Final models and Tables/Figures

Christoph Völtzke

2023

packages

```
library(readr) # read data
library(tidyverse) # data manipulation
library(RColorBrewer) # color palettes
library(xtable) # latex code for main results
library(yardstick) # ROC curves with multiple curves
library(stargazer) # latex code for descriptive stats
library(data.table) # data frame preparation for descriptive stats
```

Information

- In this document the results of the data modeling are combined over the 3 folds and estimates are calculated.
- Then the results table of the report is created with final latex code in the end.
- Then the Figure for the ROC curves over the 3 folds for each ML and data condition is created. With the final being a plot.
- Then the table for the descriptive data is included. Here the full patient data is used and not only the training or test data.
- Last an optional section is included with some further visualizations to understand the data.
  - This includes:
  - Calibration curves for multiple models
  - Outcome frequency as a Figure
  - CGM continuous profiles as a Figure to see how it fluctuates over the monitored period
- A Session Info command is the last code chunk as this is the last needed script to be run for this study.

Load Helper functions needed for analysis

```
source("Functions/calibration_helper.R")
# helper function to make calibration plots
```

All data sets needed for Main results

Load all results on evaluation metrics obtained by the model building scripts when run with different folds indicated by document name.

```
# All produced data frames from the Model building docs
full <- read_csv("Data/all_models_full_123.csv") # 1. fold
life <- read_csv("Data/all_models_lifestyle_123.csv") # 1. fold

full_1 <- read_csv("Data/all_models_full_41263.csv") # 2. fold
life_1 <- read_csv("Data/all_models_lifestyle_41263.csv") # 2. fold

full_2 <- read_csv("Data/all_models_full_2408.csv") # 3. fold
life_2 <- read_csv("Data/all_models_lifestyle_2408.csv") # 3. fold
```

Overview - Full data over Folds

```
# comparison of estimates over different folds
full[1:10,]
```

```
## # A tibble: 10 x 7
##    model           AUC Sensitivity Specificity Accuracy ML    Condition
##    <chr>         <dbl>       <dbl>       <dbl>    <dbl> <chr> <chr>
##  1 original      0.925       0.423       0.974    0.893 RF    Full_data
##  2 original_cost 0.925       0.885       0.842    0.848 RF    Full_data
##  3 smote_def     0.897       0.923       0.75     0.775 RF    Full_data
##  4 up_def        0.902       0.846       0.816    0.82  RF    Full_data
##  5 up_own        0.893       0.923       0.737    0.764 RF    Full_data
##  6 smote_own     0.914       0.923       0.783    0.803 RF    Full_data
##  7 original_own  0.927       0.923       0.809    0.826 RF    Full_data
##  8 up_ran        0.866       0.885       0.697    0.725 RF    Full_data
##  9 smote_ran     0.868       0.885       0.73     0.753 RF    Full_data
## 10 original_ran  0.93        0.885       0.888    0.888 RF    Full_data
```

```
full_1[1:10,]
```

```
## # A tibble: 10 x 7
##    model           AUC Sensitivity Specificity Accuracy ML    Condition
##    <chr>         <dbl>       <dbl>       <dbl>    <dbl> <chr> <chr>
##  1 original      0.854       0.108       1        0.815 RF    Full_data
##  2 original_cost 0.854       0.838       0.716    0.742 RF    Full_data
##  3 smote_def     0.879       0.811       0.837    0.831 RF    Full_data
##  4 up_def        0.876       0.838       0.773    0.787 RF    Full_data
##  5 up_own        0.883       0.838       0.752    0.77  RF    Full_data
##  6 smote_own     0.866       0.865       0.709    0.742 RF    Full_data
##  7 original_own  0.865       0.811       0.759    0.77  RF    Full_data
##  8 up_ran        0.887       0.865       0.759    0.781 RF    Full_data
##  9 smote_ran     0.891       0.865       0.809    0.82  RF    Full_data
## 10 original_ran  0.869       0.811       0.816    0.815 RF    Full_data
```

```
full_2[1:10,]
```

```
## # A tibble: 10 x 7
##    model           AUC Sensitivity Specificity Accuracy ML    Condition
##    <chr>         <dbl>       <dbl>       <dbl>    <dbl> <chr> <chr>
##  1 original      0.908       0.464       0.94     0.865 RF    Full_data
##  2 original_cost 0.908       0.929       0.787    0.809 RF    Full_data
##  3 smote_def     0.872       0.893       0.76     0.781 RF    Full_data
##  4 up_def        0.876       0.857       0.793    0.803 RF    Full_data
##  5 up_own        0.879       0.893       0.753    0.775 RF    Full_data
##  6 smote_own     0.834       0.893       0.693    0.725 RF    Full_data
##  7 original_own  0.92        0.929       0.8      0.82  RF    Full_data
##  8 up_ran        0.881       0.893       0.767    0.787 RF    Full_data
##  9 smote_ran     0.835       0.893       0.727    0.753 RF    Full_data
## 10 original_ran  0.921       0.964       0.8      0.826 RF    Full_data
```

Overview - Lifestyle data over Folds

```
# comparison of estimates over different folds
life[1:10,]
```

```
## # A tibble: 10 x 7
##    model           AUC Sensitivity Specificity Accuracy ML    Condition
##    <chr>         <dbl>       <dbl>       <dbl>    <dbl> <chr> <chr>
##  1 original      0.848       0.269       0.961    0.86  RF    Lifestyle
##  2 original_cost 0.848       0.885       0.711    0.736 RF    Lifestyle
##  3 smote_def     0.83        0.769       0.776    0.775 RF    Lifestyle
##  4 up_def        0.843       0.885       0.678    0.708 RF    Lifestyle
##  5 up_own        0.845       0.885       0.664    0.697 RF    Lifestyle
##  6 smote_own     0.845       0.885       0.651    0.685 RF    Lifestyle
##  7 original_own  0.841       0.885       0.697    0.725 RF    Lifestyle
##  8 up_ran        0.84        0.923       0.638    0.68  RF    Lifestyle
##  9 smote_ran     0.847       0.885       0.724    0.747 RF    Lifestyle
## 10 original_ran  0.842       0.846       0.711    0.73  RF    Lifestyle
```

```
life_1[1:10,]
```

```
## # A tibble: 10 x 7
##    model           AUC Sensitivity Specificity Accuracy ML    Condition
##    <chr>         <dbl>       <dbl>       <dbl>    <dbl> <chr> <chr>
##  1 original      0.771       0.189       0.993    0.826 RF    Lifestyle
##  2 original_cost 0.771       0.838       0.582    0.635 RF    Lifestyle
##  3 smote_def     0.775       0.784       0.638    0.669 RF    Lifestyle
##  4 up_def        0.77        0.73        0.73     0.73  RF    Lifestyle
##  5 up_own        0.768       0.811       0.61     0.652 RF    Lifestyle
##  6 smote_own     0.805       0.757       0.709    0.719 RF    Lifestyle
##  7 original_own  0.791       0.811       0.617    0.657 RF    Lifestyle
##  8 up_ran        0.768       0.757       0.674    0.691 RF    Lifestyle
##  9 smote_ran     0.774       0.838       0.624    0.669 RF    Lifestyle
## 10 original_ran  0.781       0.811       0.66     0.691 RF    Lifestyle
```

```
life_2[1:10,]
```

```
## # A tibble: 10 x 7
##    model           AUC Sensitivity Specificity Accuracy ML    Condition
##    <chr>         <dbl>       <dbl>       <dbl>    <dbl> <chr> <chr>
##  1 original      0.826       0.179       0.953    0.831 RF    Lifestyle
##  2 original_cost 0.826       0.786       0.773    0.775 RF    Lifestyle
##  3 smote_def     0.828       0.821       0.7      0.719 RF    Lifestyle
##  4 up_def        0.83        0.786       0.747    0.753 RF    Lifestyle
##  5 up_own        0.817       0.821       0.66     0.685 RF    Lifestyle
##  6 smote_own     0.824       0.786       0.773    0.775 RF    Lifestyle
##  7 original_own  0.832       0.786       0.76     0.764 RF    Lifestyle
##  8 up_ran        0.831       0.786       0.72     0.73  RF    Lifestyle
##  9 smote_ran     0.82        0.714       0.853    0.831 RF    Lifestyle
## 10 original_ran  0.834       0.821       0.767    0.775 RF    Lifestyle
```

Creating Estimates over folds

Get the mean and SD over the three folds for each of the evaluation metrics.

```
all_full <- rbind(full,full_1,full_2)

all_full <- all_full %>%
  group_by(model,Condition,ML) %>%
  summarise(mean_AUC = round(mean(AUC),3),
            mean_SENS = round(mean(Sensitivity),3),
            mean_SPEC = round(mean(Specificity),3),
            sd_AUC = round(sd(AUC),3),
            sd_SENS = round(sd(Sensitivity),3),
            sd_SPEC = round(sd(Specificity),3))
```

```
## ‘summarise()‘ has grouped output by ’model’, ’Condition’. You can override
## using the ‘.groups‘ argument.
```

```
all_life <- rbind(life,life_1,life_2)

all_life <- all_life %>%
  group_by(model,Condition,ML) %>%
  summarise(mean_AUC = round(mean(AUC),4),
            mean_SENS = round(mean(Sensitivity),4),
            mean_SPEC = round(mean(Specificity),4),
            sd_AUC = round(sd(AUC),4),
            sd_SENS = round(sd(Sensitivity),4),
            sd_SPEC = round(sd(Specificity),4))
```

```
## ‘summarise()‘ has grouped output by ’model’, ’Condition’. You can override
## using the ‘.groups‘ argument.
```

```
all_full_cross <-  rbind(all_full,all_life)
```

Table of Model Results

This is the pre-processing for the table with the best 32 models depicted in the report. In this step the best tuning strategy for each combination of sampling, ML and Condition is used.

```
data <- all_full_cross

data <- data %>%
  extract(model,c("Resample", "grid"), "([[:alnum:]]+)_([[:alnum:]]+)",remove=FALSE)
# make columns in a way that the best model performance across
# all ML, Sampling and Tuning Strategy, and data condition

data$Resample[is.na(data$Resample)] <- "original"
data$grid[is.na(data$grid)] <- "thres"

data <- data %>%
  mutate(Thres = ifelse(grid =="thres",TRUE,FALSE)) %>%
  # Always identify the no cost-sensitive learning models as these are always selected
  select(-model)
```

```
## Adding missing grouping variables: `model`
```

```
data <- data %>%
  group_by(Resample,ML,Condition) %>%
  # best tuning strategy for each combination of sampling, ML and Condition
  mutate(Best = ifelse(mean_AUC == max(mean_AUC),TRUE,FALSE)) %>%
  # first filter based on best AUC
  filter(Best == TRUE | Thres == TRUE)

data <- data %>%
  group_by(Resample,ML,Condition, Thres) %>%
  # best tuning strategy for each combination of sampling, ML and Condition
  mutate(Best_Sp = ifelse(mean_SENS == max(mean_SENS),TRUE,FALSE)) %>%
  # next filter based on best SENS
  filter(Best_Sp == TRUE | Thres == TRUE)

data <- data %>%
  group_by(Resample,ML,Condition, Thres) %>%
  # best tuning strategy for each combination of sampling, ML and Condition
  mutate(Best_Sens = ifelse(mean_SPEC == max(mean_SPEC),TRUE,FALSE)) %>%
  # last filter based on best SPEC
  filter(Best_Sens == TRUE | Thres == TRUE)
```

Pre processing with only best 32 models

Models are ordered in the way that it is structured across all MLs

```
data <- data %>%
  ungroup()

# Make sure for each ML that the models are in the right order in which I want them to be depicted in t
# This step is repeated for each of the 4 MLs

data_long_rf <- data %>%
  filter(ML == "RF") %>%
  mutate(Order =
           case_when(Condition == "Full_data" & Resample == "original" & Thres == TRUE ~ 1,
                           Condition == "Full_data" & Resample == "original" & Thres == FALSE ~ 2,
                           Condition == "Full_data" & Resample == "smote" & Thres == FALSE ~ 3,
                           Condition == "Full_data" & Resample == "up" & Thres == FALSE ~ 4,
                           Condition == "Lifestyle" & Resample == "original" & Thres == TRUE ~ 5,
                           Condition == "Lifestyle" & Resample == "original" & Thres == FALSE ~ 6,
                           Condition == "Lifestyle" & Resample == "smote" & Thres == FALSE ~ 7,
                           Condition == "Lifestyle" & Resample == "up" & Thres == FALSE ~ 8)) %>%
  arrange(Order) %>% distinct(Order, .keep_all = TRUE)
# If a model would produce the exact same estimates for multiple models only one is chosen.

colnames(data_long_rf) <- paste(colnames(data_long_rf),"RF",sep="_")

data_long_rf <- data_long_rf %>%
  select(mean_AUC_RF,mean_SENS_RF,mean_SPEC_RF,
         sd_AUC_RF,sd_SENS_RF,sd_SPEC_RF)

data_long_svm <- data %>%
  filter(ML == "SVM") %>%
  mutate(Order =
           case_when(Condition == "Full_data" & Resample == "original" & Thres == TRUE ~ 1,
                           Condition == "Full_data" & Resample == "original" & Thres == FALSE ~ 2,
                           Condition == "Full_data" & Resample == "smote" & Thres == FALSE ~ 3,
                           Condition == "Full_data" & Resample == "up" & Thres == FALSE ~ 4,
                           Condition == "Lifestyle" & Resample == "original" & Thres == TRUE ~ 5,
                           Condition == "Lifestyle" & Resample == "original" & Thres == FALSE ~ 6,
                           Condition == "Lifestyle" & Resample == "smote" & Thres == FALSE ~ 7,
                           Condition == "Lifestyle" & Resample == "up" & Thres == FALSE ~ 8)) %>%
  arrange(Order) %>% distinct(Order, .keep_all = TRUE)

colnames(data_long_svm) <- paste(colnames(data_long_svm),"SVM",sep="_")

data_long_svm <- data_long_svm %>%
  select(mean_AUC_SVM,mean_SENS_SVM,mean_SPEC_SVM,
         sd_AUC_SVM,sd_SENS_SVM,sd_SPEC_SVM)

data_long_xgb <- data %>%
  filter(ML == "XGB") %>%
  mutate(Order =
           case_when(Condition == "Full_data" & Resample == "original" & Thres == TRUE ~ 1,
                           Condition == "Full_data" & Resample == "original" & Thres == FALSE ~ 2,
                           Condition == "Full_data" & Resample == "smote" & Thres == FALSE ~ 3,
                           Condition == "Full_data" & Resample == "up" & Thres == FALSE ~ 4,
                           Condition == "Lifestyle" & Resample == "original" & Thres == TRUE ~ 5,
                           Condition == "Lifestyle" & Resample == "original" & Thres == FALSE ~ 6,
                           Condition == "Lifestyle" & Resample == "smote" & Thres == FALSE ~ 7,
                           Condition == "Lifestyle" & Resample == "up" & Thres == FALSE ~ 8)) %>%
  arrange(Order) %>% distinct(Order, .keep_all = TRUE)

colnames(data_long_xgb) <- paste(colnames(data_long_xgb),"XGB",sep="_")

data_long_xgb <- data_long_xgb %>%
  select(mean_AUC_XGB,mean_SENS_XGB,mean_SPEC_XGB,
         sd_AUC_XGB,sd_SENS_XGB,sd_SPEC_XGB)
```

```
data_long_lasso <- data %>%
  filter(ML == "Lasso") %>%
  mutate(Order =
           case_when(Condition == "Full_data" & Resample == "original" & Thres == TRUE ~ 1,
                            Condition == "Full_data" & Resample == "original" & Thres == FALSE ~ 2,
                            Condition == "Full_data" & Resample == "smote" & Thres == FALSE ~ 3,
                            Condition == "Full_data" & Resample == "up" & Thres == FALSE ~ 4,
                            Condition == "Lifestyle" & Resample == "original" & Thres == TRUE ~ 5,
                            Condition == "Lifestyle" & Resample == "original" & Thres == FALSE ~ 6,
                            Condition == "Lifestyle" & Resample == "smote" & Thres == FALSE ~ 7,
                            Condition == "Lifestyle" & Resample == "up" & Thres == FALSE ~ 8)) %>%
  arrange(Order) %>% distinct(Order, .keep_all = TRUE)


colnames(data_long_lasso) <- paste(colnames(data_long_lasso),"Lasso",sep="_")

data_long_lasso <- data_long_lasso %>%
  select(mean_AUC_Lasso,mean_SENS_Lasso,mean_SPEC_Lasso,
         sd_AUC_Lasso,sd_SENS_Lasso,sd_SPEC_Lasso)

all_table <- cbind(data_long_rf,data_long_svm,data_long_xgb,data_long_lasso)
# The four MLs are combined and then transposed so I have it exactly in the format needed for my table
all_table <- t(all_table)
```

TABLE USED IN STUDY

Column names in numbers refer to the following model building conditions.

Condition == “Full_data” & Resample == “original” & Thres == TRUE ~ 1, Condition == “Full_data” & Resample == “original” & Thres == FALSE ~ 2, Condition == “Full_data” & Resample == “smote” & Thres == FALSE ~ 3, Condition == “Full_data” & Resample == “up” & Thres == FALSE ~ 4, Condition == “Lifestyle” & Resample == “original” & Thres == TRUE ~ 5, Condition == “Lifestyle” & Resample == “original” & Thres == FALSE ~ 6, Condition == “Lifestyle” & Resample == “smote” & Thres == FALSE ~ 7, Condition == “Lifestyle” & Resample == “up” & Thres == FALSE ~ 8)

```
# Creating Latex code based on x_table package
tbl <- xtable(all_table)
tbl
```

```
## % latex table generated in R 4.2.2 by xtable 1.8-4 package
## % Thu May 11 14:56:01 2023
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrrrrrr}
##   \hline
##  & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
##   \hline
## mean\_AUC\_RF & 0.90 & 0.91 & 0.88 & 0.88 & 0.81 & 0.82 & 0.82 & 0.81 \\
##   mean\_SENS\_RF & 0.33 & 0.89 & 0.88 & 0.88 & 0.21 & 0.83 & 0.81 & 0.80 \\
##   mean\_SPEC\_RF & 0.97 & 0.83 & 0.78 & 0.75 & 0.97 & 0.69 & 0.71 & 0.72 \\
##   sd\_AUC\_RF & 0.04 & 0.03 & 0.01 & 0.01 & 0.04 & 0.03 & 0.02 & 0.04 \\
##   sd\_SENS\_RF & 0.20 & 0.08 & 0.06 & 0.04 & 0.05 & 0.05 & 0.07 & 0.08 \\
##   sd\_SPEC\_RF & 0.03 & 0.05 & 0.05 & 0.01 & 0.02 & 0.07 & 0.06 & 0.04 \\
##   mean\_AUC\_SVM & 0.89 & 0.89 & 0.86 & 0.88 & 0.78 & 0.78 & 0.78 & 0.78 \\
##   mean\_SENS\_SVM & 0.35 & 0.86 & 0.89 & 0.82 & 0.06 & 0.82 & 0.80 & 0.79 \\
##   mean\_SPEC\_SVM & 0.97 & 0.87 & 0.70 & 0.84 & 1.00 & 0.60 & 0.67 & 0.66 \\
##   sd\_AUC\_SVM & 0.04 & 0.04 & 0.04 & 0.02 & 0.05 & 0.05 & 0.05 & 0.04 \\
##   sd\_SENS\_SVM & 0.13 & 0.02 & 0.06 & 0.04 & 0.02 & 0.07 & 0.08 & 0.03 \\
##   sd\_SPEC\_SVM & 0.02 & 0.08 & 0.04 & 0.02 & 0.00 & 0.05 & 0.10 & 0.07 \\
##   mean\_AUC\_XGB & 0.89 & 0.90 & 0.89 & 0.90 & 0.81 & 0.81 & 0.79 & 0.78 \\
##   mean\_SENS\_XGB & 0.52 & 0.86 & 0.89 & 0.87 & 0.31 & 0.79 & 0.84 & 0.82 \\
##   mean\_SPEC\_XGB & 0.95 & 0.82 & 0.78 & 0.81 & 0.97 & 0.74 & 0.62 & 0.64 \\
##   sd\_AUC\_XGB & 0.04 & 0.06 & 0.04 & 0.04 & 0.03 & 0.03 & 0.04 & 0.04 \\
##   sd\_SENS\_XGB & 0.10 & 0.06 & 0.03 & 0.09 & 0.04 & 0.05 & 0.05 & 0.03 \\
##   sd\_SPEC\_XGB & 0.03 & 0.06 & 0.07 & 0.06 & 0.00 & 0.04 & 0.03 & 0.05 \\
##   mean\_AUC\_Lasso & 0.88 & 0.88 & 0.85 & 0.86 & 0.58 & 0.58 & 0.58 & 0.58 \\
##   mean\_SENS\_Lasso & 0.38 & 0.85 & 0.83 & 0.82 & 0.22 & 0.78 & 0.80 & 0.73 \\
##   mean\_SPEC\_Lasso & 0.96 & 0.78 & 0.74 & 0.76 & 0.81 & 0.45 & 0.41 & 0.49 \\
##   sd\_AUC\_Lasso & 0.06 & 0.05 & 0.07 & 0.05 & 0.05 & 0.06 & 0.03 & 0.06 \\
##   sd\_SENS\_Lasso & 0.04 & 0.04 & 0.05 & 0.06 & 0.30 & 0.08 & 0.12 & 0.04 \\
##   sd\_SPEC\_Lasso & 0.01 & 0.06 & 0.14 & 0.05 & 0.17 & 0.03 & 0.10 & 0.04 \\
##    \hline
## \end{tabular}
## \end{table}
```

ROC

Load the probabilities for each patient in the test data set

```
probabilities_full <- readRDS("Data/probs_full_123.RData")
probabilities_full_1 <- readRDS("Data/probs_full_41263.RData")
probabilities_full_2 <- readRDS("Data/probs_full_2408.RData")
probabilities_life <- readRDS("Data/probs_lifestyle_123.RData")
probabilities_life_1 <- readRDS("Data/probs_lifestyle_41263.RData")
probabilities_life_2 <- readRDS("Data/probs_lifestyle_2408.RData")
```

Pre-Processing - Making it possible to have distinct MLs, Sampling and Tuning strategies, and data conditions

This step needs to be repeated for each of the three folds. The same code with different data sets is shown here.

For 1. Fold

For each fold some further pre-processing is needed.

```r
probabilities_full$rf$ML <- "RF"
probabilities_full$svm$ML <- "SVM"
probabilities_full$xgb$ML <- "XGB"
probabilities_full$lasso$ML <- "Lasso"

probabilities_life$rf$ML <- "RF"
probabilities_life$svm$ML <- "SVM"
probabilities_life$xgb$ML <- "XGB"
probabilities_life$lasso$ML <- "Lasso"

  # Only for Full data condition
probs_full_new <- rbind(probabilities_full$rf,probabilities_full$svm,
                        probabilities_full$xgb,probabilities_full$lasso)

probs_full_new_raw <- probs_full_new %>%
  pivot_longer(cols = c("original","original_cost","smote_def",
                        "up_def","up_own","smote_own","original_own","up_ran",
                        "smote_ran","original_ran"),
               names_to = "model", values_to = "Probs")

# Here the ROC curves are caluclated based on yardstick package
probs_full_new <- probs_full_new_raw %>%
group_by(model,ML) %>%
  roc_curve(outcome,Probs, event_level="second") # here the ROC curves are obtained.
probs_full_new$Condition <- "Full_data"

probs_full_new_auc <- probs_full_new_raw %>%
group_by(model,ML) %>%
  roc_auc(outcome,Probs, event_level="second") %>%
  rename("auc" = ".estimate") %>%
  subset( select = -c(.metric,.estimator))

probs_full_new <- left_join(probs_full_new,
                            probs_full_new_auc, by= c("model","ML"))
  # THis needs to be repeated for Lifestyle Condition

probs_life_new <- rbind(probabilities_life$rf,probabilities_life$svm,
                        probabilities_life$xgb,probabilities_life$lasso)

probs_life_new_raw <- probs_life_new %>%
  pivot_longer(cols = c("original","original_cost","smote_def",
                        "up_def","up_own","smote_own","original_own","up_ran",
                        "smote_ran","original_ran"),
               names_to = "model", values_to = "Probs")

probs_life_new <- probs_life_new_raw %>%
group_by(model,ML) %>%
  roc_curve(outcome,Probs, event_level="second")
probs_life_new$Condition <- "Lifestyle"

probs_life_new_auc <- probs_life_new_raw %>%
group_by(model,ML) %>%
  roc_auc(outcome,Probs, event_level="second") %>%
  rename("auc" = ".estimate") %>%
  subset( select = -c(.metric,.estimator))

probs_life_new <- left_join(probs_life_new,
                            probs_life_new_auc, by= c("model","ML"))

probs_new <- rbind(probs_full_new,probs_life_new)

probs_new$Fold <- "Fold 1"
```

For 2. Fold

```
probabilities_full_1$rf$ML <- "RF"
probabilities_full_1$svm$ML <- "SVM"
probabilities_full_1$xgb$ML <- "XGB"
probabilities_full_1$lasso$ML <- "Lasso"

probabilities_life_1$rf$ML <- "RF"
probabilities_life_1$svm$ML <- "SVM"
probabilities_life_1$xgb$ML <- "XGB"
probabilities_life_1$lasso$ML <- "Lasso"


probs_full_new_1 <- rbind(probabilities_full_1$rf,probabilities_full_1$svm,
                          probabilities_full_1$xgb,probabilities_full_1$lasso)

probs_full_new_raw_1 <- probs_full_new_1 %>%
  pivot_longer(cols = c("original","original_cost","smote_def",
                        "up_def","up_own","smote_own","original_own","up_ran",
                        "smote_ran","original_ran"),
               names_to = "model", values_to = "Probs")

probs_full_new_1 <- probs_full_new_raw_1 %>%
group_by(model,ML) %>%
  roc_curve(outcome,Probs, event_level="second")
probs_full_new_1$Condition <- "Full_data"

probs_full_new_1_auc <- probs_full_new_raw_1 %>%
group_by(model,ML) %>%
  roc_auc(outcome,Probs, event_level="second") %>%
  rename("auc" = ".estimate") %>%
  subset( select = -c(.metric,.estimator))

probs_full_new_1 <- left_join(probs_full_new_1,
                              probs_full_new_1_auc, by= c("model","ML"))


probs_life_new_1 <- rbind(probabilities_life_1$rf,probabilities_life_1$svm,
                          probabilities_life_1$xgb,probabilities_life_1$lasso)

probs_life_new_raw_1 <- probs_life_new_1 %>%
  pivot_longer(cols = c("original","original_cost","smote_def",
                        "up_def","up_own","smote_own","original_own","up_ran",
                        "smote_ran","original_ran"),
               names_to = "model", values_to = "Probs")

probs_life_new_1 <- probs_life_new_raw_1 %>%
group_by(model,ML) %>%
  roc_curve(outcome,Probs, event_level="second")
probs_life_new_1$Condition <- "Lifestyle"

probs_life_new_1_auc <- probs_life_new_raw_1 %>%
group_by(model,ML) %>%
  roc_auc(outcome,Probs, event_level="second") %>%
  rename("auc" = ".estimate") %>%
  subset( select = -c(.metric,.estimator))

probs_life_new_1 <- left_join(probs_life_new_1,
                              probs_life_new_1_auc, by= c("model","ML"))

probs_new_fold_1 <- rbind(probs_full_new_1,probs_life_new_1)

probs_new_fold_1$Fold <- "Fold 2"
```

For 3. Fold

```
probabilities_full_2$rf$ML <- "RF"
probabilities_full_2$svm$ML <- "SVM"
probabilities_full_2$xgb$ML <- "XGB"
probabilities_full_2$lasso$ML <- "Lasso"

probabilities_life_2$rf$ML <- "RF"
probabilities_life_2$svm$ML <- "SVM"
probabilities_life_2$xgb$ML <- "XGB"
probabilities_life_2$lasso$ML <- "Lasso"


probs_full_new_2 <- rbind(probabilities_full_2$rf,probabilities_full_2$svm,
                          probabilities_full_2$xgb,probabilities_full_2$lasso)

probs_full_new_raw_2 <- probs_full_new_2 %>%
  pivot_longer(cols = c("original","original_cost","smote_def",
                        "up_def","up_own","smote_own","original_own",
                        "up_ran","smote_ran","original_ran"),
               names_to = "model", values_to = "Probs")

probs_full_new_2 <- probs_full_new_raw_2 %>%
group_by(model,ML) %>%
  roc_curve(outcome,Probs, event_level="second")
probs_full_new_2$Condition <- "Full_data"

probs_full_new_2_auc <- probs_full_new_raw_2 %>%
group_by(model,ML) %>%
  roc_auc(outcome,Probs, event_level="second") %>%
  rename("auc" = ".estimate") %>%
  subset( select = -c(.metric,.estimator))

probs_full_new_2 <- left_join(probs_full_new_2,
                              probs_full_new_2_auc, by= c("model","ML"))


probs_life_new_2 <- rbind(probabilities_life_2$rf,probabilities_life_2$svm,
                          probabilities_life_2$xgb,probabilities_life_2$lasso)

probs_life_new_raw_2 <- probs_life_new_2 %>%
  pivot_longer(cols = c("original","original_cost","smote_def",
                        "up_def","up_own","smote_own","original_own",
                        "up_ran","smote_ran","original_ran"),
               names_to = "model", values_to = "Probs")

probs_life_new_2 <- probs_life_new_raw_2 %>%
group_by(model,ML) %>%
  roc_curve(outcome,Probs, event_level="second")
probs_life_new_2$Condition <- "Lifestyle"

probs_life_new_2_auc <- probs_life_new_raw_2 %>%
group_by(model,ML) %>%
  roc_auc(outcome,Probs, event_level="second") %>%
  rename("auc" = ".estimate") %>%
  subset( select = -c(.metric,.estimator))

probs_life_new_2 <- left_join(probs_life_new_2,
                              probs_life_new_2_auc, by= c("model","ML"))

probs_new_fold_2 <- rbind(probs_full_new_2,probs_life_new_2)

probs_new_fold_2$Fold <- "Fold 3"
```

Combining all three folds

```
probs_final <- rbind(probs_new,probs_new_fold_1,probs_new_fold_2)
```

Last Pre-processing

Here the means are calculated to have the best performing model based on all model building choices, corresponding to the best model selected for the tables.

```
full$Fold <- "Fold 1"
full_1$Fold <- "Fold 2"
full_2$Fold <- "Fold 3"
life$Fold <- "Fold 1"
life_1$Fold <- "Fold 2"
```

```r
life_2$Fold <- "Fold 3"
for_roc <- rbind(full,full_1,full_2,life,life_1,life_2)
for_roc <- for_roc %>%
  group_by(model,Condition,ML) %>%
  mutate(mean_AUC = round(mean(AUC),3),
         mean_SENS = round(mean(Sensitivity),3),
         mean_SPEC = round(mean(Specificity),3)) %>%
  ungroup()
```

Again only the best performing Tuning and Sampling strategy is chosen to be depicted. However, this is depicted for each of the data folds to highlight the data variability in this data set.

```r
probs_new_final <- left_join(probs_final,for_roc,
                             by= c("model","ML","Condition","Fold"))
# combine the roc curves data with the tables data

probs_new_final <- probs_new_final %>%
  mutate(Condition = recode(Condition, "Full_data" = "Full data",
                            "Lifestyle" = "Lifestyle data")) %>%
  group_by(ML,Condition,Fold) %>%
  mutate(Best = ifelse(mean_AUC == max(mean_AUC),TRUE,FALSE)) %>%
  # first filter based on best AUC, same as Tables
  filter(Best == TRUE)

probs_new_final <- probs_new_final %>%
  group_by(ML,Condition,Fold) %>%
  mutate(Best_Sp = ifelse(mean_SENS == max(mean_SENS),TRUE,FALSE)) %>%
  # next filter based on best S
  filter(Best_Sp == TRUE)

probs_new_final <- probs_new_final %>%
  group_by(ML,Condition,Fold) %>%
  mutate(Best_Sens = ifelse(mean_SPEC == max(mean_SPEC),TRUE,FALSE)) %>%
  filter(Best_Sens == TRUE)
```

These extra dfs are containing information about the mean values which will be included in the plot.

```r
auc_values_full_mean <- probs_new_final %>%
  group_by(ML, Condition, Fold) %>%
  summarise(AUC = max(mean_AUC)) %>%
  filter(Condition == "Full data") %>%
  distinct(AUC, .keep_all = TRUE)
```

```
## ‘summarise()‘ has grouped output by ’ML’, ’Condition’. You can override using
## the ‘.groups‘ argument.
```

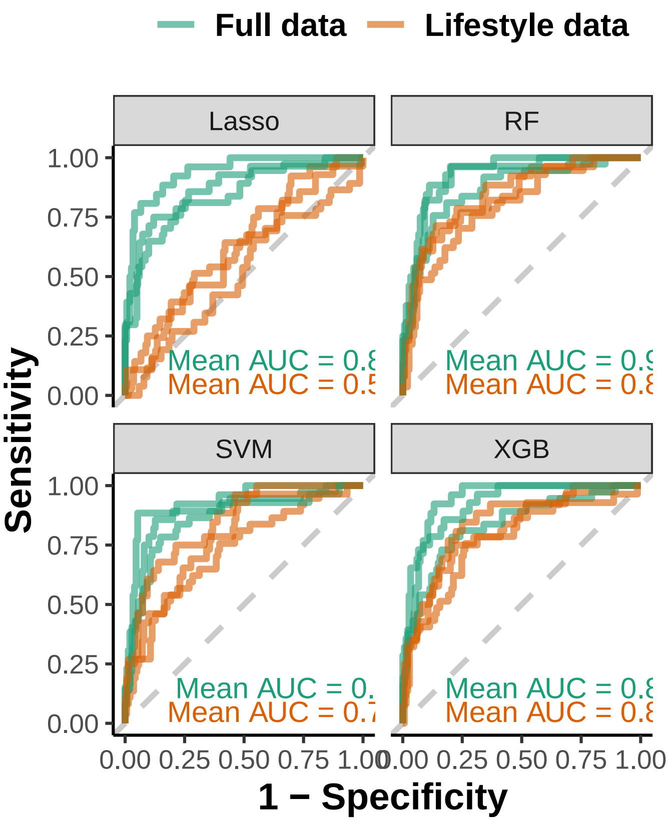
```r
auc_values_life_mean <- probs_new_final %>%
  group_by(ML, Condition, Fold) %>%
  summarise(AUC = max(mean_AUC)) %>%
  filter(Condition == "Lifestyle data") %>%
  distinct(AUC, .keep_all = TRUE)
```

```
## ‘summarise()‘ has grouped output by ’ML’, ’Condition’. You can override using
## the ‘.groups‘ argument.
```

ROC plot across all folds

The code which produces exactly the code needed for the figure in the report.

```r
ROC_compare_fold <- probs_new_final%>%
  ggplot(aes(1 - specificity, sensitivity,
             fill = Fold, color=Condition)) +
  geom_abline(lty = 2, color = "gray",
              size = 1,alpha = 0.8) +
  geom_path(alpha = 0.6, size = 1.2) +
  coord_equal() +
  labs(x = "1 - Specificity", y = "Sensitivity") +
  facet_wrap(~ ML) +
    geom_text(data = auc_values_full_mean,
              aes(label = paste0("Mean AUC = ", AUC)),
            x = 0.7, y = 0.15, show.legend = FALSE) +
  geom_text(data = auc_values_life_mean,
            aes(label = paste0("Mean AUC = ", AUC)),
            x = 0.7, y = 0.05, show.legend = FALSE) +
  theme_bw(base_size = 12) +
  theme(legend.position = "top",
        panel.border = element_blank(),  # remove panel borders
        panel.grid.major = element_blank(),  # remove major grid lines
        panel.grid.minor = element_blank(),  # remove minor grid lines
        axis.line = element_line(),  # set axis lines to bold
        axis.text = element_text(),  # set axis text to bold
        axis.title = element_text(size = 14, face = "bold"),  # set axis title to bold
        plot.background = element_blank(),  # remove plot background
        panel.background = element_blank(),  # remove panel background
        legend.text = element_text(size = 12, face = "bold"),  # set legend text to bold
        legend.title = element_blank()  # remove legend title
  ) +
  scale_color_brewer(palette = "Dark2")
print(ROC_compare_fold)
```



Descriptives Table

Estimates are obtained for all included patients on baseline stats.

```
full_data <- read_csv("Data/full_data.csv") # raw data set is loaded
```

```
## Rows: 713 Columns: 75
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): wday, weekend
## dbl (73): Night_class_hypo_out, Night_class_hyper_30, patient, mean_0_1, mea...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Preparation of dataset needed for descriptive stats
names <- c("patient","Age","Height","Weight","BMI","HbA1c","T2D_Dur")
descriptives <- full_data %>%
  select(patient,Age,Height,Weight,BMI,HbA1c,T2D_Dur)
names(descriptives) <- c(names)

DT <- data.table(descriptives)
DT[, sapply(.SD, function(x) list(mean=round(mean(x), 2))), by=patient]
```

```
##     patient Age.mean Height.mean Weight.mean BMI.mean HbA1c.mean T2D_Dur.mean
##  1:    1001       57         161          96    37.04         48           16
##  2:     601       74         165          75    27.55         66           19
##  3:     605       53         185          73    21.33         51            8
##  4:     606       68         183         116    34.64         59           13
##  5:     607       42         182          93    28.08         51            6
##  6:     608       77         165          90    33.06         94           21
##  7:     609       68         171          88    30.09         57           24
##  8:     611       64         176         101    32.61         58           21
##  9:     616       79         173          94    31.41         54            8
## 10:     617       68         180          81    25.00         53           36
## 11:     618       48         187         111    31.74         74           15
## 12:     621       65         176         110    35.51         57           12
## 13:     624       63         182          97    29.28         50            9
## 14:     628       67         188          78    22.07         50           17
## 15:     629       45         182          97    29.28         38            6
## 16:     630       71         179         118    36.83         42           28
## 17:     633       73         166         118    42.82         45            6
## 18:     634       71         161          79    30.48         63           14
## 19:     635       68         177         125    39.90         53           26
## 20:     636       64         181          93    28.39         57           24
## 21:     638       64         179          93    29.03         56           14
## 22:     639       75         158          83    33.25         77           11
## 23:     640       64         169          96    33.61         54           17
## 24:     642       81         162          71    27.05         64           26
## 25:     643       56         167          92    32.99         43            9
## 26:     644       71         182          90    27.17         42            6
## 27:     645       51         175         107    34.94         64            6
## 28:     648       66         170          67    23.18         57           13
## 29:     649       68         179          97    30.27         54           20
## 30:     650       63         177          96    30.64         57           23
## 31:     651       72         167         100    35.86         62           42
## 32:     658       47         170          76    26.30         70           23
## 33:     660       38         185         102    29.80         44           18
## 34:     662       70         161          99    38.19         51            6
## 35:     665       73         149          65    29.28         63           15
## 36:     666       52         170          93    32.18         62           12
## 37:     667       78         162          76    28.96         65           17
## 38:     671       61         195         122    32.08         69           22
## 39:     672       51         169          61    21.36         50            7
## 40:     674       68         176          98    31.64         51           14
## 41:     680       67         153          96    41.01         75           14
## 42:     683       72         168          75    26.57         53           40
## 43:     691       58         166         130    47.18         64           13
## 44:     695       80         168          78    27.64         70           25
## 45:     702       62         169          76    26.61         46            8
## 46:     709       68         161          88    33.95         53           22
## 47:     716       73         166          91    33.02         64           34
## 48:     745       71         169          80    28.01          0           11
## 49:     748       65         165          63    23.14         52           33
## 50:     751       62         189          84    23.52         54           13
## 51:     574       49         178          94    29.67         59           11
## 52:     615       67         180          92    28.40         48           23
## 53:     637       71         185         113    33.02         60           23
## 54:     641       73         189         102    28.55         46            6
## 55:     664       53         184          97    28.65         68           13
## 56:     669       74         155         106    44.12         63           18
## 57:     676       63         176          85    27.44         58           20
## 58:     677       59         177          96    30.64         71           24
## 59:     690       56         171          87    29.75         52           29
## 60:     692       72         174          97    32.04         56           13
## 61:     703       71         165          64    23.51         54           24
## 62:     708       71         177          74    23.62         56           20
## 63:     712       63         169          98    34.31         76            8
## 64:     729       76         179         105    32.77         51           34
## 65:     746       72         172          84    28.39         54            6
## 66:     749       53         171          86    29.41         60            7
## 67:     752       72         169          82    28.71         68           15
## 68:     753       63         173          90    30.07         48            3
## 69:     754       64         183         115    34.34         77            1
## 70:     598       81         168          76    26.93         72           38
## 71:     684       70         165          91    33.43         64           44
## 72:     687       70         166          99    35.93         66           30
## 73:     719       71         175          81    26.45         55           22
## 74:     699       85         164          82    30.49         54           29
## 75:     646       47         168          88    31.18         56           12
## 76:     647       50         182         115    34.72         52            7
##     patient Age.mean Height.mean Weight.mean BMI.mean HbA1c.mean T2D_Dur.mean
```

```
names(DT) <- c(names)
DT <- as.data.frame(DT)
```

Proportions and counts for factor variables

For factors a different approach is used and values need to be included manually.

```
table(full_data$Night_class_hypo_out)
```

```
##
##   0   1
## 600 113
```

```
round(table(full_data$Night_class_hypo_out)[1]/
        (table(full_data$Night_class_hypo_out)[1]+
           table(full_data$Night_class_hypo_out)[2]),3)
```

```
##     0
## 0.842
```

```
round(table(full_data$Night_class_hypo_out)[2]/
        (table(full_data$Night_class_hypo_out)[1]+
           table(full_data$Night_class_hypo_out)[2]),3)
```

```
##     1
## 0.158
```

```
table(full_data$Gender)
```

```
##
##   0   1
## 444 269
```

```
round(table(full_data$Gender)[1]/
        (table(full_data$Gender)[1]+table(full_data$Gender)[2]),3)
```

```
##     0
## 0.623
```

```
round(table(full_data$Gender)[2]/
        (table(full_data$Gender)[1]+table(full_data$Gender)[2]),3)
```

```
##     1
## 0.377
```

Latex code for descriptives

```
stargazer(DT,
          type = 'latex', min.max=FALSE, mean.sd = TRUE,
          nobs = FALSE, median = FALSE, iqr = FALSE,
          digits=1, align=T,
          title = "Summary Statistics")
```

```
##
## % Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac a
## % Date and time: Do, Mai 11, 2023 - 14:56:04
## % Requires LaTeX packages: dcolumn
## \begin{table}[!htbp] \centering
##   \caption{Summary Statistics}
##   \label{}
## \begin{tabular}{@{\extracolsep{5pt}}lD{.}{.}{-1} D{.}{.}{-1} }
## \\[-1.8ex]\hline
## \hline \\[-1.8ex]
## Statistic & \multicolumn{1}{c}{Mean} & \multicolumn{1}{c}{St. Dev.} \\
## \hline \\[-1.8ex]
## patient & 669.1 & 58.1 \\
## Age & 64.9 & 9.8 \\
## Height & 173.3 & 9.2 \\
## Weight & 91.4 & 14.8 \\
## BMI & 30.5 & 4.9 \\
## HbA1c & 56.7 & 11.9 \\
## T2D\_Dur & 17.7 & 9.9 \\
## \hline \\[-1.8ex]
## \end{tabular}
## \end{table}
```

Calibration plots - OPTIONAL

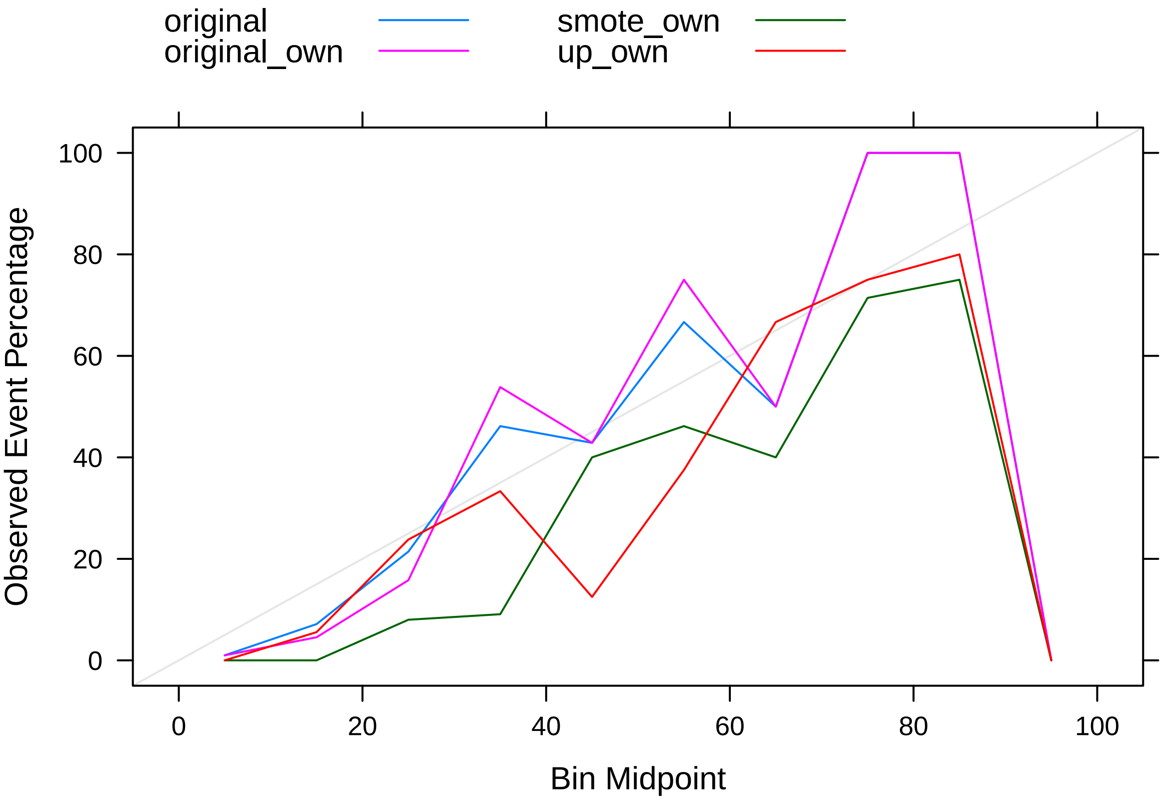
```
calibration_plots(probabilities_full$rf)
```

```
## Lade nötiges Paket: lattice
```

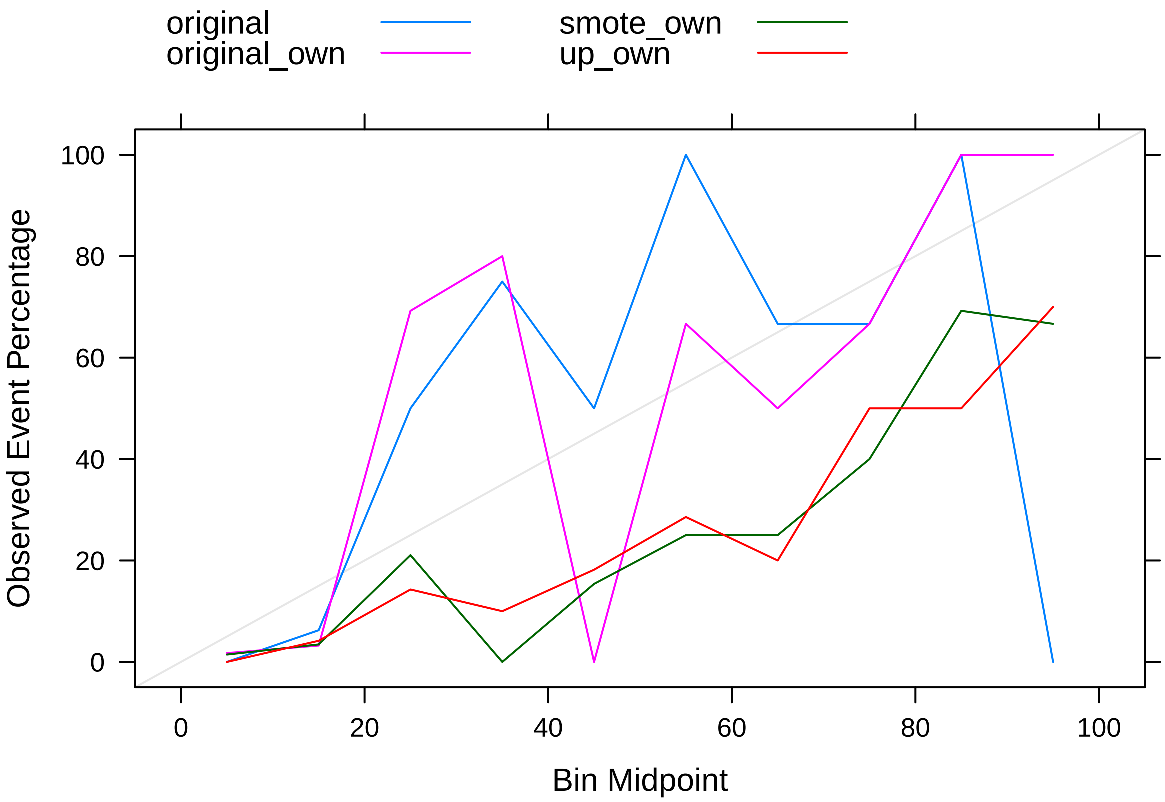
```
##
## Attache Paket: 'caret'
```

```
## Die folgenden Objekte sind maskiert von 'package:yardstick':
##
##     precision, recall, sensitivity, specificity
```

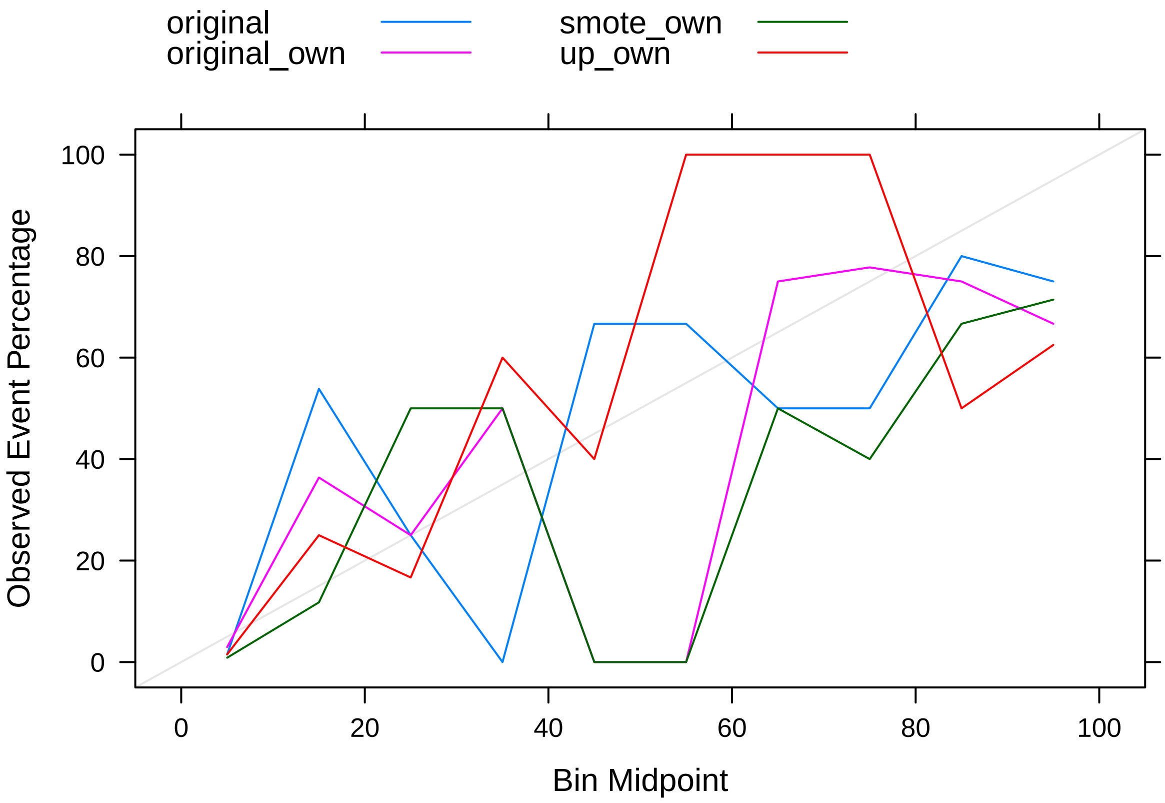
```
## Das folgende Objekt ist maskiert 'package:purrr':
##
##     lift
```



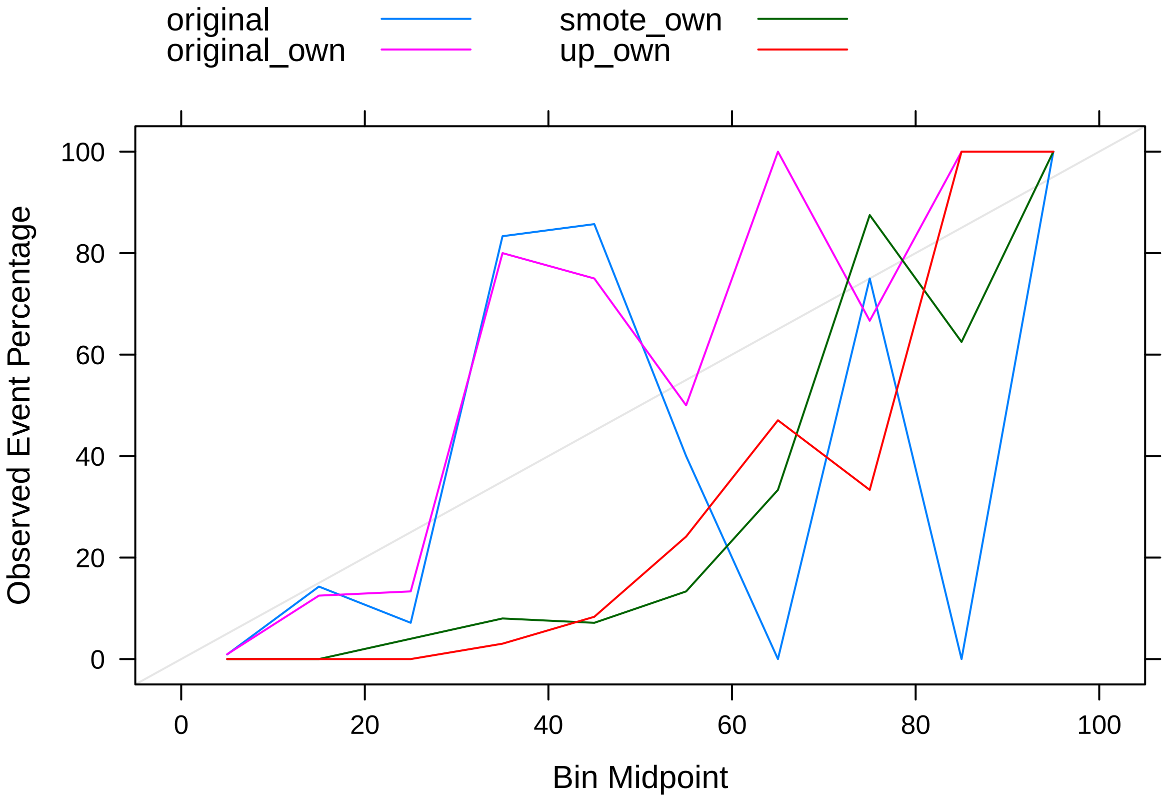
```
calibration_plots(probabilities_full$svm)
```



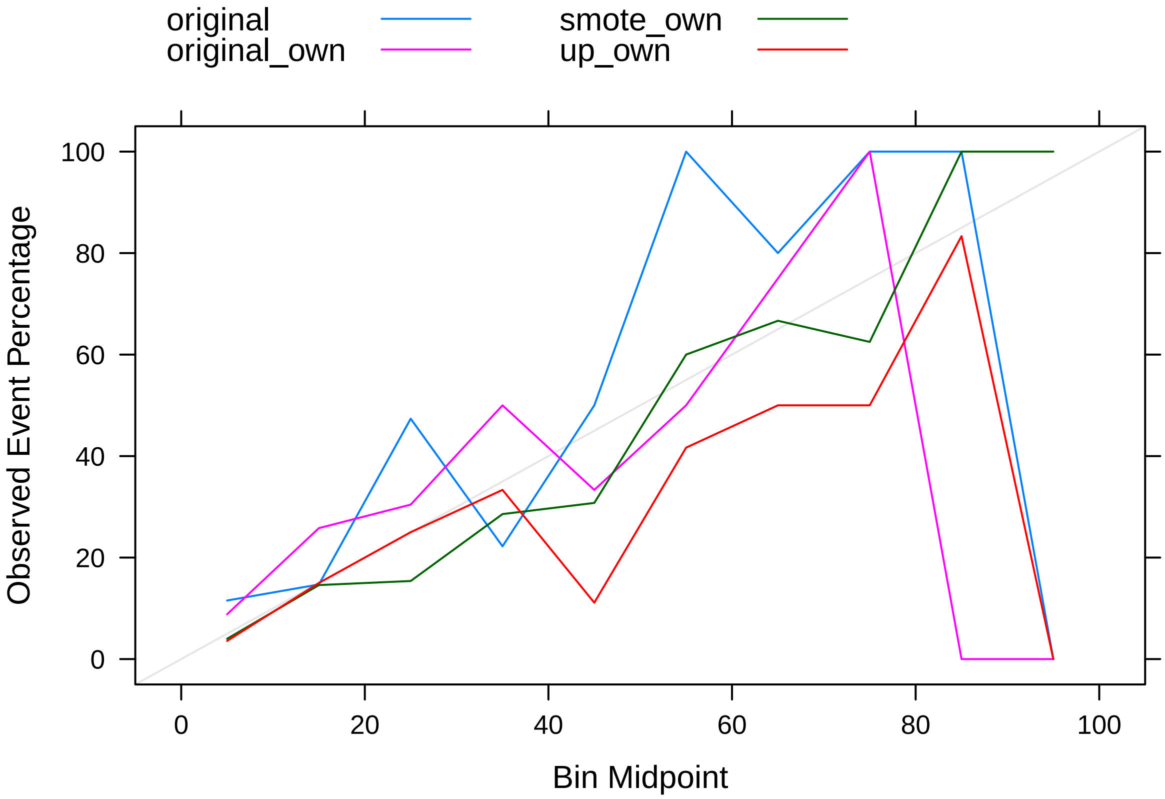
```
calibration_plots(probabilities_full$xgb)
```



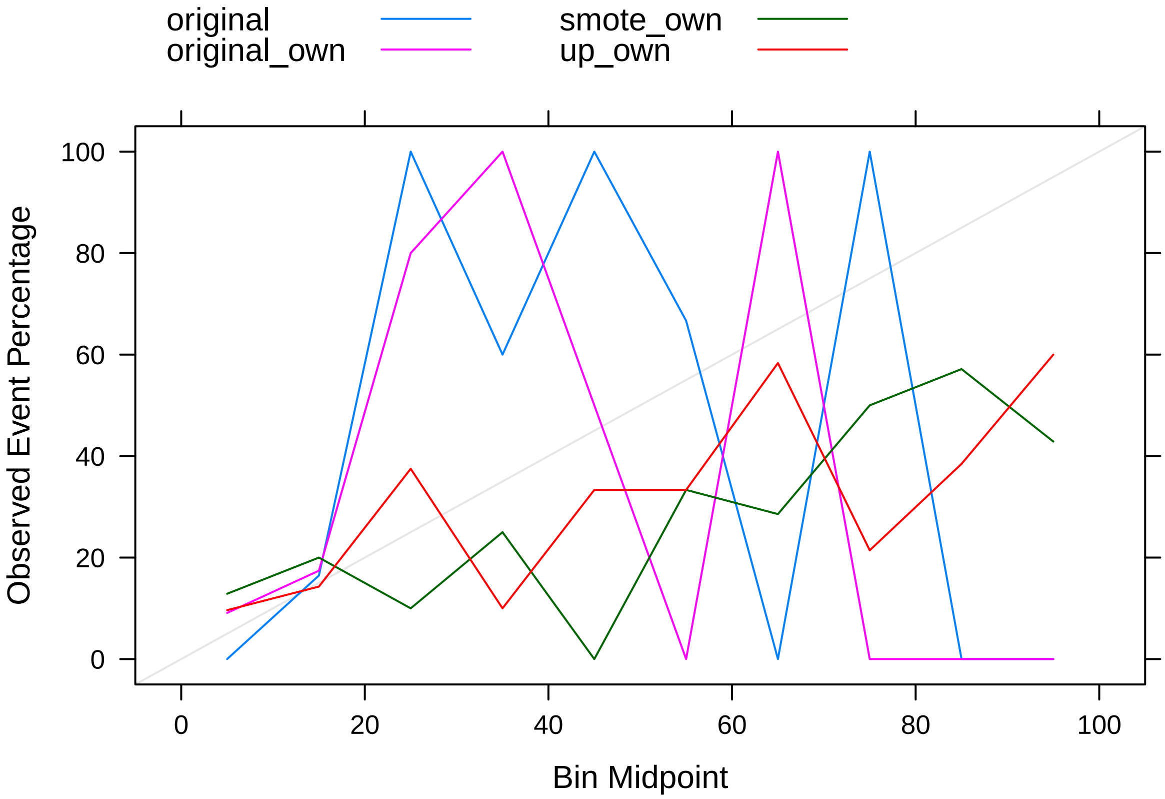
```
calibration_plots(probabilities_full$lasso)
```



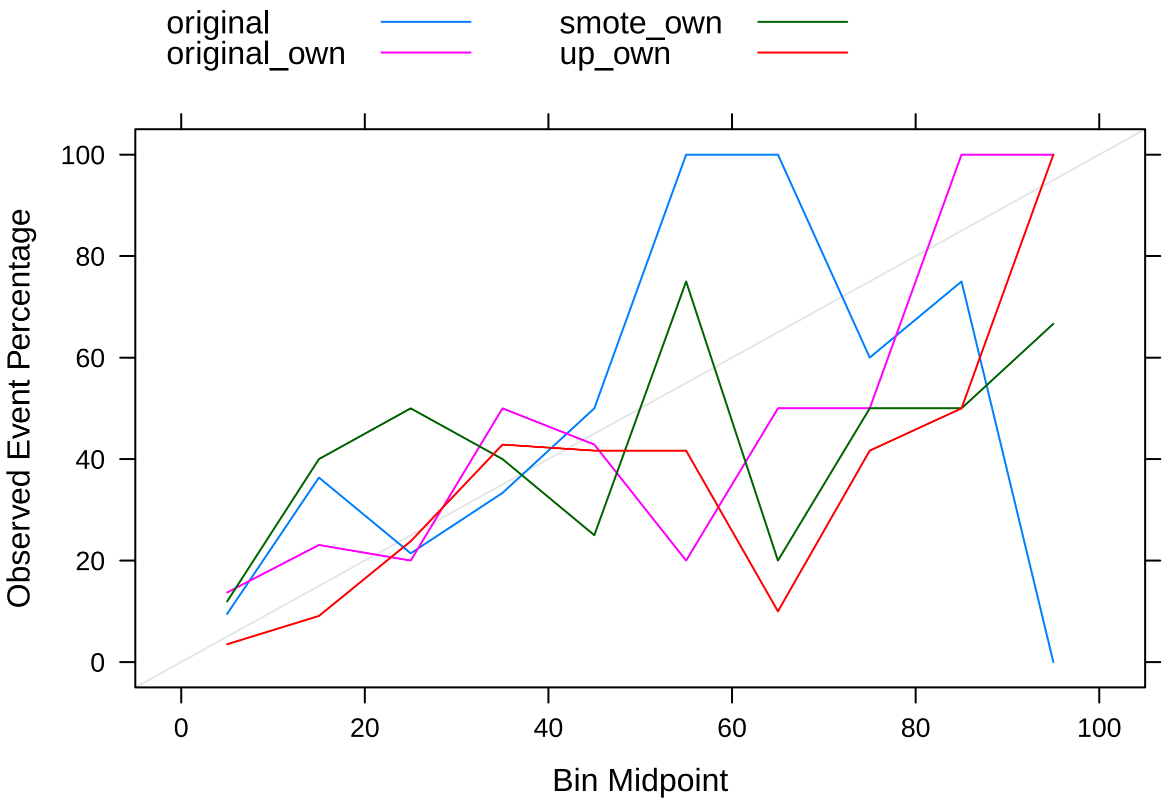
```
calibration_plots(probabilities_life$rf)
```



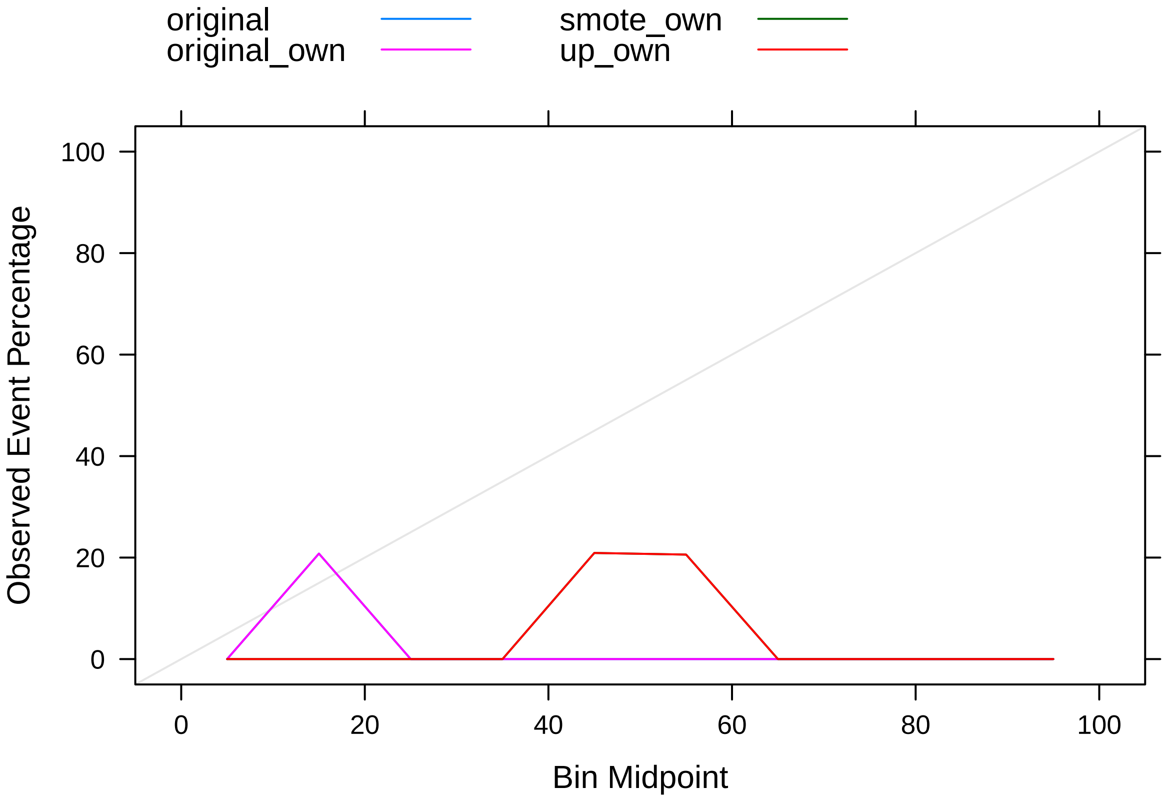
```
calibration_plots(probabilities_life$svm)
```



```
calibration_plots(probabilities_life$xgb)
```



```
calibration_plots(probabilities_life$lasso)
```



Outcome frequency and Distribution of days - OPTIONAL INFORMATION

```
descrip <- full_data
number_of_days_pp <- descrip %>%
  group_by(patient) %>%
  summarise(n= n())
sort(number_of_days_pp$n) # distributions of days per patient
```

```
##  [1]  3  4  4  5  5  5  5  5  5  6  6  6  6  6  6  7  7  7  7  7  8  8  8  8  8
## [26]  8  9  9  9  9  9  9  9  9  9  9 10 10 10 10 10 10 10 10 10 10 10 10 11 11
## [51] 11 11 11 11 11 11 12 12 12 12 12 12 12 12 12 13 13 13 13 13 13 13 14 14 14
## [76] 14
```

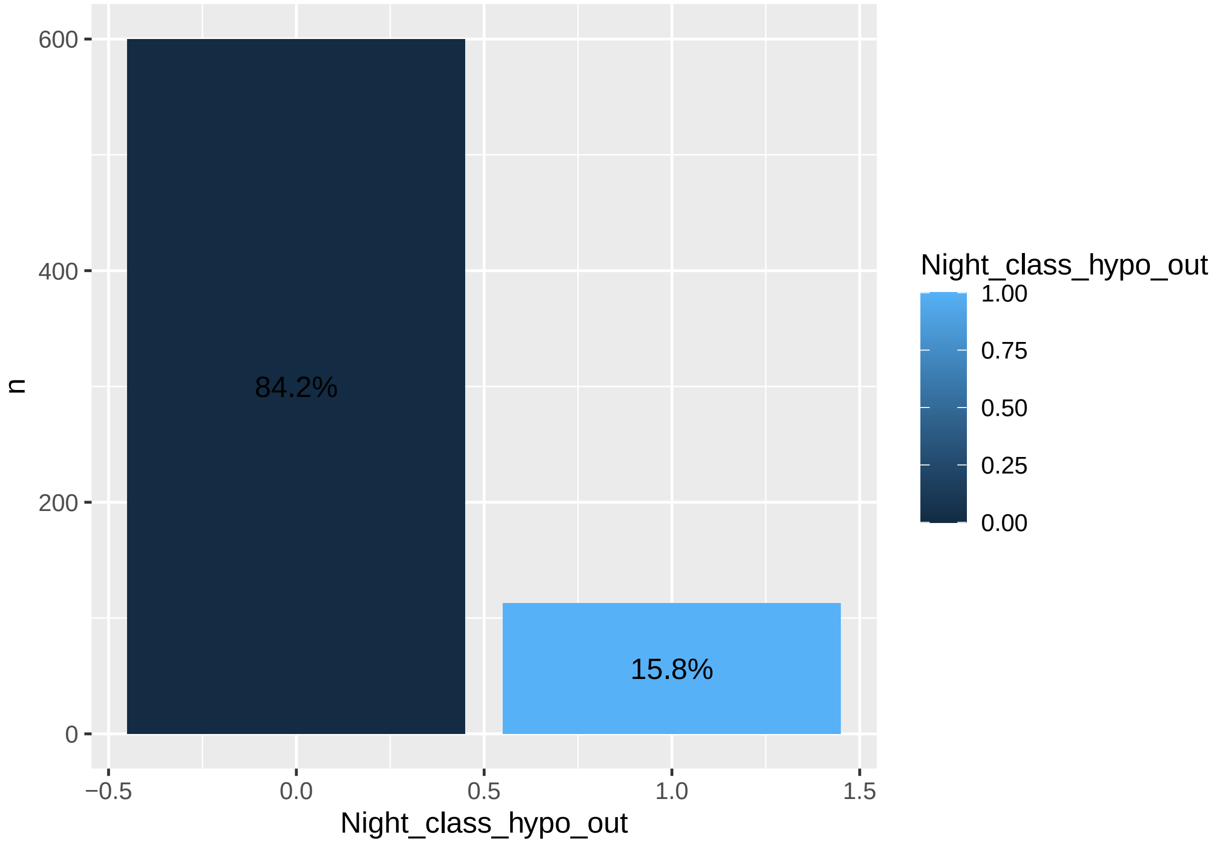
```
nrow(number_of_days_pp) # number of patients
```

```
## [1] 76
```

```
nrow(full_data) # number of individual observations
```

```
## [1] 713
```

```
# Visualization for distribution of outcome variable
ggplot(full_data %>%
      count(Night_class_hypo_out) %>% #Groups by team and role
       mutate(pct=n/sum(n)),       #Calculates % for each role
        aes(Night_class_hypo_out, n, fill=Night_class_hypo_out)) +
        geom_col(stat="identity", position="stack") +
  geom_text(aes(label=paste0(sprintf("%1.1f", pct*100),"%")),
            position=position_stack(vjust=0.5))
```



CGM visualisation

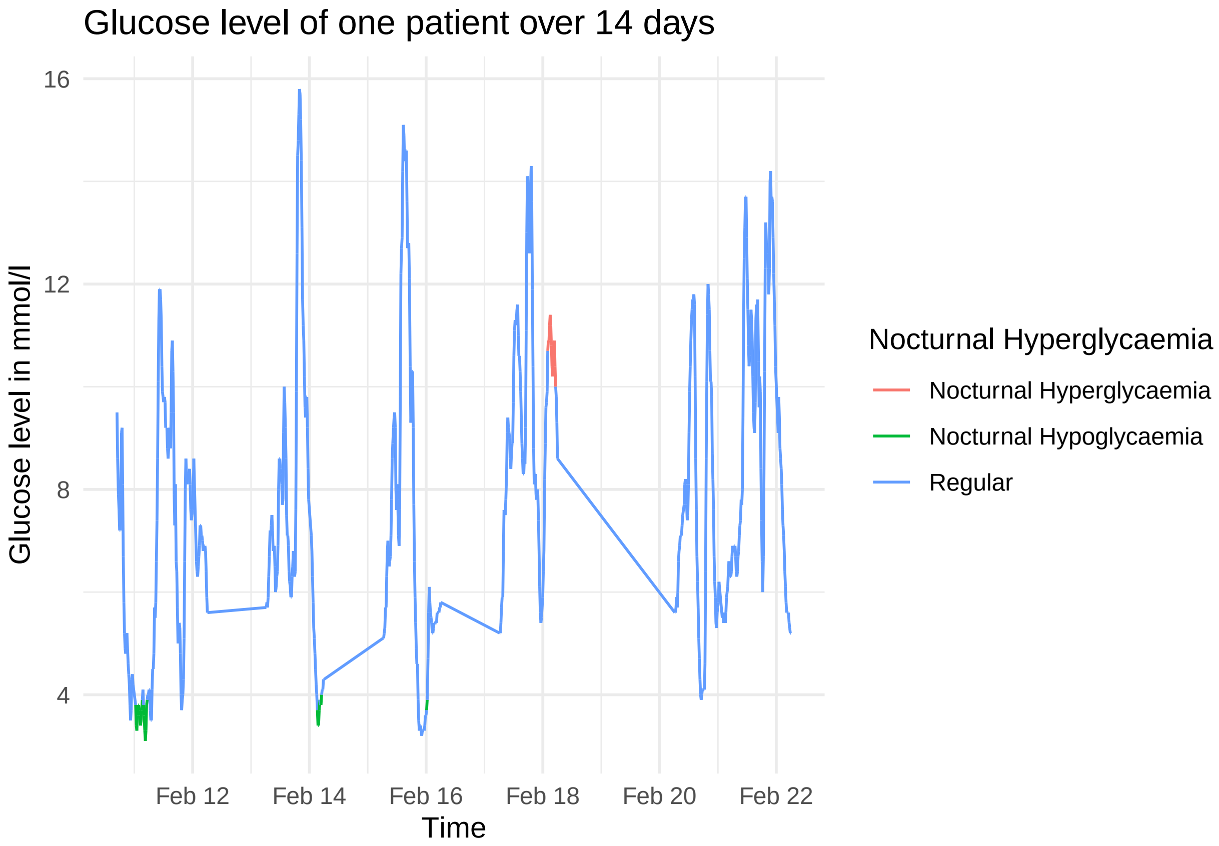
Exemplary continous profiles of patients. Nocturnal Hypo-, and Hyperglyceamia are highlighted. On top it gets clear, where periods of glucose values are missing in the intended 14 day period

```
all_patients_cgm <- read_csv("~/GitHub/Sweet_Dreams_Ahead_Machine_Learning_Models_for_Nocturnal_Hypogly
```

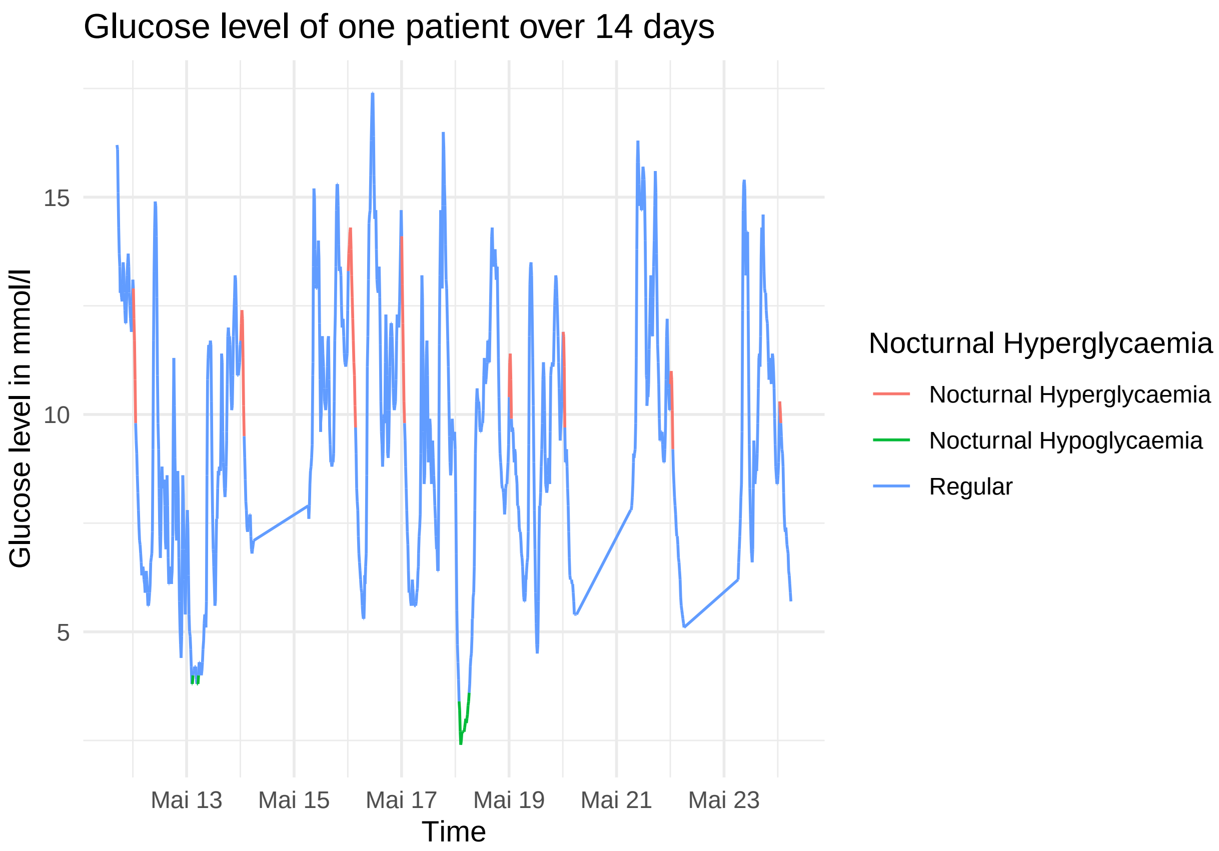
```
## Rows: 118322 Columns: 24
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (7): wday, weekend, Nocturnal, Nocturnal_dicho, Intervals_before_bedti...
## dbl  (15): patient, time_num, periods, Historie_glucose, mean_glucose_ti, me...
## dttm  (1): full_date
## date  (1): date
##
```

```
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
all_patients_cgm %>%
  filter(patient == 761) %>%
  ggplot(aes(x=full_date, y=Historie_glucose, colour=Noc_variability)) +
  geom_line(aes(group =3)) +
  theme_minimal() +
  labs(title = "Glucose level of one patient over 14 days",
       x = "Time",
       y = "Glucose level in mmol/l",
       colour = "Nocturnal Hyperglycaemia")
```



```
all_patients_cgm %>%
  filter(patient == 601) %>%
  ggplot(aes(x=full_date, y=Historie_glucose, colour=Noc_variability)) +
  geom_line(aes(group =3)) +
  theme_minimal() +
  labs(title = "Glucose level of one patient over 14 days",
       x = "Time",
       y = "Glucose level in mmol/l",
       colour = "Nocturnal Hyperglycaemia")
```


Glucose level of one patient over 14 days
Glucose level in mmol/l
15
10
5
Mai 13
Mai 15
Mai 17
Mai 19
Mai 21
Mai 23
Time
Nocturnal Hyperglycaemia
Nocturnal Hyperglycaemia
Nocturnal Hypoglycaemia
Regular