Carregamento e Pré-processamento dos Dados

Carregamento dos dados

data_train = read.csv('train.csv')

gen_sub = read.csv('gender_submission.csv')

data_test = read.csv('test.csv')

```
head(data_train)
    PassengerId Survived Pclass
## 1
               1
## 2
                        1
                               1
## 3
                               3
## 4
                               3
## 5
                               3
## 6
##
                                                    Name
                                                            Sex Age SibSp Parch
## 1
                                 Braund, Mr. Owen Harris
                                                           male 22
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female 38
                                  Heikkinen, Miss. Laina female 26
## 3
            Futrelle, Mrs. Jacques Heath (Lily May Peel) female 35
## 4
## 5
                                Allen, Mr. William Henry
                                                           male 35
                                                                        0
## 6
                                        Moran, Mr. James
                                                           male NA
##
               Ticket
                         Fare Cabin Embarked
            A/5 21171 7.2500
## 1
## 2
                                           C
            PC 17599 71.2833
                                C85
## 3 STON/02. 3101282 7.9250
                                           S
                                           S
## 4
              113803 53.1000 C123
## 5
              373450 8.0500
                                           S
                                           Q
               330877 8.4583
head(data_test)
```

```
PassengerId Pclass
##
                                                                Name
                                                                        Sex Age
            892
                      3
                                                    Kelly, Mr. James
                                                                       male 34.5
## 1
## 2
            893
                      3
                                   Wilkes, Mrs. James (Ellen Needs) female 47.0
                      2
            894
                                           Myles, Mr. Thomas Francis
                                                                       male 62.0
## 3
                      3
## 4
            895
                                                    Wirz, Mr. Albert
                                                                       male 27.0
                      3 Hirvonen, Mrs. Alexander (Helga E Lindqvist) female 22.0
## 5
            896
                      3
                                          Svensson, Mr. Johan Cervin male 14.0
## 6
            897
    SibSp Parch Ticket
                           Fare Cabin Embarked
##
## 1
                  330911 7.8292
## 2
                 363272 7.0000
                                              S
## 3
                 240276 9.6875
## 4
               0 315154 8.6625
## 5
               1 3101298 12.2875
                   7538 9.2250
## 6
head(gen_sub)
```

```
PassengerId Survived
## 1
             892
             893
                        1
## 2
```

```
## 3
             894
 ## 4
             895
                       0
             896
 ## 5
                       1
 ## 6
             897
Removendo variáveis irrelevantes
As variáveis "Name", "Ticket" e "Cabin" não serão úteis para a nossa análise.
 # Remoção de variáveis irrelevantes
 data_train = subset(data_train, select = -c(Name, Ticket, Cabin))
```

data = rbind.fill(data_train, data_test) head(data)

3 male 22

Sex Age SibSp Parch

data_test = subset(data_test, select = -c(Name, Ticket, Cabin))

Combinação dos datasets

1

PassengerId

FALSE:1309

270 123 914

Moda = S

20

data_test\$Survived = gen_sub\$Survived

PassengerId Survived Pclass

```
## 2
                    1
                          1 female 38
                                              0 71.2833
                    1
 ## 3
                          3 female 26
                                              0 7.9250
                    1
 ## 4
                          1 female 35
                                              0 53.1000
 ## 5
                                              0 8.0500
                          3 male 35
 ## 6
                          3 male NA
                                              0 8.4583
Imputação de valores faltantes
 # Identificação de valores irregulares e torná-los NA
 data[data == ''] = NA
```

0 7.2500

Fare Embarked

SibSp Parch Age Fare ## Mode :logical Mode :logical Mode :logical

Survived

Mode :logical Mode :logical Mode :logical

FALSE:1309

summary(is.na(data)) # 263 NAs em Age / 1 NA em Fare / 2 NA em Embarked

Pclass

FALSE:1309

```
Mode :logical
## FALSE:1046
                   FALSE:1309
                                   FALSE:1309
                                                  FALSE:1308
## TRUE :263
                                                   TRUE :1
    Embarked
## Mode :logical
## FALSE:1307
## TRUE :2
```

Sex

FALSE:1309

Mode :logical

```
Temos alguns dados faltantes: 263 na variável "Age", 1 na variável "Fare" e 2 na variável "Embarked".
Imputação da variável Embarked
Como Embarked é uma variável categórica, vamos imputar com a moda.
 table(data$Embarked)
 ## C Q S
```

Como a variável Age é contínua vamos analisar se podemos imputar a mediana. Iremos verificar a distribuição dos dados para cada categoria da

variável Sex sem os outliers. Caso as distribuições de Age em cada categoria de Sex forem diferentes, não podemos imputar com a mediana.

Imputação da variável Age

geom_boxplot(outliers = F) + theme(legend.position = 'none')

stat_boxplot(geom = 'errorbar', width = 0.3) +

plt1 = ggplot(data, aes(x = Age, color = Sex, fill = Sex)) +

geom_histogram(position="identity", alpha = 0.5) +

plt2 = ggplot(data, aes(x = Age, y = Sex, fill = Sex)) +

geom_histogram(position="identity", alpha = 0.5) +

plt2 = ggplot(data, aes(x = Fare, y = Sex, fill = Sex)) +

theme_classic()+ theme(legend.position="top") +

stat_boxplot(geom = 'errorbar', width = 0.3) +

scale_fill_discrete(name = "") + scale color discrete(name = "")

geom_boxplot(outliers = F) + theme(legend.position = 'none')

grid.arrange(plt1, plt2, ncol=2)

300

dummies.

data,

Criação de variáveis dummy

remove_first_dummy = TRUE,

 $data_aux = data[,-c(1,2)]$

data_aux = complete(imp, 1)

data_train = data[1:891,-1]

data_train_aux = data_train

set.seed(28051996)

set.seed(28051996)

Age

SibSp ## Parch

Fare

##

Sex_male

Pclass_2

Pclass_3

AIC: 805.04

Parch

Pclass_2

Fare

Embarked_Q -0.056267

-0.093329

-0.919468

-2.150048

Embarked_S -0.434226 0.239530 -1.813 0.06986 .

data_train_aux = data_train

Reconstrução do dataset final

data = cbind(data[,c(1,2)], data_aux)

set.seed(28051996) imp = mice(data_aux)

remove_selected_columns = TRUE

Imputação final com MICE para Fare

select_columns = c('Sex', 'Pclass', 'Embarked'),

data = dummy_cols(

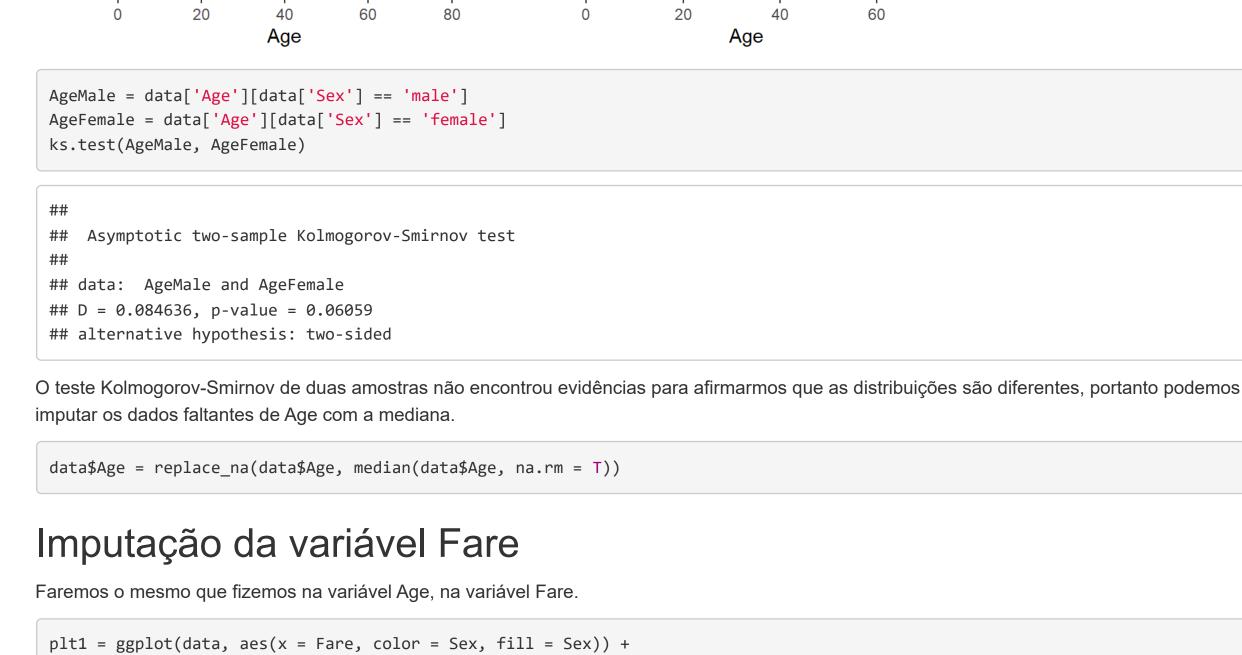
theme_classic()+ theme(legend.position="top") +

scale_fill_discrete(name = "") + scale_color_discrete(name = "")

data\$Embarked = replace na(data\$Embarked, 'S')

grid.arrange(plt1, plt2, ncol=2)

```
female male
80
                                           male
60
                                        Sex
```



female

female

male -



O teste de Kolmogorov-Smirnov para duas amostras encontrou evidências para afirmarmos que as distribuições de Fare para cada nível de Sex

Antes de utilizarmos o MICE precisamos remover as variáveis que não podemos usar (Passengerld, pois é somente um índice; e Survived, pois é

nossa variável resposta, logo comprometeria nosso modelo preditivo) e colocar as variáveis categóricas (Sex, Pclass e Embarked) como

são diferentes. Vamos utilizar MICE para imputar na observação faltante já que não há problemas em usar.

model = glm(Survived ~., family = 'binomial'(link = 'logit'), data = data_train_aux)

Vamos usar o Variance Inflation Factor para medir a multicolinearidade entre as variáveis do modelo.

0.118856 -0.785 0.43232

0.297326 -3.092 0.00199 **

0.381471 -0.148 0.88274

0.297720 -7.222 5.13e-13 ***

-2.719444 0.200977 -13.531 < 2e-16 ***

0.002339 0.002469 0.947 0.34346

Signif. codes: 0 '***' 0.001 '**' 0.05 '.' 0.1 ' ' 1

Null deviance: 1186.66 on 890 degrees of freedom

-0.093329 0.118856 -0.785 0.43232

0.002339 0.002469 0.947 0.34346

roc_obj = roc(response = data_train\$Survived, predictor = y_pred)

auc_value = auc(roc_obj) # AUC = 0,854

9.0

0.4

0.2

0.0

Sensitivity

Sex male -2.719444 0.200977 -13.531 < 2e-16 ***

Pclass_3 -2.150048 0.297720 -7.222 5.13e-13 ***

Embarked_Q -0.056267 0.381471 -0.148 0.88274

(Dispersion parameter for binomial family taken to be 1)

Residual deviance: 785.04 on 881 degrees of freedom

Verificação de multicolinearidade

data_test = data[892:1309,] Modelo preditivo usando regressão logística Definição do modelo

```
model = glm(Survived ~., family = 'binomial'(link = 'logit'), data = data_train_aux)
 vif(model)
           Age
                    SibSp
                               Parch
                                          Fare Sex_male Pclass_2 Pclass_3
     1.300121 1.264545
                          1.286850 1.610626 1.210637 2.028611 2.769354
 ## Embarked_Q Embarked_S
     1.519696 1.465691
Podemos verificar que não há multicolinearidade forte entre as variáveis. Portanto, podemos seguir com a implementação do modelo.
 summary(model)
 ## Call:
 ## glm(formula = Survived ~ ., family = binomial(link = "logit"),
        data = data_train_aux)
 ## Coefficients:
                Estimate Std. Error z value Pr(>|z|)
 ## (Intercept) 4.064159 0.472813 8.596 < 2e-16 ***
```

```
## Number of Fisher Scoring iterations: 5
As variáveis Parch, Fare e Embarked não são estatisticamente significantes, por isso podemos removê-las.
 model2 = glm(Survived ~., family = 'binomial'(link = 'logit'), data = data_train_aux)
 summary(model2)
 ##
 ## Call:
 ## glm(formula = Survived ~ ., family = binomial(link = "logit"),
 ## data = data_train_aux)
 ## Coefficients:
              Estimate Std. Error z value Pr(>|z|)
 ## (Intercept) 4.064159 0.472813 8.596 < 2e-16 ***
             ## Age
             ## SibSp
```

```
## Embarked_S -0.434226 0.239530 -1.813 0.06986 .
 ## Signif. codes: 0 '***' 0.001 '**' 0.05 '.' 0.1 ' ' 1
 ## (Dispersion parameter for binomial family taken to be 1)
        Null deviance: 1186.66 on 890 degrees of freedom
 ## Residual deviance: 785.04 on 881 degrees of freedom
 ## AIC: 805.04
 ## Number of Fisher Scoring iterations: 5
 performance::r2(model2)
 ## # R2 for Logistic Regression
 ## Tjur's R2: 0.409
As variáveis que temos não são muito boas, portanto não poderíamos esperar um R² alto.
Curva ROC
Vamos utilizar a curva ROC e o AUC para definirmos o limiar que melhor se adequa ao modelo. Utilizaremos o limiar que minimiza a raiz
quadrada do erro quadrático médio.
 y_test_pred = predict(model2, newdata = data_test, type = 'response')
 y_pred = predict(model2, type = 'response')
 set.seed(28051996)
```

0.8

plot.roc(roc_obj); legend("bottomright", legend = paste("AUC =", round(auc_value, 3)), col = "#1c61b6", lwd = 2)

AUC = 0.856

O melhor limiar será escolhido pela estatística J de Youden, que é o limiar que maximiza a soma da sensibilidade com a especificidade. threshold = coords(roc_obj, "best", ret = "threshold")[[1]] # melhor limiar y_test_pred = ifelse(y_test_pred >= threshold, 1, 0)