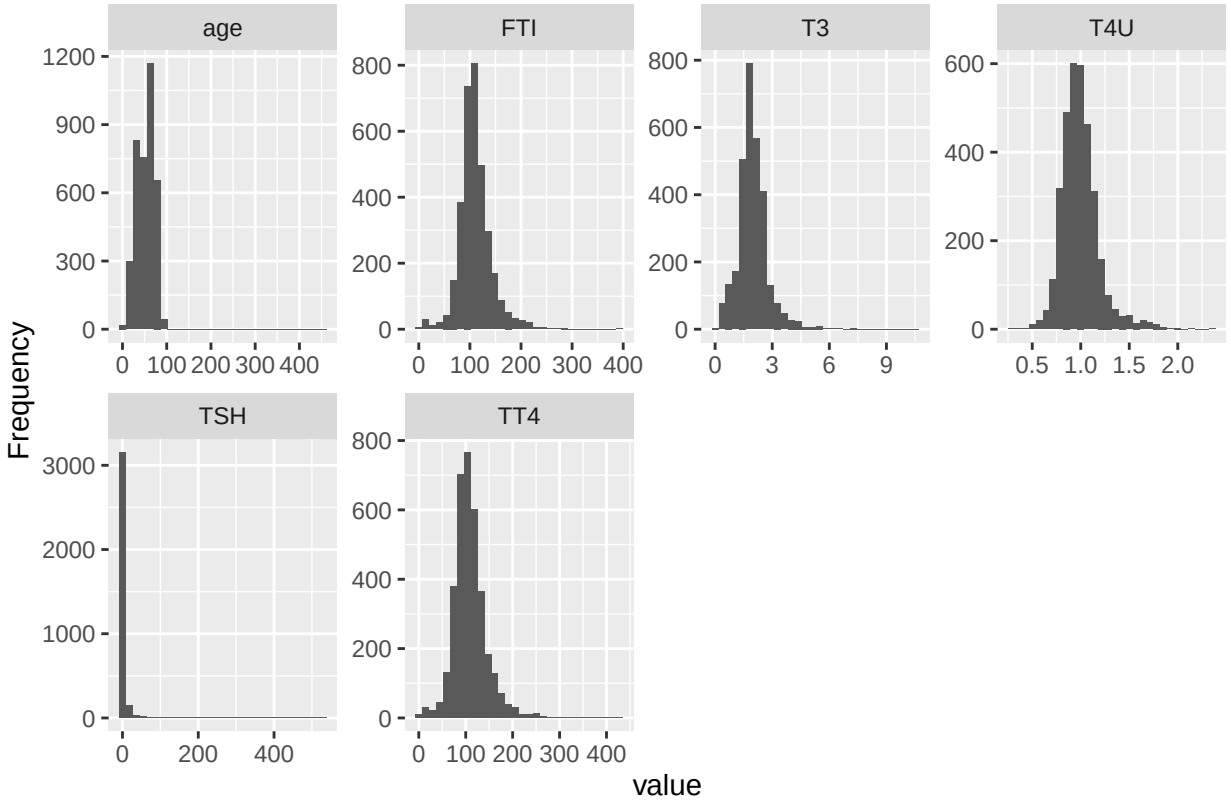
Sick dataset analysis

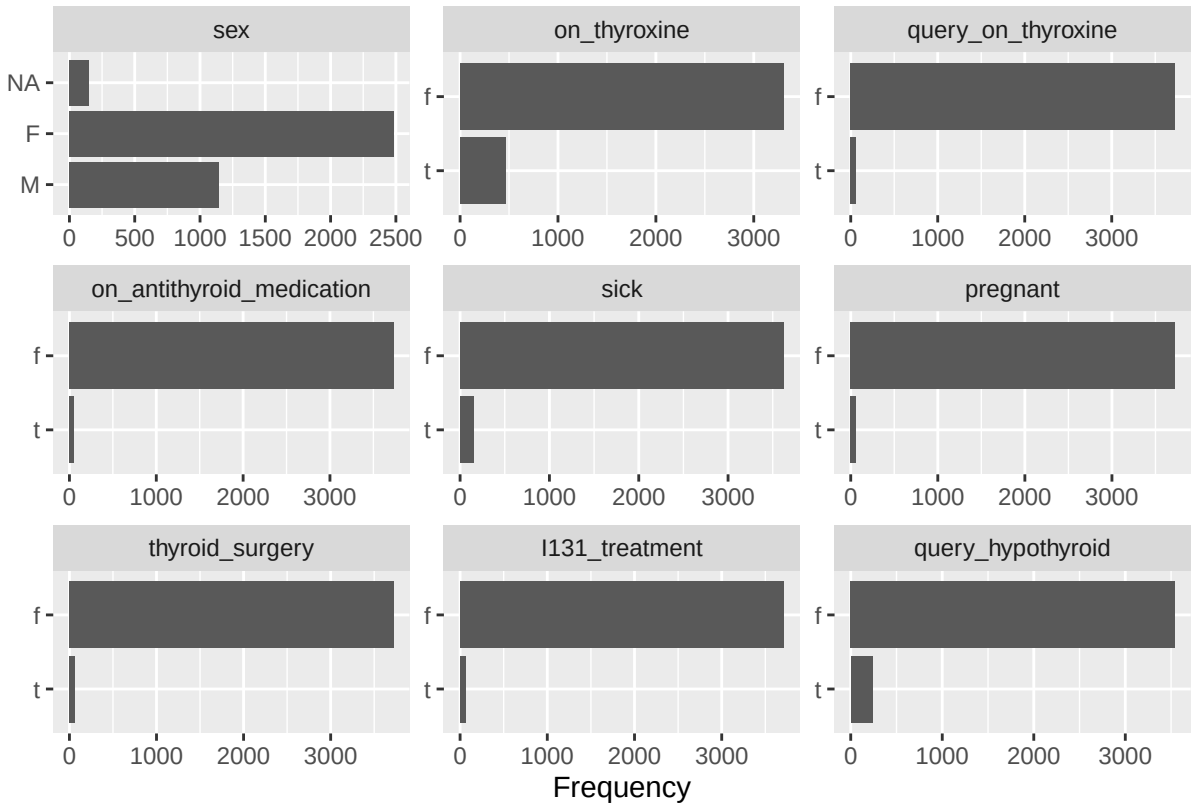
Michał Pastuszka

25 03 2020

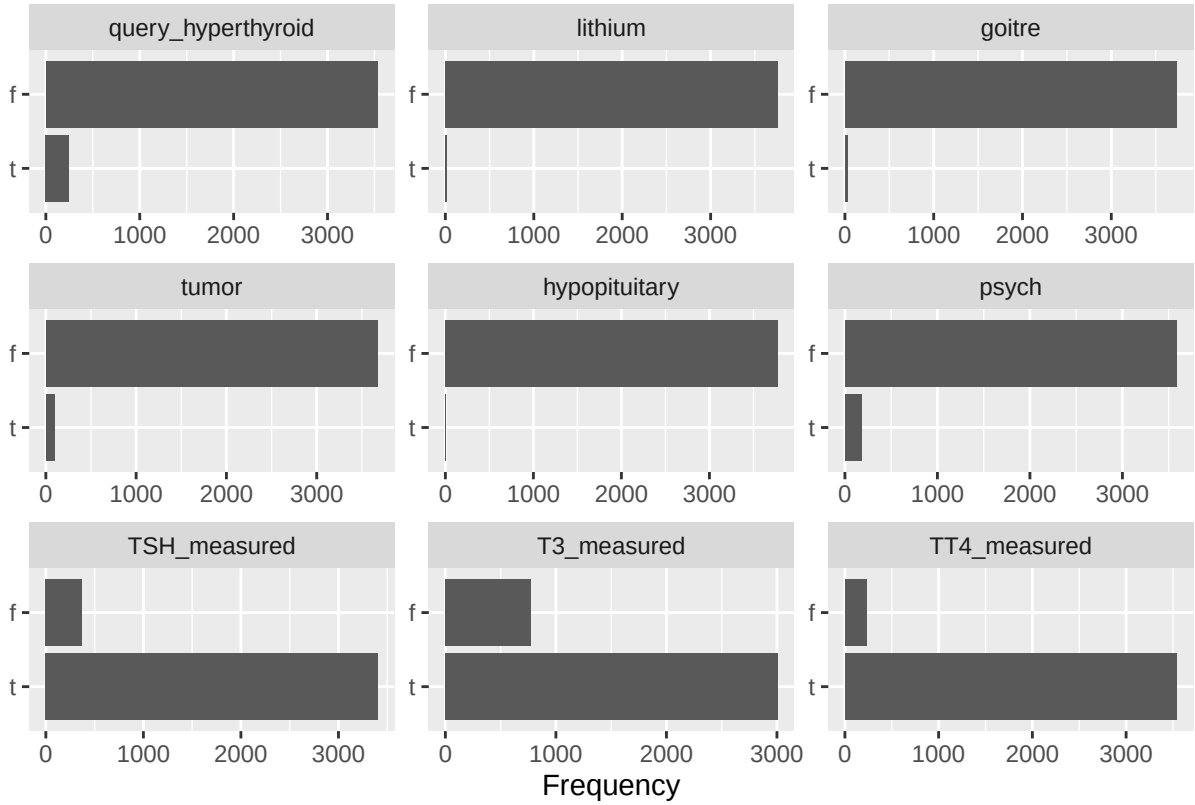
```
DataExplorer::plot_histogram(dataset_raw)
```



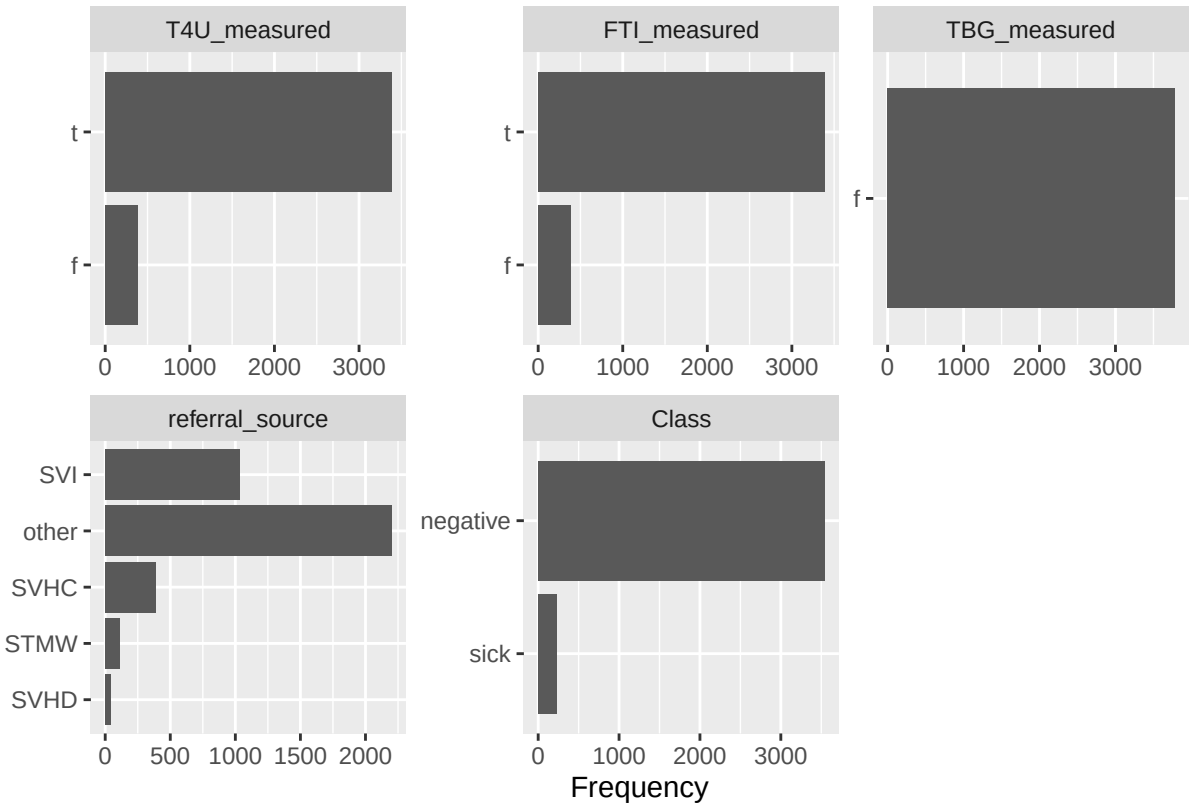

```
DataExplorer::plot_bar(dataset_raw)
```


sex
NA
F
M
0 500 1000 1500 2000 2500
on_thyroxine
f
t
0 1000 2000 3000
query_on_thyroxine
f
t
0 1000 2000 3000
on_antithyroid_medication
f
t
0 1000 2000 3000
sick
f
t
0 1000 2000 3000
pregnant
f
t
0 1000 2000 3000
thyroid_surgery
f
t
0 1000 2000 3000
I131_treatment
f
t
0 1000 2000 3000
query_hypothyroid
f
t
0 1000 2000 3000
Frequency

Page 1

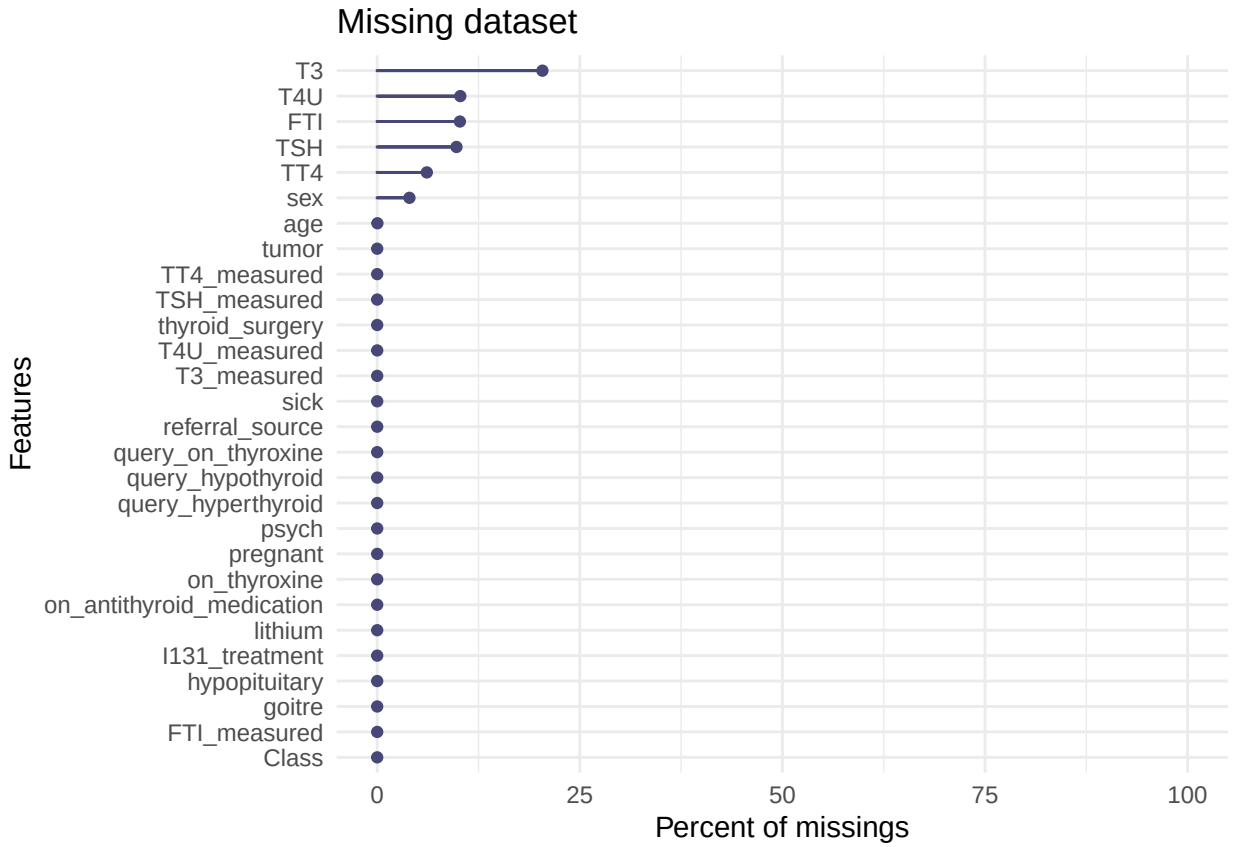

query_hyperthyroid
f
t
0 1000 2000 3000
lithium
f
t
0 1000 2000 3000
goitre
f
t
0 1000 2000 3000
tumor
f
t
0 1000 2000 3000
hypopituitary
f
t
0 1000 2000 3000
psych
f
t
0 1000 2000 3000
TSH_measured
f
t
0 1000 2000 3000
T3_measured
f
t
0 1000 2000 3000
TT4_measured
f
t
0 1000 2000 3000
Frequency

Page 2



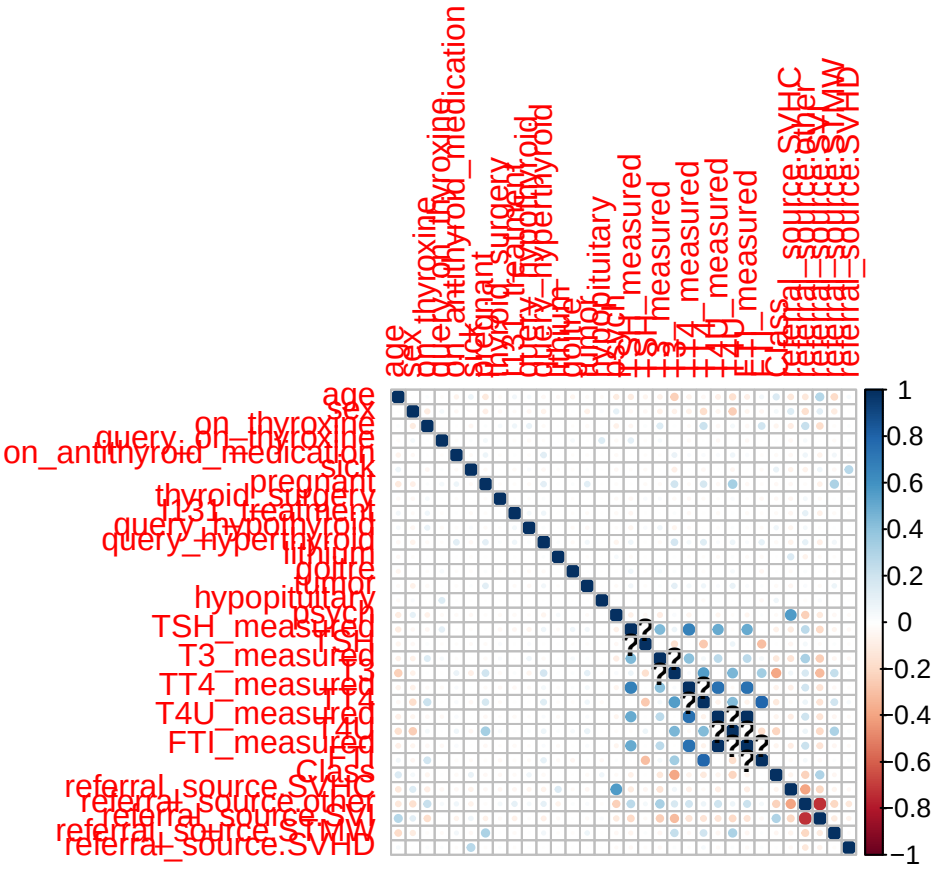
Preprocessing

```
gg_miss_var(dataset,
            show_pct = TRUE) +
  ylim(0, 100) +
  labs(title = "Missing dataset",
       x = "Features",
       y = "Percent of missings")
```




```
# variable referral_source is a vactor with more than two values - lets try one-hot encoding
dmy <- dummyVars(~referral_source , data = dataset)
new_vars <- predict(dmy, dataset)
dataset <- dataset %>% select(-referral_source)
dataset <- cbind(dataset, new_vars)
```

```
cors <- cor(sapply(dataset, as.numeric, MARGIN=2), use = "pairwise.complete.obs")
corrplot(cors)
```



Variables FTI_measured and T4U measured are very strongly correlated. We will remove one of them we will also remove referral_source.other, as it is reduntant.

```
dataset <- dataset %>%
           select(-FTI_measured) %>%
           select(-referral_source.other)
```

Splitting data

```
train_indices <- read.csv('indeksy_treningowe.txt', sep = ' ', row.names = 1, header = TRUE)
dataset_train <- dataset[train_indices$x,]
dataset_test <- dataset[-train_indices$x,]
```

```
which(dataset_train$hypopituitary=='t')
```

```
## integer(0)
```

It turns out that the data in our training set only contains one level of the variable hypopituitary. We should remove it too.

```
# variable hypopituary contains only one value in training set - we have to drop it
dataset_train <- dataset_train %>%
  select(-hypopituitary)
dataset_test <- dataset_test %>%
  select(-hypopituitary)
```

Imputation

We will impute missing values using their means calculated on training data. As there is only one missing value in the variable sex, we will replace it with female, as it is more frequent in our set.

```
# Let's start by imputing missing values with means and most frequent categories.
table(dataset$sex)
```

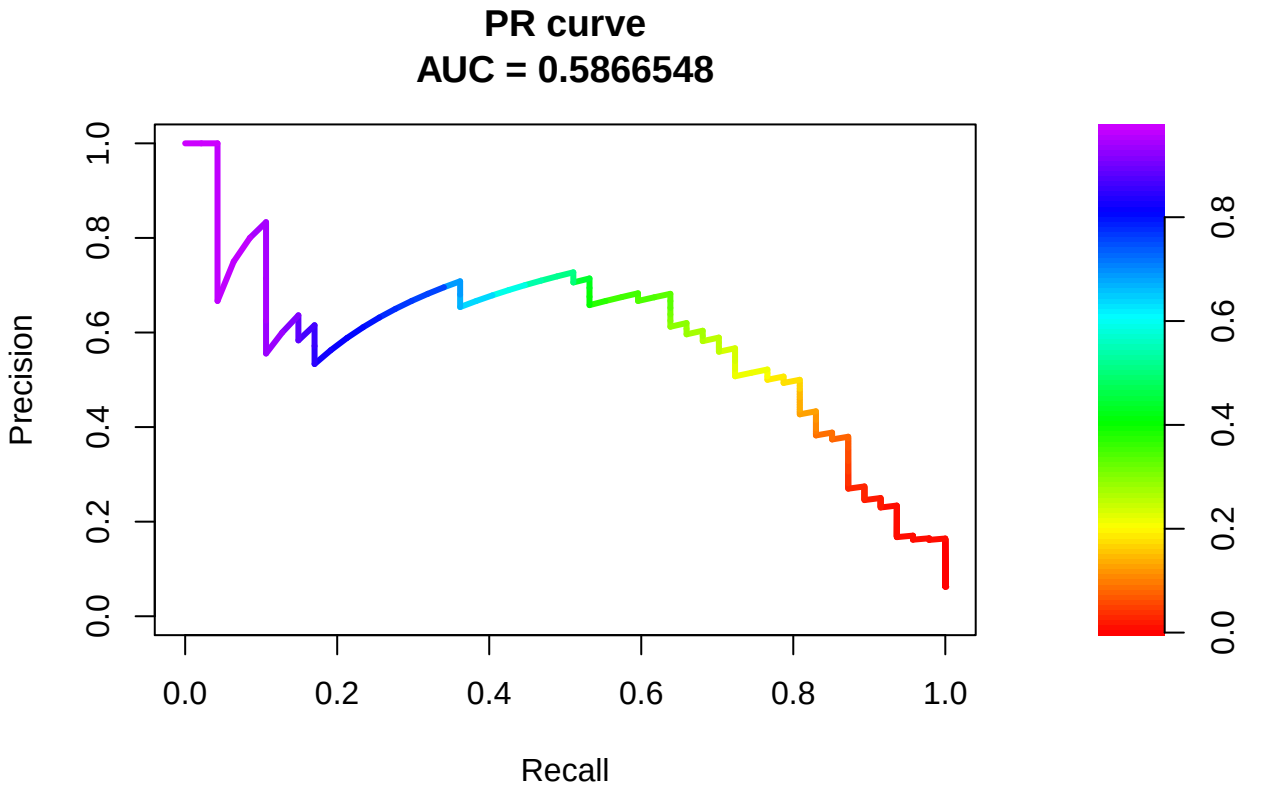
```
##
##    F    M
## 2480 1142
```

```
dataset_train$sex <- dataset_train$sex %>% replace_na('F')
dataset_test$sex <- dataset_test$sex %>% replace_na('F')
dataset_test$age <- dataset_test$age %>% replace_na(mean(dataset_train$age, na.rm = TRUE))
dataset_train_mean <- dataset_train
dataset_test_mean <- dataset_test


columns_missing <- c('T3', 'T4U', 'TSH', 'TT4', 'age', 'FTI')
for(col in columns_missing){
  avg <- mean(dataset_train_mean[,col], na.rm = TRUE)
  dataset_train_mean[,col] <- dataset_train_mean[,col] %>% replace_na(avg)
  dataset_test_mean[,col] <- dataset_test_mean[,col] %>% replace_na(avg)
}
```

Testing models

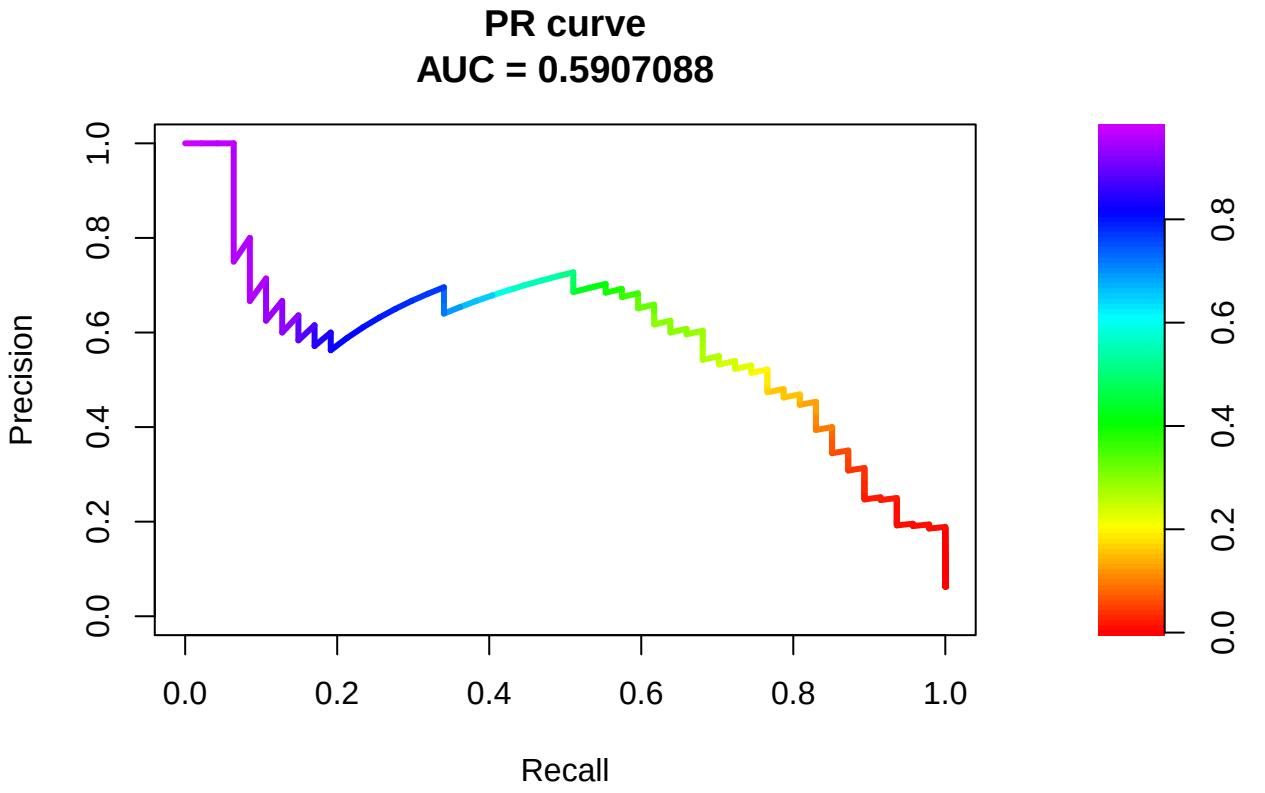
We will start using a simple logistic regression model.

```
get_prauc <- function(responses, test_set){
  fg <- responses[test_set$Class == 'sick']
  bg <- responses[test_set$Class == 'negative']
  pr <- pr.curve(scores.class0 = fg, scores.class1 = bg, curve = T)
  plot(pr)
}

logr <- glm(Class~., data=dataset_train_mean, family = 'binomial')
probs <- predict(logr, dataset_test_mean, type = 'response')

get_prauc(probs, dataset_test_mean)
```



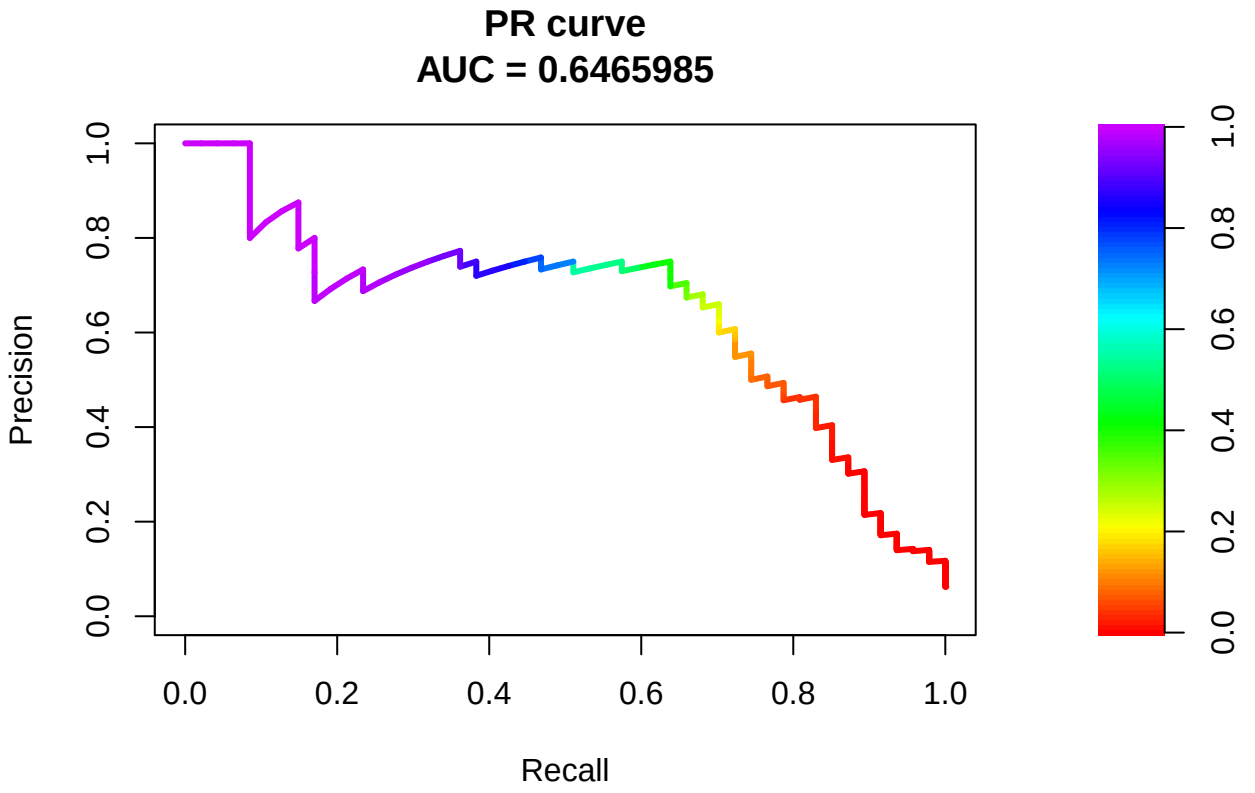
Let's see, if using a more advanced imputing technique improves our PRAUC score. We well use bagging tree imputation from the caret library, as it allows us to train it only on our training set, and apply the transformation to new data. Some more advanced libraries, such as MICE, don't have that functionality.

```
centerer_bag <- preProcess(dataset_train, method = c('bagImpute'))
dataset_train_bag <- predict(centerer_bag, dataset_train)
dataset_test_bag <- predict(centerer_bag, dataset_test)


logr <- glm(Class~., data=dataset_train_bag, family = 'binomial')
probs <- predict(logr, dataset_test_bag, type = 'response')
get_prauc(probs, dataset_test_bag)
```



We will now try to remove some outlying observations using the studentized residuals method. We will remove observations with residuals larger than 2.6.

```
logr <- glm(Class~., data=dataset_train_bag, family = 'binomial')
wh <- which(abs(rstudent(logr)) > 2.6)
logr <- glm(Class~., data=dataset_train_bag, family = 'binomial', subset = -wh)
probs <- predict(logr, dataset_test_bag, type = 'response')
get_prauc(probs, dataset_test_bag)
```

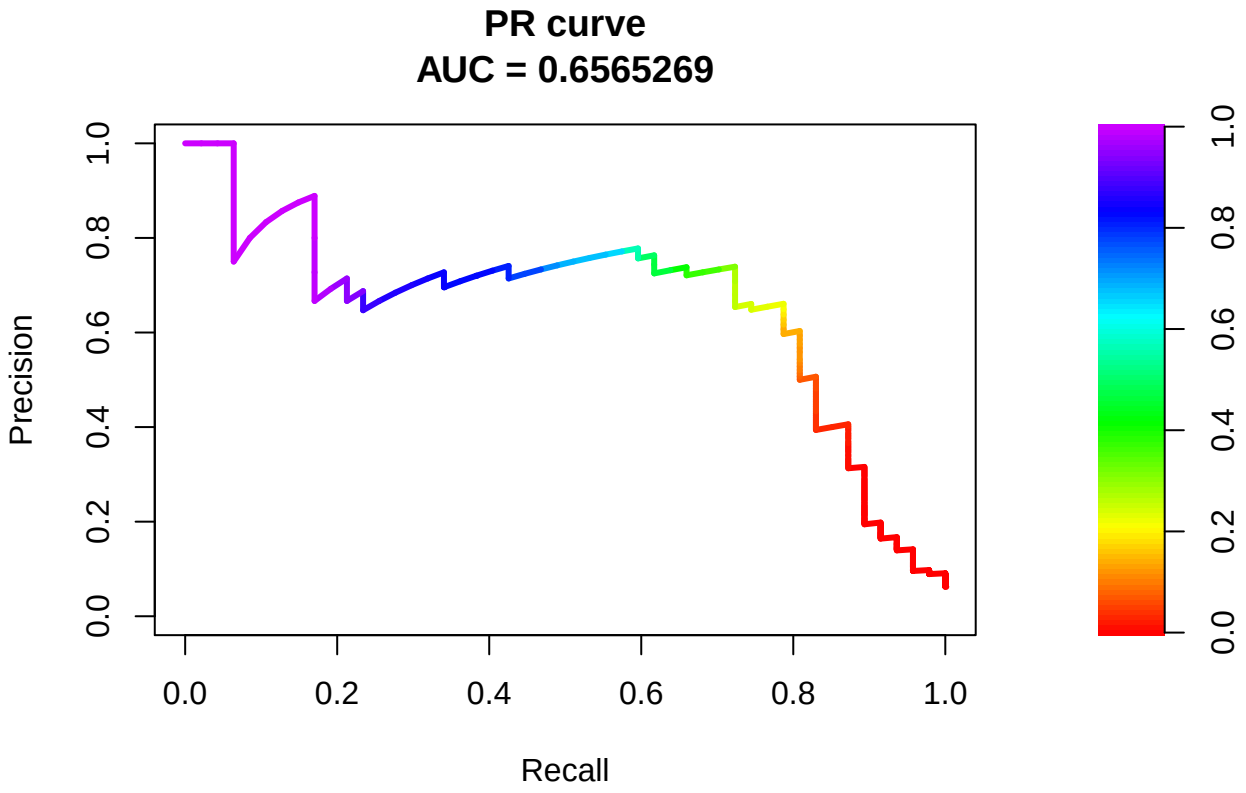


Let's try removing some variables using the Bayesian information criterion.

```
logr <- step(logr, direction = 'both', k=log(nrow(dataset_train_bag)-length(wh)), trace=FALSE)
formula(logr)
```

```
## Class ~ query_hypothyroid + TSH + T3 + referral_source.SVHC +
##     referral_source.SVI
```

```
form <- formula(logr)
probs <- predict(logr, dataset_test_bag, type = 'response')
get_prauc(probs, dataset_test_bag)
```



We were able to remove a majority of the variables, while still slightly improving our score.

Let's see of our final model:

```
logr
```

```
##
## Call:  glm(formula = Class ~ query_hypothyroid + TSH + T3 + referral_source.SVHC +
##     referral_source.SVI, family = "binomial", data = dataset_train_bag,
##     subset = -wh)
##
## Coefficients:
##          (Intercept)    query_hypothyroidt                   TSH
##               7.1850                1.7326               -0.1903
##                   T3  referral_source.SVHC   referral_source.SVI
##              -9.0855                2.7585                2.8394
##
## Degrees of Freedom: 2996 Total (i.e. Null);  2991 Residual
## Null Deviance:       1300
## Residual Deviance: 278.4     AIC: 290.4
```