SIG Project Real-Time Imputation

Steven Nijman, Thomas Debray, Maarten van Smeden, Gerko Vink, Hanne Oberman

Contents

Aim

This document contain the set-up of our SIG project titled "An evaluation of 'real-time' missing data handling in machine learning and prevailing statistical models". The aim is to compare different strategies for developing prediction models that can handle the presence of missing values real time in a single patient.

Data Generating Mechanism

We will use a prediction model with:

- 1 binary outcome (Y),
- 10 continuous predictors (X_1, X_2, ..., X_{10}).

Q: Do we need categorical predictors as well?

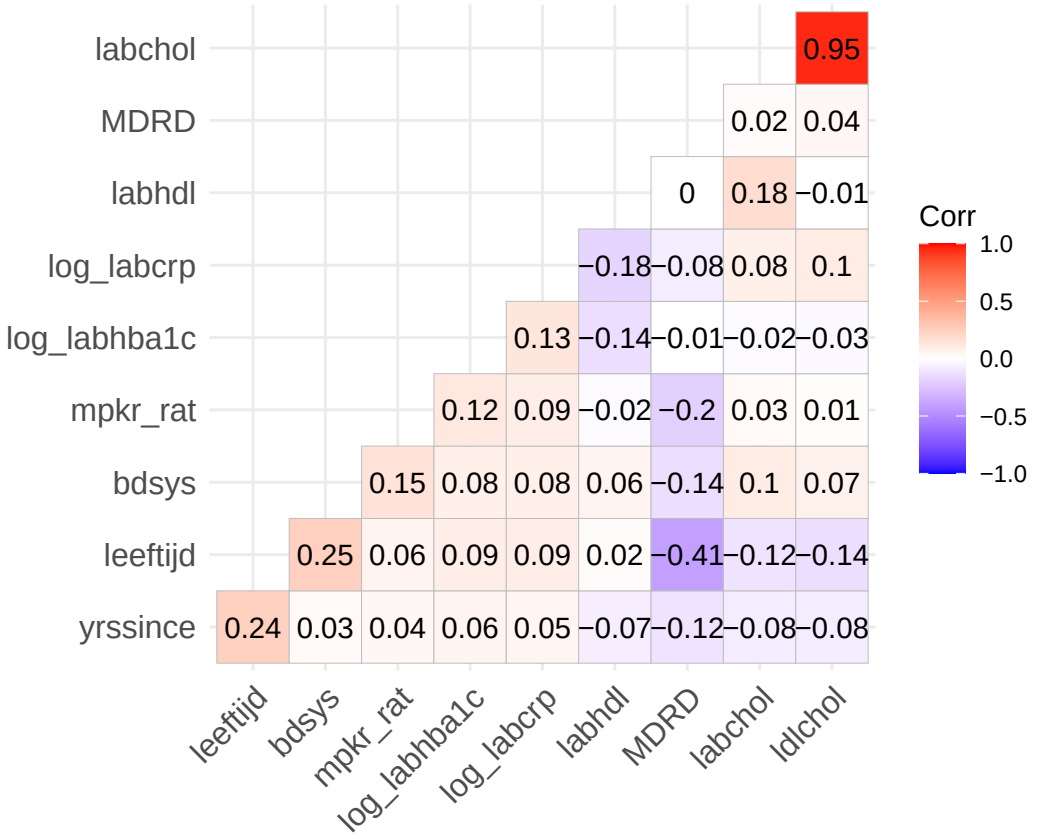
Our sample will include at least these 11 variables, with a sample size of 10.000 cases.

Q: Do we want extra 'support' variables (Z) in our sample that are not included in the prediction model?

Initially, we were going to use the SMART data as the basis for our predictor space. But this would result in the same limitations as described in Nijman et al. (2021; i.e., low correlations between predictor variables).

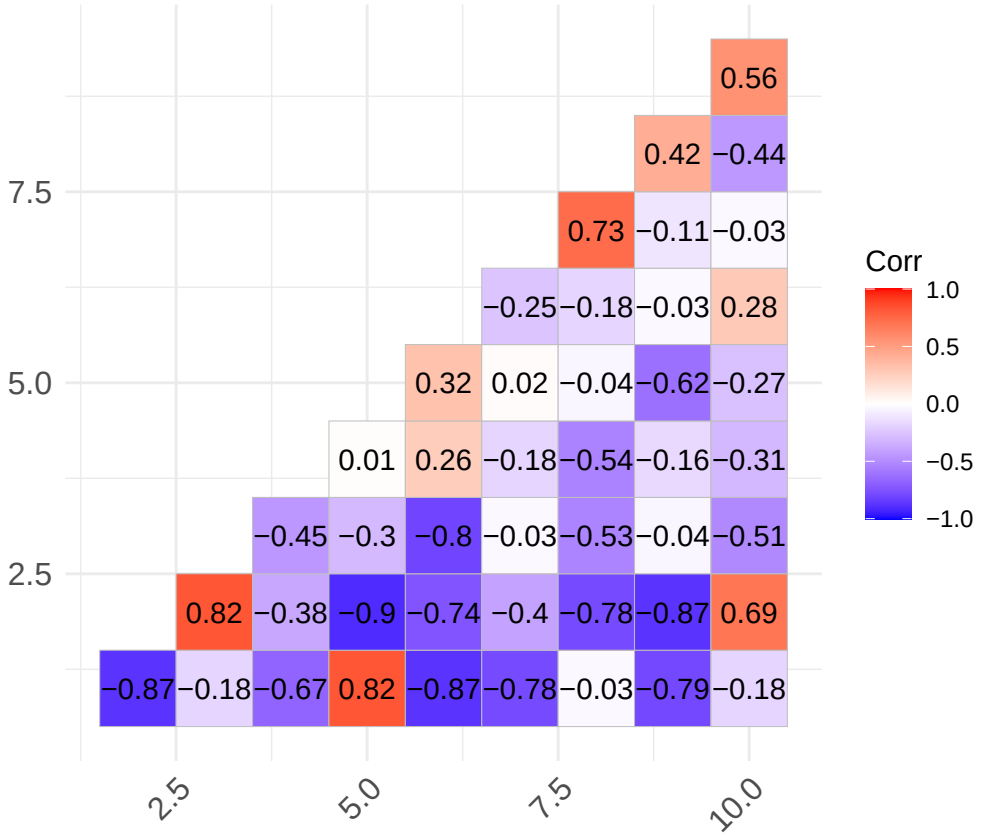
These correlations between variables from the SMART dataset are visualized below.

```
# we'll use the SMART data to define the relations between predictors
# the variance-covariance matrix is stored in the object varcov
ggcorrplot::ggcorrplot(cov2cor(varcov), hc.order = TRUE, type = "lower", lab = TRUE)
```



So instead, we'll create our own.

```
# create a variance-covariance matrix with p predictors
set.seed(11)
p <- 10
sigma <- diag(p)
sigma[lower.tri(sigma)] <- sigma[upper.tri(sigma)] <- runif(p*(p-1)/2, -.9, .9)
ggcorrplot::ggcorrplot(cov2cor(sigma), hc.order = TRUE, type = "lower", lab = TRUE)
```




```
# let's generate some data (note that this part is still based on the SMART data predictor space!)
n <- 10000
dat <- generate_sample(n, varcov)
glimpse(dat)
```

```
## Rows: 10,000
## Columns: 11
## $ Y   <int> 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, ...
## $ X1  <dbl> 9.9048242, -11.6502212, -0.3076518, -6.2776774, -18.3959809, 4....
## $ X2  <dbl> -2.39208155, -0.06163612, 0.93160755, -0.14206367, 1.86668700, ...
## $ X3  <dbl> 0.100503389, 0.117643030, -0.286659108, -0.172138871, 0.1393244...
## $ X4  <dbl> -1.667002451, -0.098932906, 1.052052195, -0.507654653, 1.492794...
## $ X5  <dbl> -9.4880225, -19.8230542, -1.3738283, 17.7636744, 1.2004914, 16....
## $ X6  <dbl> -0.013495744, -0.027731217, -0.099605192, -0.067239267, 0.01861...
## $ X7  <dbl> 13.8885410, -2.8899387, -14.5920085, -0.1586416, 10.8711534, -1...
## $ X8  <dbl> 1.6979761, -11.8147347, -9.8268114, -8.9720874, 35.2714535, 4.6...
## $ X9  <dbl> 0.63707657, -0.21483322, 0.56734021, -0.66323377, 0.29231070, -...
## $ X10 <dbl> 1.9389275, -4.9869662, -1.2976855, -6.0807333, 1.4921040, -3.31...
```

```
# check prevalence and auc
out <- check_characteristics(dat)
```

The prevalence of the outcome is 0.16. The C index/AUC of the model is 0.57.

Q: What other characteristics do we need to check to see if the data suits our purpose?

We'll add interaction terms between the first predictor and all others to create a need for ML models.

Q: Is 10 interactions enough to make the data more realistic/complex?

Q: Are additional non-linear effects needed (e.g., log-transformed predictors)?

```
# now add some interaction terms (actually 1 squared term, 9 interactions)
dat <- generate_sample(n, varcov, interaction = TRUE)
# check prevalence and auc
out <- check_characteristics(dat)
```

The prevalence of the outcome now is 0.35. The C index/AUC of the model is 0.52.

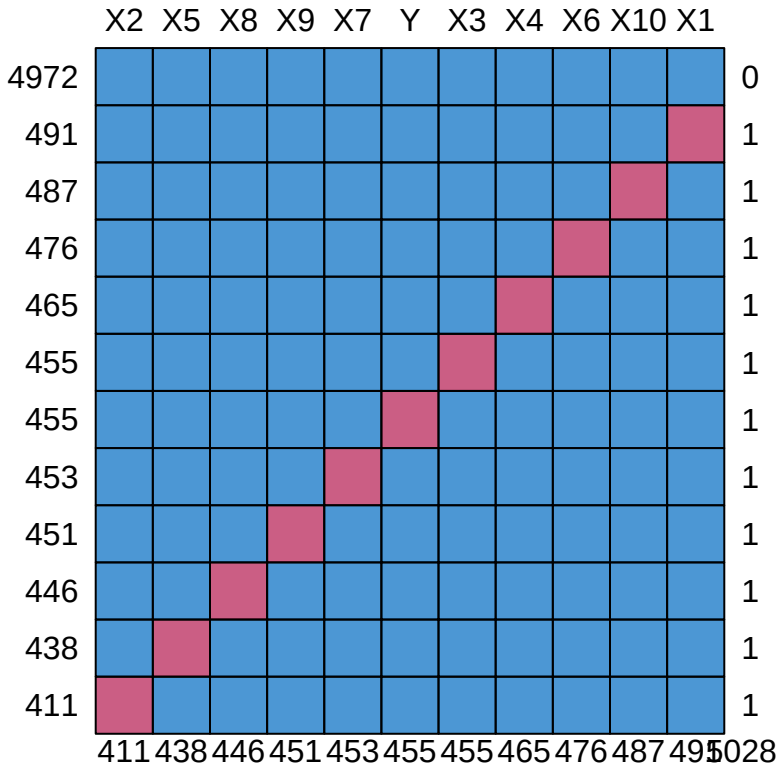
The next step is to ampute the complete set using several missing data patterns. Note that this part of the simulation pipeline is still very much under construction.

Q: Should the missing data pattern be equal for all three missing data mechanisms?

Q: Should the outcome variable contain missingness? It won't be observed anyways, right?

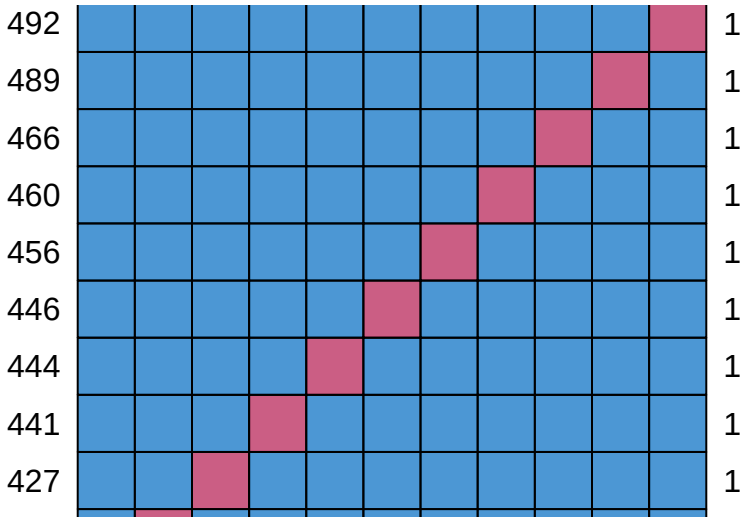
Q: What is meant with MAR situation 1 and 2 in the Word doc from our last meeting?

```
# create MCAR missingness
MCAR <- dat %>%
  mice::ampute(mech = "MCAR")
md <- mice::md.pattern(MCAR$amp)
```




```
# create MAR missingness
MAR <- dat %>%
  mice::ampute(mech = "MAR")
```

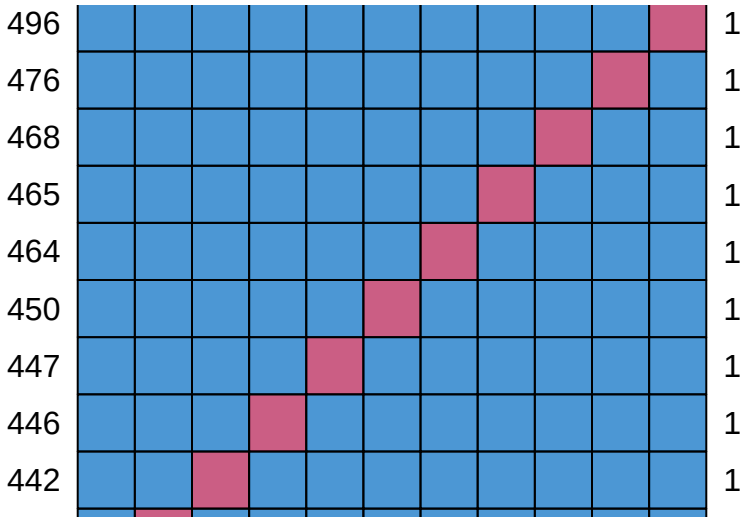
```
md <- mice::md.pattern(MAR$amp)
```




```
# create MNAR missingness for the strongest predictor
dat %>%
  glm(Y ~ ., family = "binomial", data = .) %>%
  broom::tidy() %>%
  dplyr::arrange(desc(abs(estimate)))
```

```
## # A tibble: 11 x 5
##    term         estimate std.error statistic   p.value
##    <chr>           <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept) -0.639      0.0211   -30.3    4.97e-202
##  2 X6           0.130      0.162      0.802  4.22e-  1
##  3 X2          -0.0506     0.0649    -0.779  4.36e-  1
##  4 X4           0.0504     0.0713     0.706  4.80e-  1
##  5 X3          -0.0121     0.0709    -0.170  8.65e-  1
##  6 X10          0.00428    0.00365    1.17   2.40e-  1
##  7 X1          -0.00168    0.00198   -0.850  3.96e-  1
##  8 X9           0.00136    0.0195     0.0699 9.44e-  1
##  9 X5           0.00112    0.00103    1.09   2.75e-  1
## 10 X7          -0.00110    0.00120   -0.914  3.61e-  1
## 11 X8           0.000251   0.00148    0.169  8.66e-  1
```

```
# the strongest predictor currently is X4
MNAR <- dat %>%
  mice::ampute(mech = "MNAR")
md <- mice::md.pattern(MNAR$amp)
```



Estimands