Pitfalls and Potentials in Simulation Studies
Simulation Protocol

Samuel Pawel, Lucas Kook, Kelly Reeve

September 21, 2021

1 Disclaimer

With this simulation study we will illustrate how easy it is to show statistical superiority of any method, if one has enough flexibility in changing the simulation conditions and evaluations. We propose a mock-method, termed AINET, which seems reasonable at first glance and whose evaluation could actually be published in a statistics journal. However, as we will explain in Section 7, for theoretical reasons the method should actually not be superior compared to methods of equal complexity. Throughout this protocol we will make a serious attempt at evaluating AINET using state-of-the-art simulation study methodology. After the planned simulations have been conducted, we will start to apply questionable research practices in order to illustrate how easy it is to find simulation conditions and evaluation metrics which can make a flawed method look superior.

2 Introduction

The purpose of this protocol is to describe our *a priori* plans for a comprehensive simulation study to evaluate the statistical properties of the mock-method called adaptive importance elastic net (AINET) method which is described in more detail in Section 7. We will follow the ADEMP approach (aims, data-generating process, estimands, methods, performance measures) in Morris et al. (2019). We will plan the simulation study as thoroughly as possible. However, in practice there may be difficulties and unexpected discoveries which may warrant modifications. We will clearly label these as exploratory findings in the analysis.

3 Timeline

Upon writing of this document only some preliminary evaluations have been made: After the authors came up with the method on Wednesday 28 July 2021, the method was implemented in R and evaluated on the iris data set and a few simulated data sets. These analyses showed that for particular choices of hyperparameters, the method could sometimes lead to improved predictive performance compared to standard and L1-penalized logistic regression. However, in many cases performance was actually equal or worse.

We then started to write the simulation protocol. All versions of the protocol are available on the GitHub repository (`https://github.com/SamCH93/SimPaper`), the final version will be time-stamped and tagged. As discussed in Section 8, some preliminary simulation runs will be conducted to estimate the number of simulations needed to ensure a sufficiently small Monte Carlo error. The test run will also be used to assess whether there are severe convergence problems and if so, the simulation parameters may be modified (Section 9). Also this milestone will be timestamped.

4 Aims

The aim of this simulation study is to systematically study the predictive performance of AINET for a binary prediction task. The simulation conditions should resemble typical conditions found in the development of prediction models in biomedical research. In particular we want to evaluate the performance of AINET conditional on

- low- and high-dimensional covariates
- (un-)correlated covariates
- small and large sample sizes
- varying baseline prevalences

AINET will be compared to other (penalized) binary regression models from the literature, namely

- Binary logistic regression: the simplest and most popular method for binary prediction
- Elastic net: a generalization of LASSO and ridge regression, the most widely used penalized regression methods
- Adaptive elastic net: a generalization of the most popular weighted penalized regression method (adaptive LASSO)
- Random forest: a popular, more flexible method. This method is related to AINET, see Section 7.

These cover a wide range of established methods with varying flexibility and serve as a reasonable benchmark for AINET. There are many more extensions of the adaptive elastic net in the literature (see *e.g.*, the review by Vidaurre et al., 2013). However, most of these extensions focus on variable selection and estimation instead of prediction, which is why we restrict our focus only on the four methods from above.

5 Data-generating process

In each simulation $b = 1, \dots, B$, we generate a data set consisting of n realizations, *i.e.*, $\{(y_i, \boldsymbol{x}_i)\}_{i=1}^n$. A datum $(Y, \boldsymbol{X})$ consists of a binary outcome $Y \in \{0, 1\}$ and p-dimensional covariate vector $\boldsymbol{X} \in \mathbb{R}^p$. The binary outcomes are generated by

$$Y \mid \boldsymbol{x} \sim \text{Bernoulli}\left(\text{expit}\left\{\beta_0 + \boldsymbol{x}^\top \boldsymbol{\beta}\right\}\right)$$

with $\text{expit}(z) = (1 + \exp(-z))^{-1}$ and the covariate vectors are generated by

$$\boldsymbol{X} \sim \text{N}_p\left(0, \Sigma\right)$$

with covariance matrix Σ that may vary across simulation conditions (see below). The baseline prevalence is $\text{prev} = \text{expit}(\beta_0)$. The coefficient vector $\boldsymbol{\beta}$ is generated from

$$\boldsymbol{\beta} \sim \text{N}_p(0, \text{Id})$$

once per simulation. Finally, the simulation parameters are varied fully factorially as described below, leading to a total of 240 scenarios, see below.

Sample size

The sample size used in the development of predictions models varies widely (Damen et al., 2016). We will use $n \in \{100, 500, 1000, 5000\}$, which span typical values occurring in practice. Note that previous simulation studies usually chose sample size based on the implied number of events together with the number of covariates in the model for easier interpretation (van Smeden et al., 2018; Riley et al., 2018). We will use this approach in reverse to determine the dimensionality of the parameters below.

Dimensionality

Several low- and high-dimensionality scenarios in p are considered. Previous simulation studies showed that EPV rather than the absolute sample size n and dimensionality p influences the predictive performance of a method. We will therefore define the dimensionality p via events per variable (EPV) by

$$p = \min\left\{\left\lceil \frac{n \cdot \text{prev}}{\text{EPV}} \right\rceil, 2\right\}.$$

The dimensionality is constrained to at least $p \geq 2$ because in practice prediction models are only multivariable models. EPV $\in \{20, 10, 1, 0.1, 0.05\}$ are chosen to cover scenarios with low- to high-dimensional covariates (*cf.* van Smeden et al., 2018).

Collinearity in X

We distinguish between no, low, medium and high collinearity. The diagonal elements of Σ are given by $\Sigma_{ii} = 1$ and the off-diagonal elements are set to $\Sigma_{ij} = \rho$, $\rho \in \{0, 0.3, 0.6, 0.95\}$. These values cover the typical (positive) range of correlations.

Baseline prevalence

Different baseline prevalences $\text{expit}(\beta_0) \in \{0.01, 0.05, 0.1\}$ are considered, reflecting a reasonable range of prevalences for rare to common diseases/adverse events.

Test data

In order to test the out-of-sample predictive performance, we generate a test data set of $n_{\text{test}} = 10000$ data points in each simulation b.

6 Estimands

We will estimate different quantities to evaluate overall predictive performance, calibration, and discrimination, respectively. All methods will be evaluated on independently generated test data.

6.1 Primary estimand

- **Brier score.** We compute the Brier score as

$$\overline{\text{BS}} = n_{\text{test}}^{-1} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2,$$

where $\hat{y} = \widehat{\mathbb{P}}(Y = 1 \mid \boldsymbol{x})$. Lower values indicate better predictive performance in terms of calibration and sharpness. A prediction is well-calibrated if the observed proportion of events is close to the predicted probabilities. Sharpness referes to how concentrated a predictive distribution is (*e.g.*, how long/short a prediction interval is), and the predictive goal is to maximize sharpness subject to calibration (Gneiting, 2008). The Brier score is a proper scoring rule, meaning that it is minimized if a predicted distribution is equal to the data-generating distribution (Gneiting and Raftery, 2007). Proper scoring rules thus encourage careful and honest predictions. The Brier score is therefore a principled choice for our primary estimand.

6.2 Secondary estimands

- **Scaled Brier score.** The scaled Brier score (also known as Brier skill score) is computed as

$$\overline{\text{BS}}^{*} = 1 - \overline{\text{BS}} \,/\, \overline{\text{BS}}_0$$

with $\overline{\text{BS}}_0 = \bar{y}(1 - \bar{y})$ and $\bar{y}$ the observed prevalence in the data set. The scaled Brier score takes into account that the prevalence varies across simulation conditions. Hence, the Brier score can be compared between conditions (Schmid and Griffith, 2005; Steyerberg et al., 2019).

- **Log-score.** We compute the log-score on independently generated test data,

$$\overline{\text{LS}} = -n_{\text{test}}^{-1} \sum_{i=1}^{n} \{ y_i \log(\hat{y}_i) + (1 - y_{ib}) \log(1 - \hat{y}_{ib}) \},$$

  will be used as a secondary measure of overall predictive performance. Lower values indicate better predictive performance in terms of calibration and sharpness. The log-score is a strictly proper scoring rule, however, it is more sensitive to extreme predicted probabilities compared to the Brier score (Gneiting and Raftery, 2007).

- **AUC.** The AUC is given by

$$\text{AUC} = \widehat{\mathbb{P}}(Y_i > Y_j \mid \boldsymbol{x}_i, \boldsymbol{x}_j), \; i, j = 1, \ldots, n_{\text{test}},$$

  where Y_i and Y_j denote case and non-case, respectively. The AUC is related to the area under the receiver-operating-characteristic (ROC) curve (Steyerberg et al., 2019). It will be used as a measure of discrimination and values closer to one indicate better discriminative ability. Discrimination describes the ability of a prediction model to discriminate between cases and non-cases. Other discrimination measures, such as accuracy, sensitivity, specificity, etc., are not considered because we want to evluated predictive performance in terms of probabilistic predictions instead of point predictions/classification.

- **Calibration slope $\hat{b}$.** The calibration slope $\hat{b}$ is obtained by regressing the test data outcomes y_{test} on the models' predicted logits $\operatorname{logit}(\hat{y})$, *i.e.*,

$$\operatorname{logit} \mathbb{E}[Y \mid \hat{y}] = a + b \operatorname{logit}(\hat{y}).$$

  This measure will be used to assess calibration and deviations of $\hat{b}$ from one indicate miscalibration (Steyerberg et al., 2019).

- **Calibration in the large $\hat{a}$.** We inspect calibration in the large $\hat{a}$ on independently generated test data, from the model

$$\operatorname{logit} \mathbb{E}[Y \mid \hat{y}] = a + \operatorname{logit}(\hat{y})$$

  This measure will also be used to assess calibration and deviations of $\hat{a}$ from zero indicate miscalibration (Steyerberg et al., 2019).

To facilitate comparison between simulation conditions, all estimands will also be corrected by the oracle version of the estimand, *e.g.*, the Brier score will be computed from the ground truth parameters and the simulated data $\boldsymbol{x}$, subsequently the oracle Brier score and will be subtracted from the estimated Brier score.

7 Methods

7.1 AINET

We now present the mock-method and give a superficial motivation why it could lead to improved predictive performance: Choosing the vector of penalization weights in the adaptive LASSO becomes difficult in high-dimensional settings. For instance, using absolute LASSO estimates as penalization weights omits the importance of several predictors by not selecting them, especially in the case of highly correlated predictors (Algamal and Lee, 2015). The adaptive importance elastic net (AINET) circumvents this problem by employing a random forest to estimate the penalization weights via an *a priori* chosen variable importance measure. This way, the importance of all variables enter the penalization weights simultaneously.

The penalized log-likelihood for AINET for a single observation $(y, \boldsymbol{x})$ is defined as

$$\ell_{\text{AINET}}(\beta_0, \boldsymbol{\beta}; y, \boldsymbol{x}, \alpha, \lambda, \boldsymbol{w}) = \ell(\beta_0, \boldsymbol{\beta}; y, \boldsymbol{x}) + \lambda \left(\alpha \sum_{j=1}^{p} w_j |\beta_j| + \frac{1}{2}(1 - \alpha) \sum_{j=1}^{p} w_j \beta_j^2 \right)$$

where

$$\ell(\beta_0, \boldsymbol{\beta}; y, \boldsymbol{x}) = y \log \left(\operatorname{expit} \left\{ \beta_0 + \boldsymbol{x}^\top \boldsymbol{\beta} \right\} \right) + (1 - y) \log \left(1 - \operatorname{expit} \left\{ \beta_0 + \boldsymbol{x}^\top \boldsymbol{\beta} \right\} \right)$$

denotes the log-likelihood of a binomial glm and $\boldsymbol{w}$ is derived from a random forest variable importance measure IMP as

$$w_j = 1 - \left(\frac{\mathrm{IMP}_j}{\sum_{k=1}^{p} \mathrm{IMP}_k} \right)^{\gamma},$$

where we transform IMP to be non-negative via

$$\mathrm{IMP}_j = \max\{0, \widetilde{\mathrm{IMP}}_j\}$$

and γ is a hyperparameter for the influence of the weights similar to γ hyperparameter of the adaptive elastic net. AINET is fitted by maximizing its penalized log-likelihood assuming i.i.d. observations $\{(y_i, \boldsymbol{x}_i)\}_{i=1}^{n}$, *i.e.*,

$$\underset{\beta_0, \boldsymbol{\beta}}{\arg\max} \sum_{i=1}^{n} \ell_{\mathrm{AINET}}(\beta_0, \boldsymbol{\beta}; y_i, \boldsymbol{x}_i, \alpha, \lambda, \boldsymbol{w}).$$

Per default, we choose mean decrease in the Gini coefficient for $\widetilde{\mathrm{IMP}}$. Hyperparameters of the random forest are not tuned, but kept at their default values (*e.g.*, `mtry`, `ntree`). The hyperparameter $\gamma = 1$ will stay constant for all simulations.

As outlined in Section 1, AINET is supposed to seem like a reasonable method at first glance. However, AINET cannot be expected to share desirable theoretical properties with the usual adaptive LASSO, such as oracle estimation (Zou, 2006). This is because the penalization weights $\boldsymbol{w}$ do not meet the required consistency assumption. Also in terms of prediction performance, AINET is not expected to outperform methods of comparable complexity.

7.2 Benchmark methods

- **Binary logistic regression** (McCullagh and Nelder, 2019) with and without ridge penalty for high- and low-dimensional settings, respectively. In case a ridge penalty is needed, it is tuned via 5-fold cross-validation by following the “one standard error” rule as implemented in **glmnet** (Friedman et al., 2010).

- **Elastic net** (Zou and Hastie, 2005), for which the penalized log-likelihood is given by

  $$\ell_{\mathrm{EN}}(\beta_0, \boldsymbol{\beta}; y, \boldsymbol{x}, \alpha, \lambda) = \ell(\beta_0, \boldsymbol{\beta}; y, \boldsymbol{x}) + \lambda \left(\alpha \left\| \boldsymbol{\beta} \right\|_1 + \frac{1}{2}(1 - \alpha) \left\| \boldsymbol{\beta} \right\|_2^2 \right).$$

  Here, α and λ are tuned via 5-fold cross-validation by following the “one standard error” rule.

- **Adaptive elastic net** (Zou, 2006), with penalized loss function

  $$\ell_{\mathrm{adaptive}}(\beta_0, \boldsymbol{\beta}; y, \boldsymbol{x}, \alpha, \lambda, \boldsymbol{w}) = \ell(\beta_0, \boldsymbol{\beta}; y, \boldsymbol{x}) + \lambda \left(\alpha \sum_{j=1}^{p} w_j |\beta_j| + \frac{1}{2}(1 - \alpha) \sum_{j=1}^{p} w_j \beta_j^2 \right).$$

  Here, the penalty weights $\boldsymbol{w}$ are inverse coefficient estimates from a binary logistic regression

  $$\hat{w}_j = |\hat{\vartheta}_j|^{-\gamma},$$

  where λ and α are tuned via 5-fold cross-validation by following the “one standard error” rule. The hyperparameter $\gamma = 1$ will stay constant for all simulations. In case $p > n$, we estimate the penalty weights using a ridge penalty, tuned via an additional nested 5-fold cross-validation by following the “one standard error” rule.

- **Random forests** (Breiman, 2001) for binary outcomes without hyperparameter tuning. The default parameters of **ranger** will be used (Wright and Ziegler, 2017).

8 Performance measures

The distribution of all estimands from Section 6 will be assessed visually with box- and violin-plots that are stratified by method and simulation conditions. We will also compute

- Mean
- Median
- Standard deviations
- Interquartile range
- 95% confidence intervals

for each of the estimands. Moreover, instead of "eye-balling" differences in predictive performance across methods and conditions, we will formally assess them by regressing the estimands on the method and simulation conditions (*cf.* Skrondal, 2000). To do so, we will use a fully interacted model with the interaction between the methods and the 240 simulations conditions, *i.e.*, in R notation: `estimand ~ 0 + method:scenario`. We will rank pairwise comparison between two methods within a single condition by their p-values, to more easily identify conditions where methods show differences in predictive performance. The choice of a significance level at which a method is deemed superior will be determined based on preliminary simulations. A priori we set this level to 5%.

8.1 Determining the number of simulations

We determine the number of simulation B such that the Monte Carlo standard error of the primary estimand, the mean Brier score $\overline{\text{BS}}$, is sufficiently small. The variance of $\overline{\text{BS}}$ is given by

$$\text{Var}\left(\overline{\text{BS}}\right) = B^{-1} n_{\text{test}}^{-1} \,\text{Var}\left\{(y_{ib} - \hat{y}_{ib})^2\right\}$$

and $\text{Var}\left\{(y_{ib} - \hat{y}_{ib})^2\right\}$ could be decomposed further (Bradley et al., 2008). However, the resulting expression is difficult to evaluate for our data-generating process as it dependens on several of the simulation parameters. We therefore follow a similar approach as in Morris et al. (2019) and estimate $\widehat{\text{Var}}\left\{(y_{ib} - \hat{y}_{ib})^2\right\} < V$ from an initial small simulation run with 100 simulations per condtion to get an upper bound V for worst-case variance across all simulation conditions. The number of simulations is then given by

$$B = \frac{V}{n_{\text{test}} \,\text{Var}\left(\overline{\text{BS}}\right)}.$$

Since $\overline{\text{BS}} \in [0, 1]$ we decide that we require the Monte Carlo standard error of $\overline{\text{BS}}$ to be lower than four significant digits, 0.0001, hence we obtain

$$B = V/(n_{\text{test}} \cdot 0.0001^2).$$

9 Handling exceptions

It is inevitable that convergence issues and other problems will arise in the simulation study, we will handle them as follow:

- If a method fails to converge, the simulation will be excluded from the analysis. The failing simulations will not be replaced with new simulations that successfully converge as convergence may be impossible for some scenarios.
- We will report the proportion of simulations with convergence issues for each method and discuss the potential reasons of their emergence

- In case of severe convergence issues or other problems (more than 10% of the simulations failing within a setting), we may adjust the simulation parameters post hoc. This will be indicated in the discussion of the results.
- Convergence may be possible for certain tuning parameters of a method (*e.g.*, cross-validation of LASSO may fail for some values λ while it could work for others). In this case we will choose a parameter value where the method still converges, as one would usually do with a real data set.

10 Software

The simulation study is conducted in the R language for statistical computing (R Core Team, 2020) using the most recent version (at the time of writing this protocol R version 4.1.1). AINET is implemented in the **ainet** package and available on GitHub (`https://github.com/SamCH93/SimPaper`). We use **pROC** version 1.17.0.1 to compute the AUC (Robin et al., 2011). Random forests are fitted using **ranger** version 0.13.1 (Wright and Ziegler, 2017). For penalized likelihood methods, we use **glmnet** version 4.1.1 (Friedman et al., 2010; Simon et al., 2011). The **SimDesign** package version 2.7 is used to set up simulation scenarios (Chalmers and Adkins, 2020).

References

Algamal, Z. Y. and Lee, M. H. (2015). Penalized logistic regression with the adaptive LASSO for gene selection in high-dimensional cancer classification. *Expert Systems with Applications*, 42(23):9326–9332. doi:10.1016/J.ESWA.2015.08.016.

Bradley, A. A., Schwartz, S. S., and Hashino, T. (2008). Sampling uncertainty and confidence intervals for the Brier score and Brier skill score. *Weather and Forecasting*, 23(5):992–1006. doi:10.1175/2007waf2007049.1.

Breiman, L. (2001). *Machine Learning*, 45(1):5–32. doi:10.1023/a:1010933404324.

Chalmers, R. P. and Adkins, M. C. (2020). Writing effective and reliable monte carlo simulations with the SimDesign package. *The Quantitative Methods for Psychology*, 16(4):248–280. doi:10.20982/tqmp.16.4.p248.

Damen, J. A. A. G., Hooft, L., Schuit, E., Debray, T. P. A., Collins, G. S., Tzoulaki, I., Lassale, C. M., Siontis, G. C. M., Chiocchia, V., Roberts, C., Schlüssel, M. M., Gerry, S., Black, J. A., Heus, P., van der Schouw, Y. T., Peelen, L. M., and Moons, K. G. M. (2016). Prediction models for cardiovascular disease risk in the general population: systematic review. *BMJ*, page i2416. doi:10.1136/bmj.i2416.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, 33(1):1–22. doi:10.18637/jss.v033.i01.

Gneiting, T. (2008). Editorial: Probabilistic forecasting. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 171(2):319–321. doi:10.1111/j.1467-985x.2007.00522.x.

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378. doi:10.1198/016214506000001437.

McCullagh, P. and Nelder, J. A. (2019). *Generalized linear models*. Routledge.

Morris, T. P., White, I. R., and Crowther, M. J. (2019). Using simulation studies to evaluate statistical methods. *Statistics in Medicine*, 38(11):2074–2102. doi:10.1002/sim.8086.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL `https://www.R-project.org/`.

Riley, R. D., Snell, K. I., Ensor, J., Burke, D. L., Jr, F. E. H., Moons, K. G., and Collins, G. S. (2018). Minimum sample size for developing a multivariable prediction model: PART II - binary and time-to-event outcomes. *Statistics in Medicine*, 38(7):1276–1296. doi:10.1002/sim.7992.

Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J.-C., and Müller, M. (2011). pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics*, 12:77. doi:10.1186/1471-2105-12-77.

Schmid, C. H. and Griffith, J. L. (2005). Multivariate classification rules: Calibration and discrimination. In Armitage, P. and Colton, T., editors, *Encyclopedia of Biostatistics*, volume 5, pages 3491–3497. Wiley, 2nd edition.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2011). Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent. *Journal of Statistical Software*, 39(5):1–13. doi:10.18637/jss.v039.i05.

Skrondal, A. (2000). Design and analysis of monte carlo experiments: Attacking the conventional wisdom. *Multivariate Behavioral Research*, 35(2):137–167. doi:10.1207/s15327906mbr3502_1.

Steyerberg, E. W. et al. (2019). *Clinical prediction models*. Springer.

van Smeden, M., Moons, K. G., de Groot, J. A., Collins, G. S., Altman, D. G., Eijkemans, M. J., and Reitsma, J. B. (2018). Sample size for binary logistic prediction models: Beyond events per variable criteria. *Statistical Methods in Medical Research*, 28(8):2455–2474. doi:10.1177/0962280218784726.

Vidaurre, D., Bielza, C., and Larrañaga, P. (2013). A survey of $L1$ regression. *International Statistical Review*, 81(3):361–387. doi:10.1111/insr.12023.

Wright, M. N. and Ziegler, A. (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, 77(1):1–17. doi:10.18637/jss.v077.i01.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429. doi:10.1198/016214506000000735.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320. doi:10.1111/j.1467-9868.2005.00503.x.

11 Session info

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.3 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=de_CH.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=de_CH.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=de_CH.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=de_CH.UTF-8 LC_IDENTIFICATION=C
```

```
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods
## [7] base
##
## other attached packages:
##  [1] ainet_0.0.0.9000 magrittr_2.0.1   tidyr_1.1.3
##  [4] dplyr_1.0.6      pROC_1.17.0.1    mvtnorm_1.1-2
##  [7] SimDesign_2.7    glmnet_4.1-1     Matrix_1.3-4
## [10] ranger_0.13.1    knitr_1.33
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.7        plyr_1.8.6        compiler_4.1.1
##  [4] pillar_1.6.2      highr_0.9         iterators_1.0.13
##  [7] tools_4.1.1       jsonlite_1.7.2    evaluate_0.14
## [10] lifecycle_1.0.0   tibble_3.1.4      lattice_0.20-44
## [13] pkgconfig_2.0.3   rlang_0.4.11      foreach_1.5.1
## [16] cli_3.0.1         DBI_1.1.1         curl_4.3.1
## [19] parallel_4.1.1    xfun_0.23         withr_2.4.2
## [22] stringr_1.4.0     generics_0.1.0    vctrs_0.3.8
## [25] RPushbullet_0.3.4 grid_4.1.1        tidyselect_1.1.1
## [28] glue_1.4.2        R6_2.5.1          pbapply_1.4-3
## [31] fansi_0.5.0       survival_3.2-13   sessioninfo_1.1.1
## [34] purrr_0.3.4       codetools_0.2-18  ellipsis_0.3.2
## [37] splines_4.1.1     assertthat_0.2.1  shape_1.4.6
## [40] utf8_1.2.2        stringi_1.6.2     crayon_1.4.1
```