\beta}$	Maximum likelihood point estimate of the parameter β
$\tilde{\beta}$	A draw from the estimated posterior distribution of β
ω_i	Vector of bootstrap frequencies of the donors in the sample
ρ	Pearson's correlation coefficient
R^2	Coefficient of determination
AIC	Akaike information criterion
κ_C	Cohen's kappa (Cohen, 1960)
$\psi = 1, \ldots, \Psi$	Number of predictor cells in an analysis of variance (ANOVA)

Multiple imputation	
$m = 1, \ldots, M$	Number of multiple imputations
T	Total variance of a parameter
W	Within variance of a parameter
B	Between variance of a parameter
M(C)AR	Missing (completely) at random; an assumption about the sample at hand (Rubin, 1976)

The imputation model	
β	Parameter vector of the imputation model
v	Residual of the imputation model
σ_v^2	Variance of v
$\hat{y}_h$	Predicted values from the imputation model based on $\hat{\beta}$
$\hat{\tilde{y}}_h$	Predicted values from the imputation model based on $\tilde{\beta}$

The analysis model	
γ	Parameter vector of the analysis model
u	Residual of the analysis model
σ_u^2	Variance of u
$g(Q, Y)$	Nonlinear function of (Q, Y)

Predictive mean matching and *midastouch*	
k	Number of respective closest donors in predictive mean matching with drawing probabilities larger zero
φ	Scalar distance in terms of predicted means between two data points
κ	Closeness parameter (Siddique & Belin, 2008)
w_j	Vector of drawing probabilities of length n_{don} for recipient j
ϕ	Correction factor for the total variance of the mean under approximate Bayesian bootstrap imputation (Parzen et al., 2005)
n_{eff}	Effective sample size (Kish, 1965)

Local regression	
PMSE	Prediction mean squared error: $n^{-1} \sum_h (\hat{y}_h - y_h)^2$
H_0	Neighborhood around the point P_0
q	The number of elements in the neighborhood

π	Order of a polynomial
δ	Scalar absolute distance in terms of X_h between two data points
d	Tricube distance between two data points
C	Diagonal weight matrix for weighted least squares regression with d^{-1} on the principal diagonal
Λ	Diagonal ridge penalty matrix
λ	Ridge penalty scalar
l	Regression influence vectors

Simulation studies	
rRMSE	Root mean squared error of a parameter of interest from the analysis model relative to before deletion
n_{sim}	Number of Monte Carlo simulation runs
MA(C)AR	Missing always (completely) at random; an assumption about the data generating process (Rubin (1976), Mealli & Rubin (2015))

Imputation algorithms	
PMM	Predictive mean matching (Rubin (1986), Little (1988))
ABB imputation	Approximate Bayesian bootstrap imputation, Rubin & Schenker (1986)
MIDAS	Multiple imputation using distance-aided selection of donors (Siddique & Belin (2008), Siddique & Harel (2009))
midastouch	Touched-up version of MIDAS (chapter 3)
PI	Passive imputation (van Buuren & Groothuis-Oudshoorn, 1999)
JAV	Just-another-variable imputation (von Hippel, 2009)
RF	Random forest imputation (Doove et al., 2014)
Miles	Multiple imputation via local regression (chapter 5)

Declarations / Erklärungen

I hereby declare that this dissertation is the result of my own work. It does not include work done by others, particularly not work done by dissertation consultants. Nevertheless, it builds on previously published work that is cited throughout the text and that is fully declared in the bibliography.

Hiermit versichere ich, dass ich die vorliegende Dissertation selbständig und ohne die unzulässige Hilfe Dritter, insbesondere ohne die Hilfe von Promotionsberatern, angefertigt habe. Die aus anderen Quellen direkt oder indirekt bernommenen Gedanken sind im Text als solche kenntlich gemacht und sämtlich im Literaturverzeichnis aufgeführt.

I further state that no substantial part of my dissertation has already been submitted, or, is being concurrently submitted for another degree, diploma or other qualification at the Otto-Friedrich-University Bamberg or any other University or similar institution neither in Germany nor abroad.

Hiermit erkläre ich, dass ich keine weiteren Promotionsversuche mit dieser Dissertation oder Teilen daraus unternommen habe. Die Arbeit wurde bislang weder im In- noch im Ausland einer anderen Prüfungsbehörde vorgelegt.

Parts of chapter 3 are published in
Teile von Kapitel 3 sind veröffentlicht als

Gaffert, P., Meinfelder, F. & Bosch, V. (2016). Towards an mi-proper predictive mean matching. Working Paper. `https://www.uni-bamberg.de/fileadmin/uni/fakultaeten/sowi_lehrstuehle/statistik/Personen/Dateien_Florian/properPMM.pdf`, 1-15.

Parts of the chapters 4 and 5 are published in
Teile der Kapitel 4 und 5 sind veröffentlicht als

Gaffert, P., Bosch, V. & Meinfelder, F. (2016). Interactions and Squares: Don't Transform, Just Impute! In JSM Proceedings, Survey Research Methods Section. Alexandria, VA: American Statistical Association. 2036 – 2044. Retrieved from `http://ww2.amstat.org/sections/srms/Proceedings/y2016/files/389660.pdf`.

Chapter 1

Introduction

> No institute of science and technology can guarantee discoveries or inventions, and we cannot plan or command a work of genius at will. But do we give sufficient thought to the nurture of the young investigator, to providing the right atmosphere and conditions of work and full opportunity for development? It is these things that foster invention and discovery.
>
> J.R.D. Tata

1.1 Scope

As in other areas of statistics, in data imputation, there are two types of methods, some ad hoc and some model based (Schafer, 1997, p. 1)[1]. Among the more sophisticated ad hoc methods is the random hot-deck in adjustment cells (David et al., 1986, p. 30), which will be introduced in more detail in section 2.5.1. A major advantage of this and other hot-deck procedures is that the imputed values are drawn from the empirical distribution of the observed values and are thus plausible (Andridge & Little, 2010, p. 2). The main disadvantage of ad hoc methods is that the underlying assumptions are typically implicit. In contrast, model-based methods explicitly reveal the assumptions that they require. One such model-based method is multiple imputation (Rubin, 1987), which is broadly considered to be 'simple, elegant and powerful' (van Buuren, 2012, p. xix). Multiple imputation is the theoretical framework for the contributions of this dissertation. However, explicit assumptions are not necessarily more likely to hold in real data. The three most relevant assumptions required for default, i.e., fully parametric, multiple imputation to enable consistent estimation of the parameters of interest are[2]:

1. **Missing at random[3]:** The response rates must not vary systematically after conditioning on

[1] Calibration weighting is another example. Iterative proportional fitting can be considered an ad hoc method (Deming & Stephan, 1940), and the generalized regression estimator can be considered a model-based method (Cassel et al., 1976).

[2] This list of three is based on my own experience. Nevertheless, there may be applications in which, e.g., the assumption of independent observations is more doubted than the missing at random assumption.

[3] Missing at random and distinctness are collectively required for *ignorability* (see section 2.3 and Schafer (1997, p. 10)).

the observed data, e.g., within adjustment cells, and thus must not depend on the unobserved data (van Buuren, 2012, p. 7).

2. **Distribution of the data:** In fully parametric multiple imputation, the imputed values are drawn from assumed well-defined distributions, such as the normal distribution (Schafer, 1997, p. 181).

3. **Congeniality (Meng, 1994):** The imputation model, which is used to predict the missing values of the incomplete variable, must nest all relevant analysis models[4].

The scope of the dissertation is about relaxing the distributional and the congeniality assumptions to make multiple imputation more attractive to practitioners.

1.2 Outline

Chapters 2 and 4 introduce the theoretical prerequisites for the new ideas in this dissertation. Chapter 2 places emphasis on multiple imputation (Rubin, 1987), and chapter 4 places emphasis on local regression proposed by Cleveland (1979) and Cleveland & Devlin (1988).

Chapter 3 addresses the required distributional assumption about the data. Predictive mean matching (PMM: Rubin (1986, p. 92), Little (1988, p. 291)), which combines model-based predictions with hot-deck imputations, fully relaxes this assumption. However, PMM is shown to bias multiple imputation variance estimates. Different versions of PMM are introduced, and the new *midastouch* algorithm, which is based on the ideas of Siddique & Belin (2008), is proposed. A simulation study on multivariate normal data reveals a considerable advantage of *midastouch* over the PMM implementations in the major statistical software packages.

Chapter 5 introduces the new *Miles* algorithm. Because it builds on *midastouch* from chapter 3, *Miles* does not require distributional assumptions. The congeniality assumption, however, cannot be literally relaxed. Rather, *Miles* fits an imputation model that reflects the major relations, linear or not, between the incomplete variable and its predictors. Analysis models about these major relations are approximately nested in, i.e., congenial to, the (global) imputation model resulting from the local regressions that are employed by *Miles*. A simulation study on artificial data shows that the approximately congenial *Miles* can even be superior to fully congenial alternatives.

In the final chapter 6, the newly proposed algorithms are challenged in a simulation study involving real data from the GfK SE company. The evaluations are based on a broad set of analysis models frequently used in market research. Both *midastouch* and *Miles* perform as well as the established PMM algorithm.

1.3 Contributions

The missing at random assumption

Violating the missing at random assumption can result in seriously biased estimates of the parameters of interest (Enders, 2011, p. 14). However, the practitioner has no indicator for the degree of violation in any specific application (van Buuren, 2012, p. 31). The very special nature of the real data set used in chapter 6 permits a test for the missing at random assumption, which is presented in section 6.2.4. In this setup, the missing at random assumption does not hold. Although this result cannot be generalized, the data set can be used to study the effect of a natural assumption violation in future research.

[4] or the data generating process (Xie & Meng, 2014, p. 14)

The distributional assumption

Real data generally do not fit theoretical distributions well. PMM relaxes the distributional assumption. Section 3.2 shows that this relaxation comes at the cost of biasing variance estimates toward zero. While retaining the robust properties of PMM, the newly proposed *midastouch* also does not bias variance estimates, as shown in section 3.5.3. Furthermore, when imputing complex missing patterns, *midastouch*, in contrast to PMM, does not suffer from convergence issues (section 6.3.3).

The congeniality assumption

In slightly nonlinear data, *midastouch*, although strictly speaking uncongenial, is capable of capturing the structure of the data well, and applying *Miles* does not offer any additional benefit (section 6.4). In the highly nonlinear data of section 5.5.2, *Miles* performs better than alternative approximately congenial algorithms and almost as good as the best congenial algorithm, which is the just-another-variable algorithm (von Hippel, 2009) that employs PMM.

Summary

The missing at random assumption remains a serious burden for the imputer, and the contribution of this dissertation to overcome this burden is admittedly quite small.

The newly proposed *midastouch* algorithm fully relaxes the distributional assumption and can also address some congeniality issues, such as in chapter 6, where it is applied to moderately nonlinear data. As an additional benefit, the newly proposed *Miles* works well even in highly nonlinear data, while being only slightly impaired in perfectly linear data (chapter 5).

In contrast to their competitors random forest imputation (Doove et al., 2014) and PMM, neither *Miles* nor *midastouch* suffer from variance underestimation. Furthermore, in contrast to the just-another-variable algorithm and again PMM, neither *midastouch* nor *Miles* suffer from convergence issues when applied to complex missing data patterns. Moreover, in contrast to the just-another-variable algorithm, *Miles* does not cause any consistency issues.

From a practitioner's perspective, both *midastouch* and *Miles* offer considerable robustness compared to the existing alternatives and should be chosen over these alternatives unless there is a specific reason not to do so. Now, is *midastouch* better than *Miles* or vice versa? *Miles* is superior to *midastouch* because it can also handle highly nonlinear data. However, *Miles* is considerably slower than *midastouch*. Our advice is to use *midastouch* if time is a concern and *Miles* otherwise.

1.4 Acknowledgements

I would like to thank Volker Bosch, who came up with the initial idea for this dissertation, which was to use calibration weighting formulas to perform robust data imputation (see section 4.4). It was he who infected me with the curiosity about this topic and who, over the course of the years probably spent months with me developing and rejecting ideas, trying out new things and vividly discussing practical implications. I also want to thank Susanne Rässler and Florian Meinfelder. Approximately five years ago, I took their class on multiple imputation at the *gesis* in Cologne. Although back then I lacked nearly all the fundamentals, Susi and Flo welcomed me with open arms as their Ph.D. student. Later, Susi, Flo, Volker and I met frequently at the *fusion kitchen* events and exchanged research ideas. This dissertation would not exist without their inputs.

Susi and Flo have not only introduced me to the field of multiple imputation but also to Stef van Buuren, Donald B. Rubin and Trivellore E. Raghunathan. I would like to thank Raghu for

diving into the topics of my dissertation so deeply despite joining in rather late in the process. Our discussions made some of my vague ideas turn into precise contributions.

I would like to thank my colleagues at GfK: Markus Lilienthal, Barbara Wolf, and Andreas Kersting for inspiring discussions; Sandra Keller, Christoph Schöll, and Marc Rossbach for their assistance regarding the TV data set in chapter 6; Markus Herrmann for supplying me with the technical infrastructure; Jessica Deuschel, Teodora Vrabcheva, and Erik Hirschfeld for their help with SAS; and Anette Wolfrath, Volker Bosch and Raimund Wildner for supporting my part-time work model.

Most importantly, I would like to thank my wife Veronique. Without her love and support, none of this would have been possible.

Chapter 2

Multiple Imputation

> A capacity, and taste, for reading, gives access to whatever has already been discovered by others. It is the key, or one of the keys, to the already solved problems. And not only so. It gives a relish, and facility, for successfully pursuing the [yet] unsolved ones.
>
> Abraham Lincoln

2.1 Introduction

Statistical analysis with missing data is no longer a niche problem thanks to the tireless work of, among others, Stef van Buuren and Trivellore Raghunathan, who have not only enhanced the original ideas of Rubin (1978) but also made them accessible to a broad audience through easy to read textbooks (van Buuren (2012), Raghunathan (2015)) and easy to use software (van Buuren & Groothuis-Oudshoorn (2011), Raghunathan et al. (2002)). Although it is customary to include such a theory chapter in a dissertation (e.g., Siddique (2005), Koller-Meinfelder (2009)), I have, in light of the recent advances, seriously considered dropping this chapter. The only reason for including this chapter is for you, the reader. Therefore, this is not a chapter on general concepts of missing data; rather, it shall filter the parts of the theory that are vital for understanding the *new* ideas that are presented in the subsequent chapters. In this way, I hope to save the reader some time from looking topics up elsewhere and particularly translating different notations.

2.2 The imputer's model and the analyst's model

Statistical analysis is about learning from data. One key element is to apply sensible assumptions. In the most assumption-free setting, each observation arises from a unique data generating process, and all these processes may be fundamentally different; therefore, it may be completely misleading to link their realizations to any sort of conclusion. Nothing can be learned in this assumption-free setting. Researchers apply assumptions by modeling data. A linear regression model implicitly assumes that the parameters apply to all observations or that the mean effect of an increase of one predictor on the outcome has at least some meaning. Such a model further restricts the relation

between the predictors and the outcome to be linear in the parameters rather than being arbitrary[1].

In the imputation literature, the model that is to be applied to the data, disregarding its completeness, is called the analysis model. The analysis model is derived from the research question. The purpose of performing imputation is to enable the application of the analysis models of interest, despite facing incomplete data. By filling the holes in the data set, imputation even relieves the need to adapt the analysis models to the incomplete data situation.

As will be shown in the next section, imputation uses predictive modeling. As long as the imputation model is not more restrictive than the analysis model (Schafer, 1997, p. 141) and the ignorability assumption, which is also described in the next section, is met, the incompleteness *does not bias* the conclusion of the analysis model[2]. If many different estimands are of interest, i.e., many different analysis models are to be applied to one imputed data set, then the imputation model must be very inclusive at the cost of low efficiency[3]. The term *inclusive* is used throughout this dissertation to describe an imputation model that enables at least approximately unbiased estimation of a large number of parameters on the imputed data set and thus in a broader sense than in Collins et al. (2001).

2.3 Parametric multiple imputation

A thorough treatment of multiple imputation and its underlying assumptions is already provided in Rubin (1987) and discussed in detail in Schafer (1997). This section consists of a less general example that will be revisited in the next chapter.

Let the data of interest be n independent realizations of a normal random vector (Q, Y, Z) with length p. Throughout this dissertation, Q with length $p-2$ denotes one or more predictors in both the imputation model and the analysis model. Y denotes the variable with missing values and thus the response variable in the imputation model, and Z denotes the response variable in the analysis model. The matrix of independent (Z, Q) realizations X_h with dimensions $n \times p$ is fully observed and is defined to include a leading constant column. The realization r of the random vector R takes the value 1 for all n_{obs} observed values of y_h and 0 for all $n_{mis} = n - n_{obs}$ missing values of y_h. The imputation model is the linear model

$$y_h = X_h\beta + v \qquad with \qquad v \sim N(0, \sigma_v^2 I_n), \tag{2.1}$$

where β denotes a vector of parameters of length p. In fully parametric multiple imputation, the steps of algorithm 1 are repeated $M \geqslant 2$ times to correctly reflect the uncertainty of the parameter estimates of the imputation model.

The key assumption required for this procedure is that the missing values are not governed by a different regime than the observed values. This assumption means that the imputation model in equation (2.1) is not misspecified even though it does not involve r, or, more formally, that r and v must be independent. This requirement is known as the missing at random (MAR) assumption[4]. A stricter version is the missing completely at random (MCAR) assumption, which implies that r and y must be independent. If r and v are somehow related, then the response mechanism is said to be missing not at random (MNAR). Similar relations can be defined for the data generating process, resulting in the terms missing always at random (MAAR) and missing always completely

[1] The list of necessary assumptions for the linear regression model is even much longer (Greene, 2008, p. 44).
[2] An unusual exception to this rule is superefficiency (Rubin, 1996, p. 481).
[3] In these cases it may be beneficial to use different imputation models, e.g., one for each analysis model.
[4] Strictly speaking, missing at random and an additional, rather minor, assumption, called distinctness (Schafer, 1997, p. 11), are required for the response mechanism to be *ignorable*.

Algorithm 1 Parametric multiple imputation for a single normal incomplete variable y_h and a set of complete linear predictors X_h (Little & Rubin (2002, p. 216), Greenberg (2013, p. 116)). The steps are named according to Tanner & Wong (1987, p. 531) although this algorithm is not iterative.

1. *The posterior step to draw the parameters:* First, draw from the observed data posterior distribution of the residual variance, which is $pr(\tilde{\sigma}_v^2 \mid y_i, X_i) = \Gamma^{-1}\{n_{obs}/2, (y_i - X_i\hat{\beta})'(y_i - X_i\hat{\beta})/2\}$ (Greene, 2008, p. 996). Then, draw from the observed data posterior distribution of the intercept and slope parameters, which is $pr(\tilde{\beta} \mid y_i, X_i, \tilde{\sigma}_v^2) = N_p\{\hat{\beta}, \tilde{\sigma}_v^2(X_i'X_i)^{-1}\}$. y_i and X_i refer to the fully observed subset of the data, and $\hat{\beta}$ denotes the maximum likelihood parameter estimate (Greene, 2008, p. 483).

2. *The imputation step to draw the missing values conditional on the parameters:* Draw n_{mis} times independently from the imputation model, i.e., $\tilde{y}_j \sim N(X_j\tilde{\beta}, \tilde{\sigma}_v^2)$ with $j = 1, \ldots, n_{mis}$.

at random (MACAR) (Rubin (1976), Mealli & Rubin (2015)). The work in this dissertation does not involve MNAR. Useful practical implications of the MAR assumption are derived in van Buuren (2012, p. 34).

The analysis model shall now be applied to each imputed data set. Suppose that the estimand is the mean of Y. The M different maximum likelihood estimates $(\hat{\mu}_y^{m=1}, \ldots, \hat{\mu}_y^{m=M})$ can be combined using Rubin's rules (Rubin, 1987, p. 76) by $\hat{\mu}_y = M^{-1}\sum_{m=1}^{M}\hat{\mu}_y^m$. The variance is given by

$$T = var(\hat{\mu}_y) = \underbrace{M^{-1}(n-1)^{-1}\sum_{m=1}^{M}\{var(y_h^m)\}}_{W} + (1 + M^{-1})\underbrace{(M-1)^{-1}\sum_{m=1}^{M}(\hat{\mu}_y^m - \hat{\mu}_y)^2}_{B} \tag{2.2}$$

Equation (2.2) involves an analysis of variance (ANOVA) type thinking (Rinne, 2008, p. 650). The *within* variance W is the variance as in a completely observed data set, and the *between* variance B reflects the uncertainty that is involved in estimating the imputation model parameters β and σ_v^2. If the imputation model had no parameters to estimate, e.g., $y_h = 2z_h$, then the between variance would be zero. However, note that such *restrictive* imputation models typically do not nest any relevant analysis models and thus bias their conclusions (see section 2.2). The quantity $T^{-0.5}\hat{\mu}_y$ is $t(\mu, 1, \iota)$ distributed (Rinne, 2008, p. 326) with degrees of freedom (Barnard & Rubin, 1999, p. 949)

$$\iota = \left[(M-1)^{-1}\left(1 + \frac{M}{M+1}\frac{W}{B}\right)^{-2} + \frac{n+2}{n(n-1)}\frac{T}{W}\right]^{-1}.$$

Rubin (1987, p. 118) calls imputations that yield approximately valid inferences for the parameters of interest *proper*. The detailed requirements are presented in (Schafer, 1997, p. 145).

2.4 Missing data patterns

The example in the previous section consists of only one incomplete variable. In real data applications, two or more variables are generally incomplete. We distinguish three different cases, which are also shown in figure 2.1.

1. *Monotone pattern and multivariate two patterns.* An appropriate algorithm proceeds as follows. Imputations are drawn from the imputation model, such as the one in equation 2.1 for the first incomplete variable conditional on all fully observed variables. The imputations for the second variable are drawn conditional on all fully observed variables and the first

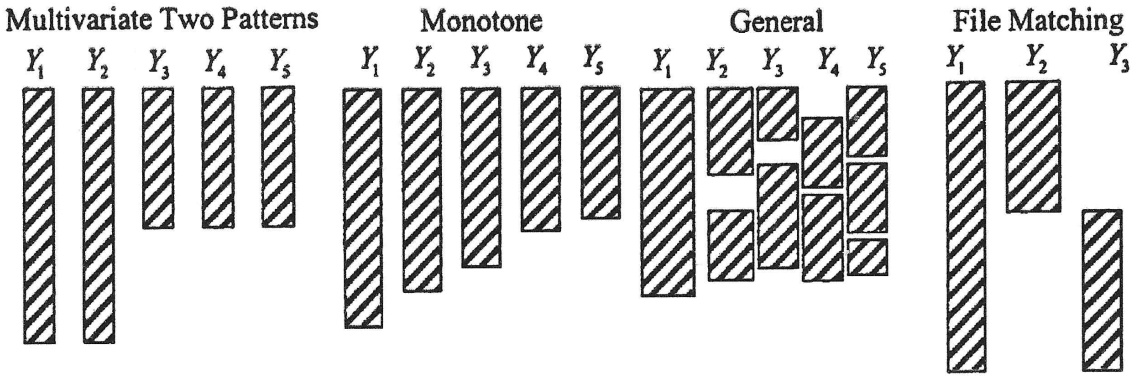

Figure 2.1: Relevant missing patterns for this text (Little & Rubin, 2002, p. 5)).

imputed variable, and so on. The last variable in the data set is imputed conditional on all variables but itself (van Buuren, 2012, p. 104). The multivariate two patterns are a special case of the monotone pattern and will be revisited in chapter 5.

2. *Swiss cheese (Andridge, 2011, p. 67) a.k.a. general pattern.* An appropriate algorithm starts with a simple random hot-deck, i.e., with drawing randomly from the observed values within each variable (van Buuren & Groothuis-Oudshoorn, 2011, p. 18), and iterates over the variables. There are two differences to the algorithm for the monotone pattern: conditioning is *always on all* other variables, and the algorithm does not stop when the end of the data set is reached. Rather, the algorithm iterates over all variables in the data set as often as required to reach convergence for the parameters of the analysis model. Kennickell (1991) was the first to apply this sequential regression algorithm, which is akin to Gibbs sampling and often referred to as fully conditional specification or chained equations (van Buuren & Groothuis-Oudshoorn, 2011). For a thorough treatment, see Raghunathan et al. (2001) and Liu et al. (2013). The Swiss cheese pattern will be revisited in chapter 6.

3. *File matching pattern.* In the file matching pattern, the complete cases maximum likelihood estimate of the parameter of interest is unobtainable. The typical example is a correlation coefficient of two variables that are never jointly observed. Although the file matching pattern, which is also known as data fusion, is very relevant in market research, it is not covered in this dissertation. For a thorough treatment, see Raessler (2002) and D'Orazio et al. (2006).

2.5 Alternatives to fully parametric algorithms

2.5.1 Hot-deck imputation

As shown in algorithm 1, in fully parametric imputation, the values are drawn from well-described distributions, which hardly fit empirical distributions. A simple solution is to impute observed values from the same variable, i.e., to provide the 'recipients' values from the 'donors'. The obvious advantage of these hot-deck procedures is that the imputed values are plausible and do not, e.g., fall outside the range. The simple random hot-deck has already been introduced above. Valid descriptive statistics can be obtained from a simple random hot-deck imputation if there is only one variable and the response mechanism is completely at random. However, unless n_{obs} is very large, confidence intervals are excessively short because the between variance component B

of equation (2.2) is ignored. The simple random hot-deck omits the posterior step (Siddique, 2005, p. 17).

A natural extension of the simple random hot-deck evolves from the presence of categorical predictors. The simple random hot-deck can then be performed within each cell, which is similar to fitting an ANOVA model containing all interactions (Lillard et al., 1982, p. 15). The imputed value can be regarded as the cell mean plus the residual of the randomly selected donor.

2.5.2 The approximate Bayesian bootstrap

Bayesian bootstrap imputation resolves the inference issue of simple random hot-deck. The absolute frequencies of the observed values serve as the parameters of a Dirichlet distribution (Rinne, 2008, p. 350). The underlying assumption is that the variable is categorical with as many categories as there are unique values. Draws from this distribution define the parameters of multinomial distributions (Rinne, 2008, p. 277). Draws from the multinomial distributions in turn yield the multiple imputations. Rubin & Schenker (1986, p. 368) provide the details.

The approximate Bayesian bootstrap imputation can be regarded as a computational shortcut of the Bayesian bootstrap imputation. In the posterior step, a bootstrap sample from the donors is drawn (Efron, 1979), and the imputations are simply drawn from this bootstrap sample (Rubin & Schenker, 1986, p. 368). This implicit modeling procedure propagates the uncertainty of the estimated parameters involved. However, Kim (2002) shows that the confidence intervals are still too small because, just like the maximum likelihood estimator, the bootstrap estimator ignores the correction for the appropriate number of degrees of freedom (Davison & Hinkley, 1997, p. 22). Therefore, for finite n_{obs}, the total parameter variance is still slightly underestimated. Parzen et al. (2005) show that multiplying the total variance estimator for the mean presented in equation (2.2) by the following factor ϕ eliminates this bias

$$\phi(n_{obs}, n_{mis}, M) = \frac{\frac{n^2}{n_{obs}} + \frac{n_{mis}}{M}\left(\frac{n-1}{n_{obs}} - \frac{n}{n_{obs}^2}\right)}{\frac{n^2}{n_{obs}} + \frac{n_{mis}}{M}\left(\frac{n-1}{n_{obs}} - \frac{n}{n_{obs}^2}\right) - \frac{n \cdot n_{mis}}{n_{obs}}\left(\frac{3}{n} + \frac{1}{n_{obs}}\right)} \geqslant 1. \tag{2.3}$$

Some criticism regarding this correction factor has been presented by Demirtas et al. (2007).

2.5.3 Predictive mean matching (PMM)

In contrast to the approximate Bayesian bootstrap, in PMM (Rubin, 1986, p. 92), only the imputation step of algorithm 1 is modified. The first implementation of PMM for general missing data problems by Little (1988) is still widely used (e.g., van Buuren & Groothuis-Oudshoorn (2011), Royston & White (2011)[5]) and is thus the key reference (see algorithm 2).

Algorithm 2 The original PMM algorithm proposed by Little (1988, p. 292).

1. Calculate the predictive mean for the n_{obs} observed elements of y_h as $\hat{y}_i = X_i\hat{\beta}$.
2. Calculate the predictive mean for the n_{mis} missing elements of y_h as $\hat{\tilde{y}}_j = X_j\tilde{\beta}$.
3. Match each element of $\hat{\tilde{y}}_j$ to its corresponding closest element of $\hat{y}_i$.
4. Impute the observed y_i of the closest matches.

[5]Both implementations deviate from the original algorithm in that they make a random draw from the closest $k > 1$ donors in the last step.

Compared to fully parametric imputation, PMM is more robust to model misspecifications (Schenker & Taylor, 1996, p. 429), namely, nonlinear associations, heteroscedastic residuals, and deviations from normality (Morris et al., 2014, p. 4). Nonetheless, the quality of PMM imputations largely depends on the availability of nearby donors; truncation of the data limits the validity of the method (Koller-Meinfelder, 2009, p. 38).

While retaining the benefits of the simple random hot-deck in cells discussed above, PMM has additional desirable properties. The most obvious such property is a more flexible imputation model, which neither requires the continuous predictors to be divided into arbitrary categories nor needs all interactions to be considered. Because the matching is not affected by variables that are not predictive, PMM can also be considered more parsimonious (David et al., 1986, p. 31).

2.5.4 Distance-aided donor selection

For the posterior step of the distance-aided donor selection algorithm proposed by Siddique (2005) and Siddique & Belin (2008), which Siddique & Harel (2009) later called MIDAS, bootstrapping is employed as originally proposed by Heitjan & Little (1991, p. 18). Maximum likelihood estimation of the linear regression imputation model parameters on M independent bootstrap samples replaces the draws from the posterior distribution (Little & Rubin, 2002, p. 216). The unique feature of the MIDAS algorithm is that it reuses the donors' bootstrap frequencies for the imputation step. For recipient j, donor i is drawn from the full donor pool with probability

$$w_{i,j} = f(\omega, \hat{\tilde{y}}_i, \hat{\tilde{y}}_j, \kappa) = \omega_i \hat{\tilde{\varphi}}_{i,j}^{-\kappa} / \sum_{i=1}^{n_{obs}} (\omega_i \hat{\tilde{\varphi}}_{i,j}^{-\kappa}), \tag{2.4}$$

where ω_i denotes the bootstrap frequency of donor i, $\hat{\tilde{\varphi}}_{i,j}$ denotes the scalar absolute distance between the predictive means of donor i and recipient j based on $\tilde{\beta}$, and κ is a closeness parameter that adjusts the importance of the distance. For $\kappa = 0$, the procedure is equivalent to the approximate Bayesian bootstrap; for $\kappa \to \infty$, the procedure becomes equivalent to nearest-neighbor matching, as in algorithm 2.

2.5.5 Random forest imputation

The choice of any model is a bias-variance trade-off. If the analysis model is known, then all parameters that are not of interest may be biased by the imputation model without any additional harm. However, if the analysis model is unknown, then it is the imputer's job to find an imputation model that neither restricts the key relations in the data nor suffers from low efficiency. Incorporating, for instance, interactions in parametric imputation models becomes inefficient very quickly because the number of parameters to estimate increases quadratically with the number of variables.

Doove et al. (2014) suggest using random forest imputation to implicitly include non-linear relations. They show that their algorithm preserves interactions that are not explicitly contained in the imputation model quite well and substantially better than posterior-step linear regression models with imputation-step PMM. This improvement, however, comes at the cost of biasing the linear effects of regression analysis models (Doove et al., 2014, p. 101).

For their implementation Doove et al. (2014) use the `R::mice` framework for sequential regressions (van Buuren & Groothuis-Oudshoorn, 2011) and the `R::randomForest` package (Liaw & Wiener, 2002), which consists of fitting classification and regression trees. A thorough theoretical treatment thereof is provided in James et al. (2013, p. 303) and the implementation is presented

Algorithm 3 The random forest imputation algorithm by Doove et al. (2014, p. 103).

1. Draw n_{tree} bootstrap samples from the donors.
2. Draw n_{tree} random samples of size $(p-1)/3$ from the $p-1$ predictor variables.
3. Fit n_{tree} trees by recursive partitioning without pruning*. Each leaf of each tree constitutes a subset of the donors.
4. Put the recipients down the trees to see in which leaves they fall.
5. Combine all leaves including the same recipient over all trees to one donor pool D_j for each recipient j.
6. For each recipient j make a random draw from D_j and impute the value of the drawn donor.

*The term pruning encompasses different algorithms that reduce the complexity of the tree to avoid overfitting.

in algorithm 3.

2.5.6 Others

There are a few other algorithms that promise to address nonlinear data. Similar to random forest imputation, the latent-class based algorithm by Akande et al. (2016) forms groups of donors and recipients such that the imputations are obtained by simply drawing donors from the same group. The `R::Hmisc::aregImpute` algorithm by Harrell (2015) (R Core Team, 2016), which is based on the theoretical work by Breiman & Friedman (1985), results in predictions of transformed y_h and employs PMM in the imputation step. Neither of the two algorithms is within the scope of the next chapters.

Chapter 3

Toward Multiple-Imputation-Proper Predictive Mean Matching

> It is by intuition that we discover and by logic we prove.
>
> Henri Poincaré

3.1 Introduction

Combining multiple imputation with predictive mean matching (PMM) promises to provide a robust imputation procedure that will yield valid inferences, thus making it highly appealing to practitioners (Heitjan & Little, 1991, p. 19). Consequently, such a combination is not only a feature but also often the default mode of imputation algorithms in all major statistical software programs (Morris et al., 2014, p. 3). Despite its preeminence in practice, skepticism regarding this combination of techniques dominates the literature. Little & Rubin (2002, p. 69) state the following about PMM

> ... properties of estimates derived from such matching procedures remain largely unexplored.

Koller-Meinfelder (2009, p. 32) notes that

> The difficult part about Predictive Mean Matching is to utilize its robust properties within the Multiple Imputation framework in a way that Rubin's combination rules still yield unbiased variance estimates.

Moreover, Morris et al. (2014, p. 5) recently warned in the same context

> ... there is thus no guarantee that Rubin's rules will be appropriate for inference.

The contrast between the theoretical uncertainty concerning the validity of this combined approach and its popularity in applications motivated the work presented in this chapter. The next section elaborates one major deviation of multiple imputation PMM algorithms from the theory of multiple imputation, which is one of the key contributions of this dissertation. The new insight sheds a

different light on well-known tuning parameters for PMM, which are presented in section 3.3. A new and more proper algorithm is proposed in section 3.4, whose empirical superiority regarding the coverages of frequentist confidence intervals is demonstrated via a simulation study in section 3.5. The insights of this chapter are published as a working paper (Gaffert et al., 2016), and citing it is spared throughout.

3.2 Why predictive mean matching is not multiple imputation proper

Using PMM for the multiple imputation of data sets causes the between variance of the parameter estimates of interest to suffer from attenuation bias.

To illustrate this situation, consider an analysis of variance example with $\psi = 1, \ldots, \Psi$ different predictor cells. Suppose that the incomplete variable Y in cell ψ is normally distributed with mean μ_ψ and variance σ^2_ψ. Furthermore, suppose that each of the Ψ cells contains a sufficient number of donors, say, five or more. Now, without loss of generality, let us examine at the recipients in the first cell. Parametric multiple imputation draws $M \geqslant 2$ times $\tilde{\sigma}^2_1$, then $\tilde{\mu}_1 \mid \tilde{\sigma}^2_1$, and then $\tilde{y}_{\psi=1} \mid (\tilde{\mu}_1, \tilde{\sigma}^2_1)$, which is efficient. A nonparametric alternative is an approximate Bayesian bootstrap imputation in cell $\psi = 1$ that proceeds as follows. It draws $M \geqslant 2$ times a bootstrap sample from the donors in the cell and draws values to impute from this bootstrap sample. The key element that these two proper procedures have in common is that the distribution from which the imputed values are drawn varies over the multiple imputations. In the parametric case the parameters of the underlying normal distributions vary, and in the nonparametric case, the composition of the empirical distribution varies.

	fully parametric	PMM	ABB imputation
Posterior step	Draw $(\tilde{\beta}, \tilde{\sigma}^2_v)$ from the imputation model $y_h = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + v$, with heteroscedastic residuals, i.e., $var(v \mid \psi = 1) = \sigma^2_{v,1}$, …, $var(v \mid \psi = 4) = \sigma^2_{v,4}$.		Within each of the $\Psi = 4$ cells draw a bootstrap sample of the $n_{obs,1}$, …, $n_{obs,4}$ donors
Imputation step	Draw from the normal imputation model: $\tilde{y}_j \mid (\tilde{\beta}, \tilde{\sigma}^2_v, x_1, x_2)$	As within each cell the predicted means $\hat{\tilde{y}}_\psi$ are identical, algorithm 2 draws $n_{mis,\psi}$ values from $n_{obs,\psi}$, i.e., a simple random hot-deck imputation within the cell	Within each cell, draw $n_{mis,\psi}$ values from the *bootstrapped* $n_{obs,\psi}$, i.e., a simple random hot-deck imputation within the *bootstrapped* cell

Table 3.1: The algorithms of proper fully parametric imputation, proper approximate Bayesian bootstrap (ABB) imputation, and PMM are compared. The underlying data situation involves two binary predictors (x_1, x_2), one incomplete variable y_h, and normal noise v. The two predictors form $\Psi = 4$ cells: $\psi(x_1 = 0, x_2 = 0) = 1, \ldots, \psi(x_1 = 1, x_2 = 1) = 4$. Ignorability is assumed. In the imputation step, PMM is very similar to ABB imputation, but it ignores the bootstrap. Because ABB imputation is approximately proper, PMM must attenuate the between imputation variance.

PMM proceeds in a considerably different manner. The recipients and the donors in cell $\psi = 1$ end up having exactly the same predicted mean[1]. Choosing the nearest neighbor ultimately consists of making a random draw from the donors in cell $\psi = 1$. This may be valid once, but the procedure is the same for all $m = 1, \ldots, M$ imputations. It thereby mimics the simple random hot-deck of

[1]This is only true if type-2 matching is applied, which slightly differs from algorithm 2. Section 3.3 presents the details.

section 2.5.1, which is known to underestimate the between variance component because it partly omits the posterior step. Table 3.1 schematically presents this reasoning.

It appears to be surprising that although PMM contains a draw from the estimated distribution of the intercept and slope parameters β (see algorithm 2), the parameter uncertainty does not propagate. In this regard, the above example is deceptive. Therefore, consider another example. For simplicity, suppose that there are two normal orthogonal predictors x_1, x_2. Now, the definition of the relevant donors is less clear than in the previous example, where it appeared obvious that all donors of $\psi = 1$ are suitable. The job of the β is simply to define the relevant 'cell'. Drawing $\tilde{\beta}$ is an important task, because the cell definition is not certain and must thus vary over the multiple imputations. Figure 3.1 displays the effect of varying β coefficients on the cell definition.

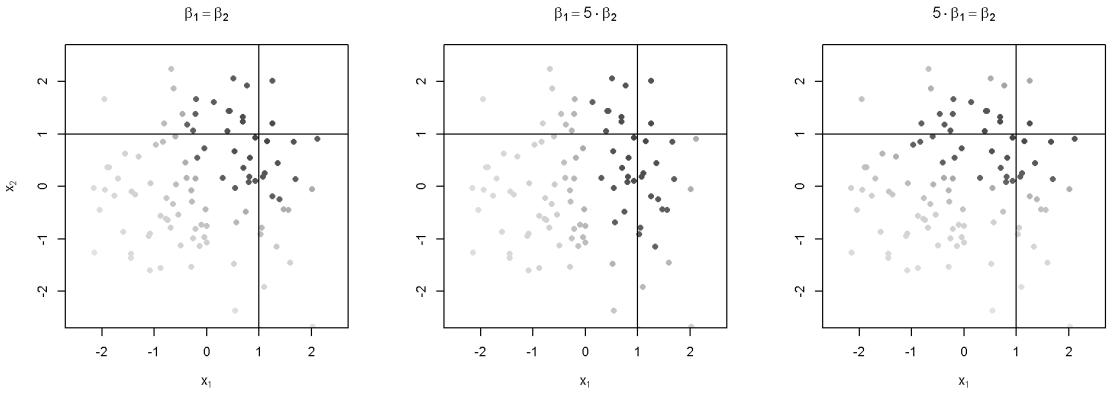

Figure 3.1: The plots show 100 random draws from a bivariate normal distribution with zero correlation. The shading indicates distances in the predictive means to one recipient $P_0(x_1 = 1, x_2 = 1)$. Different draws from the estimated distribution of the β parameters can alter the definition of the cell from which the donor is drawn. Considering distances, not frequencies, the cell is a circle in the left plot, a long ellipse in the middle plot and a wide ellipse in the right plot.

However, PMM then goes wrong. The cells are defined, i.e., we have conditioned on $\tilde{\beta}$, and all PMM does is make a random draw from the cell or even take the nearest one. It thereby ignores parameter uncertainty to a large extent. To be precise, the $\tilde{\beta}$ define the mean of the cell; however, the uncertainty in estimating the residual variance parameter σ_v^2 from the imputation model in equation (2.1) remains unconsidered. In any given cell, we observe a distribution of units in a sample, which suffers from sampling error. Thus, what is needed is some type of approximate Bayesian bootstrap imputation algorithm *after* conditioning on the $\tilde{\beta}$ parameters.

3.3 Existing ideas to make predictive mean matching proper

PMM has recently been under suspicion for underestimating the between variance component of equation (2.2). Van Buuren (2012, p. 71) and Morris et al. (2014, p. 7) criticize the selection of the nearest neighbor of algorithm 2. Selecting the nearest neighbor is a special case of general k-nearest-neighbor selection (Heitjan & Little, 1991, p. 16), which is typically applied in current statistical software programs (see table A.1). An adaptive procedure for choosing the optimal k exists (Schenker & Taylor, 1996, p. 442), but software implementations of this procedure are lacking. The attenuation bias argument is that $k = 1$ leads to selecting the same donor repeatedly across imputations. The insight of section 3.2 is that once the cell is defined, the bootstrap

frequencies are necessary to correctly reflect the between variance. The nearest neighbor selection function, however, is unable to fully capture the variance of bootstrap frequencies $var(\omega_i)$. If the nearest donor receives a bootstrap frequency that is larger than zero, then it will be selected. The exact value of the bootstrap frequency is irrelevant. It is easily found that $var(\omega_i) \geqslant var\{I(\omega_i)\}$, where I is a function that indicates whether ω_i is larger than zero. Therefore, the nearest neighbor selection is not compatible with the necessary bootstrap step. This finding underpins the criticism by van Buuren (2012) and Morris et al. (2014).

In addition to the nearest neighbor selection, van Buuren (2012, p. 71) and Morris et al. (2014, p. 7) criticize the very popular match type 2 (see table A.1). In the discussion of match types, three different types can be distinguished. Type 1 refers to the matching of $\hat{y}_i$ to $\hat{\tilde{y}}_j$, as in algorithm 2. By contrast, type 2 refers to the matching of $\hat{\tilde{y}}_i$ to $\hat{\tilde{y}}_j$ (Heitjan & Little, 1991, p. 19). Type 3 refers to a procedure in which two sets of parameters, denoted by $\{(\tilde{\beta}_1, \tilde{\sigma}^2_{v,1}), (\tilde{\beta}_2, \tilde{\sigma}^2_{v,2})\}$, are drawn from the posterior distribution, one for the donors and one for the recipients, and $\hat{\tilde{y}}_i \mid (\tilde{\beta}_1, \tilde{\sigma}^2_{v,1})$ is then matched to $\hat{\tilde{y}}_j \mid (\tilde{\beta}_2, \tilde{\sigma}^2_{v,2})$ (Royston & White, 2011; Harrell, 2015). The criticism relates to the one predictor case, where type-2 matching linked with $k = 1$ causes the M multiple imputations to be identical and therefore, prevents the uncertainty associated with parameter estimation from being propagated; again, this is an attenuation bias argument.

The insight from section 3.2 reveals that the M multiple imputations are identical only because the algorithm lacks the necessary bootstrapping. The parametric imputation step as in algorithm 1 is conditioned on one set of parameters drawn in the posterior step, as in the case of type 2. Other match types alter the cell definition and are an engineering trick that treat the symptom, which occurs in the special case of one predictor, but do not cure the disease of effectively omitting the posterior step. Consequently, the discussion on match types is dispensable, and the use of type-2 matching should be advocated for.

3.4 The proposed algorithm

Revisiting the MIDAS algorithm

In contrast to algorithm 2 and all other PMM implementations (see table A.1), the MIDAS algorithm proposed by Siddique & Belin (2008), which has been introduced in section 2.5.4, explicitly combines the two steps that are required based on the insights of section 3.2. The parameters $\tilde{\beta}$ and κ define the cell. The larger κ is, the smaller is the cell. The uncertainty involved in estimating β is correctly considered, and κ is not an estimate. However, because the within cell distribution has sampling error, equation (2.4) involves the bootstrap frequencies. The MIDAS algorithm is thus a major improvement in terms of multiple imputation theory, although its inventors have not been aware of this fact (Juned Siddique, personal communication 2016; Thomas R. Belin, written communication 2017). The proposed algorithm 4 largely builds on MIDAS. Nevertheless, other PMM algorithms could easily be adjusted to deploy the bootstrap frequencies in the imputation step.

Making predictions for recipients and donors

The magnitude of the error, which is caused by partly omitting the posterior step, depends on the magnitude of the between variance that is in turn inversely proportional to the number of available donors. Consequently, the MIDAS algorithm will be particularly beneficial when n_{obs} is small. In small samples, however, the influence of a single data point on the model parameter estimates can be considerable. Because model estimation implies minimizing the distance from the

Algorithm 4 This touched-up version of the MIDAS algorithm is named *midastouch*, which is also the name of our published R package (R Core Team, 2016). Appendix A.4 provides the source code.

1. Obtain bootstrap frequencies ω_i for the donors to introduce the between variance.
2. Draw $\tilde{\beta}$ from a weighted least-squares regression (Greene, 2008, p. 169) with the weights ω_i and calculate the according coefficient of determination $\hat{\tilde{R}}^2$.
3. Calculate the elements of the $n_{mis} \times n_{obs}$ distance matrix using the leave-one-out principle as follows: $\hat{\tilde{\varphi}}_{i,j} = |(x_{\underline{i}} - x_{\underline{j}})\tilde{\beta}_{-i}|$. Here, $x_{\underline{i}}$ denotes the row vector of X_i for the ith donor, $x_{\underline{j}}$ denotes the row vector of X_j for the jth recipient, and $\tilde{\beta}_{-i}$ denotes the weighted least-squares parameter vector from the donor sample without the ith row.
4. Calculate the closeness parameter as follows:

$$\hat{\tilde{\kappa}}(\hat{\tilde{R}}^2) = \left\{50\hat{\tilde{R}}^2 / \left(1 + \epsilon - \hat{\tilde{R}}^2\right)\right\}^{3/8}, \tag{3.1}$$

   where ϵ is a very small positive scalar number used to ensure real results for $\hat{\tilde{R}}^2 = 1$.
5. Insert ω_i, $\hat{\tilde{\varphi}}_{i,j}$, and $\hat{\tilde{\kappa}}$ from above into equation (2.4) and draw the donors.
6. Repeat the above steps $M \geqslant 2$ times, apply Rubin's rules, and multiply the total variances of the means from equation (2.2) by the correction from equation (2.3). Substitute n_{obs} with n_{eff} from equation (3.2), and thus, n with $n_{\mathit{eff}} + n_{mis}$.

model to the donor data, the model is, by construction, closer to the donors than to the recipients, particularly for small n_{obs}, i.e., residuals systematically differ between donors and recipients. For the proof, see appendix A.2. Consequently, the expectation of the residual variance added to the recipients is too small. Although this implementation is still the most common, Gelman & Hill (2011) and Meinfelder & Schnapp (2015) estimate the parameters on the full set of observations by using previously imputed values for y_j. These algorithms make in-sample predictions for both the donors and the recipients. By contrast, the proposed algorithm 4 makes only out-of-sample predictions by estimating the β parameters with the leave-one-out principle.

A flexible closeness parameter

The closeness parameter κ in equation (2.4) determines the influence of the imputation model, i.e., of the conditionality on X_h, on the donor selection. In contrast to Siddique & Belin (2008), who advocate for a fixed value, we argue that κ should reflect the goodness of fit of the imputation model such that $\partial\kappa/\partial R^2 > 0$. In other words, the probability of drawing a distant donor should decrease as the imputation model quality increases, as in equation (3.1). Its functional form is the inverse of the form of the sales response to advertising function presented by Little (1970, p. B472). Siddique & Belin (2008, p. 88) state that reasonable values for κ lie within the range $[0, 10]$, and they found in a simulation study that in a setting with $R^2 = 0.29$, the ideal value for κ is 3 (Siddique & Belin, 2008, p. 98). Equation (3.1) reflects these findings as follows:

$$\kappa(R^2 = 0) = 0, \quad \kappa(R^2 = 0.9) \approx 10, \quad \kappa(R^2 = 0.29) \approx 3$$

Fixing the attenuation bias of the approximate Bayesian bootstrap imputation

Because equation (2.4) generalizes the approximate Bayesian bootstrap imputation, it also suffers from the underestimation of the total variance for finite n_{obs} (Kim, 2002). Applying the correction

factor ϕ from equation (2.3) appears to be the most obvious solution. It applies directly to the k-nearest-neighbor distance function[2] if conducted on the bootstrapped donor sample. The available donors for each recipient, however, are no longer n_{obs}, but rather k, which causes a slight adjustment in equation (2.3): n_{obs} must be substituted by k, and n must be substituted by $k + n_{mis}$. After conditioning on the bootstrap frequencies, all donors have the same probability of being drawn. This is different for the MIDAS algorithm and for algorithm 4, because the drawing probabilities depend on the distance to the recipient. Therefore, we propose replacing n_{obs} in equation (2.3) with a measure of the effective donor sample size for each recipient $n_{j,\mathit{eff}}$ (Kish, 1965, p. 427), which is expressed as follows: $n_{j,\mathit{eff}} = n_{j,obs}^2 / \sum_i \left(w_{i,j}/\omega_i\right)^2$ (Bosch, 2005, p. 5). $w_{i,j}$ and ω_i denote the drawing probabilities from equation (2.4) and the bootstrap frequencies, respectively. Averaging over all recipients and the M imputed data sets yields

$$n_{\mathit{eff}} = \frac{1}{M n_{mis}} \sum_{m=1}^{M} \sum_{j=1}^{n_{mis}} \left[\sum_{i=1}^{n_{obs}} \left\{ \hat{\varphi}_{i,j,m}^{-\hat{\bar{\kappa}}_m} \Big/ \sum_{i=1}^{n_{obs}} \left(\omega_{i,m} \hat{\varphi}_{i,j,m}^{-\hat{\bar{\kappa}}_m}\right) \right\}^2 \right]^{-1} \tag{3.2}$$

Variance correction factors for parameters other than the mean do not yet exist; for linear regression parameters, Wu (1986, p. 1280) offers a starting point.

3.5 Simulation study

3.5.1 Simulation settings

A simulation study is conducted to assess the magnitudes of both the identified shortcomings of the existing PMM algorithms and the proposed improvements. To provide a complete picture, algorithm 4 is challenged by the multiple imputation PMM algorithms implemented in all major statistical software programs, as listed by Morris et al. (2014, p. 3)[3]. Furthermore, two benchmark algorithms are compared: a fully parametric algorithm that utilizes the additional information of a normal likelihood and a fully improper PMM algorithm that treats the maximum likelihood parameter estimates as if they were the true parameters.

For simplicity, we use the multivariate normal setting presented in section 2.3 and set all off-diagonal elements of the correlation matrix equal to each other. To address the various challenges encountered in real-world applications, we apply a full factorial design that considers the following four binary factors: we distinguish 'missing always completely at random' from 'missing always at random' and define the latter as $pr(R = 0) = \Phi\left[(1/4)\left\{Z + N(0,3)\right\}\right]$, where Φ denotes the normal cumulative distribution function (Rinne, 2008, p. 298); we consider $p - 1 = 1$ covariate, i.e., just (Y, Z) versus $p - 1 = 8$ covariates, i.e., (Y, Z) and Q with length 7; we consider $R^2 = 0$ versus $R^2 = 0.75$; and we consider $n_{obs} = 10$ versus $n_{obs} = 200$. Furthermore, we fix $M = 25$, $n_{mis} = 100$, all marginal means equal to zero, and all marginal variances equal to one, and we perform $n_{sim} = 250$ Monte Carlo simulations for each combination.

3.5.2 Simulation results

We focus on the estimates of both the mean of Y, denoted by $\hat{\mu}$, and the regression coefficient of Y in the linear regression model of Z on a constant, Y and Q, denoted by $\hat{\beta}_1$, and thereby cover the more challenging case of missing values in regression predictors (von Hippel, 2007, p. 102).

[2] k-nearest-neighbor selection means that the drawing probability for the k nearest donors is k^{-1}, and zero for all others.

[3] with the exception of Solas for technical reasons.

			950‰ confidence interval coverages			
			$n_{obs} = 10$		$n_{obs} = 200$	
Ref.	Software	Predictive mean matching command	$\hat{\mu}$	$\hat{\beta}_1$	$\hat{\mu}$	$\hat{\beta}_1$
Proposed algorithm (algorithm 4)						
1	*R::midastouch*	`mice.impute.midastouch`	936	961	945	955
2		with correction factor ϕ	973	–	972	–
3	*R::midastouch*	`mice.impute.midastouch(kappa=3)`	931	961	946	945
4		with correction factor ϕ	960	–	978	–
Predictive mean matching software listed by Morris et al. (2014, p. 3)						
5	*R::mice*	`mice.impute.pmm`	605	899	941	959
6	*R::Hmisc*	`aregImpute`	515	872	936	959
7	*R::BaBooN*	`BBPMM`	686	781	937	958
8	*R::mi*	`.pmm`	573	664	908	913
9	*SAS::proc mi*	`regpmm`	487	841	928	943
10	*SAS::MIDAS*	`MIDAS`	899	967	937	954
11	*SPSS*	`multiple imputation /impute scalemodel=PMM`	640	659	907	911
12	*Stata*	`mi impute pmm`	616	652	907	911
13	*Stata*	`ice, match`	443	727	935	958
Benchmark algorithms						
14	*R::mice*	fully parametric: `mice.impute.norm`	962	959	946	958
15	*R*	Fully ignoring between-variance PMM	382	468	877	912

Table 3.2: Simulation results. Ref.: reference; 1-8, 14, 15: R Core Team (2016); 5, 14: van Buuren & Groothuis-Oudshoorn (2011) version 2.22; 6: Harrell (2015); 7: Meinfelder & Schnapp (2015); 8: Gelman & Hill (2011) version 1.0; 9, 10: SAS Institute Inc. (2015); 10: Siddique & Harel (2009); 11: IBM Corp. (2015); 12, 13: StataCorp. (2015); 13: Royston & White (2011). The results show coverages only, because all algorithms deploy the appropriate linear regression imputation model and differences in, e.g., biases are not to be expected.

Utilizing the multiple imputation variance estimator we construct 950‰ frequentist confidence intervals (see section 2.3 and Rubin (1987, p. 21)). For each simulation run, we note whether this confidence interval covers the true parameter value. We present the key results in table 3.2 and the details in appendix A.5. For each cell in table 3.2, we average the coverages over $2^{(4-1)}n_{sim} = 2000$ simulation runs.

The most striking result is that the MIDAS algorithm outperforms all PMM algorithms implemented in the major statistical software programs[4]. The algorithm's advantage is particularly large when the uncertainty associated with the imputation model parameter estimation is considerable, i.e., when the number of donors is small, and it diminishes as the number of donors increases. This result strongly supports the findings of section 3.2.

3.5.3 The proposed algorithm

The results, particularly those for the small donor sample size $n_{obs} = 10$, indicate that our proposed modification of the MIDAS algorithm leads to a considerable improvement. This improvement appears to be true for all means introduced in section 3.4. More specifically, table 3.2 demonstrates that an improvement is achieved for the out-of-sample predictions for the donors, which can be observed by comparing row 10 to row 3; for the modified closeness parameter from equation (3.1), which can be observed by comparing row 3 to row 1; and for the application of the correction

[4]Morris et al. (2014, p. 12) show that PMM algorithms perform best when large k and type-1 matching are employed as in *R::mice* and *SAS::proc mi* (see table A.1). However, the results of these two *tuned* algorithms are not convincing.

factor ϕ from equations (2.3) and (3.2), which can be seen by comparing rows 1 and 3 to rows 2 and 4. It is striking that the coverages of the proposed algorithm do not fall below 950‰ and become closer to the ideal value of 950‰ when n_{obs} increases[5].

3.6 Conclusion and future work

The key finding of this chapter is that all but one PMM implementation systematically attenuate the between variance. In model terms, these implementations do not propagate the uncertainty involved in estimating σ_v^2; in algorithmic terms, they do not use the bootstrap frequencies in the imputation step. In this sense, the MIDAS algorithm proposed by Siddique & Belin (2008) is the exception.

The simulation study results reveal that the attenuation bias can be severe for small sample sizes. Averaging over all PMM implementations except MIDAS provides a coverage for the mean estimate of below 600‰ when n_{obs} is small. This bias can be fully avoided by applying the proposed *midastouch* algorithm.

A natural extension is to deploy other distance metrics than the one described in equation (2.4). The k-nearest-neighbor metric appears to be appropriate if the neighbors are drawn from the bootstrap sample and $k > 1$. A large k requires unequal drawing probabilities to avoid distortions of the distributions. Some reasoning is provided in appendix A.3. Alternative distance metrics have already been discussed in Siddique (2005, p. 130).

Reusing the bootstrap frequencies in the imputation step has a theoretical shortcoming. Within the cells, the sum of the bootstrap frequencies is not necessarily equal to the number of donors, which causes a deterioration of the bootstrap properties (Efron, 1979, p. 3). The obvious alternative is to draw another bootstrap sample within the cell, which in turn presumably overestimates the between imputation variance. Nevertheless, the overestimation is then, again presumably, the part of the variance of σ_v^2 that is caused by β and that is known to be negligible in most cases (Greenberg, 2013, p. 57). More research is required to resolve these conflicts.

[5]The adjustment thus appears to be slightly too large for small samples. However, unlike adjustments that are too small, adjustments that are too large are still in line with default statistical inference as in Rinne (2008, p. 505).

Chapter 4

Local Regression

> Look closer and you'll see something extraordinary, mystifying, something real and true.
>
> Zelda Fitzgerald

4.1 Introduction

Predictive mean matching (PMM) and the newly introduced *midastouch* are nonparametric algorithms that substitute the parametric *imputation* step (Little & Rubin, 2002, p. 201). This substitution is beneficial if the distributions of the incomplete variables conditional on the imputation models cannot be well described. Although there are many cases in which sensible univariate transformations enable parametric imputation (Schafer, 1997, p. 147), PMM has become very popular, primarily because it is a one-fits-all algorithm without the need for manual interventions.

In applications, nonlinear relations are likely to matter. Sensible modeling is capable of solving this issue. However, it is manual, has some arbitrary elements, and takes a considerable amount of time. This chapter introduces local regression, which is capable of automatically detecting the nonlinear relations in the data. The next chapter combines the one-fits-all modeling algorithm local regression (posterior step) and the one-fits-all *midastouch* (imputation step) to one new imputation algorithm with the name *Miles*.

This chapter starts with the fundamentals of local regression (section 4.2). Section 4.3 addresses the statistician's choices when performing local regression modeling. Section 4.4 contributes another perspective to regression analysis, which we hope facilitates an intuitive understanding. All the details of our rather simple implementation of local regression are presented with the help of the data set from Brinkman (1981) as an example in section 4.5. The chapter concludes with discussing potential future improvements in section 4.6.

4.2 Notation

Local regression dates back to Cleveland (1979), although some very early work is related (for an overview, see Cleveland & Loader (1996, p. 15)). The primary use of local regression is to smooth scatterplots (Cleveland, 1979, p. 830). A local regression model is represented by

$$Y = f(X) + v \qquad \text{with} \qquad v \sim i.i.d., \tag{4.1}$$

where Y denotes the response variable and X a set of predictor variables (Loader, 1999, p. 15). Anticipating its use in the next chapter, local regression is notated in equation (4.1) analogous to the imputation model in equation (2.1). At the global level no assumptions about $f(X)$ are required, i.e., the local regression model does not restrict the relation between X and Y globally. In other words, local regression fits any (X, Y) relation. At the local level, i.e., within a smoothing window H_0 around a certain point $x_{\underline{0}}$, $f(X)$ is a polynomial of order π. Observations outside the smoothing window are excluded from estimating $f(X)$ (Loader, 1999, p. 16). Local regression can be regarded as a Taylor series expansion (Loader (1999, p. 37), Bronstein et al. (2013, p. 455)). Consequently, the error equals the remainder of the series (Bronstein et al., 2013, p. 484). Asymptotically, as the smoothing window becomes smaller, local regression provides a perfect fit. The only assumption required is that the first π derivatives of the underlying functional relation exist locally. The following section presents the key fine-tuning options for the algorithm according to Cleveland & Loader (1996, p. 19).

4.3 Local regression modeling

4.3.1 Weight function

In general, observations are weighted in the minimization (Cleveland & Loader (1996, p. 11), Greene (2008, p.169)). The tricube function proposed by Cleveland (1979, p. 831) is by far the most popular weight function in the context of local regression (see, e.g., Cleveland & Loader (1996, p. 20)). The weights d are calculated by

$$\delta_{0,i} = \mid \left(x^*_{\underline{0}} - x^*_{\underline{i}} \right) \mid \qquad \text{with} \qquad i \in H_0 \tag{4.2a}$$

$$d_{0,i} = \left[1 - \left\{ \delta_{0,i} / \max_i(\delta_{0,i}) \right\}^3 \right]^3 + \epsilon. \tag{4.2b}$$

Dividing X_h by the respective interquartile ranges (Rinne, 2008, p. 45) provides the rescaled X^*_h that are used to calculate the distances in equation (4.2a) (Cleveland & Devlin, 1988, p. 597). The weights d_0 for observations outside the neighborhood H_0 of $x_{\underline{0}}$ are zero. The smallest weight for an observation within H_0 is $\epsilon > 0$.

4.3.2 Polynomial degree and bandwidth

Both the size of the neighborhood, also known as bandwidth (Cleveland & Loader, 1996, p. 21), and the degree of the polynomial that is fitted locally represent bias variance trade-offs. A small neighborhood and a high-order polynomial reduce bias but increase variance and vice versa. Polynomial orders discussed in the literature range from local constant fitting to local cubic fitting (Cleveland & Loader, 1996, p. 25). Cleveland & Loader (1996, p. 18) suggest polynomial mixing as averaging the coefficients from two subsequent polynomial fits, such as linear and quadratic, and remark that this procedure is equivalent to ridge regression (Rinne, 2008, p. 639) with a ridge penalty on the quadratic terms only. The bandwidth is generally estimated from the data and defined by the number of observations, the so-called nearest-neighbor bandwidth (Cleveland & Loader, 1996, p. 22), rather than by a fixed width.

We choose mixing a linear fit and a constant fit to prevent an exploding number of parameters. The number of coefficients in the linear fit is p with $p - 1$ denoting the number of predictors; in the quadratic fit, it is already $p(p + 1)/2$; and in the cubic fit, it is $p(p^2 + 3p + 2)/6$. In a data set with 25 variables the quadratic fit estimates 325 parameters and the cubic fit estimates 2925

parameters at the *local* level. Presumably, in the vast majority of applications this large number of parameters causes issues with the number of degrees of freedom.

To find the optimal nearest-neighbor bandwidth and the optimal mixing degree, we minimize cross-validation prediction mean squared errors (PMSE) as suggested in Loader (1999, p. 30). A precise description of the optimization is presented in section 4.5.

4.4 An alternative approach to regression: the influence vectors l

In the neighborhood of $x_{\underline{0}}$ local regression with polynomial mixing is simply a weighted ridge regression. The least squares minimization yields

$$\hat{\beta}_{ridge} = \left(X_i' C^{-1} X_i + \Lambda\right)^{-1} X_i' C^{-1} y_i \tag{4.3a}$$

$$\text{with} \quad C^{-1} = \begin{bmatrix} d_{0,1} & & 0 \\ & \ddots & \\ 0 & & d_{0,n_{obs}} \end{bmatrix}. \tag{4.3b}$$

A property of least squares estimators is that the predictions are always a linear combination of the observed values. This property is extensively used to derive the statistical properties of local regression (Cleveland et al., 1988, p. 95). More formally, the prediction can be written as (Greene, 2008, p. 25)

$$\hat{y}(x_{\underline{0}}) = x_{\underline{0}}' \hat{\beta}_{H_0} = l_{H_0}' y_{H_0}, \tag{4.4}$$

where the subscript H_0 refers to the subset of observations in the neighborhood of $x_{\underline{0}}$ and is dropped in the following for convenience. The influence vector l determines the linear combination and never depends on y_i. Using equations (4.4) and (4.3a) along with the rules for transposing products (Greene, 2008, p. 949), the following equation is obtained:

$$l = C^{-1} X_i \left(X_i' C^{-1} X_i + \Lambda\right)^{-1} x_{\underline{0}}. \tag{4.5}$$

Rao & Singh (1997, p. 59) show that equation (4.5) results from the following minimization problem[1]

$$\min_{l} \left[(l-d)' C (l-d) + (X_i' l - x_{\underline{0}})' \Lambda^{-1} (X_i' l - x_{\underline{0}})\right]. \tag{4.6}$$

Equation (4.6) shows that local regression essentially attempts to find a new weight vector, which is called the influence vector l, that is supposed to be as close to the distance weights d as possible (first addend) while ensuring that the influence weighted center of the observations in the neighborhood falls on $x_{\underline{0}}$ (second addend). Large ridge penalties Λ reduce the importance of the second addend and result in a prediction that is dominated by the distance weights d.

[1] Their solution is slightly more complex because it contains another additive term. However, if X_i has full column rank and a leading constant column and C is defined as in equation (4.3b), then this additive term is exactly zero. For the proof, see section B.1.

4.5 Example

4.5.1 A well known data set

To present a detailed picture of how we implement the algorithm, we continue Cleveland's tradition (Cleveland & Devlin (1988, p. 604), and many more) and use the $\texttt{NO}_\texttt{x}$ data set presented by Brinkman (1981), which is about an experiment with ethanol rather than gasoline to study the effects on the nitrogen oxide emission of an engine. The data set consists of $n = 88$ observations and $p = 3$ variables, namely, the outcome $\texttt{NO}_\texttt{x}$ and two experimentally controlled factors, the compression ratio `C` and the equivalence ratio `E`.

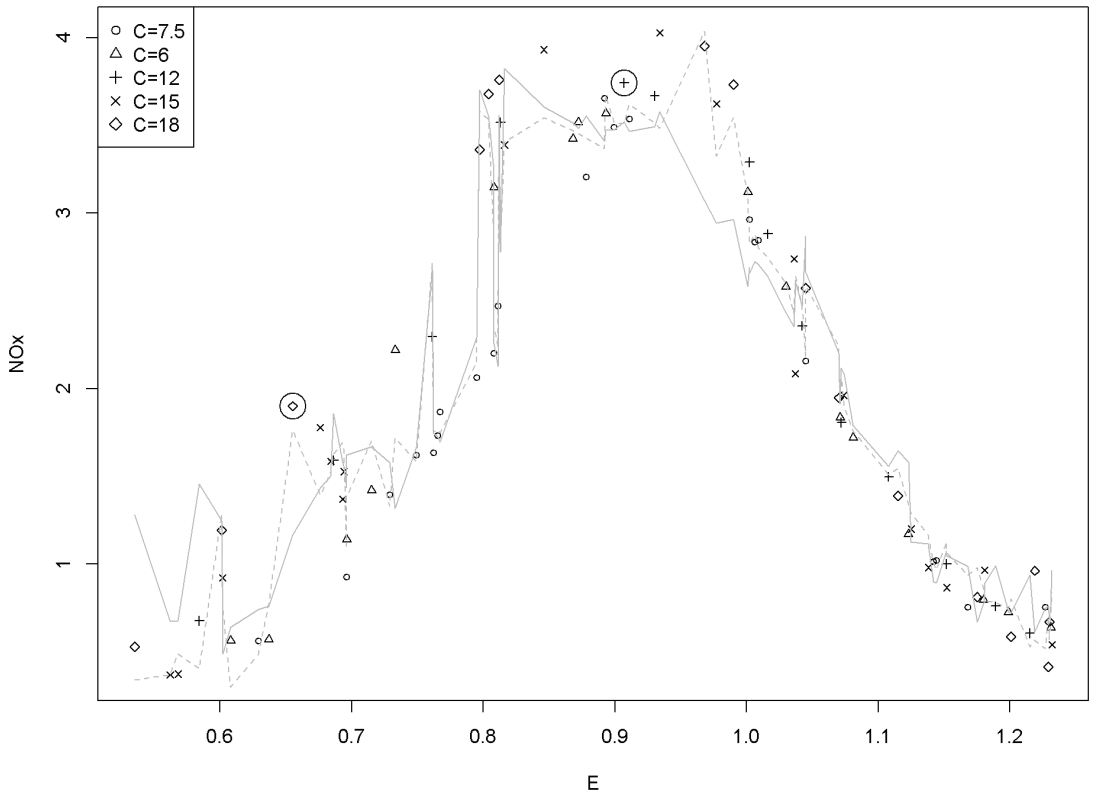

Figure 4.1: The $\texttt{NO}_\texttt{x}$ data set by Brinkman (1981) with the optimal local regression fit ($q = 7, \lambda = 0$, dashed line) and the local regression fit from the grid optimization ($q = 5, \lambda = 0.2$, solid line).

Figure 4.1 displays $\texttt{NO}_\texttt{x}$ against `E`. The dimension of `C` is shown by the symbols of the data points, because `C` only takes five different values. The solid gray line is the local regression fit in our implementation, which explains $\hat{R}^2 = 91\%$ of the variation of $\texttt{NO}_\texttt{x}$ (Rinne, 2008, p. 90). The two highlighted and randomly selected example points are the first and the last points in the data set: $P_1(\texttt{NO}_\texttt{x}, \texttt{C}, \texttt{E}) = (3.741, 12, 0.907)$ and $P_{88} = (1.9, 18, 0.655)$.

4.5.2 Optimization of the bandwidth and the polynomial mixing degree

Before calculating multivariate distances (see equation (4.2a)), the variables need to be rescaled. The interquartile ranges are $IQR_\texttt{C} = 7.5$ and $IQR_\texttt{E} = 0.36$. Before rescaling the variances are

$var(\texttt{C}) = 15.46$ and $var(\texttt{E}) = 0.04$, and after rescaling, the variances are $var(\texttt{C*}) = 0.32$ and $var(\texttt{E*}) = 0.39$.

Optimization of the bandwidth and optimization of the mixing degree involve one scalar parameter each. The parameter governing the bandwidth is the number of observations within the neighborhood q, and the parameter governing polynomial mixing is the ridge scalar λ in the weighted ridge regression equation (4.3) with

$$\Lambda = \lambda \begin{bmatrix} 0 & & & 0 \\ & \zeta_{\langle 2,2 \rangle} & & \\ & & \ddots & \\ 0 & & & \zeta_{\langle p,p \rangle} \end{bmatrix} \quad \text{with} \quad X_i' C^{-1} X_i = \begin{bmatrix} \zeta_{\langle 1,1 \rangle} & \cdots & \zeta_{\langle 1,p \rangle} \\ \vdots & \ddots & \vdots \\ \zeta_{\langle p,1 \rangle} & \cdots & \zeta_{\langle p,p \rangle} \end{bmatrix}. \tag{4.7}$$

As shown in equation (4.7), $\lambda > 0$ increases all elements of the main diagonal of the $X_i' C^{-1} X_i$ matrix except the one that corresponds to the constant in the model (van Buuren & Groothuis-Oudshoorn, 2011). A value of $\lambda = 0.1$ means that 10% are added to the main diagonal. The algorithm first draws a simple random sample of 50 from the data and builds a grid for the optimization. For each grid cell and each of the 50 data points an out-of-sample prediction is made, resulting in 900 ridge regression estimations.

$q = 5$		$q = 21$		$q = 38$		$q = 54$		$q = 71$		$q = 87$	
λ	PMSE	λ	PMSE	λ	PMSE	λ	PMSE	λ	PMSE	λ	PMSE
0.2	<u>0.1201</u>	0.1	0.2237	0.05	0.4506	0.05	0.6565	0.05	0.8387	0.05	1.0400
0.4	0.1206	0.2	0.2312	0.1	0.4927	0.1	0.7200	0.1	0.9189	0.1	1.1103
0.8	0.1209	0.4	0.2354	0.2	0.5190	0.2	0.7604	0.2	0.9718	0.2	1.1601

Table 4.1: Choosing the optimal q and the optimal λ

Table 4.1 shows that the prediction mean squared error (PMSE, Loader (1999, p. 30)) is minimal for $P_{min}(q, \lambda) = (5, 0.2)^2$.

4.5.3 Neighborhood definition and minimization

The model specification is now set. The algorithm continues by calculating the distances δ and the weights d according to equation (4.2). For this purpose the `R::RANN` library is used (R Core Team (2016), Arya et al. (2015)).

For the points P_1 and P_{88}, figure 4.2 shows that closer neighbors receive larger weights than do more remote neighbors.

The key advantage of polynomial mixing and least squares is that the minimization can be conducted analytically as described in equation (4.3) and that a large number of predictors can be digested. Thus, solving the $n = 88$ minimization problems takes less than 5ms on a single core of an `intel i7 5600U`. The relatively large regularization parameter $\lambda = 0.2$ causes the influence vectors l for the points P_1 and P_{88} to equal the rescaled weights d up to 10^{-2} (see section 4.4). That is, the local regression fit is dominated by the d weighted constant. This makes intuitive sense because with $q = 5$, the number of estimated parameters used for prediction shall be as small as possible. Figure 4.3 shows the regression lines for P_1 and P_{88}, which are almost horizontal.

[2] A full grid optimization yields a minimum PMSE of 0.0494 at $P_{min}(q, \lambda) = (7, 0)$ and an $\hat{R}^2 = 96\%$ (see figure 4.1). Although the values of λ in the grid optimization appear to be too large in this example, they might be suitable in more realistic scenarios with a large number of predictors.

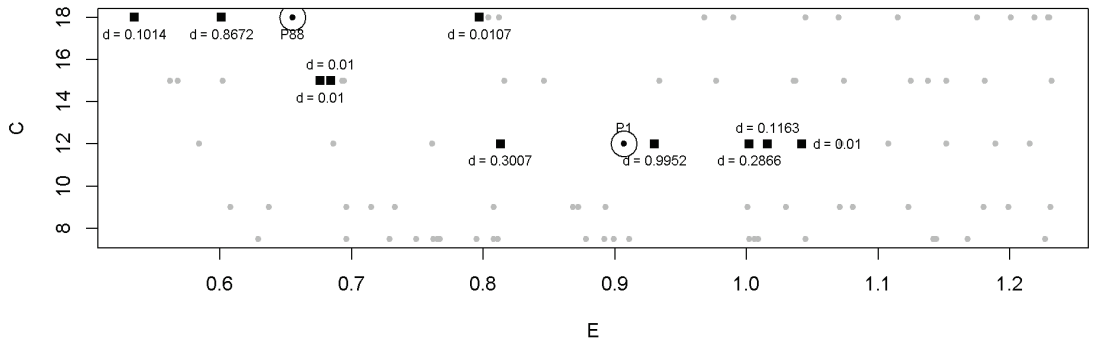

Figure 4.2: The two predictors `C` and `E` of Brinkman (1981) with highlighted P_1 and P_{88} and their respective $q = 5$ neighborhoods.

Their intersections with the vertical lines, which indicate the `E` values of the two points, mark the predicted $\texttt{NO}_\texttt{x}$.

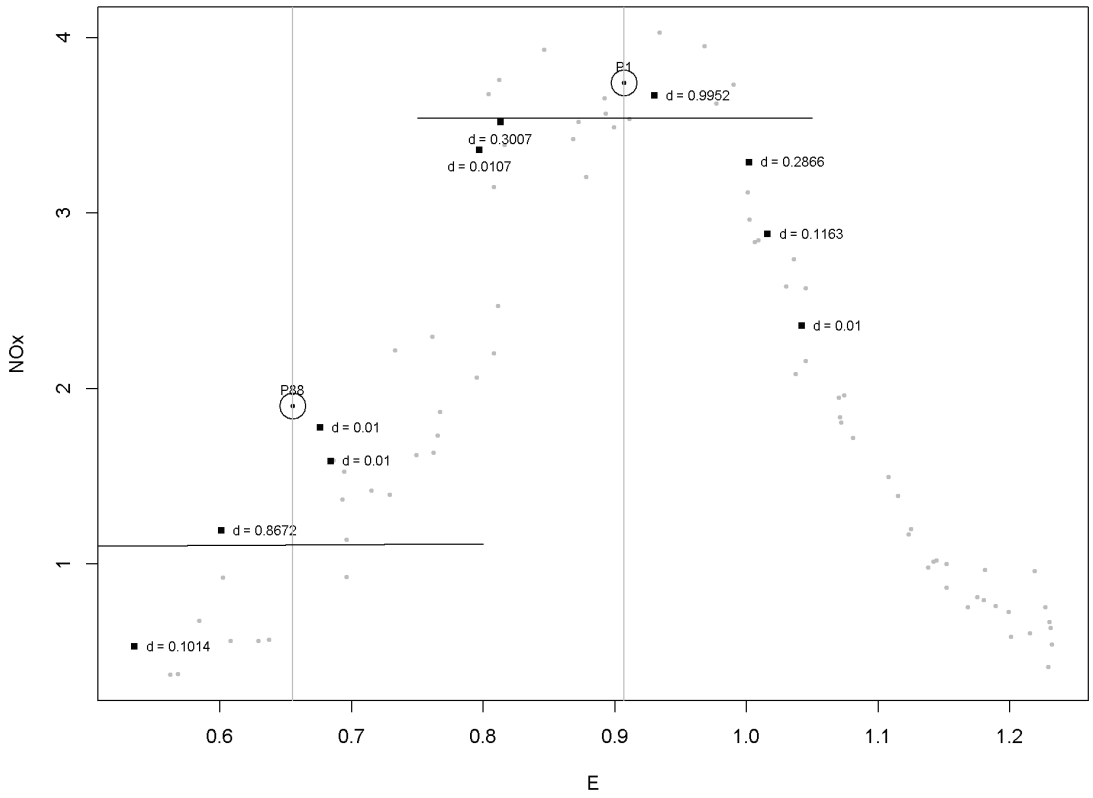

Figure 4.3: Regression lines for P_1 and P_{88}

4.6 Potential improvements

Our implementation of local regression that is described in section 4.5 is rather simple. Extensions have already been developed in different directions. Cleveland & Loader (1996, p. 31) suggest

choosing both q and λ locally rather than globally. Cleveland (1979, p. 829) has already argued to substitute least squares optimization. Whereas he suggests an outlier-robust alternative, Loader (1999, p. 59) describes how maximum likelihood procedures can be localized to better model, e.g., categorical responses.

If $x_{\underline{0}}$ falls outside the domain of its neighborhood, e.g., if all of $x_{\underline{0}}$'s q nearest neighbors are left of $x_{\underline{0}}$ itself, regression switches from interpolation to extrapolation, which causes some elements of l to be negative. Consequently, the prediction suffers from large uncertainty. One approach to address this issue might be to prevent extrapolation with existing algorithms (for an overview, see Rao & Singh (1997)), whereas another approach might be to search a surrounding neighborhood rather than only a close neighborhood H_0.

The grid optimization suggested in section 4.5.2 needs improvement. The grid is arbitrary and potentially far too imprecise in the crucial regions. Further research is needed here. One idea is to use closed-form solutions to find the optimal λ (Rinne, 2008, p. 640) conditional on the numerical optimization of q alone[3].

[3]Note that these closed-form solutions depend on the ordinary least squares estimates, which do not exist in the likely case of $p > q$.

Chapter 5

Multiple Imputation via Local Regression: The *Miles* algorithm

> Our knowledge can only be finite, while
> our ignorance must necessarily be infinite.
>
> Karl Raimund Popper

5.1 Introduction

This chapter combines the posterior step one-fits-all local regression algorithm, which is introduced in chapter 4, and the imputation step one-fits-all *midastouch* algorithm, which is introduced in chapter 3, to the multiple imputation via local regression (*Miles*) algorithm, which is a key contribution of this dissertation[1]. *Miles* is not assumption free, but it resolves two major limitations of the fully parametric approach, which is presented in section 2.3. In the posterior step it does not heavily restrict the relation between the incomplete variable and its predictors; it does, e.g., not assume a linear relation as in equation (2.1). In the imputation step, it does not require a distributional assumption for the data; it does not, e.g., require $y_h \mid X_h$ to be normally distributed as in equation (2.1). The next section presents the *Miles* algorithm.

It is tempting to believe that *Miles* is beneficial only if the analysis model is unknown to the imputer. If the imputer is aware of the analysis model, an *according*, later called *educated*, imputation model appears to be the best option, and an overly inclusive imputation model, such as *Miles*, appears to be unnecessarily inefficient (see section 2.2). However, when imputing transformed variables, the *according* imputation model procedures yield biased and inconsistent estimates in the relevant case of missing at random (MAR), as will be presented in section 5.3. Section 5.4 provides an overview of the possible approaches, which are compared in a simulation study in section 5.5. The ideas of this chapter are published in Gaffert et al. (2016); thus, this paper is not cited in the following text.

5.2 The proposed algorithm

Miles is the combination of the local regression algorithm from chapter 4 and the *midastouch*

[1] Aerts et al. (2002) proposed local models for multiple imputation. Their approach, however, relies upon Kernel smoothing and lacks the advantages of local regression (Hastie & Loader, 1993) and predictive mean matching (PMM). De Jong (2012, p. 43) first sketches some ideas of a local-regression based multiple imputation algorithm.

Algorithm 5 The *Miles* algorithm.

1. Run a Bayesian bootstrap (Rubin, 1981) and conduct all subsequent steps on the bootstrap sample.
2. For each observation, i.e., both donors and recipients, make a local regression prediction as described in chapter 4.
3. Obtain the imputations by drawing donors with probabilities given by equation (2.4) from within the recipients' neighborhoods.
4. Repeat the steps above for iterating over all variables and multiple imputations (van Buuren, 2012, p. 110).

algorithm from chapter 3. *Miles* is presented in algorithm 5, and the corresponding source code is given in appendix C (R Core Team, 2016). Because *midastouch* requires bootstrap frequencies, it appears to be natural to introduce between variance by bootstrapping the donors (the replication step in von Hippel (2007, p. 84)), i.e., to estimate the local regressions on $M \geqslant 2$ bootstrap samples. Rubin (1981, p. 132) shows that limiting the bootstrap frequencies to integers is inefficient and proposed the Bayesian bootstrap, which reaches the same degree of precision as Efron (1979)'s bootstrap faster. Otherwise, the properties are equivalent, except for very small samples[2]. That is why *Miles* employs the Bayesian bootstrap. The noninteger bootstrap frequencies from the Bayesian bootstrap ω_{BB} simply substitute ω_i in equation (2.4); thus, the *midastouch* algorithm naturally incorporates the Bayesian bootstrap.

Another specialty of *Miles* is that once the neighborhood is selected, both the modeling and the imputation step are conducted in the neighborhood. Thus, only donors within the neighborhood have positive selection probabilities according to equation (2.4). Donors outside the neighborhood have zero probability, although they may be closer in terms of the predicted mean. This property makes the algorithm truly local.

5.3 Imputing transformed variables

Using the definitions of section 2.3 and relaxing the distributional assumptions for Z, suppose that the analysis model has the following form:

$$Z = \gamma_0 + \gamma_Q Q + \gamma_Y Y + u + \sum_t \gamma_t g_t (Q, Y),$$

with u denoting independent normal noise and g_t denoting the tth nonlinear transformation term. If $g = 0$, parametric multiple imputation as presented in algorithm 1 is proper. For more complex g, there are two types of imputation algorithms. Algorithms that explicitly consider the analysis model are referred to as *educated*, and the others are referred to as *ignorant*. An *educated* imputation algorithm is referred to as *omniscient* if the analysis model exactly depicts the data generating process.

There are two educated approaches to address complex g; the passive-imputation algorithm (PI) by van Buuren & Groothuis-Oudshoorn (1999, p. 13) and the just-another-variable algorithm (JAV) by von Hippel (2009, p. 271). To introduce both algorithms, suppose that $g(Q, Y) = Y^2$. The sample data set then consists of four columns (z_h, q_h, y_h, y_h^2), where the first two are fully observed and the latter two have missing values for exactly the same observations. Recall the

[2]Rubin (1981, p. 131) shows that the variance of the mean estimator differs by the factor $n/(n-1)$.

multivariate two patterns of section 2.4. The passive-imputation algorithm proceeds as algorithm 1, but it also includes the squared term in the regression model for y_h. After the imputation of y_j, it simply computes the squares. This is why, von Hippel (2009, p. 272) names this approach 'impute (the linear terms), then transform'. The just-another-variable algorithm imputes y_j as does the passive-imputation algorithm. However, rather than calculating the square from the imputed values, it repeats the imputation procedure for y_j^2 and thereby treats the transformation as if it were *just another variable*. Logical inconsistencies between y_j and its transformations g are a natural consequence of this procedure, which can be an essential disadvantage as noted by van Buuren (2012, p. 132). If the response mechanism is completely at random (MCAR), then the just-another-variable algorithm enables consistent estimation of the parameters of interest whereas the passive-imputation algorithm does not. For the missing at random mechanism, which is by far the most relevant in applications, neither of the two approaches provides consistent estimates (Seaman et al., 2012, p. 7)[3].

5.4 Educated versus ignorant imputers

Both the PI algorithm and the JAV algorithm require the imputer to know the analysis model. This appears to be a doable requirement, because all the imputer needs to do is talk to the analyst. However, in many applications, talking to the analyst is a tricky task. Consider public use files, where one imputer at the agency provides the file but where hundreds of analysts run highly sophisticated models driven by theories from their fields (Rubin, 1996, p. 473). Some very talented imputers might actually be capable of performing this job. However, there is no doubt that this takes substantial time and effort. Now, recall that if the missing data pattern is not missing completely at random, neither of the two educated approaches provides consistent parameter estimates; therefore, what is the reward for all this work?

Doove et al. (2014) present random forest imputation, which is introduced in section 2.5.5, as an ignorant algorithm and show that it preserves ignored interaction effects well. Local regression, which serves as the posterior step for *Miles*, consistently captures a broad class of functional relations (see section 4.2). Both algorithms, random forest imputation and *Miles*, appear to be very inclusive (see section 2.2) and thus well suited to preserve the relevant relations in the data. Ignorant approaches generally require MAR, not MCAR, and do not cause inconsistencies as the JAV algorithm.

5.5 Simulation study

5.5.1 Simulation setup

Using a simulation study, the relative and absolute performances of the proposed *Miles* algorithm are assessed. The major distinction is between ignorant approaches that take the linear terms as an input only and omniscient approaches that utilize the linear terms and the relevant transformations from the analysis model, which is equivalent to the data generating process in this setting. The ignorant approaches are random forest imputation by Doove et al. (2014), the version of predictive mean matching (PMM) in van Buuren (2012, p. 68), which is recommended by Morris et al. (2014), and the proposed *Miles*; the omniscient approaches are the PI algorithm and the JAV algorithm. The omniscient approaches also utilize PMM in the imputation step to achieve better comparability with the ignorant approaches.

[3]Vink & van Buuren (2013) propose a third educated solution for the special case of a squared term.

Table 5.1: Simulation results

	Ignorant approaches						Omniscient approaches			
	Miles		PMM		RF		PI**		JAV**	
Linear only*	$\sum_t \gamma_t g_t(Q,Y) = 0$									
$\gamma_0 = 0$	151	(930)	143	(944)	175	(928)	142	(939)	143	(945)
$\gamma_Q = 1$	152	(938)	146	(953)	175	(925)	146	(948)	146	(950)
$\gamma_Y = 1$	152	(908)	139	(910)	175	(806)	141	(901)	140	(904)
Square	$\sum_t \gamma_t g_t(Q,Y) = \gamma_{Y2} Y^2$									
$\gamma_0 = 0$	177	(945)	199	(907)	168	(898)	351	(305)	145	(955)
$\gamma_Q = 1$	209	(931)	224	(947)	198	(953)	407	(951)	174	(954)
$\gamma_Y = 1$	202	(952)	225	(919)	190	(937)	418	(952)	164	(929)
$\gamma_{Y2} = 1$	279	(930)	285	(846)	269	(819)	563	(175)	233	(907)
Interaction	$\sum_t \gamma_t g_t(Q,Y) = \gamma_{QY} QY$									
$\gamma_0 = 0$	265	(939)	324	(962)	250	(934)	338	(914)	195	(919)
$\gamma_Q = 1$	267	(941)	332	(939)	252	(941)	334	(946)	205	(884)
$\gamma_Y = 1$	262	(808)	308	(717)	244	(789)	331	(872)	195	(944)
$\gamma_{QY} = 1$	263	(791)	321	(537)	248	(720)	334	(230)	188	(875)
Cube	$\sum_t \gamma_t g_t(Q,Y) = \gamma_{Y3} Y^3$									
$\gamma_0 = 0$	206	(949)	225	(957)	224	(957)	483	(949)	218	(922)
$\gamma_Q = 1$	192	(941)	206	(951)	210	(952)	438	(835)	211	(926)
$\gamma_Y = 0$	123	(963)	134	(924)	132	(954)	275	(629)	134	(917)
$\gamma_{Y3} = 1$	421	(964)	441	(951)	445	(907)	1007	(938)	457	(935)
Average	221	(922)	243	(891)	224	(895)	308	(801)	197	(924)

Table 5.2: The table presents relative root mean squared errors (rRMSE) ×100. A value of 100 means that the RMSE of the respective parameter estimate in the imputed data set is as large as the RMSE of this parameter before deletion. Coverages of 950‰ intervals are given in in parentheses. Abbreviations are: Multiple imputation via local regression (*Miles*); Predictive mean matching (PMM); Random forest imputation (RF); Passive imputation (PI); Just another variable (JAV). *When $g = 0$, PMM, PI, and JAV are identical algorithms. **The results for JAV and PI in Gaffert et al. (2016) are misleading due to an error in the implementation and are corrected here. As a consequence, in Gaffert et al. (2016) JAV looks worse and PI looks better than it really is. The PI results here are based on 50 Gibbs sampler iterations (see section 2.4).

(Q, Y) follow a standard normal distribution with a $\rho = 0.2$ correlation (Rinne, 2008, p. 201). The missingness is always at random and defined by $pr(R = 0) = \Phi[(1/4)\{Q + N(0, 3)\}]$. We fix $M = 10$, $n_{mis} = 90$, and n_{obs} as low as 60 to obtain a substantial degree of estimation uncertainty of the imputation model parameters. Throughout the different analysis models, the coefficient of determination is maintained at approximately $R^2 = 2/3$ (Rinne, 2008, p. 79), and for each model, $n_{sim} = 1000$ Monte Carlo simulation runs are performed.

5.5.2 Simulation results

Table 5.1 shows the results for all parameters of interest and all introduced imputation methods in two dimensions. The relative root mean squared error (Rinne, 2008, p. 17), abbreviated as rRMSE, is defined as the ratio of the RMSE of the imputed data sets divided by the RMSE before deletion. Small values indicate good quality. The rRMSE is a quality indicator in descriptive statistics. To obtain a quality indicator in inferential statistics, we construct 950‰ confidence intervals as in section 3.5. Good quality is indicated by coverage values of approximately 950‰.

Overall, the JAV algorithm performs the best. *Miles* keeps up with JAV in terms of coverages, but it adds approximately 12% rRMSE due to ignoring the analysis model and thereby, for this

simulation study, the data generating process. Random forest imputation performs as well as *Miles* in terms of RMSE, but it performs significantly worse in terms of coverages. Ignorant PMM and PI are clearly outperformed. JAV is the only procedure that does not ensure consistent imputation, i.e., the imputed values do not obey the transformation rules. With the focus on preserving interaction effects, Doove et al. (2014) introduce random forest imputation, which is, in this regard, slightly superior to *Miles* but clearly inferior to JAV.

5.6 Conclusion and future work

In this chapter, local regression and *midastouch* are combined to form the multiple imputation via local regression algorithm *Miles*. It is an inclusive algorithm in the sense that it attempts to capture the true nature of the data rather than preserve a predefined, e.g. linear, relation (see section 2.2).

In many practical applications, it appears to be advantageous to use an inclusive imputation algorithm. When the analysis model is unknown to the imputer, the best that the imputer can do is to preserve the major structure of the data, i.e., apply an inclusive imputation algorithm. If there are many analysis models and no one imputation model can include all parameters of interest, again, the best that the imputer can do is to apply an inclusive algorithm[4].

This chapter presents another less obvious scenario for inclusive algorithms: there is only one perfectly known analysis model that involves nonlinear relations[5]. Finding a suitable imputation model can turn out to be a serious burden for an imputer. After having established the model, the imputer can apply one of the two educated algorithms: the PI algorithm by van Buuren & Groothuis-Oudshoorn (1999) or the JAV algorithm by von Hippel (2009). The simulation results for PI are disastrous; JAV, while performing the best, has the disadvantage of inconsistent imputations. In the context of just one perfectly known analysis model, the inclusive imputation algorithms are referred to as ignorant because they are not provided with the analysis model. Consequently, ignorant algorithms, such as the proposed *Miles*, are much easier to deploy than educated ones because there is no need to worry about the functional relation for the imputer. Furthermore, because all transformations required for the analysis model are calculated on the imputed data set, inconsistencies cannot arise. These practical considerations may even outweigh the 12% rRMSE advantage of JAV over *Miles*.

JAV's advantage over *Miles* is particularly large in the case of the interaction. Thus, research is needed to improve *Miles*'s capability to capture interaction effects.

In general, the performance of the omniscient approaches is disappointing. All information to conduct a sensible imputation is available to them in the simulation study, even the true data generating process. With this valuable information, PI performs considerably worse than the three ignorant algorithms, and JAV can only slightly outperform *Miles*. This result clearly indicates a lack of suitable educated algorithms.

[4] or specify more than one imputation model.

[5] An inclusive algorithm clearly cannot be beneficial if the analysis model is both known and perfectly linear. This case is also shown in table 5.1.

Chapter 6

Real Data Simulation Study

> So, please, oh please, we beg, we pray, go throw your TV set away.
>
> Roald Dahl

6.1 Introduction

In the preceding chapters 3 and 5, simulations on artificial data were used to assess the properties of the newly proposed algorithms *midastouch* and *Miles*, respectively. Although such simulations foster the understanding of the underlying mechanics, the ultimate goal of imputation algorithms is to enable real-world statistical analyses on real-world incomplete data. Using a data set from the GfK SE company, this chapter investigates the algorithms' performance when applied in practice by answering the following research questions:

1. *midastouch*: The simulation results of chapter 3 show that for multivariate normal data the algorithm is superior to default predictive mean matching (PMM) when the number of donors is small, and does not differ from default PMM when the number of donors is large. The hypothesis is that the latter also holds in a real data set with nonlinear relations as indicated by Siddique & Belin (2008, p. 96).

2. *Miles*: The dependency structure within a real data set is usually not linear. *Miles* has been developed to capture the true structure of a data set without the need to specifying it explicitly in the imputation model. The hypothesis is that *Miles* performs best, because it approximates all kinds of nonlinear relations (see section 4.2).

Commonly applied analysis models for assessing the quality of imputation procedures are means and regression coefficients (Morris et al., 2014, p. 7). The results for analysis models that are specifically relevant in market research are also included, namely contingency tables, cluster analysis, and variances. As in section 5.5, relative root mean squared errors and coverages of confidence intervals are employed as the key performance indicators in descriptive and inferential statistics, respectively.

There are three approaches for assessing the quality of an imputation algorithm using real data. The most obvious approach is to take one data set with missing values and perform the imputation (Siddique & Belin, 2008, p. 90). The reader can learn about applying the algorithm, and plausibility checks can be conducted. However, because the values are missing, there is no way

to compare the imputed data to any truth. The second approach involves a large data set that is completely observed. Subsamples of this data set are drawn in a simulation study setup, and the missing values are created artificially (Andridge & Little, 2010, p. 17). The set of validation tools for this simulation approach is considerably larger. The biases and coverages of the parameters of analysis models can be evaluated. To overcome the drawback of a fully artificial response mechanism, Heitjan & Little (1991, p. 24) employ the complete cases of a data set with missing values rather than a fully observed data set as a third approach. In addition, they establish a model for the response indicator and use the parameter estimates to create a nearly natural missing data pattern. Their approach is very similar to the one followed in this chapter.

The next section provides the details about the data set. To ensure readability, most of the descriptive figures and tables are presented in appendices D.1 and D.2. Section 6.3 introduces the simulation setup and the analysis models. The simulation results are presented in section 6.4 and appendix D.6 before section 6.5 concludes.

6.2 The data

The GfK SE company owns household panel data in Germany. Once a year, the panelists are asked to complete a survey on their media consumption. In this chapter, the focus is on TV consumption, which is also part of the survey. Because not every panelist completes the survey, there are missing values in the variables about TV consumption. Imputation algorithms are used to enable statistical analyses of the media survey data. The survey data are introduced in section 6.2.2. In the panel households, TV consumption is also measured passively. Special smart phone devices record the sound of the TV sets in the household and transmit the audio files to a server, where they are matched to a TV program database. The technically measured data are introduced in section 6.2.1. The purpose of the survey is to learn about media consumption behavior in general; the purpose of the passive measurement is to learn about exposures to specific advertisements. Although it would be possible, the two data sets have not yet been combined. Aggregating the passively measured TV consumption data to the survey format required a substantial amount of effort as shown in section 6.2.3.

The key idea now is to define the panelists with passive measurement in place as the population of interest. Some of these panelists choose not to answer the questionnaire. In this setting, let Y_h denote the aggregated passively measured TV consumption data, which are completely observed. Let Y_h^* denote the incomplete survey TV consumption data, which are presumably more prone to measurement error than Y_h. Thus, normally, there is only Y_h^*, and imputation is required to address its missing values. Due to the considerable extra effort, however, and only for the year 2014, there is also the complete Y_h. In other words, the TV consumption is known for those who have not answered the TV consumption questions. This special data set allows us to learn about the response mechanism. Section 6.2.4 presents a test for the missing at random assumption. These insights can perhaps help improve the imputation for Y_h^* in the usual application, when Y_h is unavailable.

The data situation is very similar to the one in David et al. (1986). In their application, Y_h^* is income from the Current Population Survey, which also suffers from nonresponse and is generally imputed. Their complete Y_h is from the Internal Revenue Service and was also available for one period only. David et al. (1986) compare different imputation methods by validating the imputed Y_h^* using Y_h.

6.2.1 The passively measured data set

In the participating households, TV consumption is measured by sound recording. Audio records are matched with a database. A successful matching requires any audio record of one channel to last at least eight seconds. The database consists of the eleven most important TV channels in Germany, which are listed in table D.1 in the appendix. Other audio records are not used. The recording devices send an 'I'm alive' message at least every five minutes to distinguish between no relevant TV consumption and a broken measurement device. Panelists are required to log on and log off the measurement device before and after watching TV to indicate who is watching. An automated post-processing step assigns the most probable person or persons to a TV event if the panelist identification has not occurred. One TV event is defined by the starting date and time, the ending date and time, the channel, one panelist identification, one measurement device identification, and one household identification. When multiple people enjoy the same TV program, the same event will occur more than once with different panelist identifications. Switching channels creates a new event. In the time between 1 a.m. and 6 a.m., the measurement is unreliable because technical maintenance and data transmission are conducted.

6.2.2 The survey data set

We use data from the 2014 media survey, which was conducted in May 2014. The survey consists of questions about TV consumption, print media consumption, and Internet usage. There are four types of TV-related questions; three of them, however, are specific to the survey and not contained in the passive measurement. These questions are the following:

- *One* question on the duration of overall daily TV consumption and *one* on pay TV. Both exceed the scope of passive measurement's 11 channels.
- *One* question on different TV genres. There is no database that matches genres to viewing times, thus, this information is not provided by the passive measurement.

The only questionnaire information that can be appropriately rebuilt from the passive measurement data is the following:

> At what time of day do you usually watch the following TV channel on an ordinary
>
> 1. weekday?
> 2. Saturday?
> 3. Sunday?

The answers to each of the three types of days are provided in the rectangular structure of table 6.1.

Non-TV-related questions are discarded. Because the questionnaire is sent out to panelists, some basic claims data are available. A description thereof is provided in table D.2 in the appendix.

6.2.3 Fitting the passively measured data into the survey data format

The goal is to aggregate the TV event data into the structure of the survey data as in table 6.1. To conduct the aggregation, a few decisions must be made.

- We use the data one year prior to the survey, which means records from May, 1st 2013 through April, 30th 2014. This approach assumes that the respondents when asked for their TV consumption rather refer to the past than to the future.

time of day	ARD	ZDF	...	VIVA	another channel
06 a.m. - 09 a.m.	☐	☐	...	☐	☐
09 a.m. - 10 a.m.	☐	☐	...	☐	☐
10 a.m. - 11 a.m.	☐	☐	...	☐	☐
11 a.m. - 12 p.m.	☐	☐	...	☐	☐
12 p.m. - 01 p.m.	☐	☐	...	☐	☐
01 p.m. - 02 p.m.	☐	☐	...	☐	☐
02 p.m. - 03 p.m.	☐	☐	...	☐	☐
03 p.m. - 04 p.m.	☐	☐	...	☐	☐
04 p.m. - 05 p.m.	☐	☐	...	☐	☐
05 p.m. - 06 p.m.	☐	☐	...	☐	☐
06 p.m. - 07 p.m.	☐	☐	...	☐	☐
07 p.m. - 08 p.m.	☐	☐	...	☐	☐
08 p.m. - 09 p.m.	☐	☐	...	☐	☐
09 p.m. - 10 p.m.	☐	☐	...	☐	☐
10 p.m. - 11 p.m.	☐	☐	...	☐	☐
11 p.m. - 12 a.m.	☐	☐	...	☐	☐
12 a.m. - 01 a.m.	☐	☐	...	☐	☐
01 a.m. - 06 a.m.	☐	☐	...	☐	☐

Table 6.1: Question on TV consumption in the media survey

- Panelists with no TV event at all in the one year period are excluded from the sample; so are panelists with less than 120 days of passively measured data.
- The survey questions ask for the behavior on an *ordinary* day. This is why public holidays are excluded.

Note that this data set is not used to infer to any larger population of interest in the real world, which is why the most convenient subset is selected. Now, for each person, each time slot of interest, and each channel, the aggregation proceeds as follows.

1. Count the number of suchlike timeslots with an active measurement device for that household.
2. Count the number of suchlike timeslots in which a TV event was measured for the particular channel and for the particular person.
3. Divide the result from 2. by the result from 1.. If the ratio is at least 5%, the answer is `yes`, else the answer is `no` (see table 6.1).

Passively measured TV consumption data are available for 11916 persons living in 6136 households. Among them, 7935 persons living in 4992 households have completed the survey, and 3981 persons living in 2302 households have not. Based on the respondents only, it is possible to check how well the passively measured data match the survey data. Although both data sets should reflect the same truth in the sense that there is only one true TV consumption behavior of a particular person at a particular time, nobody would expect the two data sets to be exactly alike. The column labeled *accuracy* in table D.1 in the appendix contains the percentage of time slot cells equal to the survey data (Flach, 2012, p. 54) and ranges between 71% and 99%. In table D.1 in the appendix, the channels are sorted in descending order by their overall reaches in the data. Notably, the accuracy appears to be sorted in ascending order. In fact, Pearson's correlation (Rinne, 2008, p. 76) between the overall reach and the accuracy is $\rho = -0.99$. As the values of Cohen (1960)'s κ_C indicate this large correlation hardly implies that watching smaller TV stations can be better recalled but are rather an artifact of the accuracy measure.

6.2.4 Testing the missing at random assumption

In general applications, the assumption about missingness at random cannot be tested (van Buuren, 2012, p. 31). The joint density $pr(X, Y^*, R)$ can be factored into $pr(R \mid Y^*, X)pr(Y^* \mid X)pr(X)$, and if the missing at random assumption holds, it can even be factored into $pr(R \mid X)pr(Y^* \mid X)pr(X)$. A test for missing at random can thus be constructed by testing the null hypothesis that $pr(R \mid Y^*, X) = pr(R \mid X)$. Because Y_h^* is not completely observed, conducting this test is not feasible. In our very special TV data set, we can test the null hypothesis that $pr(R \mid Y, X) = pr(R \mid X)$, instead.

The response pattern in the survey data is created by not answering the survey at all and can thus be fully described by a single response vector. Using logit models (Greene, 2008, p. 774), both the MCAR and the MAR assumptions can be tested directly on the data (Little & Rubin, 2002, p. 16). Likelihood ratio tests for omitted variables are conducted as described in Greene (2008, p. 788). For convenience the categorical variables in table D.2 in the appendix are assumed to be continuous.

Table 6.2 shows that both assumptions can be rejected on any common significance level. The response mechanism is clearly not MCAR. Only eight fully observed covariates lead to a considerable $\hat{R}^2_{MAR} = 17.6\%$. Although the test rejects the MAR assumption, too, the model does not improve much by adding the 560 columns of Y as predictors. The AIC (Rinne, 2008, p. 635) reflects this observation. It is larger for the larger model ($\text{AIC}_{MAR} = 13588$, $\text{AIC}_{MNAR} = 13872$), which indicates that the MAR model should be chosen over the MNAR model. If the relation between Y and $R \mid X$ had been large, the likelihood ratio test and the AIC would not give contradictory answers. Thus, we conclude that the MAR assumption is not justified, however, the magnitude of the resulting bias is likely to be small.

null hypothesis	covariates in the logit models	Nagelkerke (1991) $\hat{R}^2$	χ^2	degrees of freedom	p value
MCAR	*H0*: No covariates	0%	1612	8	< 0.0001
	H1: the basic claims data (table D.2)	17,6%			
MAR	*H0*: the basic claims data	17,6%	836	560	< 0.0001
	H1: the basic claims data plus the TV consumption data (table 6.1)	25.8%			

Table 6.2: Likelihood ratio tests for the MCAR and the MAR assumption. Under the null hypothesis for the MCAR test $pr(R) = pr(R \mid X)$ holds, and under the null hypothesis for the MAR test $pr(R \mid X) = pr(R \mid Y, X)$ holds.

There are other technical approaches to test the MAR assumption when using Y_h rather than Y_h^*. Factoring $pr(X, Y, R)$ differently results in testing the null hypothesis that $pr(Y \mid X, R) = pr(Y \mid X)$. Alternatively, similar to David et al. (1986, p. 37), $Y_h \mid r = 0$ can be imputed under the MAR assumption yielding $\tilde{Y}_h \mid r = 0$. Testing the null hypothesis that the moments of $\tilde{Y} \mid R = 0$ equal those of $Y \mid R = 0$ is effectively a test for MAR. Because all Y_hs are binary, the test involves the proportions only. Certainly, the test presented in table 6.2 is the easiest to conduct because r is a vector, whereas Y_h is a matrix with 561 columns, and because it does not involve imputations at all.

6.3 The simulation setup

6.3.1 Missingness

The data set consists of the eight basic claims variables (see table D.2 in the appendix) and 561 columns with passively measured TV consumption data in the binary survey format (see table 6.1). Note that 561 is simply the product of 11 channels, 3 types of days, and 17 time slots. The 1 a.m. to 6 a.m. slot is discarded due to known measurement errors.

A total of 3981 of 11916 respondents in the data set lack the survey data, which equals 33.4%. The basic claims data are always treated as fully observed, and the TV consumption data are always treated as incomplete. Two different response mechanisms are introduced: missing always completely at random (MACAR) and missing always at random (MAAR), where the attribute 'always' indicates that the data generating process rather than one specific sample follows the respective mechanism (Mealli & Rubin, 2015, p. 998). The data set allows a definition of the response mechanisms that is close to a natural one. By defining the model of the response mechanism, MA(C)AR can be assured. The parameters governing the process need not be set but can be estimated from the data as in Heitjan & Little (1991, p. 24). Table D.3 in the appendix provides the respective parameters. The predicted probabilities from this model are used for the simulation study. They range between 0.1523 and 0.9314. The missing values across the TV consumption variables are set independently conditional on the predicted probabilities, which creates a Swiss cheese missing pattern (see section 2.4).

6.3.2 The analysis models

Exclusion and aggregation

For simplicity and for reducing the computation time, the analysis models use the binary variables (see table 6.1) from the channel Pro 7 on Sunday only. Furthermore, a variable called `VOX` is constructed as the sum of all time slots and day types of the channel VOX. Pro 7 and VOX are the largest two channels with an accuracy value in table D.1 in the appendix that is larger than 85%. The data set for imputation thus includes eight basic claims variables (see table D.2 in the appendix), 17 binary incomplete variables and one continuous incomplete variable.

Univariate statistics

Because all the data related to Pro 7 are binary, the mean is sufficient to fully describe the distribution. The focus for Pro 7 is on Sunday night from 8 p.m. to 9 p.m. with a mean value of $\mu_{\texttt{P8}} = 0.43$. The aggregate `VOX` is described by the mean $\mu_{\texttt{VOX}} = 7.56$ and the logarithm of the variance $ln\{var(\texttt{VOX})\} = 4.36$, which is assumed to be normally distributed (Koller-Meinfelder (2009, p. 53), Schafer (1997, p. 145)). Figures D.1 and D.2 in the appendix display the distributions.

Bivariate statistics

In market research, the most popular descriptive statistic is a contingency table. To show coverages in the evaluation section, the cell values are transformed to parameters of a multinomial regression model (Greene, 2008, p. 843), which are normally distributed (Greene, 2008, p. 785). The three contingency tables used for quality assessment are presented in table D.4 in the appendix and reveal that persons in larger households tend to watch more TV, particularly on Sunday night.

Multivariate regression models and nonlinearities

The focus of the simulation study with artificial data in chapter 5 was on linear regression models (Greene, 2008, p. 148). In a very similar setup, the imputation procedures are now challenged with real data. The dependent variable is `internet` (see table D.2 in the appendix). Table D.5 in the appendix presents the details. The positive coefficients indicate that media affinity dominates potential substitution between TV and online usage.

Cluster analysis

Clustering, which is focused on finding homogeneous groups, is one key analysis in market research and commonly used for, e.g., differentiating communication strategies (Punj & Stewart, 1983, p. 135). We conduct a k-means clustering (Rinne, 2008, p. 696) with two clusters on the imputed Sunday time slot data of the Pro 7 channel. The analysis of interest is the cluster means of the fully observed variable `kids18` and the imputed aggregate `VOX` . The learning from the data is that persons living in households with minors tend to watch more TV. The details are presented in table D.6 in the appendix, where the t value relates to the null hypothesis that the respective means of the two clusters are equal (Rinne, 2008, p. 528).

6.3.3 Imputation algorithms

As in chapter 5, we make the distinction between the educated and the ignorant imputer. The educated approaches take the transformations required for the regression models in section 6.3.2 into account. Due to its poor performance in chapter 5, passive imputation (PI) is excluded from further investigation. Just-another-variable PMM is the remaining educated approach. As in chapter 5 the ignorant approaches consist of PMM, random forest imputation and *Miles*. To also conclude chapter 3 the *midastouch* algorithm is implemented as an ignorant and as an educated just-another-variable (JAV) approach. All algorithms run within `R::mice` (R Core Team (2016), van Buuren & Groothuis-Oudshoorn (2011)).

In contrast to the previous simulation studies the missing pattern is now nonmonotone (see section 2.4). Thus, the algorithms must loop over the incomplete variables multiple times (van Buuren, 2012, p. 102). Thereby, it must at least be ensured that the autocorrelation is low enough for the algorithm to become independent from the arbitrary starting values (Schafer, 1997, p. 106). To assess a reasonable number of iterations, five different samples of size $n = 600$ are drawn from the $N = 11916$ population, and each of the two response mechanisms described in table D.3 in the appendix is applied. Then, we run each of the six imputation algorithms in single imputation mode and iterate over the variables 300 times. For each of the 31 parameters of interest, the first nonsignificant lag of the autocorrelation function is computed using a significance level of $\alpha = 10\%$ (Schafer (1997, p. 121), Rinne (2008, p. 400)). Table 6.3 reports the maximum of each analysis and the five data sets.

The educated approaches suffer from extremely high autocorrelation; their maximum first correlated lag is > 100. The likely reason is that the variables `P8` and `VOX` are highly correlated with their considered transformations (van Buuren, 2012, p. 113). At the $N = 11916$ level, linear regression models for the transformations with all other variables as predictors yield coefficients of determination $\hat{R}^2$ of 0.9 for the interaction, 0.99 for the squared term, and 0.98 for the cubed term. As a benchmark, the maximum $\hat{R}^2$ in the data set without the nonlinear terms is 0.65, which is for the 9 p.m. to 10 p.m. dummy for the Pro 7 channel. Because such high correlations are not unusual for typical transformations, convergence is a severe shortcoming of the just-another-variable

Analysis models	ignorant							
	PMM		*midastouch*		RF		*Miles*	
Missing Always	CAR	AR	CAR	AR	CAR	AR	CAR	AR
univariate statistics	12	4	5	3	4	2	5	4
bivariate statistics	29	7	5	4	5	4	5	4
regression coefficients	16	6	6	8	4	4	5	4
cluster means	45	3	4	3	4	3	6	2
number of iterations	88		14		8		10	

Table 6.3: First uncorrelated lag as a measure for autocorrelation

algorithm as already indicated by van Buuren (2012, p. 130). The slow convergence increases the required computation time drastically and thus forces us to exclude the educated algorithms from our simulation study.

Table 6.3 shows that ignorant PMM occasionally suffers from high autocorrelation, too. First investigations have revealed that some parameters of interest vary hardly or even not at all over many successive iterations of PMM imputation. This convergence issue is a new discovery to the best of our knowledge (Koller-Meinfelder, 2009, p. 73). To ensure stable results, the number of iterations is set to twice the maximum value of the last correlated lag. Some more insights on convergence are provided in appendix D.5.

6.3.4 Further settings

The sample size is set to $n_{obs} + n_{mis} = n = 400 + 200 = 600$, which is approximately 5% of the population size. Thus, the convenient sampling with replacement formulas still apply (Cochran, 1977, p. 25). Furthermore, as in section 5.5, the number of multiple imputations is fixed at $M = 10$, and the number of Monte Carlo simulation runs is fixed at $n_{sims} = 1000$.

6.3.5 Evaluation criteria

The estimands are introduced in section 6.3.2. As in section 5.5 the descriptive criterion is relative root mean squared error (Rinne, 2008, p. 17), abbreviated as rRMSE. The inferential criterion is coverages of 950‰ confidence intervals as in section 3.5. We use 1000 bootstrap samples of each imputed data set to assess the within variance of the parameters of interest (Davison & Hinkley, 1997, p. 22).

6.4 The simulation results

Tables D.7, D.8 and D.9 in the appendix present the results; a short summary is displayed in table 6.4. The upper part of table 6.4 shows the root mean squared errors relative to the situation before deletion. A value of 100 means that there is no increase in the root mean squared error due to the incompleteness of the data set; a value of 200 means that the root mean squared error has doubled. The lower part of table 6.4 shows the coverages of 950‰ frequentist confidence intervals. The ideal value is 950, and values below 900 are considered undesirable (van Buuren, 2012, p. 47).

All algorithms cope equally well with the MACAR mechanism and with the MAAR mechanism. This result is somewhat surprising because the MAR response mechanism (see table D.3 in the appendix) depends on nonlinear transformations of `age` and `hhsize`, which also significantly influence, e.g., `P8` and `VOX` (p values < 0.0001 for $N = 11916$) after conditioning on the (linear)

Parameter	ignorant							
	PMM		*midastouch*		RF		*Miles*	
Missing Always	CAR	AR	CAR	AR	CAR	AR	CAR	AR
	(relative root mean squared error) × 100							
univariate statistics	117	118	118	<u>117</u>	156	160	119	119
bivariate statistics	114	115	113	<u>113</u>	140	146	118	119
regression coefficients	99	100	100	100	99	99	98	<u>97</u>
cluster means	92	94	92	<u>93</u>	129	132	101	102
	coverage of 950‰ confidence interval							
univariate statistics	947	943	949	<u>949</u>	835	821	943	935
bivariate statistics	955	<u>954</u>	957	958	882	873	944	945
regression coefficients	970	<u>966</u>	969	<u>966</u>	970	967	972	968
cluster means	980	974	980	976	844	837	954	<u>954</u>

Table 6.4: Summary of the simulation results. Best in MAAR is <u>underlined</u>.

imputation model. Formally, this is a missing not at random mechanism. In applications, it is common that the researcher does not take the time to model each incomplete variable with care. Instead, as in our setup, imputation methods are employed that are somewhat robust to model misspecification, which appears to work satisfactorily in our example.

For some analysis model parameters, the relative root mean squared errors are below the theoretical threshold of 100, indicating that analyzing the imputed data set is more efficient than analyzing the data set before deletion. The *only* theoretical explanation is that the imputation model imposes meaningful restrictions, also known as superefficiency (Rubin, 1996, p. 481). The nonparametric nature of the imputation models makes it impossible to derive the implied restrictions[1]. However, it is most likely that the imputation models restrict some parameters to zero. As shown in appendix D.4, all analysis model parameters are significantly different from zero at the $N = 11916$ level. Thus, how can wrong restrictions increase efficiency? Consider the following argument: Some parameters are tiny in magnitude and barely significant even in the large data set. On a small sample ($n = 600$) imposing a zero restriction on them, which is strictly speaking wrong but not completely incorrect, may have a ridge effect, i.e., it introduces a slight bias but potentially reduces even more variance. For some empirical evidence consider the regression analysis model and specifically the coefficient for `VOX`[3] in table D.7 in the appendix. For the MACAR mechanism the root mean squared error after ignorant PMM imputation is 91% of the root mean squared error before deletion. If the restriction on the cubed term causes this increase in efficiency, then this increase must not be present in an imputed data set that results from an unrestricted imputation model. To see this, the same $n_{sim} = 1000$ incomplete data sets are imputed again with PMM. Yet, this time the imputation model comprises the cubed term in a just-another-variable fashion, i.e., it does not impose any restrictions on the cubed term[2]. The resulting root mean squared error is 121% of the root mean squared error before deletion, i.e., the analysis on the imputed data set is now less efficient than the analysis on the data set before deletion, which suits our prediction.

Because the number of donors is large, an advantage for *midastouch* over PMM cannot be expected. The simulation results based on multivariate normal data from chapter 3 reveal that *midastouch* and PMM do not differ if the number of donors is large. Table 6.4 clearly supports this finding and thus hypothesis 1.

Miles reaches approximately the same performance as PMM and *midastouch*. Hypothesis

[1]The finite number of donors probably limits the ability of, e.g., local regression to fit any functional form. However, it is difficult to state to what degree a global interaction effect can be well reflected.

[2]Because the cubed term is highly correlated 200 Gibbs sampler iterations are required for convergence.

2 is thus falsified. However, *Miles* clearly outperforms random forest imputation. The better performance comes at the expense of longer runtimes: compared to random forest imputation, *Miles* takes twelve times longer. Nevertheless, as already noted by Cleveland et al. (1988, p. 91), the local regression algorithm is embarrassingly parallel because the algorithm can be run independently on each data point.

6.5 Conclusion and future work

The simulation study in this chapter is based on a large TV consumption data set from the GfK SE company. All parameters of the simulation setup are chosen to be as realistic as possible. The parameters of the response mechanism are estimated from the data, the share of missing values is taken from the data, respondents and nonrespondents are included in the analyses, and the analysis models evaluated are the most relevant in the market research industry. Nevertheless, it is only one data set.

This chapter is the first in this dissertation to address a Swiss cheese missing pattern and thus to require Gibbs sampling (see section 2.4). Convergence diagnostics reveal an issue of PMM that is not yet understood. PMM causes analysis model parameters to vary hardly over many iterations. The simulation results appear not heavily affected. However, this is a serious issue for applications. Another issue is found for the JAV algorithm. Because the nonlinear transformations are prone to be highly correlated with the linear terms, JAV is suspected to generally suffer from high autocorrelation (van Buuren, 2012, pp. 113, 130). Consequently, the JAV procedures are excluded from the simulation study of this chapter. A potential solution for this issue is to relax the mutual dependence in the algorithm and treat the linear terms and their transformations as a monotone pattern (van Buuren, 2012, p. 211).

The newly proposed *midastouch* algorithm performs equally well as the established PMM. This result is in line with the findings of chapter 3. Because *midastouch* is superior to PMM for small data sets, reaches the same performance for larger data sets, and does not suffer from convergence issues, we argue to generally choose *midastouch* over PMM.

In this data set, the nonlinearities are not large enough to overtax the simple linear model combined with PMM or *midastouch*. This is why the newly proposed *Miles* does not provide any additional benefit in this application and performs only as good as PMM and *midastouch*. Random forest imputation, however, is clearly inferior to all competitors.

The special nature of the TV data set, which is intensively used in this chapter, allows learning about the typically untestable MAR assumption. The results in section 6.2.4 indicate that assuming MAR is much better than assuming MCAR, but perhaps not quite enough. The null hypothesis MAR can be rejected on any common significance level in favor of the alternative MNAR. A natural extension for future research is to base the simulation study on the observed MNAR mechanism to determine how severely the results are affected.

Appendices

Appendix A

Appendix to Chapter 3

A.1 Overview of existing PMM implementations

	match types		k-nearest-neighbor		parameter	predictions of	
Ref.	available	specify by	default	specify by	uncertainty	donors	recipients
1-4	2	-	n_{obs}	-	ABB	o.o.s.	o.o.s.
5	1	-	5	`donors=#`	parametric	i.s.	o.o.s.
6	1, 2, 3	`pmmtype=#`	3	`kclosest=#`	bootstrap	i.s.	o.o.s.
7	2	-	1	-	BB	i.s.	i.s.
8	2	-	1	-	parametric	i.s.	i.s.
9	1	-	5	-	parametric	i.s.	o.o.s.
10	2	-	n_{obs}	-	ABB	i.s.	o.o.s.
11	2	-	1	-	parametric	i.s.	o.o.s.
12	2	-	1	`knn(#)`	parametric	i.s.	o.o.s.
13	1, 2, 3	`matchtype=#`	3	`matchpool(#)`	parametric	i.s.	o.o.s.

Table A.1: Characteristics of existing PMM software implementations (Morris et al., 2014, p. 3). The references (Ref.) refer to table 3.2. Abbreviations are: approximate Bayesian bootstrap (ABB), Bayesian bootstrap (BB), in sample (i.s.), and out of sample (o.o.s).

A.2 Rationale for leave-one-out modeling

Consider the univariate case, in which both the donors and the recipients are drawn from the same population. The imputation model is simply the mean of Y in the donor sample, denoted by $\hat{\mu}_{obs}$. The mean squared deviation of the donors from the model is

$$\hat{V}_{don} = n_{obs}^{-1} \sum_{i=1}^{n_{obs}} (y_i - \hat{\mu}_{obs})^2.$$

Introducing the true mean by adding $0 = \mu - \mu$ yields (Cochran, 1977, p. 26)

$$\begin{aligned} \hat{V}_{don} &= n_{obs}^{-1} \sum_{i=1}^{n_{obs}} \{(y_i - \mu) - (\hat{\mu}_{obs} - \mu)\}^2 \\ &= n_{obs}^{-1} \left\{ \sum_{i=1}^{n_{obs}} (y_i - \mu)^2 \right\} - (\hat{\mu}_{obs} - \mu)^2. \end{aligned}$$

Analogously, the mean squared deviation of the recipients from the model is

$$\begin{aligned}\hat{V}_{rec} &= n_{mis}^{-1} \sum_{j=1}^{n_{mis}} \{(y_j - \mu) - (\hat{\mu}_{obs} - \mu)\}^2 \\ &= n_{mis}^{-1} \left\{ \sum_{j=1}^{n_{mis}} (y_j - \mu)^2 \right\} + \hat{\mu}_{obs}(\hat{\mu}_{obs} - 2\hat{\mu}_{mis}) - \mu(\mu - 2\hat{\mu}_{mis}),\end{aligned}$$

where $\hat{\mu}_{mis}$ denotes the unobserved mean estimate of the recipient sample. Taking the difference and utilizing the homoscedasticity assumption, we obtain

$$\begin{aligned}\hat{V}_{don} - \hat{V}_{rec} &= n_{obs}^{-1} \left\{ \sum_{i=1}^{n_{obs}} (y_i - \mu)^2 \right\} - n_{mis}^{-1} \left\{ \sum_{j=1}^{n_{mis}} (y_j - \mu)^2 \right\} + 2(\hat{\mu}_{obs} - \hat{\mu}_{mis})(\mu - \hat{\mu}_{obs}) \\ E(\hat{V}_{don} - \hat{V}_{rec}) &= 2E\{(\hat{\mu}_{obs} - \hat{\mu}_{mis})(\mu - \hat{\mu}_{obs})\}.\end{aligned}$$

For a large recipient sample, $\hat{\mu}_{mis} = E(\hat{\mu}_{mis}) = \mu$ holds, and thus, the following also holds:

$$E(\hat{V}_{don} - \hat{V}_{rec}) = -2E(\mu - \hat{\mu}_{obs})^2 \leqslant 0.$$

In other words, as long as the model, which is based on the donor sample (Rubin, 1987, p. 167), differs randomly from the true population model, the expected value of the residual variance for the donors is smaller than that for the recipients. This difference decreases as $n_{obs} \to \infty$.

A.3 Another look at choosing k for k-nearest-neighbors

We add to the discussion of choosing an optimal k by focusing on the point estimate of the variance of Y. If the domains of the donors and recipients are similar, a large k will increase the probability that recipients closer to the bounds will obtain their values from donors closer to the center. The variance of Y inevitably decreases, and thus, the estimate of the variance of Y on the imputed data is biased downwards for larger k.

To see this, suppose that the predictive mean ϖ obeys the bounds $[-0.5, 0.5]$ and that the distribution of the donors is discrete and equidistant within this range such that $\varpi_{obs} = \{-0.5 + (\Omega - 1)/(n_{obs} - 1)\}$, with $\Omega = (1, \ldots, n_{obs})$. Further suppose that the recipients are distributed in the exact same way such that $\varpi_{obs} = \varpi_{mis}$. We define $n = n_{obs} = n_{mis}$ and, for simplicity, allow it to be uneven only. We also assume that the predictive mean ϖ is the characteristic of interest. This may be the case in a multivariate setting in which the fully observed variables perfectly determine the variable with missing values.

ϖ_{mis} is imputed using ϖ_{obs}, leading to ϖ_{imp}. We wish to learn about the point estimate for the variance of ϖ_{imp} as a function of the relative size of the neighborhood from which random selection is performed for a single recipient. We define this relative size, excluding the exact nearest neighbor, as $\varsigma = (\Omega - 1)/(n - 1)$. We decompose the variance of ϖ_{imp} into a between-variance component and a within-variance component, $\Theta(\varsigma) = \Xi(\varsigma) + \Upsilon(\varsigma)$, where Ξ denotes the interrecipient variance and Υ denotes the intrarecipient variance. It follows that if the exact nearest neighbor is chosen, the interrecipient variance of ϖ_{imp} will equal the variance of ϖ_{mis},

$$\Theta(\varsigma = 0) = \Xi(\varsigma = 0) = var(\varpi_{mis}). \tag{A.1}$$

For larger ς, the intrarecipient variance increases according to the variance formula for the discrete

uniform distribution as follows (Rinne, 2008, p. 372):

$$\Upsilon(\varsigma) = -\Delta\Upsilon(\varsigma) = \varsigma^2/12 + \varsigma/\{6(n-1)\}.$$

The interrecipient variance is equivalent to the variance of the expectations. The expectation of a uniform distribution is the mean of its bounds. Because the range of ϖ is limited on both sides, the interrecipient variance decreases with increasing ς. More specifically, we see that for the left side, i.e., $\varpi^i_{mis} < 0$,

$$E\left\{\varpi^i_{imp} \mid \varsigma, \varpi^i_{mis} < (\varsigma-1)/2\right\} = (\varsigma-1)/2. \tag{A.2}$$

We assume that the mean of ϖ is known to be zero. We can then write, based on the left-hand side,

$$var(\varpi_{mis}) - \Xi(\varsigma) = \Delta\Xi(\varsigma) = 2n^{-1}\sum_{i=1}^{(n-1)/2}\left[(\varpi^i_{mis})^2 - \{E(\varpi^i_{imp})\}^2\right]. \tag{A.3}$$

We now focus solely on the part of the left-hand side for which $\varpi^i_{mis} < (\varsigma-1)/2$ holds. We may ignore the rest because all corresponding elements of the sum in equation (A.3) are zero. Then, using the assumption of equidistance and equation (A.2), we obtain

$$\Delta\Xi(\varsigma) = 2n^{-1}\sum_{i=1}^{(n-1)\varsigma/2}\left\{\left(\frac{\varsigma-1}{2}\right)^2 - \frac{(\varsigma-1)i}{n-1} + \frac{i^2}{(n-1)^2} - \left(\frac{\varsigma-1}{2}\right)^2\right\}. \tag{A.4}$$

The last term in equation (A.4) is equal to the last term in (A.3) and cancels out. Some rewriting reveals a series that allows further simplification (Bronstein et al., 2013, p. 20):

$$\Delta\Xi(\varsigma) = 2n^{-1}\left\{(1-\varsigma)/(n-1)\sum_{i=1}^{(n-1)\varsigma/2}(i) + (n-1)^{-2}\sum_{i=1}^{(n-1)\varsigma/2}\left(i^2\right)\right\}.$$

Further algebra leads to the third-order polynomial

$$\Delta\Xi(\varsigma) = \varsigma\{\varsigma(n-1)+2\}\{2\varsigma(n-1)-3n+2\}/\{-12n(n-1)\}.$$

Adding $\Delta\Upsilon(\varsigma)$ results in

$$\Delta\Theta(\varsigma) = \varsigma(\varsigma-1)\{\varsigma(n-1)+2\}/(6n),$$

which has two obvious roots: one at $\varsigma = 0$, as already seen from equation (A.1), and one at $\varsigma = 1$, where $\Xi = 0$. The third root does not exist given the limits on n and ς. The first derivative is

$$\partial\Delta\Theta(\varsigma)/\partial\varsigma = \{3\varsigma^2(n-1) - 2\varsigma(n-3) - 2\}/(6n).$$

For $n \to \infty$, $\Delta\Theta(\varsigma)$ has a minimum at $P_{min}(\varsigma = 2/3, \Delta\Theta = -2/81)$ and a falling inflection point at $P_{infl}(\varsigma = 1/3, \Delta\Theta = -1/81)$. We conclude that the point estimate for the variance of ϖ_{imp} is biased downwards for all ς except $\varsigma = 0$ and $\varsigma = 1$.

A.4 R-Code for *midastouch*

```
mice.impute.midastouch <- function(y, ry, x, ridge = 1e-05, midas.kappa = NULL, outout = NULL,
neff = NULL, debug = NULL, ...) {

  #+ auxiliaries +#
  if(!is.null(debug)){midastouch.inputlist <- list(y = y, ry = ry, x = x, omega = NULL)}
  sminx <- .Machine$double.eps^(1/4)
```

```
#+ ensure data format +#
x <- data.matrix(x)
storage.mode(x) <- "numeric"
X <- cbind(1, x)
y <- as.numeric(y)

#+ get data dimensions +#
nobs <- sum(ry) ; nmis <- sum(!ry) ; n <- length(ry)
obsind <- which(ry) ; misind <- which(!ry)
m <- ncol(X)
yobs <- y[obsind]
Xobs <- X[obsind,,drop=FALSE]
Xmis <- X[misind,,drop=FALSE]

#+ P-Step +#
##++++ bootstrap
omega <- bootfunc.plain(nobs)
if(!is.null(debug)){
  midastouch.inputlist$omega <- omega
  assign(x = "midastouch.inputlist",value = midastouch.inputlist,envir = get(debug))
}

##++++ beta estimation
CX <- omega * Xobs
XCX <- crossprod(Xobs,CX)
if(ridge > 0){    diag(XCX) <- diag(XCX) * (1+c(0,rep(ridge,m-1)))  }

#= check if any diagonal element is exactly zero  ===========#
diag0 <- which(diag(XCX) == 0)                              #==#
if(length(diag0)>0){diag(XCX)[diag0] <- max(sminx,ridge)} #==#
#============================================================#

Xy <- crossprod(CX,yobs)
beta <- solve(XCX,Xy)
yhat.obs <- c(Xobs %*% beta)

##++++ kappa estimation
if(is.null(midas.kappa)){
  mean.y <- crossprod(yobs,omega)/nobs
  eps <- yobs - yhat.obs
  r2 <- 1 - c(crossprod(omega, eps^2) / crossprod(omega,(yobs - mean.y)^2))
  ##slight deviation from the paper to ensure real results
  ##   paper: a tiny delta is added to the denominator
  ##   R Code: min function is used, note that this correction gets active for r2>.999 only
  midas.kappa <- min((50*r2 / (1-r2))^(3/8),100)
  ##if r2 cannot be determined (eg zero variance in yhat), use 3 as suggested by Siddique/Belin
  if(is.na(midas.kappa)){midas.kappa <- 3}
}

#+ I-Step +#
if(is.null(outout)){ outout <- ifelse(nobs>250,FALSE,TRUE)  }
if(outout){
  ##++++ P-step if out of sample predictions for donors
  ## estimate one model per donor by leave-one-out
  XXarray_pre <- t(t(apply(X = Xobs,MARGIN = 1,FUN = tcrossprod)) * omega)
  ridgeind <- c(1:(m-1))*(m+1)+1
  if(ridge > 0){
    XXarray_pre[ridgeind,] <- XXarray_pre[ridgeind,] * (1+ridge)
  }
  XXarray <- c(XCX) - XXarray_pre

  #= check if any diagonal element is exactly zero ======================#
  diag0 <- which(XXarray[ridgeind,] == 0)                              #==#
  if(length(diag0) > 0){XXarray[ridgeind,][diag0] <- max(sminx,ridge)} #==#
  #======================================================================#

  Xyarray <- c(Xy) - t(Xobs * yobs * omega)
  BETAarray <- apply(rbind(XXarray,Xyarray),2,function(x,m){
   solve(a = matrix(head(x,m^2),m),b = tail(x,m))},m=m)
  YHATdon <- rowSums(Xobs * t(BETAarray))
  ## each recipient has nobs different yhats
  YHATrec <- Xmis %*% BETAarray
  ##++++ distance calculations
  dist.mat <- YHATdon - t(YHATrec)
}else{
  yhat.mis <- c(Xmis %*% beta)
  dist.mat <- yhat.obs - matrix(data = yhat.mis,nrow = nobs,ncol = nmis,byrow = TRUE)
}

##++++ convert distances to drawing probs // ensure real results
delta.mat <- 1/((abs(dist.mat))^midas.kappa)
delta.mat <- minmax(delta.mat)
probs <- delta.mat * omega
csums <- minmax(colSums(probs,na.rm = TRUE))
probs <- t(t(probs)/csums)
```

```
  #+ calculate neff +#
  if(!is.null(neff)){
    if(!exists("midastouch.neff",envir = get(neff))){
     assign(x = "midastouch.neff",value = list(),envir = get(neff))}
    midastouch.neff <- get("midastouch.neff",envir = get(neff))
    midastouch.neff[[length(midastouch.neff)+1]] <- mean(1/rowSums((t(delta.mat)/csums)^2))
    assign(x = "midastouch.neff",value = midastouch.neff,envir = get(neff))
  }

  #+ return result +#
  index <- apply(probs,2,sample,x = nobs,size = 1, replace = FALSE)
  yimp <- y[obsind][index]
  return(yimp)
}

bootfunc.plain <- function(n){
  random <- sample(n,replace = TRUE)
  weights <- as.numeric(table(factor(random,levels = c(1:n))))
  return(weights)
}

minmax <- function(x,domin=TRUE,domax=TRUE){
  maxx <- sqrt(.Machine$double.xmax)
  minx <- sqrt(.Machine$double.eps)
  if(domin){      x <- pmin(x,maxx)   }
  if(domax){      x <- pmax(x,minx)   }
  return(x)
}
```

A.5 Detailed simulation results

The concept of multiple imputation relies on the propagation of the uncertainty associated with the estimation of the parameters of the imputation model. Thus, to check whether multiple imputation PMM algorithms perform multiple imputation properly, those parameters should be uncertain. Because the degree of uncertainty primarily depends on the donor sample size n_{obs}, we present the detailed simulation results, split by n_{obs}, in tables A.2 and A.3. Each cell in these tables contains a 950‰ frequentist confidence interval coverage averaged over $2^{(4-2)}n_{sim} = 1000$ simulation runs.

It it worth noting that the proposed *midastouch* algorithm does not fall below 950‰ in any of the splits.

Match types and k-nearest-neighbors

With one predictor only, i.e., for $p - 1 = 1$, some algorithms perform as poorly as the deliberately poor benchmark that does not propagate parameter uncertainty at all. All of these algorithms, presented in rows 8, 11, and 12 in table A.2 and table A.3, rely on both type-2 matching and $k = 1$-nearest-neighbor imputation. The observed attenuation bias for these algorithms buttresses the criticism offered by van Buuren (2012). Although the MIDAS algorithm also involves type-2 matching, it outperforms the poor benchmark.

In appendix A.3 we argue that the point estimate for the variance of Y is biased downwards for large values of k. For the simulation runs with $n_{obs} = 10$, the mean point estimates for the variance of Y are 0.846 and 0.729 for all PMM implementations in the software listed by Morris et al. (2014, p. 3) for $k = 1$ and $k > 1$, respectively. This difference is highly significant. Both estimates are well below the true variance of 1 because the relatively small number of donors causes the domain of X_h to be smaller for the donors than for the recipients. This is a case of truncation. For the runs with $n_{obs} = 200$, the differences diminish because k is small relative to the number of donors; the mean point estimates are 0.996 and 0.992. For the proposed algorithm, the mean point estimates for the variance of Y are 0.821 and 1 for $n_{obs} = 10$ and $n_{obs} = 200$, respectively.

	950‰ confidence interval coverages													
	Response mechanism				Number of covariates				Coefficient of determination				Overall	
	MACAR		MAAR		$p-1=1$		$p-1=8$		$R^2=0$		$R^2=0.75$			
Ref.	$\hat{\mu}$	$\hat{\beta}_1$	$\hat{\mu}$	$\hat{\beta}_1$	$\hat{\mu}$	$\hat{\beta}_1$	$\hat{\mu}$	$\hat{\beta}_1$	$\hat{\mu}$	$\hat{\beta}_1$	$\hat{\mu}$	$\hat{\beta}_1$	$\hat{\mu}$	$\hat{\beta}_1$
Proposed algorithm (algorithm 4)														
1	967	960	905	962	948	958	924	964	976	962	896	960	936	961
2	991	–	955	–	972	–	974	–	988	–	958	–	973	–
3	968	959	894	963	951	961	911	961	970	950	892	972	931	961
4	985	–	934	–	963	–	956	–	990	–	929	–	960	–
Predictive mean matching software listed by Morris et al. (2014, p. 3)														
5	732	910	477	887	598	836	611	961	700	950	509	847	605	899
6	587	900	442	844	464	776	565	968	564	934	465	810	515	872
7	771	794	600	768	658	650	714	912	764	831	607	731	686	781
8	704	685	442	642	396	351	750	976	583	639	564	688	573	664
9	616	840	358	841	436	724	539	957	557	855	418	827	487	841
10	960	971	838	963	873	953	925	981	954	957	844	977	899	967
11	718	675	561	643	396	352	883	966	604	631	675	687	640	659
12	704	667	528	637	396	351	836	953	583	624	650	680	616	652
13	579	731	309	723	446	500	440	954	575	978	312	476	443	727
Benchmark algorithms														
14	964	956	960	961	970	946	954	971	962	948	962	969	962	959
15	479	469	285	466	396	351	367	585	313	438	451	498	382	468

Table A.2: Coverages for $n_{obs} = 10$ split by the three remaining binary factors. The references (Ref.) refer to table 3.2. Abbreviations are: missing always (completely) at random (MA(C)AR).

	950‰ confidence interval coverages													
	Response mechanism				Number of covariates				Coefficient of determination				Overall	
	MACAR		MAAR		$p-1=1$		$p-1=8$		$R^2=0$		$R^2=0.75$			
Ref.	$\hat{\mu}$	$\hat{\beta}_1$	$\hat{\mu}$	$\hat{\beta}_1$	$\hat{\mu}$	$\hat{\beta}_1$	$\hat{\mu}$	$\hat{\beta}_1$	$\hat{\mu}$	$\hat{\beta}_1$	$\hat{\mu}$	$\hat{\beta}_1$	$\hat{\mu}$	$\hat{\beta}_1$
Proposed algorithm (algorithm 4)														
1	948	960	942	950	942	954	948	956	953	959	936	951	945	955
2	974	–	969	–	966	–	977	–	960	–	983	–	972	–
3	952	949	940	942	948	934	944	956	957	933	935	957	946	945
4	982	–	973	–	975	–	980	–	987	–	968	–	978	–
Predictive mean matching software listed by Morris et al. (2014, p. 3)														
5	947	965	934	952	936	961	945	956	942	959	940	958	941	959
6	939	967	932	950	921	961	950	956	930	960	942	957	936	959
7	940	966	933	949	933	959	940	956	939	962	934	953	937	958
8	908	929	907	897	874	866	942	960	895	902	920	924	908	913
9	931	953	925	932	918	933	938	952	927	935	929	950	928	943
10	939	959	934	949	930	952	943	956	940	946	933	962	937	954
11	907	927	906	895	874	866	939	956	892	901	921	921	907	911
12	908	927	906	895	874	866	940	956	893	902	921	920	907	911
13	943	966	926	950	932	959	937	957	936	960	933	956	935	958
Benchmark algorithms														
14	950	966	942	949	945	959	947	956	951	957	942	958	946	958
15	886	927	868	896	875	866	879	957	843	903	911	920	877	912

Table A.3: Coverages for $n_{obs} = 200$ split by the three remaining binary factors. The references (Ref.) refer to table 3.2. Abbreviations are: missing always (completely) at random (MA(C)AR).

Appendix B

Appendix to Chapter 4

B.1 Proof related to section 4.4

Solving equation (4.6) leads Rao & Singh (1997, p. 59) to the following result

$$l = C^{-1}X_i \left(X_i'C^{-1}X_i + \Lambda\right)^{-1} x_{\underline{0}} + \left\{d - C^{-1}X_i \left(X_i'C^{-1}X_i + \Lambda\right)^{-1} X_i'd\right\}. \tag{B.1}$$

In contrast to equation (4.5) equation (B.1) comprises the term in the curly braces, which is zero if

$$d = C^{-1}X_i \left(X_i'C^{-1}X_i + \Lambda\right)^{-1} X_i'd \tag{B.2}$$

is true. Assuming X_i has full column rank, we can multiply both sides from the left as follows

$$\left\{\left(X_i'C^{-1}X_i\right)^{-1} X_i'\right\} d = \left(X_i'C^{-1}X_i + \Lambda\right)^{-1} X_i'd.$$

Switching the sides of the equation and multiplying again from the left gives

$$X_i'd = X_i'd + \Lambda \left(X_i'C^{-1}X_i\right)^{-1} X_i'd.$$

With C defined as in equation (4.3b) we can write

$$0 = \Lambda \left\{\left(X_i'C^{-1}X_i\right)^{-1} X_i'C^{-1}(1, 1, \ldots, 1)'\right\}. \tag{B.3}$$

Note that the term in the curly braces in equation (B.3) is equivalent to the weighted least squares estimator with a constant response. The corresponding minimization problem is given by

$$\min_{\beta_{WLS}^{const}} \left[\{(1, 1, \ldots, 1)' - X_i\beta_{WLS}^{const}\}'\{(1, 1, \ldots, 1)' - X_i\beta_{WLS}^{const}\}\right]. \tag{B.4}$$

Assuming X_i has a leading constant column, the solution for equation (B.4) is

$$\beta_{WLS}^{const} = (1, 0, \ldots, 0)'. \tag{B.5}$$

To see that, we can just plug it in the regression equation

$$(1, 1, \ldots, 1)' = (1, 1, \ldots, 1)' + X_{-1}(0, 0, \ldots, 0)',$$

where X_{-1} denotes the X_i matrix without the constant. In this solution the residual sum of squares, weighted or unweighted, is zero. Because weighted regression minimizes the sum of squared residuals, (B.5) must be a solution; it does not get smaller than zero. The assumption of full column rank for X_i means that there is no linear combination of all X_{-1} that equals the constant. In other words, there is no other solution to the problem in equation (B.4) than equation (B.5). For equation (B.3) to be true, we require the diagonal matrix Λ to be zero at its first element, as described in section 4.3.2.

We have shown that the weighted ridge regression is equivalent to the minimization problem in Rao & Singh (1997). This proof comes down to showing that equation (B.2) holds. It does under the following assumptions, X_i must have full column rank and a leading constant column, C must be defined as in equation (4.3b) and the diagonal ridge penalty matrix must not penalize the intercept parameter as in equation (4.7).

Appendix C

Appendix to Chapter 5

C.1 R Code for *Miles*

```
mice.impute.li <- function(y, ry, x, boot = TRUE, kgran = 6, rgran = 3, rescale = 1, midastouch=TRUE,...) {

  #data preparation --------------------
  #++ ensure data format ----------
  x <- data.matrix(x)
  storage.mode(x) <- "numeric"
  y <- as.numeric(y)
  #++ get data dimensions ----------
  n <- length(y)
  donind <- which(ry) ; ndon <- length(donind)
  recind <- which(!ry); nrec <- length(recind)
  if(ndon==1){return(rep(y[donind],(n-1)))}
  #++ remove cols with zero variance ----------
  remcons.ind <- removecons(X = x)
  X <- x[,remcons.ind,drop=FALSE]

  #Bayesian bootstrap weights donors and recipients --------------------
  if(boot){bw <- bootfunc(n)}else{bw <- NULL}

  #rescale x --------------------
  if(rescale>0){
    if(rescale == 2 && ncol(X)>1){ #double robust
      Xr <- cbind(1,X)
      if(!is.null(bw)){
        Xrbw <- Xr * bw
      }else{
        Xrbw <- Xr
      }
      beta <- abs(c(solve(crossprod(Xrbw,Xr),crossprod(Xrbw,ry),tol = 0)))[-1]
      mbeta <- max(beta) ; mbeta <- ifelse(test = mbeta == 0,yes = 1,no = mbeta)
      resc.factors <- beta / mbeta
      X <- t(t(X)*resc.factors)
    }else{ #iqr rescaling
      iqrvec <- rescale.quartiles(X = X,weight = bw)
      X <- t(t(X)/iqrvec)
    }
  }

  #optimization over k and ridge --------------------
  #++ select possible solutions ----------
  kvec <- kopt(ndon = ndon,kgran = kgran)
  rmat <- ropt(kvec = kvec,rgran = rgran,ncolX = ncol(X))
  nk <- length(kvec) ; nr <- ncol(rmat)
  #++ if only one solution is feasible no optimization necessary ----------
  if((nk*nr) == 1){
    noopt <- TRUE
    k <- kvec
    r <- rmat
  }else{
    noopt <- FALSE
    maxk <- kvec[length(kvec)]
    #+ draw a donorsample to measure the fit [faster than using all donors] +#
    sampledonind_ <- sort(sample(x = c(1:ndon),size = sampledon(ndon = ndon),replace = FALSE))
    sampledonind <- donind[sampledonind_]
    nsdon <- length(sampledonind)
    #++ build modeltable for k = max(kvec) ----------
    mto <- modeltable(Xdon_full = X[donind,],Xdon_sample = cbind(sampledonind_,X[sampledonind,]),k = maxk)
    mto <- mt.idconvert(mt = mto,donind = donind,recind = recind,k = maxk,ndon = nsdon,nrec = 0)
```

```
    #++ loop over possible k's ----------
    yhato <- array(dim = c(nsdon,nk,nr))
    for(kind in c(1:nk)){
      kloop <- kvec[kind]
      #++ reduce modeltable and adjust weights ----------
      if(kind<nk){
        mti <- rep(c(rep(TRUE,kloop),rep(FALSE,(maxk-kloop))),nsdon)
        mtk <- mto[mti,]
        mtk[,4] <- adjust.weights(id = mtk[,1],dist = mtk[,3],k = kloop)
      }else{          mtk <- mto      }
      #++ create a lookup table for each ID ----------
      ipmk <- ipm.create(id = mtk[,1],k = kloop)
      #++ consider bootstrap weights ----------
      if(boot){mtk[,4] <- mtk[,4] * bw[mtk[,2]]}
      #++ loop over possible r's ----------
      for(rind in c(1:nr)){
        #+ calculate yhat +#
        yhato[,kind,rind] <- (yhat.calc(X = X,y = y,mt = mtk,ipm = ipmk,r = rmat[kind,rind]))[sampledonind]
      }
    }
    #++ choose the k and r that fit best ----------
    fits <- apply(yhato - y[sampledonind],c(2,3),crossprod)
    optinds <- which(fits == min(fits),arr.ind = TRUE)
    optindex <- which.max(optinds[,1]) #if unsure, take the maximum k
    k_optind <- optinds[optindex,1]
    r_optind <- optinds[optindex,2]
    k <- kvec[k_optind]
    r <- rmat[k_optind,r_optind]
  }
  cat(paste0("\nOptimal k is ",k,", optimal r is ",r,".\n"))
  if(k>=ndon){k <- ndon - 1}

  #model table (donor selection + design weights) --------------------
  #++ NN if k == 1 ----------
  if(k == 1){recmodel <- FALSE}
  #++ generate ----------
  mt <- modeltable(Xdon_full = X[donind,,drop=FALSE],Xrec = X[recind,,drop=FALSE],k = k)
  #++ map back to real overall indices and sort ----------
  mt <- mt.idconvert(mt = mt,donind = donind,recind = recind,k = k,ndon = ndon,nrec = nrec)
  #++ create a lookup table for each ID ----------
  ipm <- ipm.create(id = mt[,1],k = k)
  #++ utilize predictions from the optimization for speedup ----------
  if(!noopt){ipm.small <- ipm[-sampledonind,]}else{ipm.small <- ipm}
  #++ consider bootstrap weights ----------
  if(boot){mt[,4] <- mt[,4] * bw[mt[,2]]}

  #model and midastouch --------------------
  #++ NN if k == 1 ----------
  if(k == 1){
    matchind <- mt[,2][recind]
    cat("\nk == 1, NN applied instead of li - be aware that MI does not work with NN\n")
  }else{
    #calculate yhat ----------
    yhat <- yhat.calc(X = X,y = y,mt = mt,ipm = ipm.small,r = r)
    #utilize predictions from the optimization ----------
    if(!noopt){yhat[sampledonind] <- yhato[,k_optind,r_optind]}
    #match to NN in yhat (PMM step) ----------
    if(!midastouch){
     matchind <- NNindex(yhat = yhat,mt = mt,ipm = ipm,k = k,donind = donind,recind = recind)
    } else {
     matchind <- midasindex(y = y,yhat = yhat,mt = mt,ipm = ipm,
     k = k,donind = donind,recind = recind,bw = bw) }
  }
  yimp <- y[matchind]

  #return --------------------
  return(yimp)
}

bootfunc <- function(n){
  random <- runif(n = n - 1)
  sorted <- c(0,sort(random),1)
  weights <- diff(sorted) * n
  return(weights)
}

modeltable <- function(Xdon_full,Xrec=NULL,Xdon_sample=NULL,k){
  if(!is.null(Xdon_sample)){

    #Donors from Donors k optimization --------------------
    donids <- Xdon_sample[,1]
    #++ initialize ----------
    ndon <- nrow(Xdon_sample)
    M <- matrix(nrow = ndon * (k+1),ncol = 5)
    colnames(M) <- c("ID","modelid","dist","maxdist","w")
    M[,1] <- rep(donids,each=k+1)
    #++ RANN distance calculation for the donors ----------
```

```
    RANNobj <- nn2(data = Xdon_full,query = Xdon_sample[,-1],k = k + 1)
    #++ write to M ----------
    M[,2] <- c(t(RANNobj$nn.idx))
    M[,3] <- c(t(RANNobj$nn.dists))
    #++ kick out self matches --> finally: nrow(MD) == ndon_r x k ----------
    selfpos <- M[,1] == M[,2]
    M <- M[!selfpos,]
    #++ add maxdist ----------
    M[,4] <- rep(x = M[c(1:ndon)*k,3],each = k)

  }else{
    #Donors from Donors --------------------
    #++ initialize ----------
    ndon <- nrow(Xdon_full)
    MDD <- matrix(nrow = ndon * (k+1),ncol = 5)
    c1ndon <- c(1:ndon)
    colnames(MDD) <- c("ID","modelid","dist","maxdist","w")
    MDD[,1] <- dvec <- rep(c1ndon,each=k+1)
    #++ RANN distance calculation for the donors ----------
    RANNobj <- nn2(data = Xdon_full,k = k + 1)
    #++ write to MDD ----------
    MDD[,2] <- c(t(RANNobj$nn.idx))
    MDD[,3] <- c(t(RANNobj$nn.dists))
    #++ kick out self matches ----------
    selfpos <- MDD[,1] == MDD[,2]
    if(sum(selfpos)<ndon){
      selfpos[((c1ndon-1) * (k+1) + 1)[setdiff(x = c1ndon,dvec[selfpos])]] <- TRUE
    }
    MDD <- MDD[!selfpos,,drop=FALSE]
    #++ add maxdist ----------
    MDD[,4] <- rep(x = MDD[c(1:ndon)*k,3],each = k)

    #Recipients from Donors --------------------
    #++ initialize ----------
    nrec <- nrow(Xrec)
    MRD <- matrix(nrow = nrec * k,ncol = 5)
    colnames(MRD) <- c("ID","modelid","dist","maxdist","w")
    MRD[,1] <- rep(x = c(1:nrec),each = k)
    #++ RANN distance calculation recipients from donors ----------
    RANNobj <- nn2(data = Xdon_full,query = Xrec,k = k)
    #++ write to MRD ----------
    MRD[,2] <- c(t(RANNobj$nn.idx))
    MRD[,3] <- c(t(RANNobj$nn.dists))
    #++ add maxdist ----------
    MRD[,4] <- rep(x = MRD[c(1:nrec)*k,3],each = k)

    #Combine --------------------
    M <- rbind(MDD,MRD)
  }

  #Calculate design weights by tricube distance --------------------
  M[(M[,4] == 0),4] <- 1
  M[,5] <- (1-(M[,3]/M[,4])^3)^3 + .01

  #reduce to the necessary and return --------------------
  M <- M[,c(1,2,3,5)]
  return(M)
}

yhat.calc <- function(X,y,mt,ipm,r){
  yhat <- vector(mode = "numeric",length = nrow(X))
  mid <- mt[,2]
  w0 <- mt[,4]
  for(i in c(1:nrow(ipm))){
    indexmt  <- c(ipm[i,2]:ipm[i,3])
    indexi   <- ipm[i,1]
    midindex <- mid[indexmt]
    yi <- y[midindex]
    if(.Call("bycol_all_equal_double2",matrix(data = yi,ncol = 1))){
      yhat[indexi] <- yi[1]
    }else{
      di <- w0[indexmt]
      Xi = X[midindex,,drop=FALSE]
      remcons <- removecons(X = Xi)
      nc <- length(remcons)
      if(nc == 0){
        yhat[indexi] <- weighted.mean(x = yi,w = di)
      }else{
        Xi = cbind(1,Xi[,remcons,drop=FALSE])
        tvec <- c(1,X[indexi,remcons])
        l <- c(1,rep((1+r),nc))
        a = .Call("Xt_D_Xv2",Xi,di,l)
        b = crossprod(x = Xi,y = di*yi)
        beta <- solve(a = a,b = b,tol = 0)
        yhat[indexi] <- crossprod(tvec,beta)
      }
    }
```

```
  }
  return(yhat)
}

midasindex <- function(y,yhat,mt,ipm,k,donind,recind,bw){
  c1nrec <- c(1:length(recind))
  yhat.k <- cbind(1,yhat[donind])
  y.k <- y[donind]
  beta.k <- try(solve(crossprod(yhat.k),crossprod(yhat.k,y.k),tol=0),silent = TRUE)
  if(class(beta.k) != "try-error"){
    r2.k <- vx(yhat.k %*% beta.k) / vx(y.k)
    kappa <- (50 * r2.k / (1.001 - r2.k))^(3/8)
  }else{
    kappa <- 0
  }
  index.r <- c(apply(ipm[recind,],1,function(x){c(x[2]:x[3])}))
  index.d <- mt[index.r,2]
  d <- abs(rep(yhat[recind],each=k) - yhat[index.d])
  d <- d / mean(d)
  dk <- 1/d^k
  if(!is.null(bw)){
    wdk <- minmax(dk * bw[index.d])
    wdk[is.nan(wdk)] <- 0
  }else{                wdk <- minmax(dk)             }
  draw <- do.call(c,lapply(split(wdk,rep(c1nrec,each=k)),sample,x=k,size=1,replace=FALSE))
  ydonind <- index.d[k*(c1nrec-1) + draw]
  return(ydonind)
}

rescale.quartiles <- function(X,weight) {
  iqrvec <- apply(X = X,2,wiqr,w=weight)
  iqrvec[iqrvec == 0] <- 1
  return(iqrvec)
}

wiqr <- function(x,w=NULL){
  ox <- order(x)
  if(!is.null(w)){
    ssw <- cumsum(w[ox])
    limits <- c(.25,.75)*ssw[length(ssw)]
    index <- pmax(findInterval(limits,ssw),1)
    index <- index + (ssw[index+1] -limits < limits - ssw[index])
  } else  {
    index <- round(c(.25,.75) * length(x))
  }
  return(diff(x[ox][index]))
}

vx <- function(x){xstd <- x-mean(x);return(c(crossprod(xstd)/(length(x)-1)))}

removecons = function(X){
  if(ncol(X) == 1){
    index <- 1
  }else{
    colranges <- !(.Call("bycol_all_equal_double2",X))
    ivl0 <- which(colranges)
    if(length(ivl0) == 0){ index <- 1
    }else{ index <- ivl0 }
  }
  return(index)
}

mt.idconvert <- function(mt,donind,recind,k,ndon,nrec){
  don_r <- c(rep(TRUE,ndon*k),rep(FALSE,nrec*k))
  id <- mt[,1]
  id[don_r] <- donind[id[don_r]]
  id[!don_r] <- recind[id[!don_r]]
  modelid <- mt[,2]
  modelid <- donind[modelid]
  mt[,1] <- id
  mt[,2] <- modelid
  mt <- mt[order(mt[,1]),]
  return(mt)
}

ipm.create <- function(id,k){
  length.unique.id <- length(id)/k
  ind.pos.mat <- matrix(nrow = length.unique.id,ncol = 3) ; colnames(ind.pos.mat) <- c("id","start","stop")
  ind.pos.mat[,3] <- aux <- c(1:length.unique.id) * k
  ind.pos.mat[,2] <- c(1,aux[-length.unique.id]+1)
  ind.pos.mat[,1] <- id[ind.pos.mat[,2]]
  return(ind.pos.mat)
}

adjust.weights <- function(id,dist,k){
  indk <- c(1:(length(id)/k)) * k
  maxdist <- rep(dist[indk],each=k)
```

```
  maxdist[maxdist == 0] <- 1
  w <- (1-(dist/maxdist)^3)^3 + .01
  return(w)
}

kopt <- function(ndon,kgran){
  if(ndon <= 6){
    k <- ndon-1
  }else{
    if((ndon-6)<kgran){   k <- c(5:(ndon-1))
    }else{ k <- c(round(5 + (0:(kgran-2))*(ndon-6)/(kgran-1)),ndon-1) }
  }
  return(k)
}

ropt <- function(kvec,rgran,ncolX){
  rfac <- matrix(2^c(0:(rgran-1)),nrow = length(kvec),ncol = rgran,byrow=TRUE)
  rulez <- matrix(c(10,30,max(2*ncolX,100),Inf,.2,.1,.05,.025),ncol=2)
  starts <- rulez[,2][(findInterval(x = kvec,vec = rulez[,1]) + 1)]
  r <- starts * rfac
  return(r)
}

sampledon <- function(ndon){
  rulez <- matrix(c(50,500,2000,Inf,ndon,50,.1*ndon,200),ncol=2)
  sdon <- rulez[,2][findInterval(ndon,rulez[,1]) + 1]
  return(sdon)
}
```

C.2 C Code for *Miles*

Markus Lilienthal wrote this `C` code based on my `R` Code for speeding up the *Miles* algorithm.

```
#include <R.h>
#include <Rinternals.h>
#include <Rdefines.h>
#include <Rmath.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>

SEXP Xt_D_X (SEXP X, SEXP d1, SEXP d2, SEXP ind1, SEXP ind2){
  //computes t(X)%*%diag(d1)%*%X+diag(d2) for rows ind1 and cols ind2 (C numbering with 0 as first index)

  double *p_X, *p_d1, *p_d2, *p_res;
  int *p_ind1, *p_ind2;

  /*pointers to data arrays*/
  p_X    = NUMERIC_POINTER(X);
  p_d1   = NUMERIC_POINTER(d1);
  p_d2   = NUMERIC_POINTER(d2);
  p_ind1 = INTEGER_POINTER(ind1);
  p_ind2 = INTEGER_POINTER(ind2);

  /*get matrix dimensions of X*/
  SEXP X_dim = getAttrib(X,R_DimSymbol);
  int *p_X_dim = INTEGER_POINTER(X_dim);

  /*get length of ind1 and ind2*/
  int ind1_length, ind2_length;
  ind1_length = length(ind1);
  ind2_length = length(ind2);

  /*allocate result object*/
  SEXP res = PROTECT(allocMatrix(REALSXP,ind2_length,ind2_length));
  p_res    = NUMERIC_POINTER(res);

  /*multiplication*/
  int i,j,k;
  int ind_res = 0;
  int aux_ind1, aux_ind2;
  double res_i;
  for (i=0; i<ind2_length; i++){ //column i of result
    aux_ind2 = p_X_dim[0]*p_ind2[i];
    for (j=0; j<ind2_length; j++){ //row j of result
      res_i = 0;
      aux_ind1 = p_X_dim[0]*p_ind2[j];
      for (k=0; k<ind1_length; k++){
        res_i += p_X[p_ind1[k]+aux_ind1] * p_X[p_ind1[k]+aux_ind2] * p_d1[k];
      }
      p_res[ind_res] = i==j ? res_i+p_d2[i] : res_i;
```

```
      ind_res++;
    }
  }

  UNPROTECT(1);
  return(res);

}

SEXP Xt_D_Xv2 (SEXP X, SEXP d1, SEXP d2){
  //computes t(X)%*%diag(d1)%*%X+diag(d2)

  double *p_X, *p_d1, *p_d2, *p_res;

  /*pointers to data arrays*/
  p_X    = NUMERIC_POINTER(X);
  p_d1   = NUMERIC_POINTER(d1);
  p_d2   = NUMERIC_POINTER(d2);

  /*get matrix dimensions of X*/
  SEXP X_dim = getAttrib(X,R_DimSymbol);
  int *p_X_dim = INTEGER_POINTER(X_dim);

  /*allocate result object*/
  SEXP res = PROTECT(allocMatrix(REALSXP,p_X_dim[1],p_X_dim[1]));
  p_res    = NUMERIC_POINTER(res);

  /*multiplication*/
  int i,j,k;
  int ind_res = 0;
  int aux_ind1, aux_ind2;
  double res_i;
  for (i=0; i<p_X_dim[1]; i++){ //column i of result
    for (j=0; j<p_X_dim[1]; j++){ //row j of result
      res_i = 0;
      aux_ind1 = p_X_dim[0]*j;
      aux_ind2 = p_X_dim[0]*i;
      for (k=0; k<p_X_dim[0]; k++){
        res_i += p_X[aux_ind1] * p_X[aux_ind2] * p_d1[k];
        aux_ind1++;
        aux_ind2++;
      }
      p_res[ind_res] = i==j ? res_i*p_d2[i] : res_i;
      ind_res++;
    }

  }

  UNPROTECT(1);
  return(res);

}

SEXP bycol_all_equal_double2(SEXP x){
  double *p_x;
  int *p_res;
  int i,j,col;
  int *p_dim;

  p_x    = NUMERIC_POINTER(x);
  p_dim  = INTEGER_POINTER(getAttrib(x,R_DimSymbol));

  SEXP res = PROTECT(allocVector(LGLSXP,p_dim[1]));
  p_res    = LOGICAL_POINTER(res);

  for (col=0;col<p_dim[1];col++){
    p_res[col] = 1;
    if (ISNAN(p_x[p_dim[0]*col])){
      j = p_dim[0]*col+1;
      while (ISNAN(p_x[j]) && j<p_dim[0]*(col+1)) j++;
      for(i=j;i<p_dim[0]*(col+1) && p_res[col]==1;i++){
        if (!ISNAN(p_x[i]) && p_x[i]!=p_x[j]) p_res[col] = 0;
      }
    }
    else{
      for(i=p_dim[0]*col+1;i<p_dim[0]*(col+1) && p_res[col]==1;i++){
        if (!ISNAN(p_x[i]) && p_x[i]!=p_x[p_dim[0]*col]) p_res[col] = 0;
      }
    }
  }

  UNPROTECT(1);
  return(res);
}
```

Appendix D

Appendix to Chapter 6

D.1 Passively measured TV consumption data

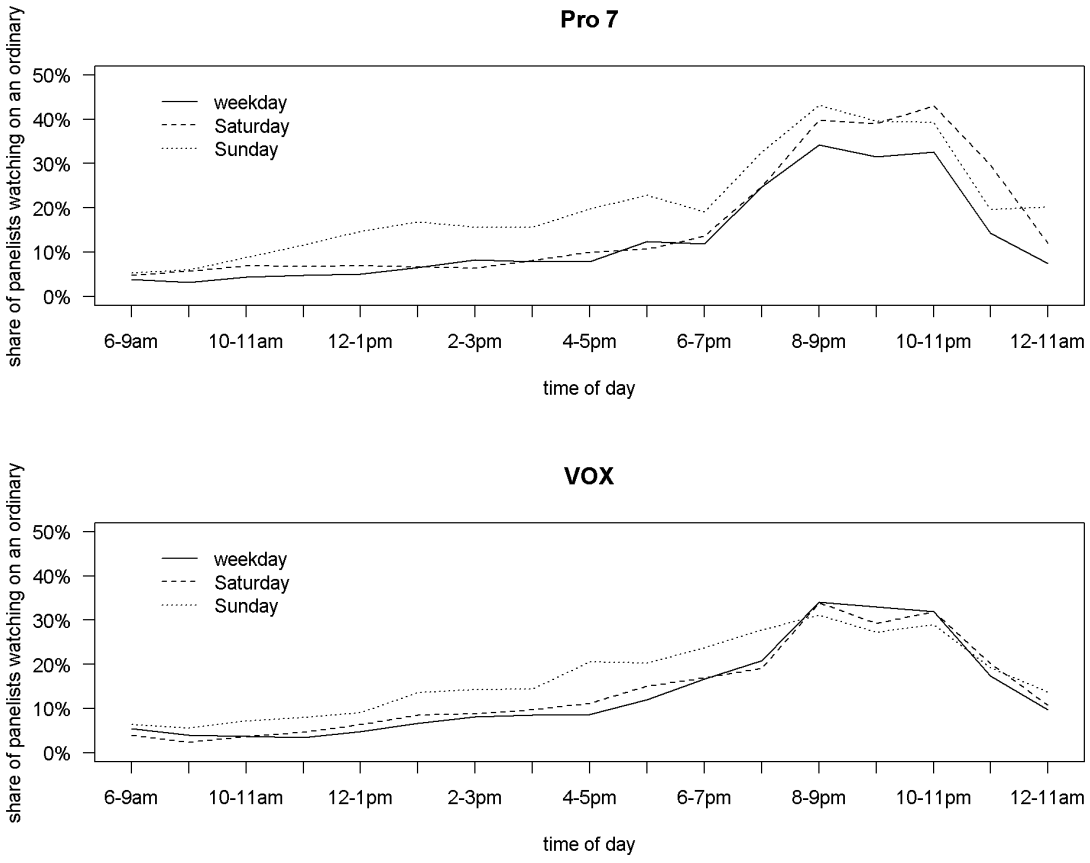

Figure D.1: TV consumption on an ordinary day by time of the day and channel

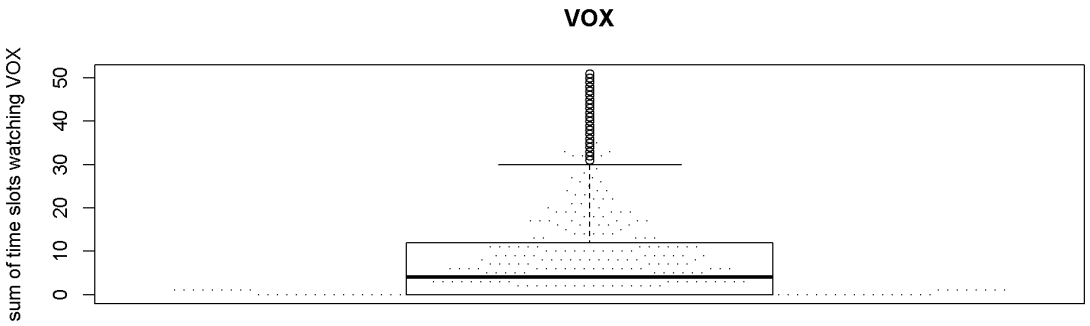

Figure D.2: Population boxplot (Rinne, 2008, p. 49) and beeswarm plot (Eklund, 2016) of a 2.5% simple random sample.

D.2 Descriptive statistics

channel	broadcasting since	URL	average reaches of time slots on weekdays, Saturdays, and Sundays			accuracy relative to survey data	κ_C (Cohen, 1960)	included in simulation
ARD	1952	`www.ard.de`	30%	36%	41%	71%	0.25	no
ZDF	1963	`www.zdf.de`	30%	33%	36%	73%	0.25	no
RTL	1984	`www.rtl.de`	25%	27%	35%	75%	0.22	no
SAT 1	1984	`www.sat1.de`	19%	20%	21%	82%	0.22	no
Pro 7	1989	`www.prosieben.de`	12%	15%	19%	86%	0.25	yes
VOX	1993	`www.vox.de`	13%	14%	17%	86%	0.17	yes
Kabel 1	1992	`www.kabeleins.de`	11%	12%	15%	87%	0.15	no
RTL 2	1993	`www.rtl2.de`	10%	12%	15%	88%	0.10	no
Super RTL	1995	`www.superrtl.de`	4%	5%	6%	95%	0.21	no
Tele 5	2002	`www.tele5.de`	3%	3%	3%	97%	0.17	no
VIVA	1993	`www.viva.tv`	1%	1%	1%	99%	0.06	no

Table D.1: Passively measured data on the TV channels

variable		statistics			description
categorical		counts		shares	
`total`					panelists with passive TV
		11916		100%	measurement in place
`female`					gender
female		5987		50%	`code: 1`
male		5929		50%	`code: 0`
`employ`					employment type
fulltime		4434		37%	`code: 1`
halftime or student		2545		21%	`code: 0.5`
parttime		348		3%	`code: 0.2`
not employed		4589		39%	`code: 0`
continuous	Mean	Median	Min	Max	
`age`					age in years on April, 30th 2013
	50	52	14	99	
`citysize`					home town population count
	35	281	1	2200	average of classes /1000
`hhsize`					number of persons in household
	2.59	2	1	8	
`kids6`					number of kids aged 6 or
	0.13	0	0	3	younger in household
`kids18`					number of kids aged 18 or
	0.50	0	0	6	younger in household
`internet`					average number of days with
	4.89	7	0	7	Internet usage per week

Table D.2: Basic claims data

D.3 Modeling the response mechanism

	MACAR		MAAR	
covariate	$exp\left(\hat{\beta}\right)$	z value	$exp\left(\hat{\beta}\right)$	z value
`intercept`	1.9932	35.5137	0.6792	−1.1104
`female`	-	-	1.1371	2.9203
`employ`	-	-	0.7520	−4.7139
`age`	-	-	1.0815	8.0142
`citysize`	-	-	0.9999	−2.7551
`hhsize`	-	-	0.5646	−4.9363
`kids6`	-	-	1.1801	2.9827
`kids18`	-	-	1.2295	5.0186
`internet`	-	-	1.0262	3.0280
`age`^{2}	-	-	0.9996	−5.2049
`hhsize`^{2}	-	-	1.0398	3.3061
`age` × `hhsize`	-	-	0.9954	−3.1992

Table D.3: Parameters estimates for the response mechanism ($N = 11916$). These estimates are used to delete observations within the simulation study, thereby mimicking natural nonresponse. The estimate for the intercept in the MACAR case of approximately 2 means that it is twice as likely to be observed than to be missing.

D.4 'Population' results for the analysis models

The $N = 11916$ data set is a sample of a larger population of TV consumers in Germany. The statistical inference conducted on this data set refers to this larger population. Nevertheless, for the simulation study the same data set is declared the *population*. Therefore, rather than n, N denotes the number of observations.

table 1:	`householdsize (hh)` > 2 versus `Pro_7_Sunday_8to9pm (P8)`			
table cells	`hh` $\leqslant 2$ & `P8` $= 0$	`hh` > 2 & `P8` $= 0$	`hh` $\leqslant 2$ & `P8` $= 1$	`hh` > 2 & `P8` $= 1$
frequencies	4493	2276	2505	2642
$exp\left(\hat{\beta}_{mlogit}\right)$	1	0.5066	0.5575	0.5880
table 2:	`householdsize (hh)` > 2 versus `VOX` > 3			
table cells	`hh` $\leqslant 2$ & `VOX` $\leqslant 3$	`hh` > 2 & `VOX` $\leqslant 3$	`hh` $\leqslant 2$ & `VOX` > 3	`hh` > 2 & `VOX` > 3
frequencies	3295	2127	3703	2791
$exp\left(\hat{\beta}_{mlogit}\right)$	1	0.6455	1.1238	0.8470
table 3:	`Pro_7_Sunday_8to9pm (P8)` versus `VOX` > 3			
table cells	`P8` $= 0$ & `VOX` $\leqslant 3$	`P8` $= 1$ & `VOX` $\leqslant 3$	`P8` $= 0$ & `VOX` > 3	`P8` $= 1$ & `VOX` > 3
frequencies	4259	1163	2510	3984
$exp\left(\hat{\beta}_{mlogit}\right)$	1	0.2731	0.5893	0.9354

Table D.4: Contingency tables ($N = 11916$).

Model	Parameter	$\hat{\beta}$	t value
Linear	`Intercept`	7.503162	90.0992
	`age`	−0.053390	−37.7711
	`Pro_7_Sunday_8to9pm (P8)`	0.160513	3.2308
Square	`Intercept`	7.520440	94.8994
	`age`	−0.054274	−39.8622
	`VOX`	0.024037	3.4557
	`VOX` ^{2}	−0.000629	−2.7476
Interaction	`Intercept`	7.812346	77.7167
	`age`	−0.059054	−33.7441
	`Pro_7_Sunday_8to9pm (P8)`	−0.620408	−4.1095
	`age` × `Pro_7_Sunday_8to9pm (P8)`	0.016220	5.4770
Cube	`Intercept`	7.531351	96.2085
	`age`	−0.054289	−39.8800
	`VOX`	0.016516	3.6630
	`VOX` ^{3}	−0.000012	−2.7894

Table D.5: Regression models ($N = 11916$).

Variable name	Mean in cluster 1	Mean in cluster 2	t value
`kids18`	0.4152	0.6363	13.18
`VOX`	4.4499	12.6131	50.05
number of cases	8067	3849	-

Table D.6: Clustering ($N = 11916$).

D.5 Convergence plots

In section 6.3.3 two key findings regarding convergence issues are presented: Due to the high correlations of the variables with their transformations, convergence is extremely slow for educated algorithms; and (ignorant) PMM sometimes shows very odd convergence behavior. In fact, over many iterations there is no variance of analysis model parameter estimates at all.

In this appendix section, one sample of size 600 is drawn from the population and the response mechanism is MCAR. The estimand is $\mu_{\texttt{P8}}$. Instead of conducting the default simple random hot-deck imputation to get the algorithm started, the missing values are initially filled with the column minimum values. Then, for a single imputation with each of the six imputation algorithms, 500 Gibbs sampler iterations are conducted. The evolution of the parameter estimates over the iterations is presented in figure D.3. The respective sample autocorrelation functions are presented in figure D.4. Finally, figure D.5 focuses on the ignorant PMM algorithm.

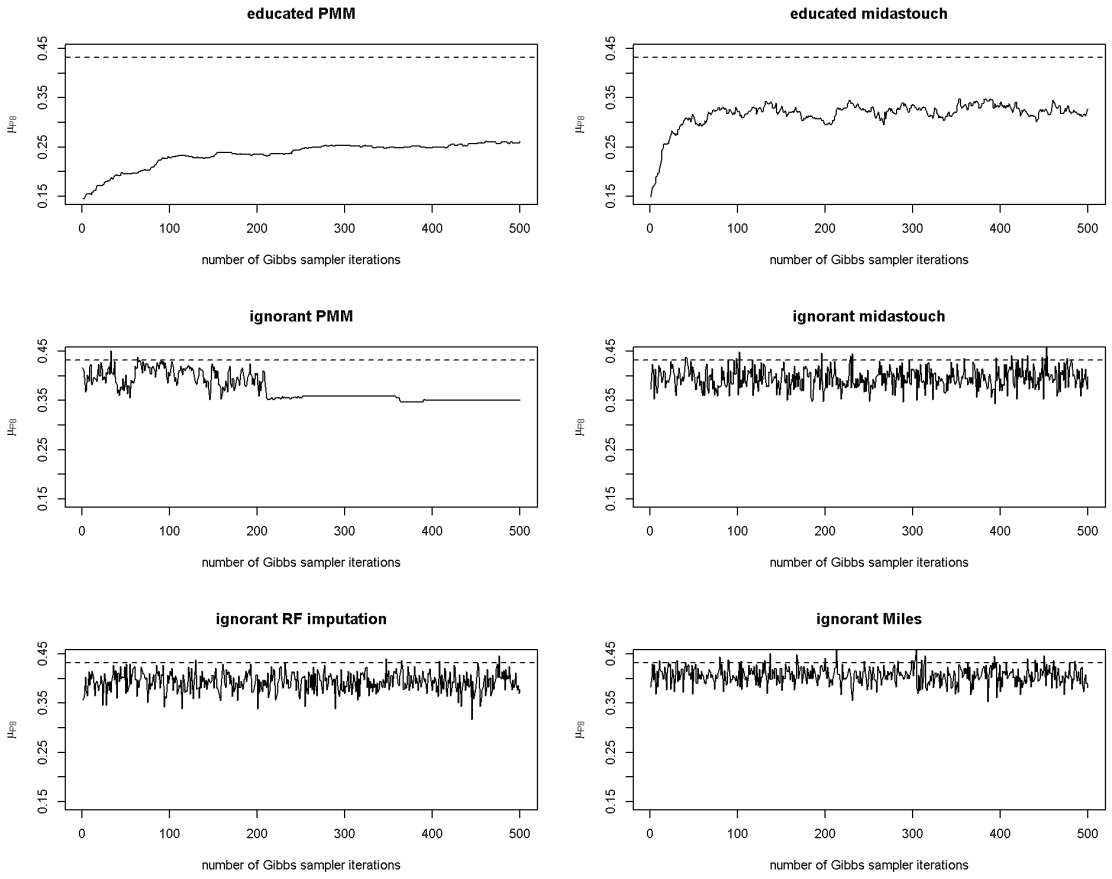

Figure D.3: Convergence plots for the parameter $\mu_{\texttt{P8}}$ in one $n = 600$ sample. The dashed line marks the value of $\mu_{\texttt{P8}}$ in the population. To better see the dependence on the starting values, the missing values are initially imputed by the column minimum values, which is zero for the variable `P8`. For the educated procedures the 500 iterations are clearly insuffient to get even close to the true value. Ignorant PMM shows very odd behavior beyond the 200th iteration for no obvious reason. The other three ignorant procedures do not show any trend.

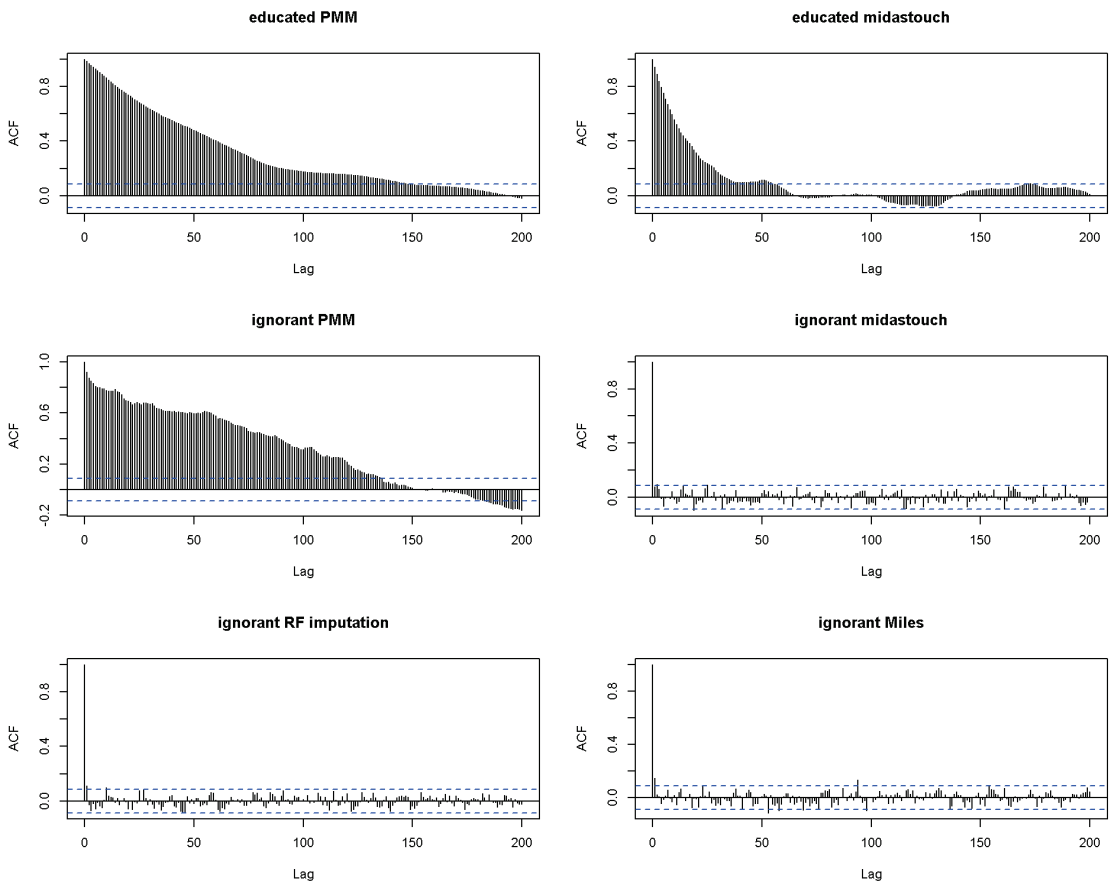

Figure D.4: Plots of the autocorrelation function of the series in figure D.3 with $\alpha = 5\%$ confidence intervals. The educated algorithms have a long memory, which is probably caused by the high correlations between the incomplete variables and their transformations. Ignorant PMM seems to have a very long memory, too. The other three ignorant procedures have essentially no memory at all. I.e., employing them to sample from the posterior distribution of μ_{P8} is extremly efficient.

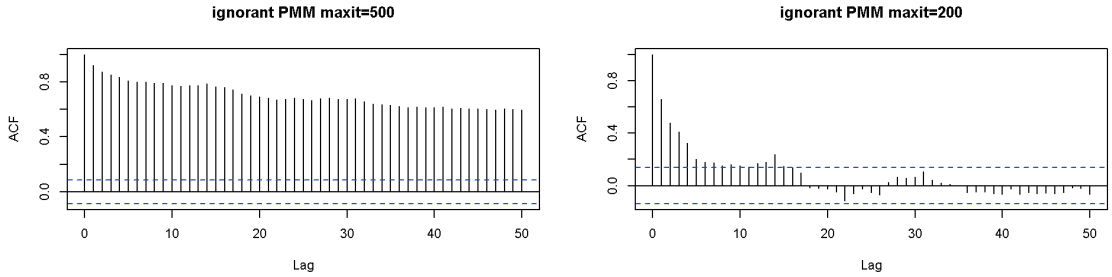

Figure D.5: Plots of the autocorrelation function of the series in figure D.3 for the ignorant PMM algorithm. The left plot shows the autocorrelation function based on the entire series with 500 iterations. The right plot is based on the same series, but only on its first 200 data points, i.e., before the odd behavior occurs. The extreme autocorrelation is clearly driven by the odd behavior. However, even disregarding this issue, the values of the autocorrelation function of PMM are much larger than those of the other ignorant algorithms in figure D.4

D.6 Detailed simulation results

Parameter		PMM		*midastouch*		RF		*Miles*	
	Missing Always	CAR	AR	CAR	AR	CAR	AR	CAR	AR
univariate statistics									
	average	118		118		158		119	
	$\mu_{\texttt{P8}}$	111	110	111	109	146	147	114	112
	$\mu_{\texttt{VOX}}$	120	122	120	120	186	194	120	122
	$ln\{var(\texttt{VOX})\}$	121	122	122	123	135	139	122	123
bivariate statistics									
	average	115		113		143		118	
table 1	$\texttt{hh} > 2$ & $\texttt{P8} = 0$	102	103	101	103	102	102	101	101
	$\texttt{hh} \leqslant 2$ & $\texttt{P8} = 1$	107	107	106	105	126	135	107	107
	$\texttt{hh} > 2$ & $\texttt{P8} = 1$	106	104	106	104	124	121	107	105
table 2	$\texttt{hh} > 2$ & $\texttt{VOX} \leqslant 3$	104	107	104	105	100	97	102	99
	$\texttt{hh} \leqslant 2$ & $\texttt{VOX} > 3$	115	120	115	116	167	187	116	117
	$\texttt{hh} > 2$ & $\texttt{VOX} > 3$	111	112	110	111	157	161	111	112
table 3	$\texttt{P8} = 1$ & $\texttt{VOX} \leqslant 3$	130	132	128	129	190	200	150	162
	$\texttt{P8} = 0$ & $\texttt{VOX} > 3$	135	137	130	134	144	159	146	146
	$\texttt{P8} = 1$ & $\texttt{VOX} > 3$	115	115	116	114	154	149	121	119
regression coefficients									
	average	100		100		99		98	
Linear	`Intercept`	100	98	101	98	98	98	99	99
	`age`	100	98	100	98	99	99	99	99
	`P8`	103	105	103	105	96	97	99	101
Square	`Intercept`	104	99	104	99	101	98	102	99
	`age`	101	99	101	99	100	100	100	100
	`VOX`	100	104	100	104	101	100	97	95
	$\texttt{VOX}^2$	90	95	92	97	101	100	94	92
Inter-	`Intercept`	99	103	100	103	99	100	98	99
action	`age`	99	102	100	102	100	100	99	99
	`P8`	95	100	96	100	94	95	95	93
	`age` × `P8`	94	97	95	98	92	92	94	92
Cube	`Intercept`	104	100	104	99	101	99	102	99
	`age`	101	99	101	99	100	100	100	100
	`VOX`	105	108	104	107	101	100	99	98
	$\texttt{VOX}^3$	91	96	93	98	103	102	95	93
cluster means									
	average	93		93		130		101	
`kids18`	$\mu_{\texttt{kids18}}$ |cluster1	89	92	89	91	107	105	95	93
	$\mu_{\texttt{kids18}}$ |cluster2	92	90	93	91	97	101	96	98
`VOX`	$\mu_{\texttt{VOX}}$ |cluster1	92	96	91	95	134	147	94	97
	$\mu_{\texttt{VOX}}$ |cluster2	93	97	95	95	177	176	120	120

Table D.7: Simulation results: (relative root mean squared error) × 100

Parameter		PMM		*midastouch*		RF		*Miles*	
	Missing Always	CAR	AR	CAR	AR	CAR	AR	CAR	AR
univariate statistics									
	average	0		0		−2		−1	
	$\mu_{\texttt{P8}}$	0	0	0	0	2	2	−1	0
	$\mu_{\texttt{VOX}}$	0	0	0	−1	−7	−7	−1	−1
	$ln\{var(\texttt{VOX})\}$	0	0	0	0	−1	−1	0	−1
bivariate statistics									
	average	−1		0		6		−1	
table 1	$\texttt{hh} > 2$ & $\texttt{P8} = 0$	−1	−2	−1	−2	0	−2	−2	−2
	$\texttt{hh} \leqslant 2$ & $\texttt{P8} = 1$	0	−2	1	−1	−6	−10	1	−2
	$\texttt{hh} > 2$ & $\texttt{P8} = 1$	1	0	2	0	−4	−6	3	0
table 2	$\texttt{hh} > 2$ & $\texttt{VOX} \leqslant 3$	−4	−8	−2	−6	−1	2	−1	−4
	$\texttt{hh} \leqslant 2$ & $\texttt{VOX} > 3$	13	20	5	11	−109	−128	−7	−4
	$\texttt{hh} > 2$ & $\texttt{VOX} > 3$	−2	−1	1	2	73	76	5	9
table 3	$\texttt{P8} = 1$ & $\texttt{VOX} \leqslant 3$	−7	−8	−6	−8	−17	−18	−12	−13
	$\texttt{P8} = 0$ & $\texttt{VOX} > 3$	−12	−12	−11	−10	14	19	−18	−16
	$\texttt{P8} = 1$ & $\texttt{VOX} > 3$	5	−1	14	8	112	106	29	24
regression coefficients									
	average	−7		−6		−4		−7	
Linear	`Intercept`	0	0	0	0	0	0	0	0
	`age`	0	0	0	0	0	0	0	0
	`P8`	−3	−8	−4	−8	−3	−7	−1	−4
Square	`Intercept`	0	0	0	0	0	0	0	0
	`age`	0	0	0	0	0	0	0	0
	`VOX`	−23	−26	−20	−23	−10	−15	−22	−25
	$\texttt{VOX}^2$	−32	−33	−29	−29	−13	−14	−28	−29
Inter-	`Intercept`	0	0	0	0	0	0	0	0
action	`age`	1	0	1	0	0	−1	0	0
	`P8`	7	1	6	1	0	−8	2	−3
	`age` × `P8`	5	−1	4	−1	−1	−9	1	−4
Cube	`Intercept`	0	0	0	0	0	0	0	0
	`age`	0	0	0	0	0	0	0	0
	`VOX`	−18	−21	−15	−18	−8	−13	−19	−22
	$\texttt{VOX}^3$	−31	−29	−27	−25	−11	−10	−26	−25
cluster means*									
	average	−2		−2		−7		−2	
`kids18`	$\mu_{\texttt{kids18}}$ |cluster1	−4	−3	−4	−3	−7	−7	−5	−4
	$\mu_{\texttt{kids18}}$ |cluster2	1	2	1	2	4	5	3	3
`VOX`	$\mu_{\texttt{VOX}}$ |cluster1	0	−1	−1	−1	−10	−11	1	0
	$\mu_{\texttt{VOX}}$ |cluster2	−4	−5	−4	−5	−15	−15	−8	−9

Table D.8: Simulation results: bias relative to the population parameter (in %): $100 \cdot \{\sum(\hat{\gamma} - \gamma)\}/(n_{sims} \cdot \gamma)$.

Parameter		PMM		*midastouch*		RF		*Miles*	
	Missing Always	CAR	AR	CAR	AR	CAR	AR	CAR	AR
univariate statistics									
	average	945		949		828		939	
	$\mu_{\texttt{P8}}$	948	949	950	958	852	856	944	949
	$\mu_{\texttt{VOX}}$	954	942	951	944	740	707	945	930
	$ln\{var(\texttt{VOX})\}$	938	939	945	946	913	901	940	927
bivariate statistics									
	average	954		957		878		944	
table 1	$\texttt{hh} > 2$ & $\texttt{P8} = 0$	962	971	965	972	972	972	966	971
	$\texttt{hh} \leqslant 2$ & $\texttt{P8} = 1$	963	963	964	967	910	902	963	967
	$\texttt{hh} > 2$ & $\texttt{P8} = 1$	957	965	957	968	915	914	955	968
table 2	$\texttt{hh} > 2$ & $\texttt{VOX} \leqslant 3$	967	967	969	967	968	967	972	972
	$\texttt{hh} \leqslant 2$ & $\texttt{VOX} > 3$	967	956	963	962	821	795	966	961
	$\texttt{hh} > 2$ & $\texttt{VOX} > 3$	956	964	959	963	843	826	958	958
table 3	$\texttt{P8} = 1$ & $\texttt{VOX} \leqslant 3$	925	919	933	930	725	700	873	845
	$\texttt{P8} = 0$ & $\texttt{VOX} > 3$	944	930	953	940	926	910	903	917
	$\texttt{P8} = 1$ & $\texttt{VOX} > 3$	952	947	946	950	860	869	936	944
regression coefficients									
	average	968		968		968		970	
Linear	`Intercept`	951	953	957	956	960	960	958	956
	`age`	956	958	956	956	960	960	959	957
	`P8`	965	963	962	955	971	979	974	968
Square	`Intercept`	957	960	956	964	955	963	959	965
	`age`	960	952	958	951	961	953	962	952
	`VOX`	985	985	985	985	987	982	986	983
	$\texttt{VOX}^2$	992	990	990	993	984	986	987	987
Inter-	`Intercept`	962	953	964	954	965	960	967	961
action	`age`	958	951	957	948	953	946	957	947
	`P8`	985	971	987	972	987	968	989	973
	`age` × `P8`	984	972	984	973	984	975	984	984
Cube	`Intercept`	958	962	955	961	956	962	961	963
	`age`	959	952	959	951	961	952	961	952
	`VOX`	982	984	983	983	983	984	991	985
	$\texttt{VOX}^3$	990	989	987	991	979	977	987	986
cluster means*									
	average	976		978		841		954	
`kids18`	$\mu_{\texttt{kids18}}$ |cluster1	980	976	982	978	941	930	970	969
	$\mu_{\texttt{kids18}}$ |cluster2	984	986	986	989	974	973	981	982
`VOX`	$\mu_{\texttt{VOX}}$ |cluster1	988	979	989	981	876	849	987	979
	$\mu_{\texttt{VOX}}$ |cluster2	966	953	963	957	587	595	876	884

Table D.9: Simulation results: coverage of 950‰ confidence intervals. *It is uncertain, whether the clustering fulfills the conditions of Yang & Kim (2016, p. 246), and therefore whether Rubin's combining rules are appropriate.

Bibliography

Aerts, M., Claeskens, G., Hens, N. & Molenberghs, G. (2002). Local multiple imputation. *Biometrika* **89**, 375–88.

Akande, O., Li, F. & Reiter, J. (2016). An empirical comparison of multiple imputation methods for categorical data. To appear.

Andridge, R. R. & Little, R. J. A. (2010). A review of hot deck imputation for survey non-response. *International Statistical Review* **78**, 40–64.

Andridge, R. R. (2011). Quantifying the impact of fixed effects modeling of clusters in multiple imputation for cluster randomized trials. *Biometrical Journal* **53**, 57–74.

Arya, S., Mount, D., Kemp, S. E. & Jefferis, G. (2015). *RANN: Fast Nearest Neighbour Search (Wraps Arya and Mount's ANN Library)*. R package version 2.5. `https://CRAN.R-project.org/package=RANN`.

Barnard, J. & Rubin, D. B. (1999). Small-sample degrees of freedom with multiple imputation. *Biometrika* **86**, 948–55.

Bosch, V. (2005). A generalized measure of effective sample size. Technical report, GfK AG, Nuremberg, Germany.

Breiman, L. & Friedman, J. (1985). Estimating optimal transformations in multiple regression and correlation (with discussion). *Journal of the American Statistical Association* **80**, 580–619.

Brinkman, N. D. (1981). Ethanol fuel - a single-cylinder engine study of efficiency and exhaust emissions. *SAE Transactions* **90**, 1410–24.

Bronstein, I. N., Semendjajew, K. A., Musiol, G. & Muehlig, H. (2013). *Taschenbuch der Mathematik*. Haan-Gruiten: Europa-Lehrmittel, 9th ed.

Burgette, L. F. & Reiter, L. F. (2010). Multiple imputation for missing data via sequential regression trees. *American Journal of Epidemiology* **172**, 1070–6.

Cassel, C. M., Sarndal, C. E. & Wretman, J. H. (1976). Some results on generalized difference estimation and generalized regression estimation for finite populations. *Biometrika* **63**, 615–20.

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association* **74**, 829–36.

Cleveland, W. S. & Devlin, S. J. (1988). Locally weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical Association* **83**, 596–610.

Cleveland, W. S., Devlin, S. J. & Grosse, E. (1988). Regression by local fitting: Methods, properties, and computational algorithms. *Journal of Econometrics* **37**, 87–114.

Cleveland, W. S. & Loader, C. R. (1996). Smoothing by local regression: Principles and methods. In *Statistical Theory and Computational Aspects of Smoothing* (W. Hardle and M. G. Schimek, eds.), 10–49. Heidelberg: Physica.

Cochran, W. G. (1977). *Sampling Techniques*. New York: Wiley, 3rd ed.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* **20**, 37–46.

Collins, L. M., Schafer, J. L. & Kam, C. M. (2001). A comparison of inclusive and restrictive strategies in modern missing-data procedures. *Psychological Methods* **6**, 330–51.

David, M., Little, R. J. A., Samuhel, M. E. & Triest, R.-K. (1986). Alternative methods for CPS income imputation. *Journal of the American Statistical Association* **81**, 29–41.

Davison, A. C. & Hinkley, D. V. (1997). *Bootstrap Methods and Their Application*. Cambridge, U.K.: Cambridge University Press.

de Jong, R. N. (2012). Robust Multiple Imputation. *Dissertation* Hamburg University.

Deming, W. E. & Stephan, F. F. (1940). On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *The Annals of Mathematical Statistics* **11**, 427–44.

Demirtas, H., Arguelles, L. M., Chung, H. & Hedeker, D. (2007). On the performance of bias-reduction techniques for variance estimation in approximate Bayesian bootstrap imputation. *Computational Statistics & Data Analysis* **51**, 4064–8.

Doove, L. L., van Buuren, S. & Dusseldorp, E. (2014). Recursive partitioning for missing data imputation in the presence of interaction effects. *Computational Statistics & Data Analysis* **72**, 92–104.

D'Orazio, M., Di Zio, M. & Scanu, M. (2006). Statistical Matching: Theory and Practice. New York: Wiley.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics* **7**, 1–26.

Eklund, A. (2016). *beeswarm: The Bee Swarm Plot: An Alternative to Stripchart*. R package version 0.2.3. `http://www.cbs.dtu.dk/~eklund/beeswarm/`.

Enders, C. (2011). Missing not at random models for latent growth curve analysis. *Psychological Methods* **16**, 1–16.

Flach, P. (2012). *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*. Cambridge, U.K.: Cambridge University Press.

Gaffert, P., Bosch, V. & Meinfelder, F. (2016). Interactions and squares: Don't transform, just impute! *Proceedings of the survey research methods section of the American Statistical Association*. To appear.

Gaffert, P., Meinfelder, F. & Bosch, V. (2016). Towards an mi-proper predictive mean matching. *Working Paper.* https://www.uni-bamberg.de/fileadmin/uni/fakultaeten/sowi_lehrstuehle/statistik/Personen/Dateien_Florian/properPMM.pdf, 1–15.

Gelman, A. & Hill, J. (2011). Opening windows to the black box. *Journal of Statistical Software* **40**, 1–31.

Greenberg, E. (2013). *Introduction to Bayesian Econometrics.* Cambridge, U.K.: Cambridge University Press, 2nd ed.

Greene, W. H. (2008). *Econometric Analysis.* Upper Saddle River, NJ: Prentice Hall, 6th ed.

Harrell, F. E. (2015). *Hmisc: Harrell Miscellaneous.* R package version 3.16-0. http://biostat.mc.vanderbilt.edu/wiki/Main/Hmisc.

Hastie, T. & Loader, C. (1993). Local regression: Automatic kernel carpentry. *Statistical Science* **8**, 120–43.

Heitjan, D. F. & Little, R. J. A. (1991). Multiple imputation for the fatal accident reporting system. *Journal of the Royal Statistical Society* C **40**, 13–29.

IBM Corp. (2015). *IBM SPSS Statistics for Windows: Version 23.0.* Armonk, NY: IBM Corp. http://www-01.ibm.com/support/docview.wss?uid=swg24038592.

James, G., Witten, D., Hastie, T. & Tibshirani, R. (2013). An Introduction to Statistical Learning. New York: Springer.

Kennickell, A. B. (1991). Imputation of the 1989 survey of consumer finances: Stochastic relaxation and multiple imputation. *Proceedings of the survey research methods section of the American Statistical Association.* 1–10.

Kim, J. K. (2002). A note on approximate Bayesian bootstrap imputation. *Biometrika* **89**, 470–7.

Kish, L. (1965). *Survey Sampling.* New York: Wiley.

Koller-Meinfelder, F. (2009). Analysis of Incomplete Survey Data - Multiple Imputation via Bayesian Bootstrap Predictive Mean Matching. *Dissertation* Otto-Friedrich-University Bamberg.

Liaw, A. & Wiener, M. (2002). Classification and regression by randomForest. *R News* **2**, 18–22.

Lillard, L., Smith, J. P. & Welch, F. (1982). What do we really know about wages? The importance of nonreporting and census imputation. Technical report, Rand Corporation, Santa Monica, CA.

Little, R. J. A. (1988). Missing-data adjustments in large surveys. *Journal of Business & Economic Statistics* **6**, 287–96.

Little, R. J. A. & Rubin, D. B. (2002). *Statistical Analysis with Missing Data.* Hoboken, NJ: Wiley, 2nd ed.

Little, J. D. C. (1970). Models and managers: The concept of a decision calculus. *Management Science* **16**, B466–85.

Liu, J., Gelman, A., Hill, J. & Su, Y.-S. (2013). On the stationary distribution of iterative imputations. *Biometrika* **101**, 155–73.

Loader, C. R. (1999). *Local Regression and Likelihood*. New York: Springer.

Mealli, F. & Rubin, D. B. (2015). Clarifying missing at random and related definitions, and implications when coupled with exchangeability. *Biometrika* **102**, 995–1000.

Meinfelder, F. & Schnapp, T. (2015). *BaBooN: Bayesian Bootstrap Predictive Mean Matching*. R package version 0.2-0. `https://cran.r-project.org/web/packages/BaBooN/index.html`.

Meng, X. L. (1994). Multiple-imputation inferences with uncongenial sources of input (with discussion). *Statistical Science* **10**, 538–73.

Morris, T. P., White, I. R. & Royston, P. (2014). Tuning multiple imputation by predictive mean matching and local residual draws. *BMC Medical Research Methodology* **14**, 1–13.

Nagelkerke, N. J. D. (1991). A note on a general definition of the coefficient of determination. *Biometrika* **78**, 691–2.

Parzen, M., Lipsitz, S. R. & Fitzmaurice, G. M. (2005). A note on reducing the bias of the approximate Bayesian bootstrap imputation variance estimator. *Biometrika* **92**, 971–4.

Punj, G. & Stewart, D. W. (1983). Cluster analysis in marketing research: Review and suggestions for application. *Journal of Marketing Research* **20**, 134–48.

R Core Team (2016). *A Language and Environment for Statistical Computing: Version 3.3.2*. Vienna, Austria: R Foundation for Statistical Computing. `https://www.R-project.org`.

Raessler, S. (2002). *Statistical Matching: A Frequentist Theory. Practical Applications, and Alternative Bayesian Approaches*. New York: Springer.

Raghunathan, T. E., Lepkowski, J. M., Van Hoewyk, J. & Solenberger, P. (2001). A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey Methodology* **27**, 85–96.

Raghunathan, T. E., Solenberger, P. W. & Van Hoewyk, J. (2002). IVEware: Imputation and Variance Estimation Software. Ann Arbor, MI: Survey Methodology Program, Survey Research Center, Institute for Social Research, University of Michigan. `http://www.isr.umich.edu/src/smp/ive/`.

Raghunathan, T. E. (2015). *Missing Data Analysis in Practice*. Boca Raton, FL: Chapman & Hall/CRC.

Rao, J. N. K. & Singh, A. C. (1997). A ridge shrinkage method for range restricted weight calibration in survey sampling. *Proceedings of the survey research methods section of the American Statistical Association*. 57–64.

Rinne, H. (2008). *Taschenbuch der Statistik*. Frankfurt: Harri Deutsch, 4th ed.

Royston, P. & White, I. R. (2011). Multiple imputation by chained equations (MICE): Implementation in Stata. *Journal of Statistical Software* **45**, 1–20.

Rubin, D. B. (1976). Inference and missing data. *Biometrika* **63**, 581–92.

Rubin, D. B. (1978). Multiple imputations in sample surveys - a phenomenological Bayesian approach to nonresponse. *Proceedings of the survey research methods section of the American Statistical Association*. 20–34.

Rubin, D. B. (1981). The Bayesian bootstrap. *The Annals of Statistics* **9**, 130–4.

Rubin, D. B. (1986). Statistical matching using file concatenation with adjusted weights and multiple imputations. *Journal of Business & Economic Statistics* **4**, 87–94.

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

Rubin, D. B. & Schenker, N. (1986). Multiple imputation for interval estimation from simple random samples with ignorable nonresponse. *Journal of the American Statistical Association* **81**, 366–74.

Rubin, D. B. (1996). Multiple imputation after 18+ years. *Journal of the American Statistical Association* **91**, 473–89.

SAS Institute Inc. (2015). *SAS Software University Edition: Version 9.4*. Cary, NC: SAS Institute Inc. `https://www.sas.com/en_us/software/university-edition.html`.

Schafer, J. L. (1997). *Analysis of Incomplete Multivariate Data*. Boca Raton, FL: Chapman & Hall/CRC.

Schenker, N. & Taylor, J. M. G. (1996). Partially parametric techniques for multiple imputation. *Computational Statistics & Data Analysis* **22**, 425–46.

Seaman, S. R., Bartlett, J. W. & White, I. R. (2012). Multiple imputation of missing covariates with non-linear effects and interactions: An evaluation of statistical methods. *BMC Medical Research Methodology* **12**, 1–13.

Siddique, J. (2005). Multiple Imputation using an Iterative Hot-Deck with Distance-Based Donor Selection. *Dissertation* University of California Los Angeles.

Siddique, J. & Belin, T. R. (2008). Multiple imputation using an iterative hot-deck with distance-based donor selection. *Statistics in Medicine* **27**, 83–102.

Siddique, J. & Harel, O. (2009). MIDAS: A SAS macro for multiple imputation using distance-aided selection of donors. *Journal of Statistical Software* **29**, 1–18.

StataCorp. (2015). *Stata Statistical Software: Release 14*. College Station, TX: StataCorp LP. `http://www.stata.com/`.

Tanner, M. A. & Wong, W. H. (1987). The calculation of posterior distributions by data augmentation). *Journal of the American Statistical Association* **82**, 528–540.

van Buuren, S. & Groothuis-Oudshoorn, K. (1999). Flexible multivariate imputation by mice. Technical report, TNO Prevention and Health, Leiden, The Netherlands.

van Buuren, S. & Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software* **45**, 1–67.

van Buuren, S. (2012). *Flexible Imputation of Missing Data*. Boca Raton, FL: Chapman & Hall/CRC.

Vink, G. & van Buuren, S. (2013). Multiple imputation of squared terms. *Sociological Methods & Research* **42**, 598–607.

von Hippel, P. T. (2007). Regression with missing Ys: An improved strategy for analyzing multiply imputed data. *Sociological Methodology* **37**, 83–117.

von Hippel, P. T. (2009). How to impute interactions, squares, and other transformed variables. *Sociological Methodology* **39**, 265–91.

Wu, C. F. J. (1986). Jackknife, bootstrap and other resampling methods in regression analysis. *The Annals of Statistics* **14**, 1261–95.

Xie, X. & Meng, X. L. (2014). Dissecting multiple imputation from a multi-phase inference perspective: What happens when God's, imputer's and analyst's models are uncongenial? *Statistica Sinica* to appear.

Yang, S. & Kim, J. K. (2016). A note on multiple imputation for method of moments estimation. *Biometrika* **103**, 244–51.