Model Averaging and Model Selection after Multiple Imputation using the *R*-package `MAMI`

Version: May 6, 2019
Author: Michael Schomaker[1], with support from Christian Heumann

NEW: speed up estimation using parallelization, see Section 6.1.

Contents

1 Background **2**
1.1 Installation . . . 2
1.2 Citation . . . 2
1.3 Motivation . . . 3
1.3.1 Model Averaging . . . 3
1.3.2 Multiple Imputation . . . 3
1.3.3 Model Averaging (or Model Selection) after Multiple Imputation . . . 4

2 Model Choice **5**
2.1 Imputation Model . . . 5
2.2 Analysis Model . . . 6

3 Choice of Model Averaging (or Selection) Method **7**
3.1 Model Averaging . . . 7
3.1.1 Criterion Based Model Averaging . . . 7
3.1.2 Mallow's Model Averaging . . . 8
3.1.3 Lasso Averaging . . . 10
3.1.4 Jackknife Model Averaging . . . 11
3.2 Model Selection . . . 12
3.2.1 Criterion Based Model Selection . . . 12
3.2.2 LASSO and Shrinkage Based Model Selection . . . 13

4 Inference **13**

5 Analysis and Interpretation **14**
5.1 Example . . . 14
5.2 Suggested Reporting in Publications . . . 17

6 Miscellaneous **20**
6.1 Computation Time . . . 20
6.2 Limitations of the Approach . . . 22
6.3 Optimal Model Averaging for Prediction: Super Learning . . . 22
6.4 Miscellaneous . . . 23

Index **23**

[1]University of Cape Town, Centre for Infectious Disease Epidemiology and Research, Observatory, 7925; Cape Town, South Africa, michael.schomaker@uct.ac.za

1 Background

The *R*-package `MAMI` offers an implementation of the methodology proposed in

Schomaker, M. and C. Heumann (2014). Model selection and model averaging after multiple imputation. *Computational Statistics and Data Analysis 71*, 758–770.

It is essentially a summary of functions used in the paper, with some useful extensions. The main function of the package (`mami()`) performs model selection/averaging on multiply imputed datasets and combines the resulting estimates. The package also provides access to less frequently used model averaging techniques and offers integrated bootstrap estimation. The package is useful if

- one wants to perform model selection or model averaging on multiply imputed data and the analysis model of interest is either the linear model, the logistic model, the Poisson model, or the Cox proportional hazards model, possibly with a random intercept.
- one wants to obtain bootstrap confidence intervals for model selection or model averaging estimators (with or without missing data/imputation) – to address model selection uncertainty and to discover relationships of small effect size, see Table 1 in Schomaker and Heumann (2014).
- one wants to compare different model selection and averaging techniques, easily with the same syntax.

The package is of limited use under the following circumstances:

- if one is interested in model selection or averaging for models other than those listed above, for example parametric survival models, additive models, time-series analysis, and many others.
- if one decides for a specific model selection or averaging technique not provided by the package, see Section 3 for more details.
- if the model selection/averaging problem is computationally too intensive, see Section 6.1 for more details.

1.1 Installation

The package can be downloaded at *R*−forge:

http://mami.r-forge.r-project.org/
https://r-forge.r-project.org/R/?group_id=2152

Or, simply type:

```
install.packages("MAMI", repos=c("http://R-Forge.R-project.org",
"http://cran.at.r-project.org"), dependencies=TRUE)
```

The latter option is recommended as `mami()` depends on a couple of other packages.

1.2 Citation

If you use `MAMI`, please cite it. For suggested citations use

```
citation("MAMI")
print(citation("MAMI"), bibtex=T)
```

It is always recommended to cite the methodological reference:

Schomaker, M. and C. Heumann (2014). Model selection and model averaging after multiple imputation. *Computational Statistics and Data Analysis 71*, 758–770.

Additionally, the package and this manual can be cited as well:

Schomaker, M. (2017). *MAMI: Model averaging (and model selection) after multiple Imputation.* R package version 0.9.12.; `http://mami.r-forge.r-project.org/`

Schomaker, M. and C. Heumann (2017). *Model averaging and model selection after multiple imputation using the R-package MAMI*; `http://mami.r-forge.r-project.org/`

1.3 Motivation

1.3.1 Model Averaging

The motivation for variable selection in regression models is based on the rationale that associational relationships between variables are best understood by reducing the model's dimension. The problem with this approach is that (i) regression parameters after model selection are often biased and (ii) the respective standard errors are too small because they do not reflect the uncertainty related to the model selection process (Leeb and Pötscher, 2005; Burnham and Anderson, 2002). It has been proposed (Chatfield, 1995; Draper, 1995; Hoeting et al., 1999) that the drawback of model selection can be overcome by model averaging. With model averaging, one calculates a weighted average $\hat{\bar{\beta}} = \sum_\kappa w_\kappa \hat{\beta}_\kappa$ from the k parameter estimates $\hat{\beta}_\kappa$ ($\kappa = 1, \ldots, k$) of a set of candidate (regression) models $\mathcal{M} = \{M_1, \ldots, M_k\}$, where the weights are calculated in a way such that 'better' models receive a higher weight. A popular weight choice would be based on the exponential AIC,

$$w_\kappa^{\text{AIC}} = \frac{\exp(-\frac{1}{2}\text{AIC}_\kappa)}{\sum_{\kappa=1}^{k} \exp(-\frac{1}{2}\text{AIC}_\kappa)}, \tag{1}$$

where AIC_κ is the AIC value related to model $M_\kappa \in \mathcal{M}$ (Buckland et al., 1997) and $\sum_\kappa w_\kappa^{\text{AIC}} = 1$. It has been suggested [2] to estimate the variance of the scalar $\hat{\bar{\beta}}_j \in \hat{\bar{\beta}}$ via

$$\widehat{\text{Var}}(\hat{\bar{\beta}}_j) = \left\{ \sum_{\kappa=1}^{k} w_\kappa \sqrt{\widehat{\text{Var}}(\hat{\beta}_{j,\kappa}|M_\kappa) + (\hat{\beta}_{j,\kappa} - \hat{\bar{\beta}}_j)^2} \right\}^2, \tag{2}$$

where $\hat{\beta}_{j,\kappa}$ is the j^{th} regression coefficient of the κ^{th} candidate model. This approach tackles problem (ii), the incorporation of model selection uncertainty into the standard errors of the regression parameters; but it may not necessarily tackle problem (i) as the regression parameters may still be biased. There are multiple different suggestions on how the weights can be calculated, and those implemented in `mami()` are explained in Section 3. Note that model selection can be viewed as a special case of model averaging where the "best" model receives weight 1 (and all others a weight of 0). All implemented model selection options are listed in Section 3 too.

1.3.2 Multiple Imputation

Multiple imputation (MI) is a popular method to address missing data. Based on assumptions about the data distribution (and the mechanism which gives rise to the missing data) missing

[2] While formula (2) from Buckland et al. (1997) is the most popular choice to calculate standard errors in model averaging, it has also been criticized that the coverage probability of interval estimates based on (2) can be biased (Hjort and Claeskens, 2003).

values can be imputed by means of draws from the posterior predictive distribution of the unobserved data given the observed data. This procedure is repeated to create M imputed data sets, the (regression) analysis is then conducted on each of these data sets and the M results (M point and M variance estimates) are combined by a set of simple rules:

$$\hat{\beta}_{\text{MI}} = \frac{1}{M}\sum_{m=1}^{M}\hat{\beta}^{(m)} \tag{3}$$

and

$$\widehat{\text{Cov}}(\hat{\beta}_{\text{MI}}) \;=\; \widehat{W} + \frac{M+1}{M}\hat{B} = \frac{1}{M}\sum_{m=1}^{M}\widehat{\text{Cov}}(\hat{\beta}^{(m)}) + \frac{M+1}{M(M-1)}\sum_{m=1}^{M}(\hat{\beta}^{(m)} - \hat{\beta}_{\text{MI}})(\hat{\beta}^{(m)} - \hat{\beta}_{\text{MI}})' \tag{4}$$

where $\hat{\beta}^{(m)}$ refers to the estimate of β in the mth imputed set of data $\mathcal{D}^{(m)}$, $m = 1, \ldots, M$, $\widehat{W} = M^{-1}\sum_m \widehat{\text{Cov}}(\hat{\beta}^{(m)})$ is the average within imputation covariance, and $\hat{B} = (M-1)^{-1}\sum_m(\hat{\beta}^{(m)} - \hat{\beta}_{\text{MI}})(\hat{\beta}^{(m)} - \hat{\beta}_{\text{MI}})'$ the between imputation covariance. Confidence intervals are constructed on a t_R-distribution with approximately $R = (M-1)[1+\{M\hat{W}/(M+1)\hat{V}\}]^2$ degrees of freedom (Rubin and Schenker, 1986), though there are alternative approximations, especially for small samples (Lipsitz et al., 2002). More details on imputation can be found in Rubin (1996) and White et al. (2011), among others.

1.3.3 Model Averaging (or Model Selection) after Multiple Imputation

How can model averaging and model selection be applied to multiply imputed data? The detailed motivation can be found in Schomaker and Heumann (2014). The basic results for model *averaging* are

$$\hat{\bar{\beta}}_{\text{MI}} \;=\; \frac{1}{M}\sum_{m=1}^{M}\hat{\bar{\beta}}^{(m)} \quad \text{with} \quad \hat{\bar{\beta}}^{(m)} = \sum_{\kappa=1}^{k} w_\kappa^{(m)}\hat{\beta}_\kappa^{(m)} \tag{5}$$

and applies to any weight choice. If the variance of the model averaging estimator is estimated via (2), the overall variance of the estimator after multiple imputation relates to

$$\begin{aligned}\widehat{\text{Var}}(\hat{\bar{\beta}}_{j,\text{MI}}) \;=\; & \frac{1}{M}\sum_{m=1}^{M}\left\{\sum_{\kappa=1}^{k} w_\kappa\sqrt{\widehat{\text{Var}}(\hat{\beta}_{j,\kappa}^{(m)}) + (\hat{\beta}_{j,\kappa}^{(m)} - \hat{\bar{\beta}}_j^{(m)})^2}\right\}^2 + \\ & \frac{M+1}{M(M-1)}\sum_{m=1}^{M}(\hat{\bar{\beta}}_j^{(m)} - \hat{\bar{\beta}}_{j,\text{MI}})^2\,.\end{aligned} \tag{6}$$

Confidence intervals could then again be estimated based on a t_R distribution (as explained above) or, alternatively, via bootstrapping – see Section 4, and Table 1 in Schomaker and Heumann (2014), for more details.

Model *selection* after imputation works essentially the same, except that parameters associated with variables which have not been selected are assumed to be 0. With this assumption, a variable will be formally selected if it is selected in at least one imputed set of data, but its overall impact will depend on how often it is chosen. Here, confidence intervals will almost always be too narrow if (6) is applied (because of model selection uncertainty) and bootstrapping is recommended.

As a consequence for model selection (and model averaging), effects of variables which are not supported throughout imputed data sets (and candidate models) will simply be less pronounced.

MAMI estimates the point estimates (5), together with confidence intervals that are either based on (6), or a Bayesian variation thereof (Section 3.1.1), or bootstrapping (preferred option, Section 4).

In addition, a variable importance measure (averaged over the M imputed data sets) will be calculated: this measure, see also Burnham and Anderson (2002), simply sums up the weights w_κ of those candidate models M_κ that contain the relevant variable, and lies between 0 (unimportant) and 1 (very important). It is similar to the Bayesian posterior effect probability.

Results could be interpreted and reported as suggested in Section 5.

MAMI's main function is `mami()`. It is recommended to get familiar with the function's syntax by typing `?mami` and running the examples at the bottom of the help page. Briefly, the syntax is

```
mami(X, missing.data=c("imputed","none","CC"),
  model=c("gaussian","binomial","poisson","cox"), outcome=1, id=NULL,
  method=c("MA.criterion","MS.criterion","LASSO","LAE","MMA"),
  criterion=c("AIC","BIC","BIC+","CV","GCV"), kfold=5, cvr=F,
  inference=c("standard","+boot"), B=20, X.org=NULL, CI=0.95,
  var.remove=NULL, aweights=NULL, add.factor=NULL, add.interaction=NULL,
  add.transformation=NULL, add.stratum=NULL, ncores=1,
  candidate.models=c("all","restricted","very restricted"), screen=0,
  report.exp=FALSE, print.time=FALSE, print.warnings=TRUE, ...)
```

```
?mami
example(mami)
```

Using the above framework requires decisions with respect to the following:

- A decision about the imputation procedure, see Section 2.1.
  - this decision is being made before applying `mami`
- A decision about the (regression) analysis model, see Section 2.2.
  - this is being specified by means of the following options: `model, id` as well as sometimes `var.remove, outcome, aweights, add.factor, add.interaction, add.transformation, add.stratum, screen` .
- A decision about the model selection or model averaging approach, i.e. the determination of the weights and the set of candidate models, see Section 3.
  - this is being specified by means of the following options: `method, criterion` as well as sometimes `kfold, candidate.models` .
- A decision about confidence interval calculation, see Section 4.
  - this is being specified by means of the following options: `inference` as well as sometimes `B, X.org, CI` .

2 Model Choice

2.1 Imputation Model

MAMI does not impute. The user would typically provide the data `X` in one of the following formats:

- An object of class `amelia`, generated by imputation with `Amelia II` [author's recommendation]. See Honaker and King (2010) and Honaker et al. (2011) for more details.
- An object of class `mids`, generated by imputation with `mice`. See, for example, van Buuren and Groothuis-Oudshoorn (2011) for more details.
- A list of data sets imputed with any other package or software. With this choice bootstrap confidence intervals, as explained in Section 4, can not be calculated. If the user has used other software than *R* to generate the imputations, one option would be to i) save the imputed data sets as *.csv* files, ii) import them to *R* with `read.csv()`, and then iii) create a list of these data sets which can then be used by `mami()`.
- A single data frame. This can be of interest when
  - one is interested in a comparison with results from a complete case analysis (`missing.data = "CC"`),
  - or if one has no missing data (`missing.data="none"`) but is interested in post model selection/averaging intervals using bootstrapping .

2.2 Analysis Model

The analysis model can (currently) be any of the following: the linear regression model, the logistic regression model, the Poisson regression model, or the Cox proportional hazards regression model. Thus, β refers to the parameter vector of a generalized linear model or survival model. It is being specified with the option `model` and can be either `"gaussian"`, `"binomial"`, `"poisson"`, or `"cox"`.

If the data are not cross-sectional, but longitudinal, it is possible to specify this with a variable which indicates the cross-sectional unit, i.e. via the option `id`. Then, a (generalized) linear mixed model with random intercept is fit [or a Cox model with frailty]. When dealing with longitudinal data the following needs to be considered:

- the imputation model needs to account for the longitudinal structure of the data. For longitudinal data, there are only semi-satisfactory imputation procedures available at the moment. The best might be to use the "time series - cross section" facilities in `Amelia II`, specified with the `ts`, `cs` and `intercs` options; see Honaker et al. (2011, Section 4.5) for more details. This approach often works well, though it does not explicitly take into account the time-ordering in the data. Alternatively, if there are not many time points, reshaping the data with `reshape()`, from long to wide format, could be an option in some applications.
- The bootstrap needs to be facilitated on the id-level. `mami()` does this.

The full analysis model (before model selection or averaging) is the one specified via `model` and `id`. It includes all variables in the data set `X`, and uses the first column as the outcome variable, unless any of the below options are specified.

- `var.remove` – either a vector of character strings or integers, specifying the variables or columns which are part of the data but should not be considered in the model selection/averaging procedure.
- `screen` – number of variables which should be removed in an initial screening step with LASSO.

- `outcome` – a character vector or integer specifying the variable or column which should be treated as outcome variable. For survival models, two variables (time to event, event) need to be specified.
- `add.transformation` – a vector of character strings, specifying transformations of variables which should be added to the analysis models.
- `add.interaction` – a list of either character strings or integers, specifying the variables which should be added as interactions in the analysis model.
- `add.factor` – a list of either character strings or integers which indicates the (categorical) variables that should be treated as a factor. Rarely needed as `mami()` already searches for existing factor variables.
- `aweights` – a weight vector to be used in the analysis model.
- `add.stratum` – a character vector or integer specifying the variable used as a stratum in Cox regression analyses.

Example 2 in `?mami` shows how these options could be used.

Future releases of `MAMI` are likely going to contain the option to specify quasi-likelihood models. Other analysis models, like GEEs or parametric survival models may be offered too; however, mixed models with random slope or additive models not, because the best use of model averaging in this context is not clear yet.

3 Choice of Model Averaging (or Selection) Method

The model averaging (selection) method is chosen by the combination of the `method` and `criterion` options.

3.1 Model Averaging

3.1.1 Criterion Based Model Averaging

Criterion based model averaging means essentially using the weights (1), with any information type criterion. This can be utilized by picking:

`method="MA.criterion"` and either
`criterion="AIC"` or `criterion="BIC"` or `criterion="BIC+"`.

Model averaging is utilized with the package `MuMIn` (Barton, 2017) for `criterion="AIC/BIC"` and with `BMA` (Raftery et al., 2017) for `criterion="BIC+"`.

`MuMIn` evaluates (by default) *all* possible candidate models $\mathcal{M}$ (i.e. 2^p for p variables), whereas `BMA` uses a subset of models based on a leaps and bounds algorithm in conjunction with "Occam's razor", see Hoeting et al. (1999) for more details.

There are several implications by using the above methodology:

- With many variables, the computation time can be (too) large if "all" candidate models are evaluated. A solution to this is restricting the number of candidate models and/or parallelization. Section 6.1 gives an overview on how this can be facilitated. Briefly, by i) restricting candidate models by accessing `dredge` from `MuMIn`, ii) or using `criterion="BIC+"` [the number of candidate models is printed], or iii) using `criterion="BIC+"` and accessing `bic.surv` or `bic.glm` from `BMA` to adjust Occam's window, iv) using the option `candidate.models`, by v) screening variables with option `screen`, or vi) by parallel computing using option `ncores`.

- If there is no proper likelihood function, then it can be argued that using an information type criterion, and therefore criterion based model averaging, does not make sense. In particular, survival models, such as Cox's proportional hazards model, use a *partial* likelihood, due to the censoring inherent in survival data. Nevertheless, a pragmatic solution is to simply adopt the partial likelihood; if BIC is used, one has to decide what n means, i.e. the number of subjects or the number of uncensored subjects, see Hoeting et al. (1999) and Volinsky et al. (1997) for more details. Similarly, the definition of information criteria in mixed models is not entirely clear and we use the implementation from package `lme4`.

- From the Bayesian perspective the quality of a regression model $M_\kappa \in \mathcal{M}$ may be judged upon its estimated posterior probability that this model is correct, that is

$$\Pr(M_\kappa|\mathbf{y}) \quad \propto \quad \Pr(M_\kappa) \int \Pr(\mathbf{y}|M_\kappa, \beta_\kappa) \cdot \Pr(\beta_\kappa|M_\kappa)\, d\beta_\kappa \,,$$

  where $\Pr(M_\kappa)$ is the prior probability for the model M_κ to be correct, $\Pr(\mathbf{y}|M_\kappa, \beta_\kappa) = \mathcal{L}(\beta)$ represents the maximized likelihood, and $\Pr(\beta_\kappa|M_\kappa)$ reflects the prior of β_κ for model M_κ. Since, for a large sample size, $\Pr(M_\kappa|\mathbf{y})$ can be approximated via the Bayes-Criterion of Schwarz (BCS, BIC), it is often suggested that the weight (1) is used for the construction of the Bayesian Model Averaging estimator, but with BIC, instead of AIC. The BCS corresponds to $-2\mathcal{L}(\hat{\beta}) + \ln n \cdot p$, where p corresponds to the number of parameters. Note the following:

  - The `BMA` implementation does not allow the specification of $\Pr(M_\kappa)$ and assumes equal prior probabilities for each model to be correct.
  - Broadly, using option `BIC+` uses variance estimation based on variance decomposition (Draper, 1995) such as the law of total variance, i.e. using

$$\widehat{\mathrm{Var}}(\hat{\bar{\beta}}_j) \quad = \quad \hat{\mathrm{E}}_{\mathcal{M}}(\widehat{\mathrm{Var}}(\hat{\beta}_{j,\kappa}|y, M_\kappa)) + \widehat{\mathrm{Var}}_{\mathcal{M}}(\hat{\mathrm{E}}(\hat{\beta}_{j,\kappa}|y, M_\kappa)) \tag{7}$$

    which is similar, but not identical to (2). This means standard errors and confidence intervals are not identical when comparing options `BIC` and `BIC+` – even if the candidate models are the same (which practically never happens). However, in most situations the confidence intervals will be very close.
  - Using `BIC+` and passing on the option `OR=Inf` (to `bic.glm` or `bic.surv`) means considering *all* candidate models. Can be of interest when considering a full Bayesian approach.
  - For `BIC+`, the variable importance measure displayed by `mami` essentially refers to the posterior effect probability (averaged over the imputed data sets) calculated by `BMA`; see Hoeting et al. (1999), among others, for more details.

- The number of candidate models is $2^p - 1$ if the Cox model ist the analysis model of interest. This is because the "intercept only" model is not evaluated, as there is no natural intercept (estimation of the baseline hazard is treated as a nuisance parameter).

3.1.2 Mallow's Model Averaging

Mallow's model averaging (MMA) refers to the approach described by Hansen (2007). This estimator is an example of *optimal* model averaging, which may be of particular interest from a predictive point of view, rather than explanatory point of view. It is implemented in the functions `mma()` and `jma()`[3] and can be used within `mami()` when the option `method="MMA"` is chosen. It can only be used for the linear model.

[3]The original implementation in `mma` is almost identical to the more recent version in `jma`, where the latter is a robust expansion of Bruce Hansen's file at `http://www.ssc.wisc.edu/~bhansen/progs/ecnmt_07.html` and allows for a non-nested model setup (where computationally feasible). However, the former is a bit more stable and allows variance estimation according to (2) when using option `variance="BA"`.

Hansen considers a situation of k nested linear regression models for k variables. Let $\hat{\tilde{\beta}} = \sum_{\kappa=1}^{k} w_\kappa \hat{\beta}_\kappa$ be a model averaging estimator with $\hat{\mu}_w = X_k \hat{\tilde{\beta}}$. Based on similar thoughts as in the construction of Mallow's C_p, Hansen suggests to minimize the mean squared (prediction) error by minimizing the following criterion:

$$\tilde{C}_p \quad = \quad (y - X_k \hat{\tilde{\beta}})'(y - X_k \hat{\tilde{\beta}}) + 2\sigma^2 K_w \,, \tag{8}$$

where $K_w = \mathrm{tr}\,(P_w)$, $P_w = \sum_{\kappa=1}^{k} w_\kappa P_\kappa$, $P_\kappa = X_\kappa (X_\kappa' X_\kappa)^{-1} X_\kappa'$, and σ^2 is the variance which needs to be estimated from the full model. Consequently, the weights are chosen such that $\tilde{C}_p$ is minimized

$$w_\kappa^{\mathrm{MMA}} \quad = \quad \arg \min_{w_\kappa \in \mathcal{H}} \tilde{C}_p \,, \tag{9}$$

with $\mathcal{H} = \{(w_1, \ldots, w_k) \in [0,1]^k : \sum_{\kappa=1}^{k} w_\kappa = 1\}$. Since the first part of (8) is quadratic in w_κ and the second one linear, one can obtain the model averaging estimator by means of quadratic programming (i.e. the package `quadprog`).

The assumptions of a discrete weight set and nested regression models sound restrictive, but it has been shown that both assumptions are not necessarily required and MMA can be applied to non-nested regression models as well; given that this is computationally feasible (Wan et al., 2010).

Note the following:

- The MMA estimator is based on the weights (9) rather than (1).
- Using `mami` with `method="MMA"` uses the function `jma`[4]. By default, no standard errors and confidence intervals are calculated. This is because the motivation for Mallow's model averaging is prediction (Schomaker and Heumann, 2019). If one is interested in bootstrap standard errors (which may not achieve nominal coverage), then the options `calc.var="boot"` and `bsa` (for the number of bootstrap samples) can be passed on to `jma`. For example, in Example 1 in the `mami` help file one could specify:

```
mami(freetrade_imp, method="MMA",outcome="tariff",
add.factor=c("country"),calc.var="boot",bsa=200)
```

- The `mma` and `jma` functions can be compared easily. An example is

```
data(Prostate)
jma(y=Prostate[,9],x=Prostate[,-9],ma.method="MMA")
mma(Prostate, formula=lpsa .,ycol="lpsa")
```

- In the above described nested model setup there are k candidate models and the MMA estimates depend upon the ordering of these regressors. The non-nested setup allows for the evaluation all 2^k candidate models and is not affected by the ordering of the regressors; however, MMA is much more unstable under such a setup and for large k estimation may not be computationally feasible. The default in `jma` is to use the nested setup (`model.subset="nested"`), but it is also possible to use all candidate models for weight calculation (`model.subset="nested"`):

```
jma(y=Prostate[,9],x=Prostate[,-9],ma.method="MMA",model.subset="all")
```

The function `mma` allows only evaluation of the nested setup. Even in this simpler setting, matrices which are not positive definite yield to the failure of the optimization problem.

[4] since version 0.9.13

Both `mma` and `jma` prevent this by looking for a "close" positive definite matrix that can be used, based on `make.positive.definite` from package `corpcor`, and a warning is printed.

3.1.3 Lasso Averaging

Shrinkage estimation, for example via the LASSO (Tibshirani, 1996), can be used for model selection. This requires the choice of a tuning parameter which comes with tuning parameter selection uncertainty. LASSO averaging estimation (LAE), or more general shrinkage averaging estimation (Schomaker, 2012), is a way to combine shrinkage estimators with different tuning parameters. This is implemented in `MAMI` in the `lae` function and can be used in `mami()` by calling the option `method="LAE"`.

Consider the LASSO estimator for a simple linear model:

$$\hat{\beta}_{\text{LE}}(\lambda) \quad = \quad \arg\min \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} . \tag{10}$$

The complexity parameter $\lambda \geq 0$ tunes the amount of shrinkage and is typically estimated via the generalized cross validation criterion (GCV) or any other cross validation criterion (CV). The larger the value of λ, the greater the amount of shrinkage since the estimated coefficients are shrunk towards zero.

Consider a sequence of candidate tuning parameters $\boldsymbol{\lambda} = \{\lambda_1, \ldots, \lambda_L\}$. If each estimator $\hat{\beta}_{\text{LE}}(\lambda_i)$ obtains a specific weight w_{λ_i}, then a *LASSO averaging estimator* takes the form

$$\hat{\bar{\beta}}_{\text{LAE}} \quad = \quad \sum_{i=1}^{L} w_{\lambda_i} \hat{\beta}_{\text{LE}}(\lambda_i) = \mathbf{w}_\lambda \hat{\boldsymbol{B}}_{\text{LE}} \, , \tag{11}$$

where $\lambda_i \in [0, c]$, $c > 0$ is a suitable constant, $\hat{\boldsymbol{B}}_{\text{LE}} = (\hat{\beta}_{\text{LE}}(\lambda_1), \ldots, \hat{\beta}_{\text{LE}}(\lambda_L))'$ is the $L \times p$ matrix of the LASSO estimators, $\mathbf{w}_\lambda = (w_{\lambda_1}, \ldots, w_{\lambda_L})$ is an $1 \times L$ weight vector, $\mathbf{w}_\lambda \in \mathcal{W}$ and $\mathcal{W} = \{\mathbf{w}_\lambda \in [0,1]^L : \mathbf{1}'\mathbf{w}_\lambda = 1\}$.

One could choose the weights

$$\hat{\mathbf{w}}_\lambda^{\text{OCV}} \quad = \quad \arg \min_{\mathbf{w}_\lambda \in \mathcal{W}} OCV_k \tag{12}$$

with

$$\begin{aligned} OCV_k \quad &= \quad \frac{1}{n} \tilde{\boldsymbol{\epsilon}}_\kappa(w)' \tilde{\boldsymbol{\epsilon}}_\kappa(w) \\ &\propto \quad \mathbf{w}_\lambda \mathbf{E}_k' \mathbf{E}_k \mathbf{w}_\lambda{}' \, , \end{aligned} \tag{13}$$

referring to an optimal cross validation (OCV) based criterion and $\mathbf{E}_k = (\tilde{\boldsymbol{\epsilon}}_k(\lambda_1), \ldots, \tilde{\boldsymbol{\epsilon}}_k(\lambda_L))$ is the $n \times L$ matrix of the cross-validation residuals for the L competing tuning parameters (given a specific loss function). More details can be found in Schomaker (2012). Using LAE with `method="LAE"` has the following implications:

- LAE can be used for the linear model, the logistic model and the Poisson model – but not for longitudinal or survival data.
- The model averaging estimator is based on the weights (12) rather than (1).
- The variance of the LAE estimator is calculated according to (2), but the candidate models refer to the different choices of the tuning parameter. The variance of each single LASSO estimator $\hat{\beta}_{\text{LE}}(\lambda_i)$ is based on bootstrapping. The default is 100 bootstrap samples as not for all problems confidence intervals of the LASSO are of interest. The option `B.var` (in `lae`, or called from `mami`) can be used to adjust the number of bootstrap

samples. Note that the LASSO estimator is a shrinkage estimator which trades decreased variance with increased bias. Therefore simple bootstrap based confidence intervals will not achieve nominal coverage (a warning is printed), and using it for model averaging (another shrinkage estimator) will not change this.

- The LASSO implementation `cv.glmnet` from package `glmnet` is used. Therefore, any arguments can passed to this function: for example `alpha` (to call the Elastic Net or Ridge estimator [see Example 4, `?mami`]; or the λ sequence (`nlambda`).

- The k of k-fold cross-validation (for the definition of OCV_k) can be specified with the option `kfold` [both in `lae` and `mami`]. The default split of the data for cross validation is based on the sequence $1, 2, ..., k, 1, 2, ...$ The option `cvr=T` randomly shuffles the split. The default `cvr=F` is meant to make results reproducible and comparable when using both `method="LAE"` and `method="LASSO"`.

- Note that the deviance is used as the loss function to calculate the cross validation residuals for logisitic and Poisson regression models. However, the weights for Lasso averaging have to be determined based on a squared loss function.

- Example 3 under `?mami` as well as the examples under `?lae` give examples of the use of LAE.

3.1.4 Jackknife Model Averaging

Jackknife Model Averaging (JMA), as suggested in Hansen and Racine (2012) for linear models, has been implemented in the function `jma` and builds on leave-one-out (LOO) cross validation. JMA is implemented in `MAMI` in the `jma` function and can be used in `mami()` by calling the option `method="JMA"`.

JMA works as follows: for Model M_κ the LOO residual vector is $\tilde{\epsilon}^\kappa = y - \hat{y}^\kappa$, with $\hat{y}_i^\kappa = x_i^\kappa (X_{(-i)}^{\kappa'} X_{(-i)}^\kappa)^{-1} X_{(-i)}^{\kappa'} y_{(-i)}$ where the index $_{(-i)}$ describes that the respective matrix excludes observation i, $i = 1, \ldots, n$. It can be shown that there is a simple algebraic relationship which allows the computation of the LOO residuals in one rather than n operations:

$$\tilde{\epsilon}^\kappa = D_\kappa \hat{\epsilon}^\kappa \tag{14}$$

where $\hat{\epsilon}^\kappa$ is the standard least squares residual vector $y - P_\kappa y$ with the hat matrix $P = X(X'X)^{-1}X'$; and D_κ is a $n \times n$ diagonal matrix with $D_{ii,\kappa} = (1 - P_{ii,\kappa})^{-1}$, $i = 1, ..., n$.

For k candidate models, which may be nested or not (as in the MMA-setup described in Section 3.1.2), the linear weighted LOO residuals are $\tilde{\epsilon}_w = \sum_\kappa w_\kappa \tilde{\epsilon}^\kappa$, $\kappa = 1, \ldots, k$. An estimate of the true expected squared error is $\text{CV}_w = n^{-1} \tilde{\epsilon}_w' \tilde{\epsilon}_w$ and an appropriate weight choice would thus be

$$w^{\text{JMA}} \quad = \quad \arg\min_{w \in \mathcal{H}} \text{CV}_w \,, \tag{15}$$

As with MMA, the weights can be obtained with quadratic programming. Similar considerations as with MMA apply:

- The JMA estimator is based on the weights (15) rather than (1).

- Using `mami` with `method="JMA"` uses the function `jma`. By default, no standard errors and confidence intervals are calculated. This is because the motivation for Jackknife model averaging is prediction (Schomaker and Heumann, 2019). If one is interested in bootstrap standard errors (which may not achieve nominal coverage), then the options `calc.var="boot"` and `bsa` (for the number of bootstrap samples) can be passed on to `jma` (see example in Section 3.1.2).

- In the nested model setup there are k candidate models and the JMA estimates depend upon the ordering of these regressors. The non-nested setup allows for the evaluation all 2^k candidate models and is not affected by the ordering of the regressors; however, JMA is much more unstable under such a setup and for large k (say > 20) estimation may not be computationally feasible. The default in `jma` is to use the nested setup (`model.subset="nested"`), but it is also possible to use all candidate models for weight calculation (`model.subset="nested"`), see example in Section 3.1.2.

3.2 Model Selection

Model selection means assigning a weight of 1 to the model which is optimal with respect to a specific criterion. In many situations it is likely that there is model selection uncertainty in the sense that different samples would lead to different model choices: sometimes a variable is included, and sometimes not. Sometimes, this can make the distribution of parameter estimates bimodal (Schomaker and Heumann, 2014) – and bootstrap based confidence intervals (Section 4) together with graphical summaries (Section 5) can therefore be a good choice when applying model selection.

3.2.1 Criterion Based Model Selection

`MAMI` offers model selection based on the following options:

`method="MS.criterion"` [or sometimes `method="MA.criterion"`] and
`criterion="AIC"` or `criterion="BIC"` or `criterion="GCV"` or `criterion="CV"`.

- `criterion="AIC"` chooses the model with the smallest AIC based on a stepwise search with `stepAIC` from `MASS`. Note that the number of candidate models is therefore not 2^p as with `method="MA.criterion"` [where model selection results are displayed in addition to model averaging results]. This means that model selection results can potentially differ between the two approaches. The latter is preferred if computationally feasible. For longitudinal data, AIC based model selection can only be utilized with `method="MA.criterion"` and not with the quicker `method="MS.criterion"`.

- `criterion="BIC"` chooses the model with the smallest BIC based on a stepwise search with `stepAIC` from `MASS`. The same considerations as above apply here as well, i.e. with respect to longitudinal data and potentially differing results to those displayed after model averaging.

- For both AIC and BIC the considerations from Section 3.1.1 apply. For example, the critique of these criteria in survival analysis and mixed models.

- `criterion="CV"` together with `kfold` uses the k-fold cross validation error for model selection based on a squared loss function. This approach is implemented using `dredge` in `MuMIn`, and therefore all 2^p models are being evaluated. This may be time-consuming and `criterion="GCV"` (see below) is an alternative. The default split of the data for cross validation is based on the sequence $1, 2, ..., k, 1, 2, ...$ The option `cvr=T` randomly shuffles the split. Cross validation can not be used for survival models. It can be used for longitudinal data, i.e. in the context of mixed models, but one needs to be aware that the cross validation predictions are using the average intercept within each subject.

- `criterion="GCV"` uses generalized cross validation, i.e. an approximation to leave-one-out cross-validation, for model selection and is much quicker than `criterion="CV"`. It can't be used for survival data, but for longitudinal data. Again, the implementation utilizes `dredge` from `MuMIn` and thus all 2^p candidate models are being evaluated.

3.2.2 LASSO and Shrinkage Based Model Selection

Model selection can be done with the LASSO estimator as introduced in (10). This can be achieved with the following option:

`method="LASSO"` and `kfold`

As opposed to LASSO averaging, LASSO based model selection can be used for not only the linear, logistic and Poisson model, but also for the Cox proportional hazards model. The implementation is based on `cv.glmnet` from package `glmnet`.

Essentially the same considerations as in Section 3.1.3 apply. Briefly:

- The variance of each single LASSO estimator is based on bootstrapping, with a default of $\max(\texttt{B}, 100)$ bootstrap samples. This can be lowered in future releases, as the standard error may not necessarily be of interest.
- One might argue that confidence intervals for the LASSO estimator are not meaningful. For sure, they are not corrected for the bias introduced by shrinkage and won't achieve nominal coverage.
- The data is split into a sequence of $1, 2, ..., k, 1, 2, ...$ and the tuning parameter which minimizes the k-fold cross validation error (chosen via `kfold`) is being used. The option `cvr=T` randomly shuffles the split.
- The loss function to calculate the cross validation error is the squared error loss for linear models, the deviance for logistic and Poisson models, and the partial likelihood for the Cox model.
- To use general Elastic Net type model selection pass on `alpha` to `cv.glmnet`, e.g. `alpha=0.5`.

4 Inference

In general, confidence intervals based on (6) are calculated (or they are based on the very similar (7) if `criterion="BIC+"`)[5]. Schomaker and Heumann (2014, Table 3) show that these intervals can work quite well for criterion based model averaging estimators; however, they will underestimate the variance for model selection estimators. Bootstrap confidence intervals can help to improve the coverage of model selection estimators. Moreover, distributions post model selection and post model averaging are often non-symmetric (Hjort and Claeskens, 2003, Schomaker and Heumann, 2014, Fig. 2-4) which is another motivation for bootstrapping. When applying bootstrapping one has of course to impute the data in each bootstrap sample. Bootstrap model selection/averaging confidence intervals, after imputation, can thus be generated as follows:

1)	Create B bootstrap samples of the original data (including missing observations)
2)	Generate M imputed sets of data for each bootstrap sample
3)	In each bootstrap sample calculate a model averaging (selection) estimator of the regression parameters using equation (5) – based on the application of a particular model averaging/selection scheme on the multiply imputed data
4)	Construct $1 - \alpha$ confidence intervals based on the $\alpha/2$ and $1 - \alpha/2$ percentiles of the empirical distribution of the B point estimates produced in step 3

[5] If `method="MMA"` or `method="JMA"` no confidence intervals are calculated by default; see Sections 3.1.2 and 3.1.4 on details how to calculate confidence intervals with these methods. Note that also for these methods (additional) bootstrap confidence intervals as discussed in this Section can, in principle, be calculated (though they may not always be meaningful as MMA and JMA are pure prediction methods).

If the option `inference="+boot"` is chosen, confidence intervals according to the above algorithm are being generated in addition to the standard confidence intervals from (6). If computationally feasible, it is recommended to implement this approach and plot the results (see also Section 5.1).

- The option `inference="+boot"` needs to be complemented with the original unimputed data (option `X.org`) [because of step 2] and the number of bootstrap samples to be drawn (option `B`) .
- The point estimates reported from `mami` are still the point estimates according to (5), and not the arithmetic mean of the bootstrap samples as suggested by Table 1 in Schomaker and Heumann (2014), though the latter are reported by `print.mami` as well.
- By default a 95% confidence interval is reported, but this can be changed with the option `CI`.
- Confidence intervals are based on a t_R distribution as explained in Section 1.3.2. If $M = 1$, for example under no missing data or a complete case analysis, R is assumed to be infinity under model averaging and thus a standard normal distribution is used. For model selection, an appropriate t-distribution is used, if meaningful for the respective model class.
- More about combining bootstrapping and multiple imputation can be found in Schomaker and Heumann (2018).

5 Analysis and Interpretation

5.1 Example

For a reasonably realistic example one could look at the hypothetical HIV data set provided by `MAMI`. It contains typical variables of HIV treatment research such as follow-up time (`futime`), event of death (`dead`), baseline variables such as CD4 count (`cd4`), WHO stage (`stage`), weight (`weight`), and many others. The data looks as follows:

patient	hospital	futime	dead	sex	age	cd4	cd4slope6	weight	period	haem	stage	tb	cm
1	3	58	0	0	39	128	1.8887	65	2004	13.2184	2	0	0
2	3	1717	1	1	38	77	-6.5082	NA	2001	10.7261	3	0	NA
3	3	1941	0	1	34	124	12.0466	62	2001	NA	3	NA	0
4	3	512	0	1	22	147	8.9213	63	2004	NA	2	0	0
5	3	766	0	1	28	187	4.6059	NA	2004	12.7076	3	0	NA
6	3	2242	0	1	36	10	8.8271	NA	2001	9.4803	3	NA	NA

Since data of CD4 count, haemoglobin, WHO stage, tuberculosis and cryptococcal meningitis (all at baseline) are missing, one could impute the data under a missing at random assumption. This could be done with `Amelia II` in R, see Honaker et al. (2011) for details.

```
library(MAMI)
library(Amelia)
data(HIV)
HIV_imp <- amelia(HIV, m=5, idvars="patient",logs=c("futime","cd4"),
    noms=c("hospital","sex","dead","tb","cm"), ords=c("period","stage"),
    bounds=matrix(c(3,7,9,11,0,0,0,0,3000,5000,200,150),ncol=3,nrow=4))
```

To select a model with AIC on the multiply imputed data, we could simply pass it to `mami`. Picking the options `model="cox"` and `outcome=c("futime","dead")` makes it clear that we are interested in risk factors for the hazard of death, modeled by a Cox proportional hazards model. With `method="MS.criterion"` and `criterion="AIC"` we make clear that

we are interested in model selection with AIC. To ensure that the categorical variables are treated as such we use `add.factor`. Now, we may already know that the influence of CD4 count on mortality is typically squared, and that haemoglobin levels mean different things for men and women, and therefore add the respective transformations and interactions with `add.transformation` and `add.interaction`. Two variables (patient identifier, and average CD4 slope) may not be of interest and be removed. We want the results to be reported as hazard ratios (rather than coefficients), and thus use `report.exp=T`. The code looks as follows.

```
# Model selection after imputation
mami(HIV_imp, model="cox", outcome=c("futime","dead"), method="MS.criterion",
    criterion="AIC", add.factor=c("period","hospital","stage"),
    add.transformation=c("cd4^2"), add.interaction=list(c("haem","sex")),
    report.exp=TRUE, var.remove=c("patient","cd4slope6"))
```

To get a sense how results would vary if we selected the model with LASSO rather than AIC, we simply replace `method="MS.criterion"` with `method="LASSO"`.

```
# Model selection after imputation with LASSO
mami(HIV_imp, model="cox", outcome=c("futime","dead"), method="LASSO",
    add.factor=c("period","hospital","stage"),
    add.transformation=c("cd4^2"), add.interaction=list(c("haem","sex")),
    report.exp=TRUE, var.remove=c("patient","cd4slope6"))
```

Both methods suggest to pick all variables, but at the same time confidence intervals for some variables are very wide. Let's say we are interested in model selection uncertainty, therefore use model averaging, but with the more parsimonious BIC instead. We may pick `BIC+` for a reduced set of candidate models and therefore quicker results.

```
# Model averaging after imputation
mami(HIV_imp, model="cox", outcome=c("futime","dead"), method="MA.criterion",
    criterion="BIC+", add.factor=c("period","hospital","stage"),
    add.transformation=c("cd4^2"), add.interaction=list(c("haem","sex")),
    report.exp=TRUE, var.remove=c("patient","cd4slope6"))
```

The output looks as follows:

```
Estimates for model averaging:

                    Estimate Lower CI  Upper CI
factor.hospital..4        --       --        --
factor.hospital..5        --       --        --
sex                 0.914877  0.75895  1.102839
age                    1.024 0.977467  1.072749
cd4                 0.998027 0.994104  1.001965
wt                  0.974976 0.927272  1.025134
factor.period..2004 0.999213 0.993614  1.004844
factor.period..2007 0.999764 0.998083  1.001448
haem                0.890771 0.707211  1.121974
factor.stage..3     1.425054 0.651179  3.118619
factor.stage..4      2.40972 0.420057 13.823716
tb                  1.445979 0.600062  3.484396
cm                  1.011204 0.957709  1.067687
I.cd4.2.                   1 0.999999  1.000002
sex.haem            0.988829 0.964419  1.013856

Estimates for model selection:
```

```
                   Estimate Lower CI  Upper CI
factor.hospital..4       --       --        --
factor.hospital..5       --       --        --
sex                 0.94812 0.613547   1.46514
age                1.023815 0.977524  1.072298
cd4                 0.99802 0.994124  1.001931
wt                  0.97507 0.927527  1.025051
factor.period..2004      --       --        --
factor.period..2007      --       --        --
haem               0.892626 0.710438  1.121534
factor.stage..3    1.414049 0.641343  3.117733
factor.stage..4    2.385434 0.423874 13.424494
tb                  1.44305 0.516333  4.033044
cm                       --       --        --
I.cd4.2.                 --       --        --
sex.haem           0.984011 0.925989  1.045668

Posterior effect probabilities:

 age  factor.stage.     wt  haem    cd4    tb  sex.haem
1.00           1.00   1.00  1.00   0.90  0.82      0.42

 sex   I.cd4.2.    cm   factor.period. factor.hospital.
0.35       0.14  0.03             0.00             0.00
```

One can see from the results of both model selection and averaging that the variable `hospital` is not picked and has a variable importance of 0. Some of the variables which had wider confidence intervals with AIC, are now not being selected anymore by BIC (e.g. cryptococcal meningitis and $CD4^2$). The variable importance (i.e. posterior effect probabilities) suggest that there is some uncertainty around variables such as the added interaction, or sex.

So, what to do? We could repeat the analysis with bootstrap confidence intervals to get a better sense of the model selection and model averaging results. Therefore we use the options `inference="+boot", B=200, X.org=HIV`. We know that biologically it makes sense that co-infections are relevant to predict mortality but that the associations in the data are maybe not strong enough to immediately pick them. Thus, we could opt to do model averaging with AIC rather than BIC, because the latter has a more parsimonious approach. Knowing that `mami` utilizes `dredge` from `MuMIn`, we could pass on the option `subset` to specify that candidate models should contain squared CD4 count only if linear CD4 count is contained, and that the interaction of sex and haemoglobin should only be contained in the models where the main effect is contained too. This can be utilized with dependency chains (`dc`), see `?dredge` for help. Then, the syntax looks as follows:

```
# Model averaging after imputation + bootstrapping
m1 <- mami(HIV_imp,
   model="cox", outcome=c("futime","dead"),
   method="MA.criterion", criterion="AIC",
   add.factor=c("period","hospital","stage"),
   add.transformation=c("cd4^2"), add.interaction=list(c("haem","sex")),
   report.exp=TRUE, var.remove=c("patient","cd4slope6"),
   inference="+boot", B=200, X.org=HIV,
   subset = dc("cd4","I(cd4^2)") && dc("sex","sex:haem")
            && dc("haem","sex:haem"),
   print.time=TRUE)
summary(m1)
plot(m1)
```

```
> summary(m1)

...

Estimates for model selection (based on 200 bootstrap samples):

                    Estimate      LCI       UCI Boot LCI Boot UCI   VI
age                 1.023907 0.977561  1.072449 1.011644 1.034206    1
cd4                 0.997417 0.991887  1.002979 0.994914 0.999347 0.99
cm                  1.078447 0.581898  1.998714 0.796985 1.881647 0.36
factor(hospital)4         --       --        -- 0.678480 1.000000 0.23
factor(hospital)5         --       --        -- 0.783300 1.143629   --
factor(period)2004  0.760642 0.444489  1.301663 0.586118 1.000000  0.7
factor(period)2007  0.908518 0.740873  1.114098 0.545100 1.424140   --
factor(stage)3      1.382479 0.647652  2.951041 0.961486 1.961763    1
factor(stage)4      2.286076 0.440729 11.857959 1.520232 3.193898   --
haem                0.879414 0.680912  1.135784 0.830303 0.953194    1
haem:sex                  --       --        -- 0.898716 1.016400 0.31
I(cd4^2)            1.000002 0.999994  1.000009 1.000000 1.000005 0.58
sex                 0.763531 0.448949  1.298543 0.591503 2.268933 0.94
tb                  1.514913 0.636557  3.605274 1.044047 1.903004 0.94
wt                  0.975424  0.92847  1.024753 0.965503 0.985913    1
```

The results we got a now much more nuanced: estimates of coefficients from variables we were unsure about, have non-normal distributions. For example, the 95% bootstrap confidence interval of the hazard ratio of cryptococcal meningitis is bimodal, see also Figures 1a and 1b [which are provided by `plot(m1)`]. Comprehensive model selection and averaging with AIC suggests a moderate effect of variables such as the transformation, but no major role of the interaction. This would have potentially remained undiscovered without bootstrapping. Figure 1 shows that from an explorative perspective post-model averaging/selection plots can help us to get us a better sense of which variables are potentially relevant and which not.

Given the variety of options how to perform and report model selection and averaging results, we give some guidance in the next section on how to report results.

5.2 Suggested Reporting in Publications

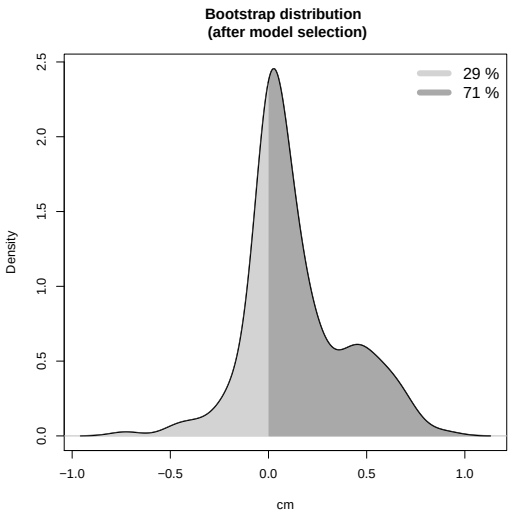
By default, `print.mami` lists a big variety of results and all of them could be reported. However, a more concise approach may be better understood.

Since model averaging is not always well-known, as opposed to model selection, a good option could be to report i) the results of the full model (without model selection, as estimates are consistent), ii) the results after the preferred model selection procedure has been applied – together with bootstrap confidence intervals (crucial, such that the intervals reflect model selection uncertainty), and iii) the variable importance measure related to the weights of model averaging (to get a sense of model selection uncertainty). A template is given in Table 1 (use `summary()`) .

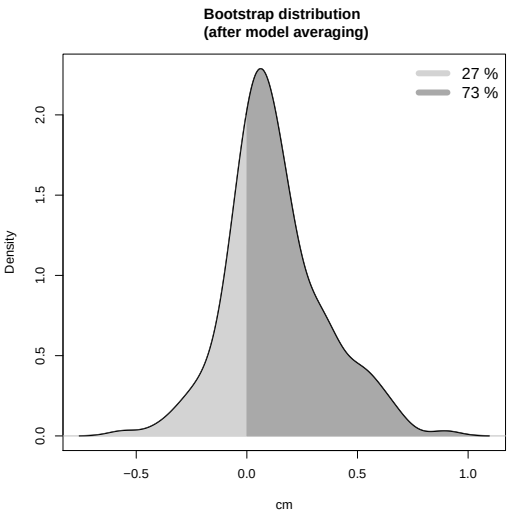
For example, the results reported in Porter et al. (2015) – see Figure 2 – are very similar to the suggested template, except that confidence intervals are not based on bootstrapping and univariate results are reported as well.

Karamchand et al. (2016) do not report results from model averaging, e.g. via variable importance measures, but are otherwise similar in their approach on how to report results of model selection after imputation; see Figure 3.

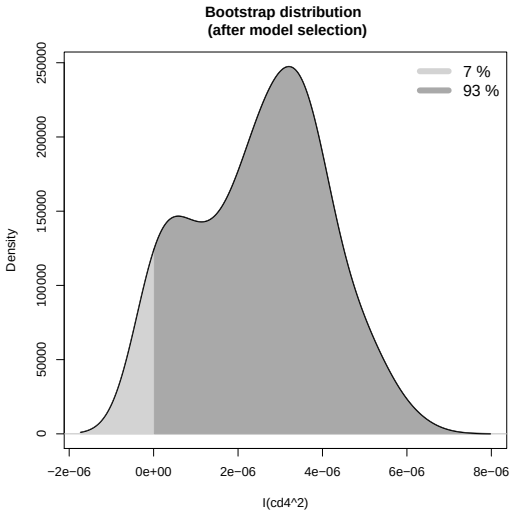
Another good example is the table from Visser et al. (2012) [Figure 4], where Bayesian posterior effect probabilities (based on Bayesian Model Averaging after imputation) are given, but no results of model selection.



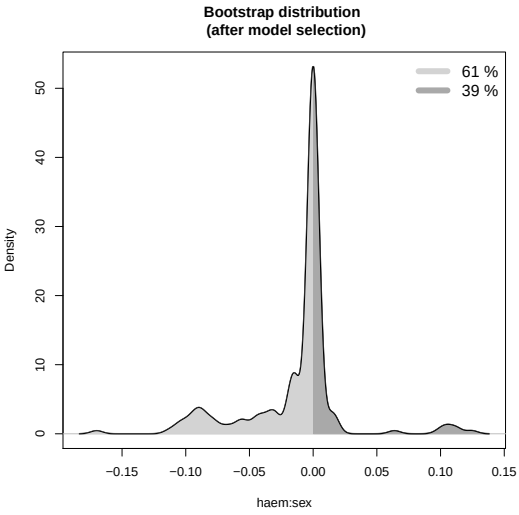
(a) cryptococcal meningitis



(b) cryptococcal meningitis



(c) squared CD4



(d) sex:haemoglobin (interaction)

Figure 1: Bootstrap distribution, after model selection and model averaging

Table 1: A template to report results from `mami` (use `summary()`)

	Full Model		Model Selection		
Variable	β	95% CI	β	95% CI	VI
Variable 1					
Variable 2					
Variable 3					
Variable 4					
Variable 5					
⋮	⋮	⋮	⋮	⋮	⋮

[†]Footnotes on screening, model assumptions etc.

TABLE 3. Survival Analysis of Imputed Data: Cox Regression for Predictors of Mortality Stratified by Cohort

	Univariate			Multivariate*			Model Selection †		
Variable	HR	*P*	95% CI	HR	*P*	95% CI	HR	95% CI	VI
Female gender	0.97	0.716	0.83 to 1.14	—	—	—	—	—	0.31
Age at initiation, mo									
0–2	Reference			Reference			Reference		
3–5	0.90	0.417	0.71 to 1.15	0.87	0.268	0.68 to 1.11	—	—	0.22
6–11	0.98	0.898	0.78 to 1.25	0.84	0.161	0.66 to 1.07	—	—	—
Nonsevere immune suppression (WHO 2006)	Reference			Reference			Reference		
Severe immune suppression (WHO 2006)	2.51	0.000	1.66 to 3.79	2.19	0.000	1.44 to 3.33	2.15	1.42 to 3.27	1
WHO stage 1 or 2	Reference			Reference			Reference		
WHO stage 3 or 4	1.89	0.000	1.47 to 2.45	1.36	0.023	1.04 to 1.78	1.35	1.04 to 1.77	0.87
Mild or moderate anemia (DAIDS 2009)	Reference			Reference			Reference		
Severe anemia (DAIDS 2009)	1.57	0.003	1.18 to 2.10	1.34	0.062	0.98 to 1.82	1.29	0.82 to 2.05	0.79
WAZ category									
Greater than −2	Reference			Reference			Reference		
−2 to −3	1.40	0.015	1.07 to 1.84	1.29	0.063	0.99 to 1.71	1.29	0.99 to 1.71	1
Lesser than −3	2.55	0.000	2.04 to 3.19	2.23	0.000	1.78 to 2.80	2.22	1.78 to 2.79	—
ART initiation before 2010	Reference			Reference			Reference		
ART initiated from the start of 2010	0.65	0.000	0.52 to 0.83	0.75	0.015	0.59 to 0.94	0.75	0.59 to 0.95	0.88
‡Viral load ≤1 million copies/mL	Reference			Reference			Reference		
‡Viral load >1 million copies/mL	1.30	0.045	1.01 to 1.68	1.17	0.267	0.88 to 1.56	1.14	0.79 to 1.62	0.56

*Adjusted for age, weight-for-age category, severe immunosuppression, WHO stage 3 or 4, severe anemia, and initiation from the start of 2010.
†Using AIC and model averaging.
‡Modeled on a subset of infants from South African cohorts, adjusted for variables as above.
CI, confidence interval; DAIDS, Division of AIDS; HR, hazard ratio; VI, variable importance; WAZ, weight-for-age z-score; WHO, World Health Organization.

Figure 2: Table 3 from Porter et al. (2015)

TABLE 3. Univariate, Multivariate, and Model Selection Results for the Cox Regression Model of Associations With Incident Diabetes

		Univariate		Multivariate		
Variable	Category	HR (95% CI)	*P*	HR (95% CI)	*P*	Model Selection (AIC)
Nonnucleoside reverse transcriptase inhibitor	Efavirenz	1.40 (1.22–1.60)	<0.001	1.27 (1.10–1.47)	0.001	1.27 (1.10–1.46)
	Nevirapine	Referent		Referent		Referent
Nucleoside reverse transcriptase inhibitor	Zidovudine	1.30 (1.15–1.46)	<0.001	1.35 (1.19–1.52)	<0.001	1.37 (1.21–1.54)
	Stavudine	1.53 (1.32–1.78)	<0.001	1.60 (1.38–1.87)	<0.001	1.64 (1.41–1.91)
	Other	Referent		Referent		Referent
Exposure to other diabetogenic drugs		1.68 (1.51–1.86)	<0.001	1.53 (1.37–1.70)	<0.001	1.53 (1.38–1.71)
Baseline age (yr)	19–24	0.35 (0.20–0.60)	<0.001	0.47 (0.27–0.81)	0.007	0.46 (0.27–0.80)
	25–34	0.64 (0.56–0.73)	<0.001	0.71 (0.62–0.82)	<0.001	0.71 (0.62–0.81)
	35–44	Referent		Referent		Referent
	45–54	1.50 (1.33–1.70)	<0.001	1.38 (1.21–1.56)	<0.001	1.36 (1.20–1.54)
	≥55	1.86 (1.50–2.31)	<0.001	1.64 (1.32–2.04)	<0.001	1.57 (1.26–1.95)
Sex	Male	1.41 (1.27–1.56)	<0.001	1.47 (1.32–1.64)	<0.001	1.44 (1.29–1.61)
	Female	Referent		Referent		Referent
Baseline body mass index (BMI) quartile (kg/m/m)	10–17	0.38 (0.23–0.63)	0.001	0.33 (0.19–0.56)	0.001	0.32 (0.19–0.55)
	18–24	0.65 (0.58–0.74)	<0.001	0.61 (0.53–0.69)	<0.001	0.60 (0.53–0.69)
	25–34	Referent		Referent		Referent
	35+	1.45 (1.16–1.81)	0.002	1.58 (1.26–1.97)	<0.001	1.58 (1.27–1.97)
Baseline CD4 count (cells/μl)	0–199	1.04 (0.92–1.17)	0.534	1.08 (0.95–1.23)	0.220	
	200–349	Referent		Referent		Excluded by AIC
	350+	1.25 (1.06–1.47)	0.007	1.10 (0.91–1.34)	0.324	
Baseline viral load (copies/ml)	0–999	1.31 (1.14–1.51)	<0.001	1.24 (1.05–1.47)	0.011	1.28 (1.11–1.47)
	1000–99,999	Referent		Referent		Referent
	100,000–999,999	1.07 (0.95–1.21)	0.261	1.03 (0.91–1.16)	0.674	1.03 (0.91–1.17)
	≥1,000,000	1.15 (0.89–1.48)	0.285	1.15 (0.89–1.49)	0.278	1.14 (0.89–1.48)

Drug switches within first-line regimen included in the model, with censoring at switch to second line therapy. All results are based on multiple imputations.
AIC = Akaike Information Criterion, CI = confidence interval, HR = hazard ratio.

Figure 3: Table 3 from Karamchand et al. (2016)

Table 2. Cox proportional hazards regression analysis of baseline variables associated with sputum culture conversion after multiple imputation*.

	Unadjusted Hazard ratio (95% CI)*	Adjusted Hazard ratio (95%CI)*	Posterior Effect Probability#
Age	0.99 (0.98–1.02)	0.98 (0.94–1.02)	1.1
Male sex	1.06 (0.57–1.97)	2.38 (0.88–6.25)	14.04
HIV-positive	1.62 (0.67–3.92)	0.65 (0.14–3.12)	0.43
Time to culture detection (days)	1.09 (1.03–1.16)	1.11 (1.02–1.2)	80.08
Sputum smear grading	0.79 (0.55–1.13)	0.75 (0.48–1.18)	10.08
Presence of lung cavities	0.32 (0.12–0.81)	0.13 (0.02–0.95)	87.61
No. of lung zones affected by cavities	0.92 (0.70–1.22)	0.99 (0.54–1.83)	9.52
W-Beijing genotype	0.62 (0.34–1.10)	0.51 (0.25–1.07)	41.24
Ever smoker	0.45 (0.25–0.82)	0.32 (0.1–1.02)	91.52
Alcohol misuse	1.13 (0.64–1.99)	1.67 (0.73–3.79)	10.97
Body Mass Index (kg/m^2)	1.03 (0.94–1.13)	1.11 (0.96–1.31)	0.07
Haemoglobin (g/dl)	0.86 (0.71–1.03)	0.77 (0.58–1.03)	7.89
Albumin (g/l)	0.96 (0.90–1.03)	1.01 (0.89–1.15)	1.78
C Reactive Protein (CRP) (mg/l)	1.01 (0.99–1.01)	1.01 (0.99–1.02)	5.51
Change in CRP (baseline to week 2)	1.01(0.99–1.02)	0.99 (0.97–1.02)	0.98
Total Lymphocyte count ($\times 10^9$/l)	0.86 (0.54–1.37)	0.81 (0.42–1.56)	5.42

*Likelihood of sputum clearance per unit change in predictor variable.
#Posterior effect probability after Bayesian Model averaging; this is the posterior probability that the Hazard in the Cox regression model for a variable is not one, taking model selection uncertainty into account.
doi:10.1371/journal.pone.0029588.t002

Figure 4: Table 2 from Visser et al. (2012)

6 Miscellaneous

6.1 Computation Time

In many settings, the computation time can be long; particularly when performing model averaging. When applying bootstrapping to obtain confidence intervals, estimation can take even longer. To decrease computation time several options are possible. each of these options either decreases the number of candidate models and/or parallelize the computations.

(i) If model averaging or model selection is based on a criterion, and `mami` accesses `dredge` from package `MuMIn` as described in Section 3.1.1 and Section 3.2.1, then restrictions on the number of candidate models can be passed on to `dredge`. Type `?dredge` to learn more, and possibly see Example 4 in `?mami` and the last example in Section 5.1.

(ii) For criterion based model averaging (or selection with cross validation) as described in (i), the option `candidate.models` can be used as well. This is a simple wrapper to reduce the maximum amount of variables in the candidate models to half of all variables (`candidate.models="restricted"`) or a fourth of all variables (`candidate.models = "very restricted"`). Should be used only if one is sure that this is appropriate.

(iii) As indicated above, the option `criterion="BIC+"` utilizes Bayesian Model Averaging by calling the package `BMA`. This implementation does not evaluate all candidate models, but uses a branch-and-bound algorithm to reduce the number of candidate models. The number of candidate models (in each imputed dataset) is printed by `mami`. Thus, `criterion="BIC+"` is a viable alternative to `criterion="BIC"` if the number of candidate models is large.

(iv) Similarly, when using `criterion="BIC+"` and therefore accessing `bic.surv` or `bic.glm` from `BMA`, one can adjust Occam's Window (Hoeting et al., 1999) using the option `OR`, i.e. based on the ratio between the posterior model probabilities of the "best" model and the candidate model under consideration, models with ratios greater than C are rejected.

Since within `BMA`'s branch-and-bound algorithm rejection of a model means also that nested submodels can be rejected, this leads to reduced complexity. The default of C is 20, and a reduction down to 5 can possibly be o.k. in complex settings. See Hoeting et al. (1999, Section3) for more details, and Example 4 from `?mami` for a practical example.

(v) For a very large amount of variables a pre-screening step can be an option. This implies that the model selection (averaging) estimator is then conditional on this pre-screening step. An efficient way to screen variables is to use the LASSO estimator as introduced in (10). This is implemented in the package by the option `screen`. One has to specify the amount of variables that should be excluded before model selection/averaging after imputation is utilized. Then, based on the LASSO path an appropriate amount of variables are excluded and `mami` reports which one.

- Screening is done on the first imputed data set.
- `mami` deals automatically with the consequences related to suggested interactions, transformations etc.
- Screening works for all model classes, but only for cross-sectional data.
- Screening might not work in conjunction with complex sub-options, for example when accessing `subset` in `dredge`.
- Screening is recommended if the amount of variables is very large. For example, when dealing with 100 variables, and having the knowledge that only a small subset is likely relevant, one could screen away 60 variables and then use `criterion="BIC+"` afterwards.
- If screening is utilized, it is suggested to report the excluded variables in a footnote.

(vi) The best option to save time is parallelization using the option `ncores`. One has to simply supply the number of available cores (threads) and `mami` automatically parallelizes parts of the code to speed up calculations.

- Note that this (experimental) options is still in its infancy. It has been tested heavily, but mostly on Windows machines. Please report any bugs.
- There will be no speed up for very simple problems, or the calculation may even take longer. However, you can expect a considerable speed up if
  i) you have multiply imputed data sets (i.e. $M >> 1$) and/or
  ii) you are using option `inference="+boot"` for bootstrap confidence intervals (except in conjunction with `mice`) or
  iii) $M = 1$ and model averaging is criterion based (`"MA.criterion"`) and very complex (many models and/or complex models).
- In the example from Section 5.1, utilizing 7 cores on a Window's machine improved the computation time by a factor of 4.2.
- If a Cox model is fit it may be necessary to load `library(survival)` first; if bootstrapping is used it may be needed to load `library(boot)` first; and if $M = 1$ (complete cases, no missing data, one imputation), and also `method="MA.criterion"` or `method="MS.criterion"` and `criterion` is `"CV"` or `"GCV"`, it is needed to first load `library(snow)`. The same applies in the context of mixed models. `mami` will notify the user if a needed package has not been loaded.
- Currently the cluster for parallelization is set up (and stopped) automatically, mostly for convenience. In future releases there will be more flexible options for advanced users and different computing environments.
- `set.seed()` will not work with parallelization. Thus, bootstrap confidence intervals are not fully reproducible. There are ways around this, and guidance will be given in future.
- Using parallelization means that less warning messages are printed from `mami`.

6.2 Limitations of the Approach

- Model Selection and model averaging estimators are generally biased. Improving the coverage probability with the approach implemented in `mami` doesn't alter this conclusion. If the number of variables is reasonable, fitting the full model in addition to the selected/averaged model is recommended.
- The implementation in `mami` is meant to be useful, stable and flexible. However, for extremely complex situations, or more complex models, `mami` does not work. Here, a manual implementation is needed.
- Re-sampling has limitations, is not always valid, and this applies to model averaging too. Good references are Leeb and Pötscher (2005, 2006, 2008) and Pötscher (2006).
- There are simple pragmatic alternatives to our suggested approach: for example selecting only variables which are selected in each imputed data set; or stacking the imputed data, see Wood et al. (2008).

6.3 Optimal Model Averaging for Prediction: Super Learning

The motivation for optimal model averaging, i.e. Mallow's Model Averaging, Jackknife Model Averaging, and Shrinkage Averaging Estimation, is essentially prediction. For this reason `predict` methods are available: `predict.mma`, `predict.jma`, `predict.lae`. For example:

```
data(Prostate)
m1 <- mma(Prostate,lpsa~.,ycol='lpsa')
predict(m1)
predict(m1, newdata=Prostate[1,])
```

Depending on the specific problem, optimal model averaging may be a good prediction algorithm or not. To choose and combine the best prediction methods, *super learning* can be used. Super learning means considering a set of prediction algorithms, for example regression models, shrinkage estimators or regression trees. Instead of choosing the algorithm with the smallest cross validation error, super learning chooses a weighted combination of different algorithms, that is the weighted combination which minimizes the cross validation error. It can be shown that this weighted combination will perform at least as good as the best algorithm, if not better (Van der Laan et al., 2008). One may interpret this procedure as model averaging in a broader sense. The interested reader is referred to Van der Laan and Petersen (2007) and Van der Laan and Rose (2011), and the references therein, for more details.

Briefly, `MAMI` contains several wrappers that can be used for super learning. They are listed and explained by typing:

```
listSLWrappers()
```

Note that the package `SuperLearner` (Polley et al., 2017) is required. A simple example from `?listSLWrappers` would be:

```
library(SuperLearner) # needs to be installed
SL.library <- c('SL.glm','SL.stepAIC', 'SL.mean', 'SL.mma.int', 'SL.jma')
SL.library2 <- c('SL.glm','SL.stepAIC', 'SL.mean', 'SL.lae2')

data(Prostate)
P1 <- SuperLearner(Y=Prostate[,9], X=Prostate[,-9], SL.library =
SL.library, verbose=T)
P2 <- SuperLearner(Y=Prostate[,5], X=Prostate[,-5], family='binomial',
SL.library = SL.library2, verbose=T)
P2$coef
P2$SL.predict
```

In the above example both a continuous outcome (P1) and a binary outcome (P2) is predicted – using generalized linear models, the arithmetic mean, models selected by AIC, LASSO averaging including squared varriables ('SL.lae2'), among others. To see the weight each algorithm contributes to the prediction type `P2$coef`. The prediction itself is `P2$SL.predict`.

Super learning is often used for estimation of causal estimands with targeted maximum likelihood estimation. The implemented wrappers can be used, and have been tested, with package `tmle` (Gruber and van der Laan, 2012) and `ltmle` (Lendle et al., 2017).

6.4 Miscellaneous

Other options in `mami`.

`report.exp`: strongly recommended to set `report.exp=T` when fitting a logistic model, a Poisson model or Cox model to obtain the odds ratio, incidence rate ratio and hazard ratio respectively. Can also be of interest when the outcome of a linear model is log-transformed and thus a log-linear model is interpreted.

`print.warnings`: if set as `TRUE`, `mami` prints not only warnings but also plenty of information on the progress of the fitting procedure and on implied assumptions. The default is `TRUE` to be as transparent as possible, but may be set as `FALSE` in simulations.

`print.time`: primarily intended to forecast the time when `inference="+boot"`. Simply multiplies the time of the model selection/averaging procedure on the original data times the intended bootstrap runs.

Index

BMA, 7
    bic.glm
      OR, 20
MAMI
    jma, 8, 11
      bsa, 9, 11
      calc.var, 9, 11
      model.setup, 9, 12
    lae, 10
      B.var, 10
    mami, 5
      B, 13, 14
      CI, 14
      X.org, 14
      add.factor, 7
      add.interaction, 7
      add.stratum, 7
      add.transformation, 7
      aweights, 7
      candidate.models, 20
      criterion, 7
      cvr, 11–13
      id, 6
      inference, 14
      kfold, 11, 13
      method, 7, 8, 10, 11, 13
      model, 6
      outcome, 7
      print.time, 23
      print.warnings, 23
      report.exp, 23
      screen, 6, 21
      var.remove, 6
    mma, 8
    plot.mami, 17
    predict, 22
    print.mami, 14, 17
    summary.mami, 16, 17
MASS
    stepAIC, 12
MuMIn, 7
    dredge, 12, 20
      dc, 16
      subset, 16
corpcor, 10
glmnet
    cv.glmnet, 11
      alpha, 11, 13
      nlambda, 11
lme4, 8
quadprog, 9

analysis model, 6

bootstrapping, 10, 13

citation, 2
cross validation, 10

imputation model, 5
installation, 2
interpretation, 14

Jackknife Model Averaging, 11

LASSO, 10

model averaging, 7
    criterion based, 7
    JMA, 11
    LASSO, 10
    MMA, 8
model selection, 12
    criterion based, 12
    LASSO, 13
multiple imputation, 3

ncores, 21

parallelization, 21

super learner, 22

TMLE, 23

References

Barton, K. (2017). *MuMIn: Multi-Model Inference*. R package version 1.16.6/r405.

Buckland, S. T., K. P. Burnham, and N. H. Augustin (1997). Model selection: an integral part of inference. *Biometrics 53*, 603–618.

Burnham, K. and D. Anderson (2002). *Model selection and multimodel inference. A practical information-theoretic approach*. Springer, New York.

Chatfield, C. (1995). Model uncertainty, data mining and statistical inference. *Journal of the Royal Statistical Society A 158*, 419–466.

Draper, D. (1995). Assessment and propagation of model uncertainty. *Journal of the Royal Statistical Society B 57*, 45–97.

Gruber, S. and M. J. van der Laan (2012). tmle: An R package for targeted maximum likelihood estimation. *Journal of Statistical Software 51*(13), 1–35.

Hansen, B. E. (2007). Least squares model averaging. *Econometrica 75*, 1175–1189.

Hansen, B. E. and J. Racine (2012). Jackknife model averaging. *Journal of Econometrics, 167*, 38–46.

Hjort, L. and G. Claeskens (2003). Frequentist model average estimators. *Journal of the American Statistical Association 98*, 879–945.

Hoeting, J. A., D. Madigan, A. E. Raftery, and C. T. Volinsky (1999). Bayesian model averaging: a tutorial. *Statistical Science 14*, 382–417.

Honaker, J. and G. King (2010). What to do about missing values in time series cross-section data. *American Journal of Political Science 54*, 561–581.

Honaker, J., G. King, and M. Blackwell (2011). Amelia II: A program for missing data. *Journal of Statistical Software 45*(7), 1–47.

Karamchand, S., R. Leisegang, M. Schomaker, G. Maartens, L. Walters, M. Hislop, J. A. Dave, N. S. Levitt, and K. Cohen (2016). Risk factors for incident diabetes in a cohort taking first-line nonnucleoside reverse transcriptase inhibitor-based antiretroviral therapy. *Medicine (Baltimore) 95*(9), e2844.

Leeb, H. and B. M. Pötscher (2005). Model selection and inference: facts and fiction. *Econometric Theory 21*, 21–59.

Leeb, H. and B. M. Pötscher (2006). Can one estimate the conditional distribution of post-model-selection estimators? *Annals of Statistics 34*, 2554–2591.

Leeb, H. and B. M. Pötscher (2008). Can one estimate the unconditional distribution of post-model-selection estimators? *Econometric Theory 24*, 338–376.

Lendle, S. D., J. Schwab, M. L. Petersen, and M. J. van der Laan (2017). ltmle: An R package implementing targeted minimum loss-based estimation for longitudinal data. *Journal of Statistical Software 81*(1), 1–21.

Lipsitz, S., M. Parzen, and L. Zhao (2002). A degrees-of-freedom approximation in multiple imputation. *Journal of Statistical Computation and Simulation 72*, 309–318.

Polley, E., E. LeDell, C. Kennedy, and M. van der Laan (2017). *SuperLearner: Super Learner Prediction*. R package version 2.0-22.

Porter, M., M. A. Davies, M. K. Mapani, H. Rabie, S. Phiri, J. Nuttall, L. Fairlie, K. G. Technau, K. Stinson, R. Wood, M. Wellington, A. D. Haas, J. Giddy, F. Tanser, and B. Eley (2015). Outcomes of infants starting antiretroviral therapy in southern africa, 2004-2012. *Journal of Acquired Immune Deficiency Syndromes 69*(5), 593–601.

Pötscher, B. (2006). The distribution of model averaging estimators and an impossibility result regarding its estimation. In H. Ho, C. Ing, and T. Lai (Eds.), *IMS Lecture Notes: Time series and related topics*, Volume 52, pp. 113–129.

Raftery, A., J. Hoeting, C. Volinsky, I. Painter, and K. Y. Yeung (2017). *BMA: Bayesian Model Averaging*. R package version 3.18.7.

Rubin, D. and N. Schenker (1986). Multiple imputation for interval estimation from simple random samples with ignorable nonresponse. *Journal of the American Statistical Association 81*, 366–374.

Rubin, D. B. (1996). Multiple imputation after 18+ years. *Journal of the American Statistical Association 91*(434), 473–489.

Schomaker, M. (2012). Shrinkage averaging estimation. *Statistical Papers 53*, 1015–1034.

Schomaker, M. and C. Heumann (2014). Model selection and model averaging after multiple imputation. *Computational Statistics and Data Analysis 71*, 758–770.

Schomaker, M. and C. Heumann (2018). Bootstrap inference when using multiple imputation. *Statistics in Medicine 37*(14), 2252–2266.

Schomaker, M. and C. Heumann (2019). When and when not to use optimal model averaging. *Statistical Papers in press*.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B 58*, 267–288.

van Buuren, S. and K. Groothuis-Oudshoorn (2011). mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software 45*(3), 1–67.

Van der Laan, M. and M. Petersen (2007). Statistical learning of origin-specific statistically optimal individualized treatment rules. *International Journal of Biostatistics 3*, Article 3.

Van der Laan, M., E. Polley, and A. Hubbard (2008). Super learner. *Statistical Applications in Genetics and Molecular Biology 6*, Article 25.

Van der Laan, M. and S. Rose (2011). *Targeted Learning*. Springer.

Visser, M., M. Stead, G. Walzl, R. Warren, M. Schomaker, H. Grewal, E. Swart, and G. Maartens (2012). Baseline predictors of sputum conversion in pulmonary tuberculosis: importance of cavities, smoking, time to detection and W-Beijing genotype. *PLoS ONE 7*, e29588.

Volinsky, C. T., D. Madigan, A. E. Raftery, and R. A. Kronmal (1997). Bayesian model averaging in proportional hazard models. assessing the risk of a stroke. *Journal of the Royal Statistical Society Series C-Applied Statistics 46*(4), 433–448.

Wan, A. T. K., X. Zhang, and G. H. Zou (2010). Least squares model averaging by mallows criterion. *Journal of Econometrics 156*, 277–283.

White, I., P. Royston, and A. Wood (2011). Multiple imputation using chained equations. *Statistics in Medicine 30*, 377–399.

Wood, A., I. White, and P. Royston (2008). How should variable selection be performed with multiply imputed data? *Statistics in Medicine 27*, 3227–3246.