

Aprendizagem Estatística em Altas Dimensões

[MAE0501/MAE5904/IBI5904]

Ícaro Maia Santos de Castro¹
Rayssa de Carvalho Roberto²
Rodrigo Aoyama Nakahara³
Rodrigo Araujo⁴
Vitor Hugo Vieira de Lima⁵

Novembro de 2020

Sumário

Importando os dados / Limpando / Inspeccionando	2
Renomear coluna e ajustar níveis de categórica	2
Visualização dos Dados	3
Análise dos Missings	3
Descritiva inicial	3
Possibilidades de imputação	9
train / test	16
Proporção de diabetes (Benign / Malignant)	16
Análise Descritiva	16
Distribuição da variável Diabetes	16
Correlação entre cada variável	17
Modelagem	19
SVM	19
RandomForest	22

¹Número USP: 11866921

²Número USP: 10940828

³Número USP: 3510922

⁴Número USP: 9299208

⁵Número USP: 10263886

Importando os dados / Limpando / Inspeccionando

```
diabetes <- read_csv("C:\\Users\\Rodrigo Araujo\\Documents\\IME-USP\\Aprendizagem Estatística em Altas I

## Parsed with column specification:
## cols(
##   Pregnancies = col_double(),
##   Glucose = col_double(),
##   BloodPressure = col_double(),
##   SkinThickness = col_double(),
##   Insulin = col_double(),
##   BMI = col_double(),
##   DiabetesPedigreeFunction = col_double(),
##   Age = col_double(),
##   Outcome = col_double()
## )

#diabetes <- read.csv("diabetes.csv")
head(diabetes) %>% kable(caption="Dados.")
```

Tabela 1: Dados.

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	148	72	35	0	33,6	0,63	50	1
1	85	66	29	0	26,6	0,35	31	0
8	183	64	0	0	23,3	0,67	32	1
1	89	66	23	94	28,1	0,17	21	0
0	137	40	35	168	43,1	2,29	33	1
5	116	74	0	0	25,6	0,20	30	0

```
dim(diabetes)
```

```
## [1] 768 9
```

Renomear coluna e ajustar níveis de categórica

```
diabetes[, 2:6][diabetes[, 2:6] == 0] <- NA

colnames(diabetes)[9] <- "diabetes"

diabetes$diabetes <- as.factor(diabetes$diabetes)

levels(diabetes$diabetes) <- c("No", "Yes")
```

Visualização dos Dados

Estrutura dos Dados

```
str(diabetes)
```

```
## tibble [768 x 9] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Pregnancies      : num [1:768] 6 1 8 1 0 5 3 10 2 8 ...
##  $ Glucose           : num [1:768] 148 85 183 89 137 116 78 115 197 125 ...
##  $ BloodPressure     : num [1:768] 72 66 64 66 40 74 50 NA 70 96 ...
##  $ SkinThickness     : num [1:768] 35 29 NA 23 35 NA 32 NA 45 NA ...
##  $ Insulin           : num [1:768] NA NA NA 94 168 NA 88 NA 543 NA ...
##  $ BMI               : num [1:768] 33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 NA ...
##  $ DiabetesPedigreeFunction: num [1:768] 0.627 0.351 0.672 0.167 2.288 ...
##  $ Age               : num [1:768] 50 31 32 21 33 30 26 29 53 54 ...
##  $ diabetes          : Factor w/ 2 levels "No","Yes": 2 1 2 1 2 1 2 1 2 2 ...
## - attr(*, "spec")=
##   .. cols(
##     .. Pregnancies = col_double(),
##     .. Glucose = col_double(),
##     .. BloodPressure = col_double(),
##     .. SkinThickness = col_double(),
##     .. Insulin = col_double(),
##     .. BMI = col_double(),
##     .. DiabetesPedigreeFunction = col_double(),
##     .. Age = col_double(),
##     .. Outcome = col_double()
##   .. )
```

Análise dos Missings

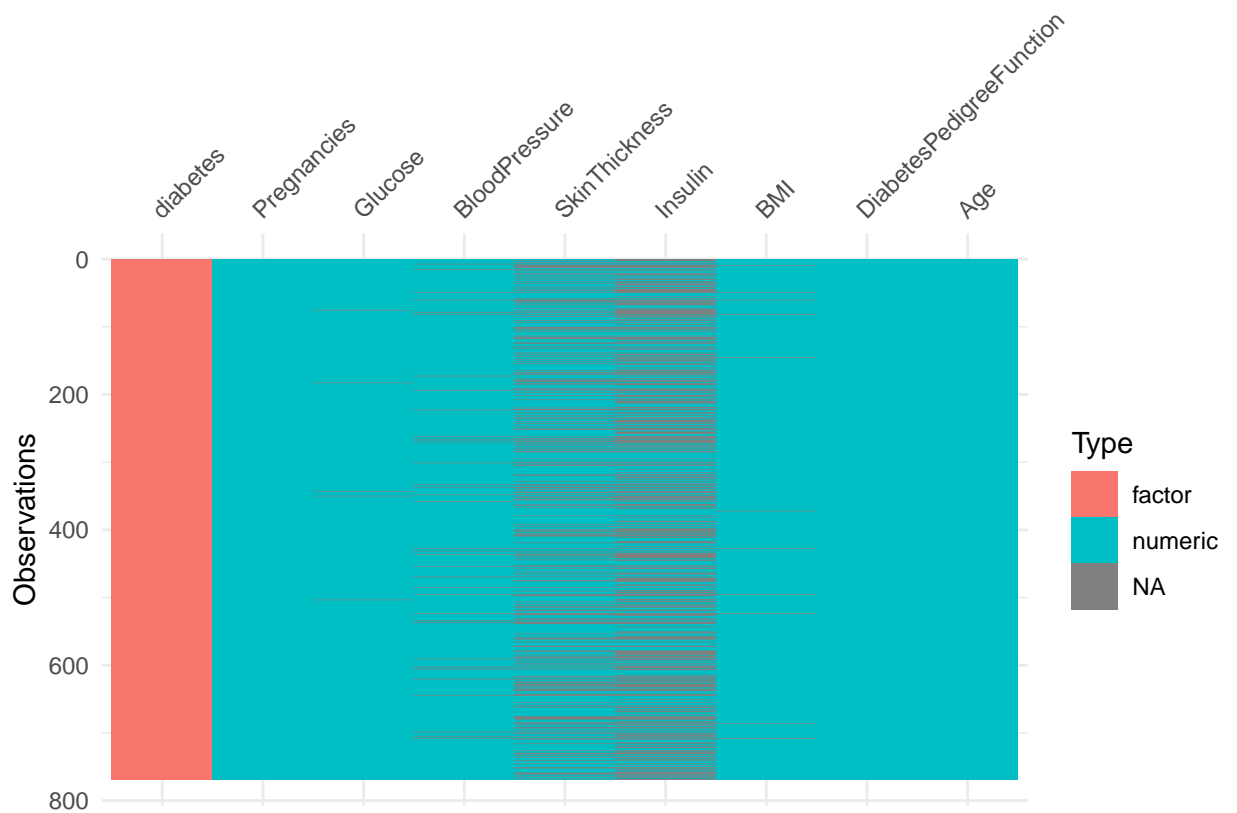
Descritiva inicial

```
summary(diabetes)
```

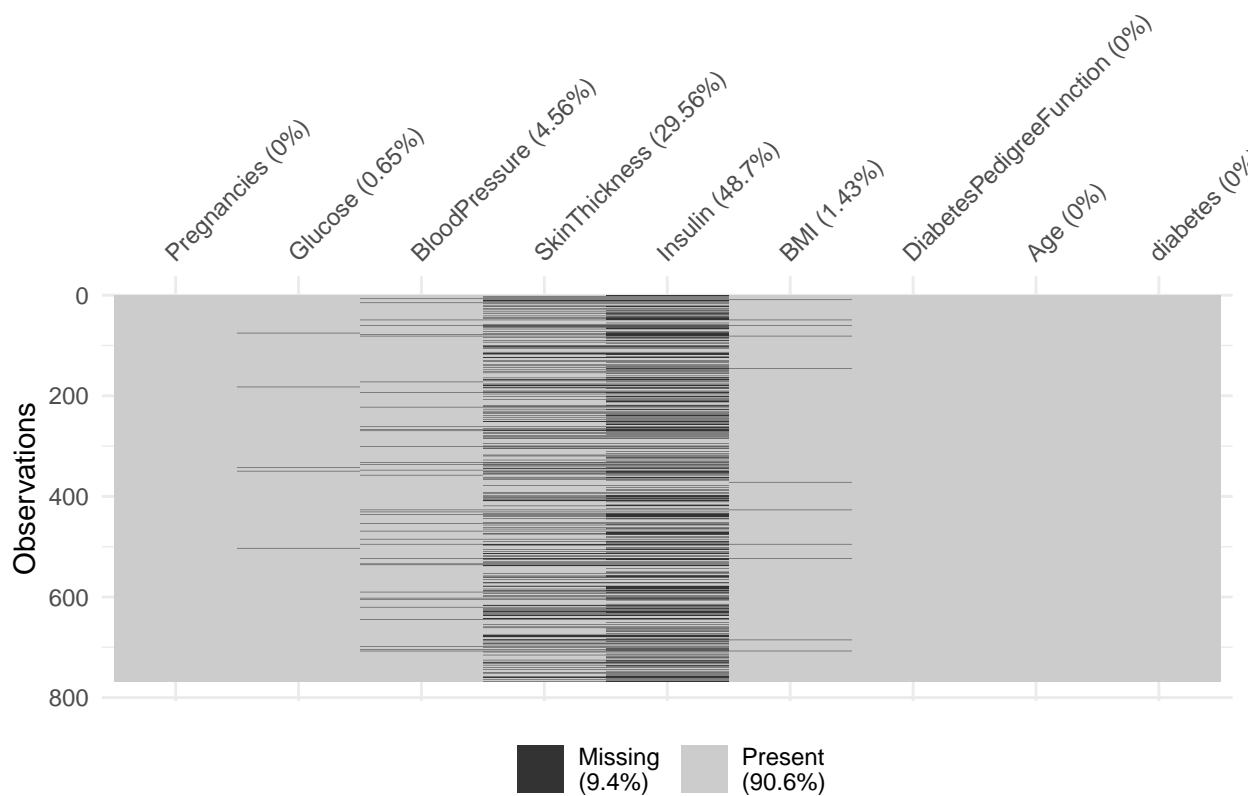
```
##   Pregnancies      Glucose      BloodPressure      SkinThickness
##   Min.   : 0.000    Min.   : 44.0    Min.   : 24.00    Min.   : 7.00
##   1st Qu.: 1.000    1st Qu.: 99.0    1st Qu.: 64.00    1st Qu.:22.00
##   Median : 3.000    Median :117.0    Median : 72.00    Median :29.00
##   Mean   : 3.845    Mean   :121.7    Mean   : 72.41    Mean   :29.15
##   3rd Qu.: 6.000    3rd Qu.:141.0    3rd Qu.: 80.00    3rd Qu.:36.00
##   Max.   :17.000    Max.   :199.0    Max.   :122.00    Max.   :99.00
##               NA's      :5      NA's      :35      NA's      :227
##   Insulin          BMI      DiabetesPedigreeFunction      Age
##   Min.   : 14.00    Min.   :18.20    Min.   :0.0780    Min.   :21.00
##   1st Qu.: 76.25    1st Qu.:27.50    1st Qu.:0.2437    1st Qu.:24.00
##   Median :125.00    Median :32.30    Median :0.3725    Median :29.00
```

```
## Mean :155.55 Mean :32.46 Mean :0.4719 Mean :33.24
## 3rd Qu.:190.00 3rd Qu.:36.60 3rd Qu.:0.6262 3rd Qu.:41.00
## Max. :846.00 Max. :67.10 Max. :2.4200 Max. :81.00
## NA's :374 NA's :11
## diabetes
## No :500
## Yes:268
##
##
##
##
```

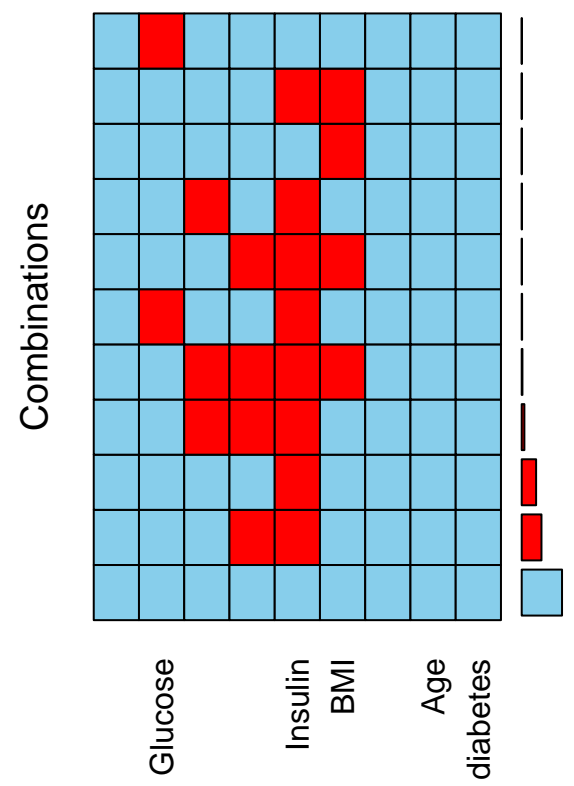
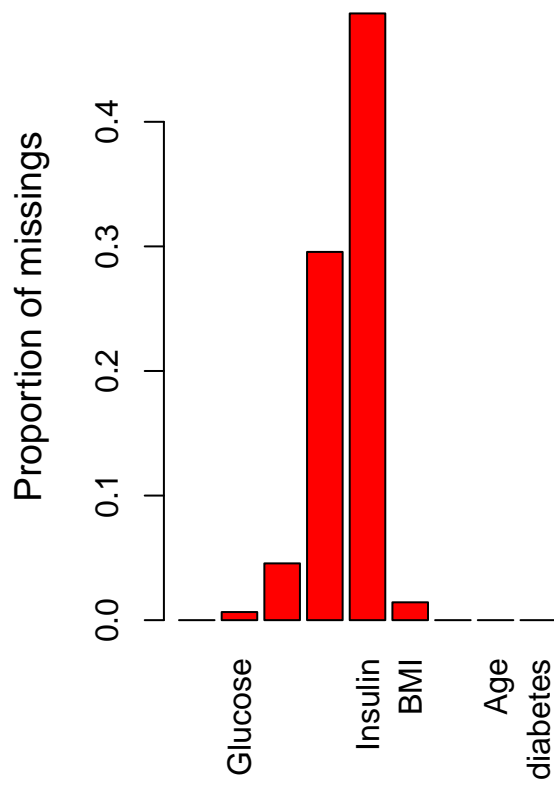
```
vis_dat(diabetes)
```



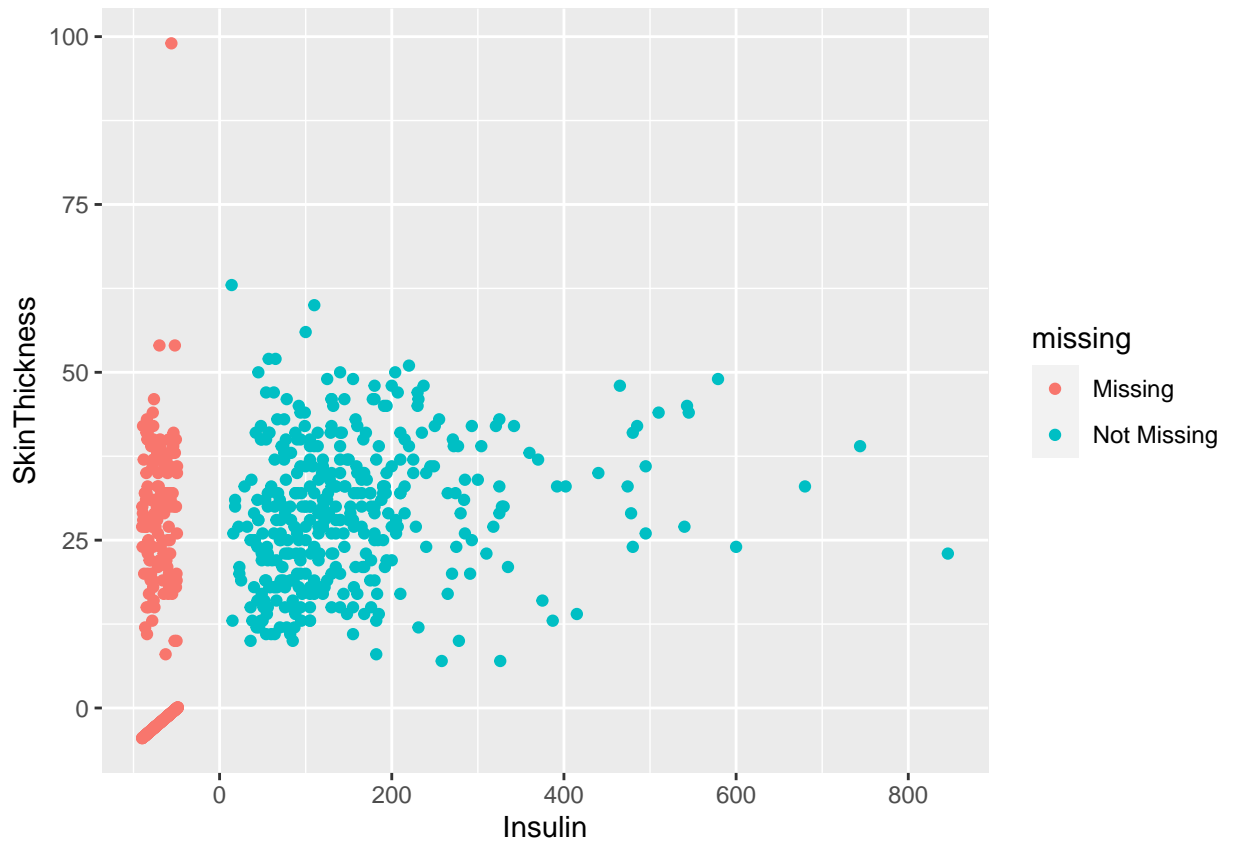
```
vis_miss(diabetes)
```



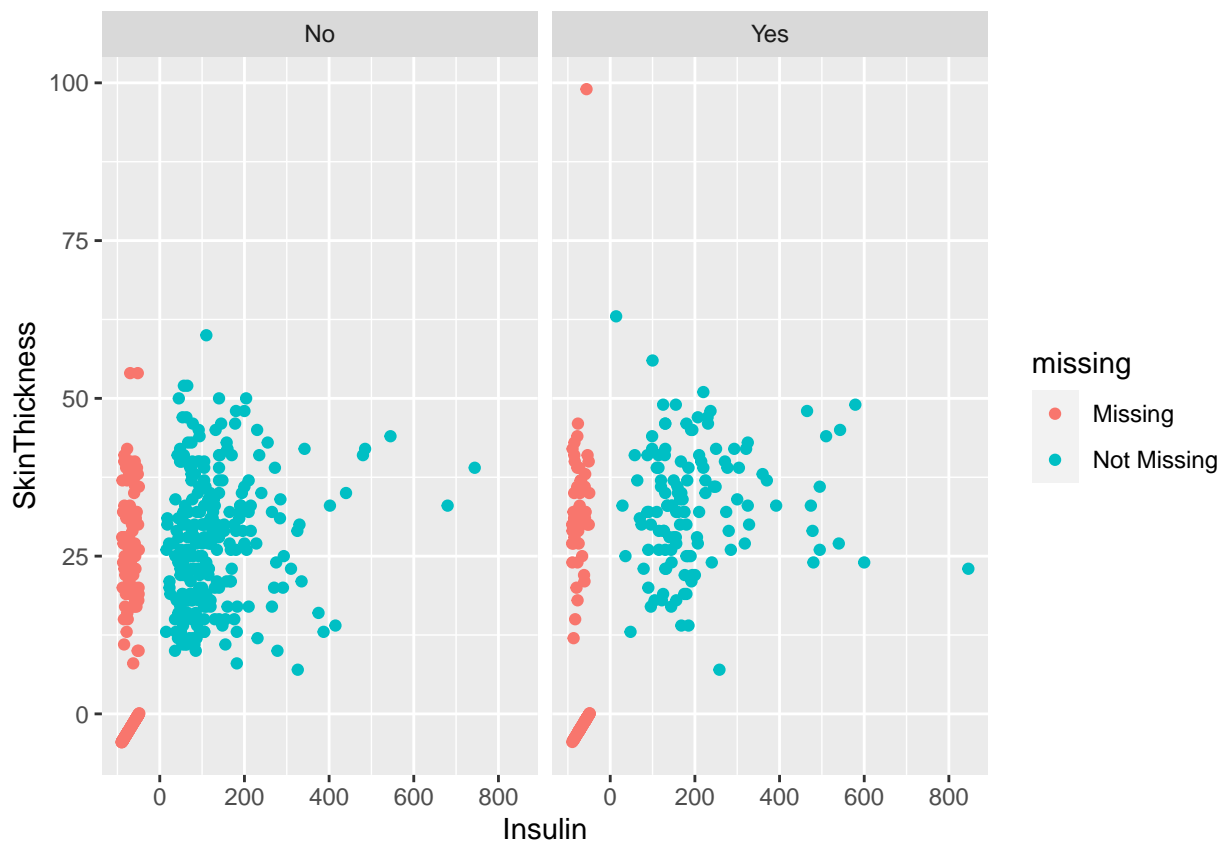
```
aggr(diabetes) # Missings têm padrões?
```



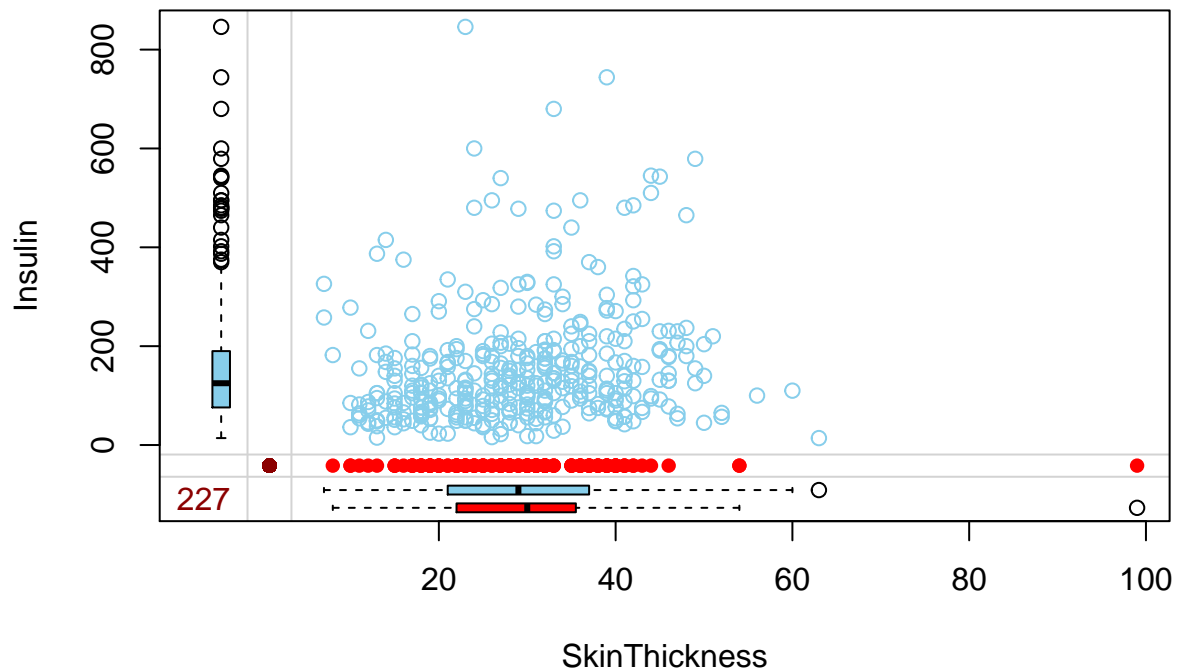
```
ggplot(diabetes, aes(x = Insulin, y = SkinThickness)) + # Padrão de missing entre 2 vars com mais missings
  geom_miss_point()
```



```
ggplot(diabetes, aes(x = Insulin, y = SkinThickness)) + # Padrão de missing entre 2 vars com mais missi  
  geom_miss_point() + facet_wrap(~ diabetes)
```



```
marginplot(diabetes[c(4,5)])
```

Possibilidades de imputação

```
library(mice)
```

```
## Warning: package 'mice' was built under R version 4.0.3
```

```
##
```

```
## Attaching package: 'mice'
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
## cbind, rbind
```

```
imputacao1 <- mice(data = diabetes , m = 5, maxit = 50, meth = 'pmm', seed = 25)
```

```
##
```

```
## iter imp variable
```

```
## 1 1 Glucose BloodPressure SkinThickness Insulin BMI
```

```
## 1 2 Glucose BloodPressure SkinThickness Insulin BMI
```

```
## 1 3 Glucose BloodPressure SkinThickness Insulin BMI
```

[illegible]

[illegible]

[illegible]

[illegible]

```
## 44 5 Glucose BloodPressure SkinThickness Insulin BMI
## 45 1 Glucose BloodPressure SkinThickness Insulin BMI
## 45 2 Glucose BloodPressure SkinThickness Insulin BMI
## 45 3 Glucose BloodPressure SkinThickness Insulin BMI
## 45 4 Glucose BloodPressure SkinThickness Insulin BMI
## 45 5 Glucose BloodPressure SkinThickness Insulin BMI
## 46 1 Glucose BloodPressure SkinThickness Insulin BMI
## 46 2 Glucose BloodPressure SkinThickness Insulin BMI
## 46 3 Glucose BloodPressure SkinThickness Insulin BMI
## 46 4 Glucose BloodPressure SkinThickness Insulin BMI
## 46 5 Glucose BloodPressure SkinThickness Insulin BMI
## 47 1 Glucose BloodPressure SkinThickness Insulin BMI
## 47 2 Glucose BloodPressure SkinThickness Insulin BMI
## 47 3 Glucose BloodPressure SkinThickness Insulin BMI
## 47 4 Glucose BloodPressure SkinThickness Insulin BMI
## 47 5 Glucose BloodPressure SkinThickness Insulin BMI
## 48 1 Glucose BloodPressure SkinThickness Insulin BMI
## 48 2 Glucose BloodPressure SkinThickness Insulin BMI
## 48 3 Glucose BloodPressure SkinThickness Insulin BMI
## 48 4 Glucose BloodPressure SkinThickness Insulin BMI
## 48 5 Glucose BloodPressure SkinThickness Insulin BMI
## 49 1 Glucose BloodPressure SkinThickness Insulin BMI
## 49 2 Glucose BloodPressure SkinThickness Insulin BMI
## 49 3 Glucose BloodPressure SkinThickness Insulin BMI
## 49 4 Glucose BloodPressure SkinThickness Insulin BMI
## 49 5 Glucose BloodPressure SkinThickness Insulin BMI
## 50 1 Glucose BloodPressure SkinThickness Insulin BMI
## 50 2 Glucose BloodPressure SkinThickness Insulin BMI
## 50 3 Glucose BloodPressure SkinThickness Insulin BMI
## 50 4 Glucose BloodPressure SkinThickness Insulin BMI
## 50 5 Glucose BloodPressure SkinThickness Insulin BMI
```

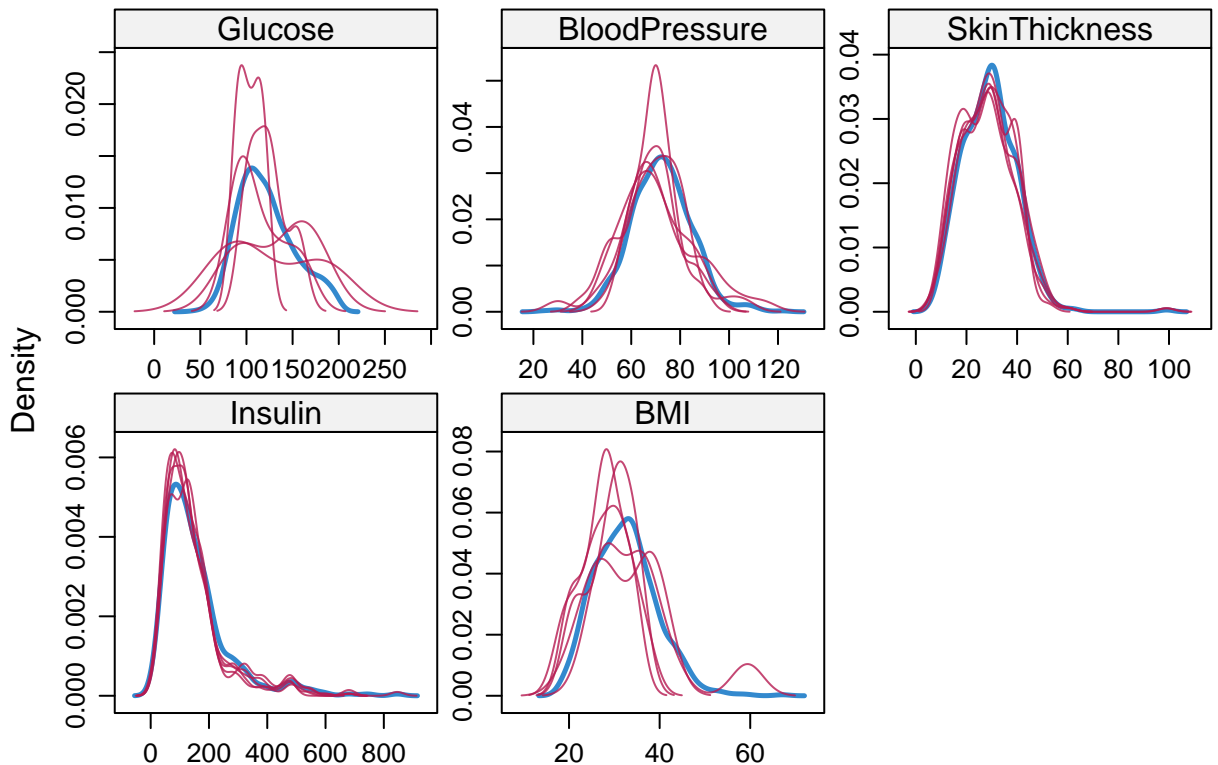
```
summary(imputacao1)
```

```
## Class: mids
## Number of multiple imputations: 5
## Imputation methods:
##           Pregnancies           Glucose           BloodPressure
##           "pmm"              "pmm"              "pmm"
##           SkinThickness       Insulin              BMI
##           "pmm"              "pmm"              "pmm"
## DiabetesPedigreeFunction      Age              diabetes
##           "pmm"              "pmm"              "pmm"
## PredictorMatrix:
##           Pregnancies Glucose BloodPressure SkinThickness Insulin BMI
## Pregnancies           0         1           1           1         1  1
## Glucose               1         0           1           1         1  1
## BloodPressure         1         1           0           1         1  1
## SkinThickness         1         1           1           0         1  1
## Insulin               1         1           1           1         0  1
## BMI                   1         1           1           1         1  0
##           DiabetesPedigreeFunction Age diabetes
## Pregnancies           1         1           1
```

```
## Glucose          1    1    1
## BloodPressure    1    1    1
## SkinThickness    1    1    1
## Insulin          1    1    1
## BMI              1    1    1
```

```
imputado1 <- complete(imputacao1, 1)
imputado2 <- complete(imputacao1, 2)
imputado3 <- complete(imputacao1, 3)
imputado4 <- complete(imputacao1, 4)
imputado5 <- complete(imputacao1, 5)
```

```
library(lattice)
densityplot(imputacao1)
```



```
diabetes$Glucose <- apply(cbind(imputado1$Glucose, imputado2$Glucose, imputado3$Glucose, imputado4$Glucose, imputado5$Glucose), 1, FUN=function(x){
  densityplot(x)
})
diabetes$BloodPressure <- apply(cbind(imputado1$BloodPressure, imputado2$BloodPressure, imputado3$BloodPressure, imputado4$BloodPressure, imputado5$BloodPressure), 1, FUN=function(x){
  densityplot(x)
})
diabetes$SkinThickness <- apply(cbind(imputado1$SkinThickness, imputado2$SkinThickness, imputado3$SkinThickness, imputado4$SkinThickness, imputado5$SkinThickness), 1, FUN=function(x){
  densityplot(x)
})
diabetes$Insulin <- apply(cbind(imputado1$Insulin, imputado2$Insulin, imputado3$Insulin, imputado4$Insulin, imputado5$Insulin), 1, FUN=function(x){
  densityplot(x)
})
diabetes$BMI <- apply(cbind(imputado1$BMI, imputado2$BMI, imputado3$BMI, imputado4$BMI, imputado5$BMI), 1, FUN=function(x){
  densityplot(x)
})
```

train / test

```
# para reprodução
set.seed(23)
nrows <- nrow(diabetes)
index <- sample(1:nrows, 0.7 * nrows) # shuffle and divide
# train <- diab                        # 768 test data (100%)
train <- diabetes[index,]              # 537 test data (70%)
test <- diabetes[-index,]              # 231 test data (30%)
```

Proporção de diabetes (Benign / Malignant)

train

```
prop.table(table(train$diabetes))
```

```
##
##           No           Yes
## 0.6405959 0.3594041
```

test

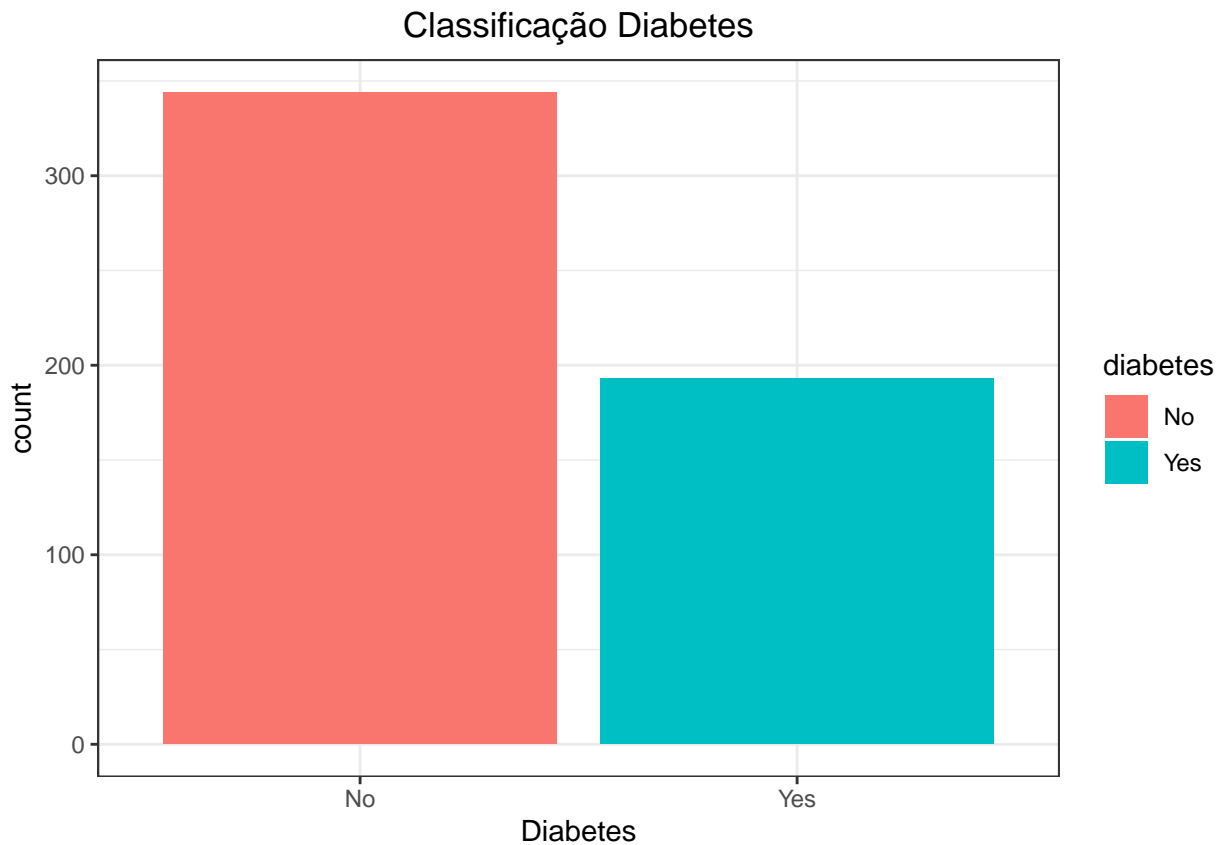
```
prop.table(table(test$diabetes))
```

```
##
##           No           Yes
## 0.6753247 0.3246753
```

Análise Descritiva

Distribuição da variável Diabetes

```
ggplot(train, aes(diabetes, fill = diabetes)) +
  geom_bar() +
  theme_bw() +
  labs(title = "Classificação Diabetes", x = "Diabetes") +
  theme(plot.title = element_text(hjust = 0.5))
```

Correlação entre cada variável

```
library(PerformanceAnalytics)
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
```

```
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
## as.Date, as.Date.numeric
```

```
##
```

```
## Attaching package: 'xts'
```

```
## The following objects are masked from 'package:dplyr':
```

```
##
```

```
## first, last
```

```
##
```

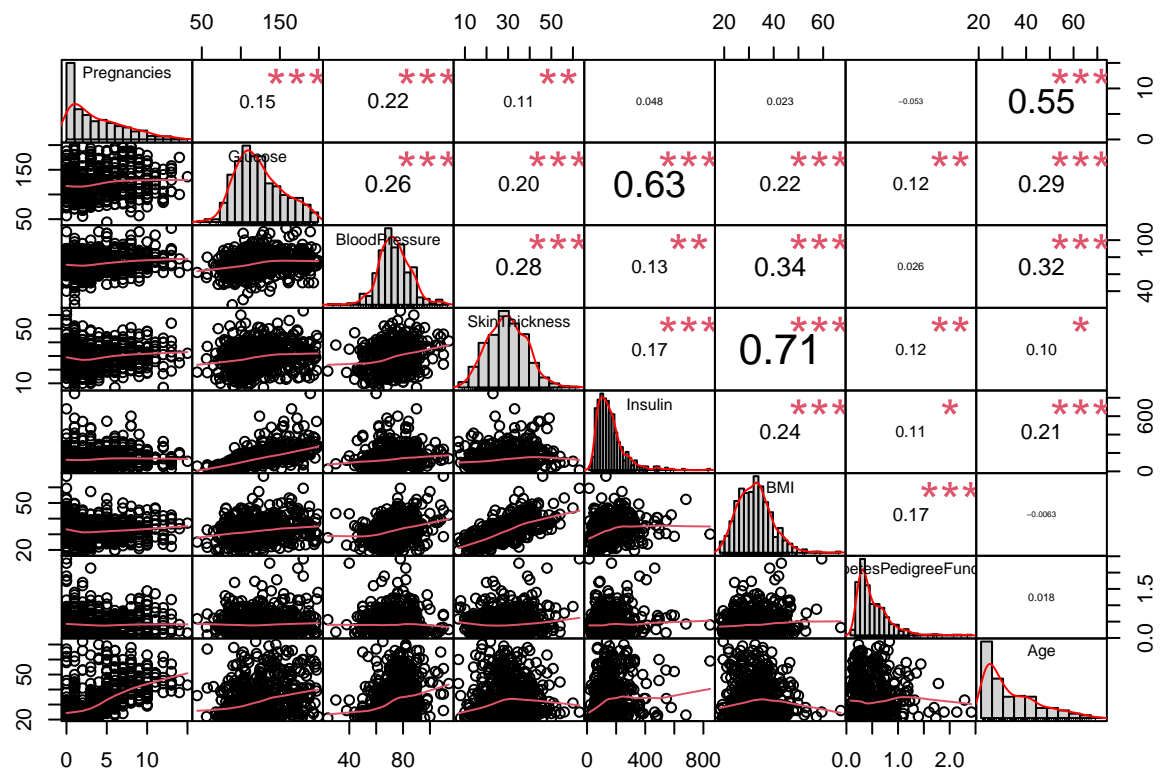
```
## Attaching package: 'PerformanceAnalytics'
```

```
## The following object is masked from 'package:graphics':
```

```
##
```

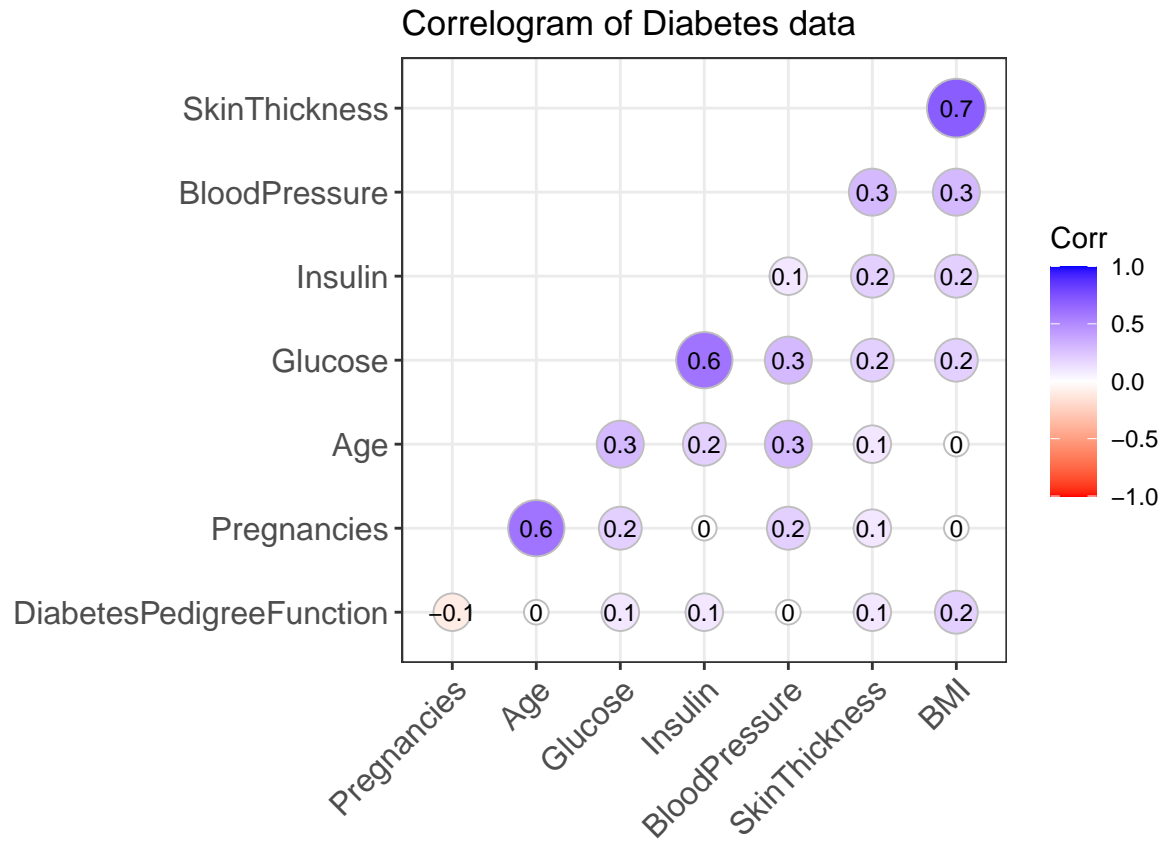
```
## legend
```

```
chart.Correlation(train[,-9], histogram=TRUE, col="grey10", pch=1, main="Correlação entre as variáveis")
```



```
library(ggcorrplot)
```

```
corr<-round(cor(train[,-9]),1)
ggcorrplot(corr, hc.order = TRUE,
  type = "lower",
  lab = TRUE,
  lab_size = 3,
  method="circle",
  colors = c("red", "white", "blue"),
  title="Correlogram of Diabetes data",
  ggtheme=theme_bw)
```



Modelagem

SVM

```
library(caret)
```

```
##
## Attaching package: 'caret'

## The following object is masked from 'package:vegan':
##
##   tolerance

## The following objects are masked from 'package:MLmetrics':
##
##   MAE, RMSE

## The following object is masked from 'package:survival':
##
##   cluster
```

```
## The following object is masked from 'package:purrr':  
##  
## lift
```

```
library(e1071)
```

```
##  
## Attaching package: 'e1071'
```

```
## The following objects are masked from 'package:PerformanceAnalytics':  
##  
## kurtosis, skewness
```

```
## The following object is masked from 'package:Hmisc':  
##  
## impute
```

```
set.seed(123)
```

```
linear.tune <- e1071::tune.svm(diabetes ~.,  
                             data = train,  
                             kernel = 'linear',  
                             cost = c(0.001, 0.01, 0.1, 1, 5, 10))  
summary(linear.tune)
```

```
##  
## Parameter tuning of 'svm':  
##  
## - sampling method: 10-fold cross validation  
##  
## - best parameters:  
## cost  
## 0.01  
##  
## - best performance: 0.2309574  
##  
## - Detailed performance results:  
## cost error dispersion  
## 1 1e-03 0.3591894 0.07461689  
## 2 1e-02 0.2309574 0.05370355  
## 3 1e-01 0.2310273 0.05114627  
## 4 1e+00 0.2365828 0.05083653  
## 5 5e+00 0.2347310 0.05025263  
## 6 1e+01 0.2347310 0.05025263
```

Matriz de Confusão

```
svm.real <- test$diabetes  
  
best.linear <- linear.tune$best.model
```

```
tune.test <- predict(best.linear, test[, -9])

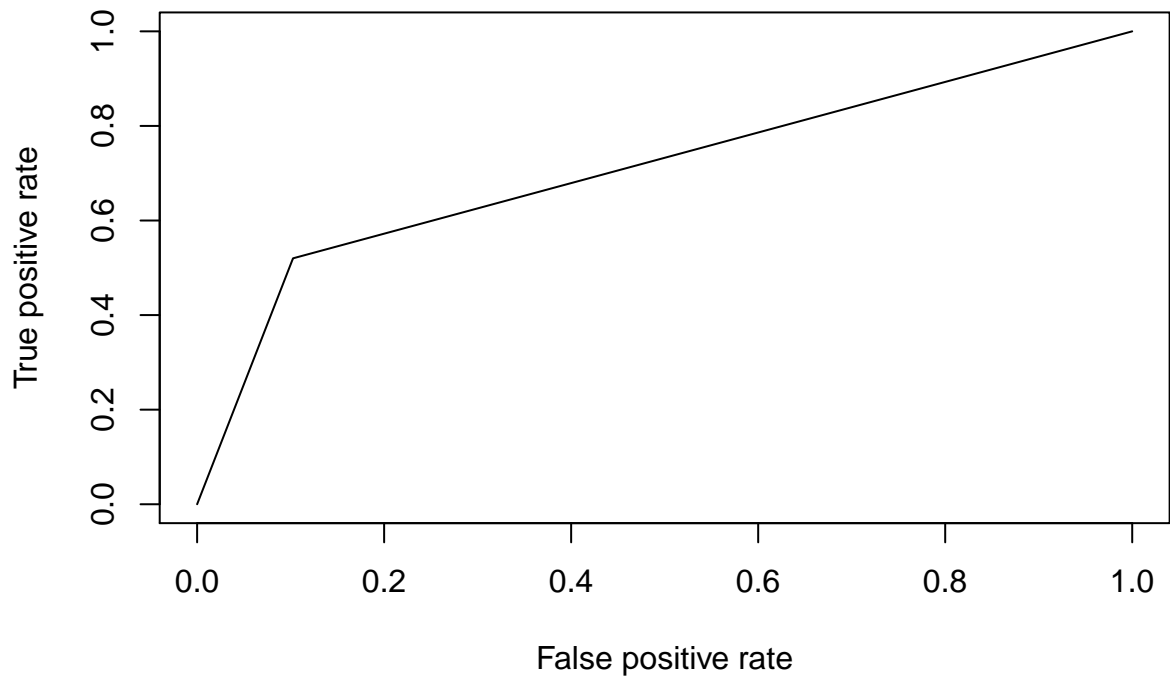
caret::confusionMatrix(data = tune.test,
                        reference = svm.real,
                        positive = 'Yes')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Yes
##      No  140  36
##      Yes   16  39
##
##              Accuracy : 0.7749
##              95% CI : (0.7155, 0.8271)
##      No Information Rate : 0.6753
##      P-Value [Acc > NIR] : 0.000570
##
##              Kappa : 0.4485
##
##      Mcnemar's Test P-Value : 0.008418
##
##              Sensitivity : 0.5200
##              Specificity : 0.8974
##      Pos Pred Value : 0.7091
##      Neg Pred Value : 0.7955
##      Prevalence : 0.3247
##      Detection Rate : 0.1688
##      Detection Prevalence : 0.2381
##      Balanced Accuracy : 0.7087
##
##      'Positive' Class : Yes
##
```

Curva ROC

```
svm.predobj <- ROCR::prediction(predictions = as.numeric(x = tune.test ),
                              labels      = as.numeric(x = svm.real))
svm.perform <- ROCR::performance(prediction.obj = svm.predobj,
                                measure       = 'tpr',
                                x.measure    = 'fpr')
plot(x = svm.perform, main = 'ROC curve')
```

ROC curve



```
MLmetrics::F1_Score(y_pred = tune.test ,  
                    y_true  = svm.real,  
                    positive = "Yes"); pROC::auc(response = as.numeric(x = svm.real),  
                                                predictor = as.numeric(x = tune.test))
```

```
## [1] 0.6
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

```
## Area under the curve: 0.7087
```

RandomForest

```
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'

## The following object is masked from 'package:psych':
##
## outlier

## The following object is masked from 'package:dplyr':
##
## combine

## The following object is masked from 'package:ggplot2':
##
## margin

learn_rf <- randomForest(diabetes~., data=train, ntree=500, proximity=T, importance=T)

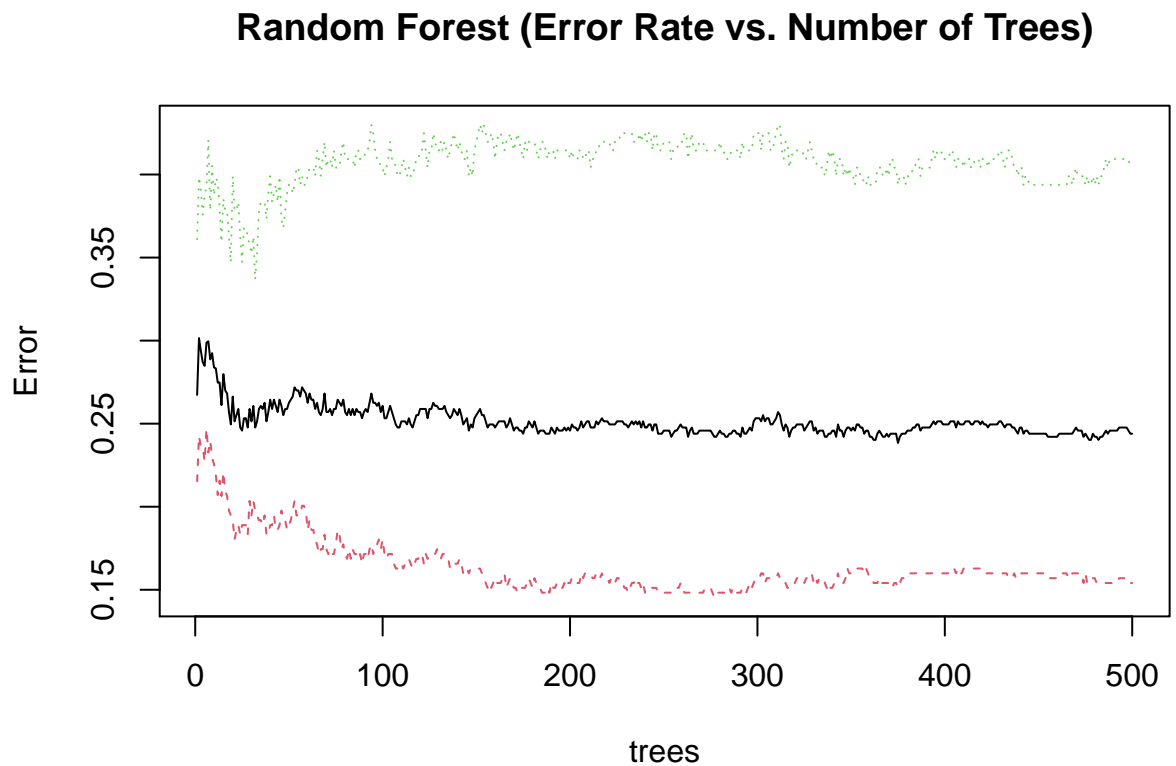
pre_rf <- predict(learn_rf, test[, -9])

cm_rf <- confusionMatrix(pre_rf, test$diabetes)

cm_rf

## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Yes
##      No  127  30
##      Yes   29  45
##
##           Accuracy : 0.7446
##           95% CI : (0.6833, 0.7995)
##      No Information Rate : 0.6753
##      P-Value [Acc > NIR] : 0.01342
##
##           Kappa : 0.4155
##
##  Mcnemar's Test P-Value : 1.00000
##
##           Sensitivity : 0.8141
##           Specificity : 0.6000
##           Pos Pred Value : 0.8089
##           Neg Pred Value : 0.6081
##           Prevalence : 0.6753
##           Detection Rate : 0.5498
##           Detection Prevalence : 0.6797
##           Balanced Accuracy : 0.7071
##
##           'Positive' Class : No
##
```

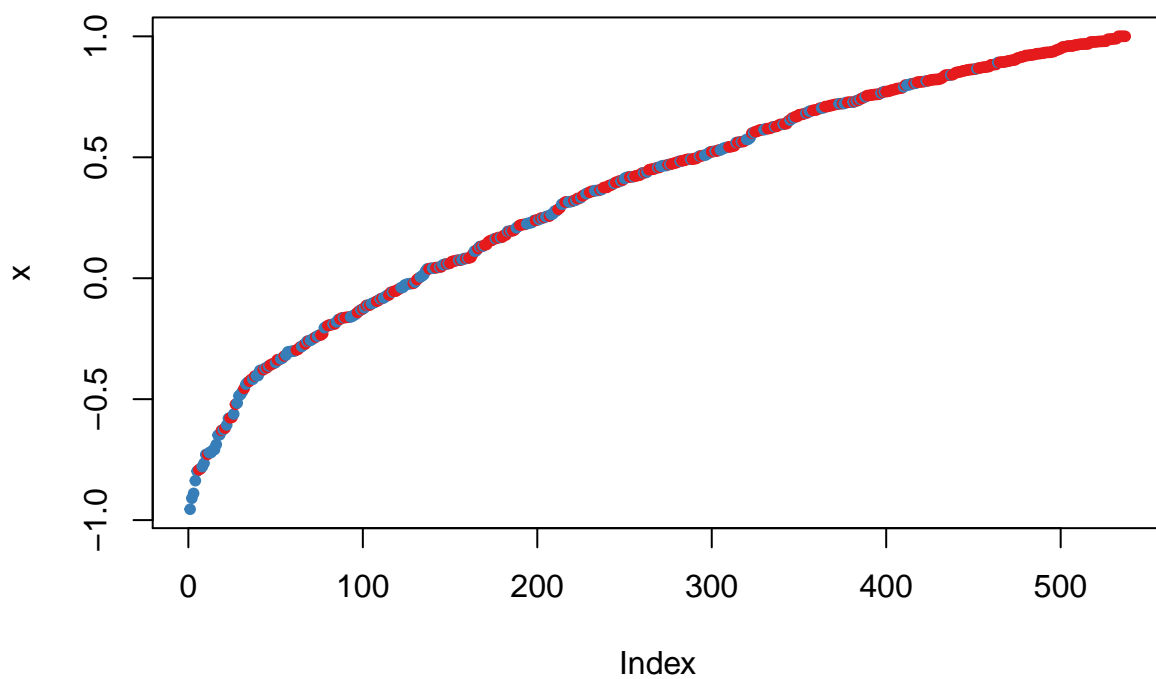
```
plot(learn_rf, main="Random Forest (Error Rate vs. Number of Trees)")
```



Prediction Plot

```
plot(margin(learn_rf, test$diabetes))
```

```
## Warning in RColorBrewer::brewer.pal(nlevs, "Set1"): minimal value for n is 3, returning requested pa
```

Variance Importance Plot

```
varImpPlot(learn_rf)
```

learn_rf

