Kaggle - Titanic

Marco Antônio Aragão Rocha

Objetivo

Meu objetivo é modelar a base de passageiros do Titanic e prever quais tripulantes sobreviveram ao naufrágio ou não, para isso irei classificar como 0 ou 1 no conjunto de teste. A métrica utilizada pelo problema é a acurácia que consiste em (TP+TN) / (TP+FP +TN +FN)

Importando os Dados

Obs: Nesse problema do kaggle as bases de treino e teste já foram separadas mas como o target não é visível na base de teste disponibilizada, visando ter uma validação mais robusta, também irei dividir a base de treino em uma base de teste e outro de treino assim será possível acompanhar de perto as métricas do modelo.

```
# Carregando pacotes
library(tidyverse)
library(pROC)
# Importando a base de dados
dados <- read_csv("train.csv")
```

Sanity nos Dados

Verificar a quantidade de Nas por coluna

```
qtd_na <- colSums(is.na(dados))
qtd_na
```

```
PassengerId    Survived      Pclass        Name         Sex         Age
          0           0           0           0           0         177
      SibSp       Parch      Ticket        Fare       Cabin    Embarked
          0           0           0           0         687           2
```

As variáveis Age e Cabin apresentam muitos Nas, é nescessário pensar em formas de lidar com isso

Análise Exploratória dos Dados

O Primeiro passo para entender melhor os dados e os problemas e tirar insights é por meio da análise exploratória que sera feita a seguir.

```
head(dados)
```

```
# A tibble: 6 × 12
  PassengerId Survived Pclass Name    Sex     Age SibSp Parch Ticket
Fare Cabin
```

```
        <dbl>    <dbl>  <dbl> <chr>   <chr> <dbl> <dbl> <dbl> <chr>
<dbl> <chr>
1           1        0      3 Braund… male     22     1     0 A/5 2…
7.25 <NA>
2           2        1      1 Cuming… fema…    38     1     0 PC 17… 71.3
C85
3           3        1      3 Heikki… fema…    26     0     0 STON/…
7.92 <NA>
4           4        1      1 Futrel… fema…    35     1     0 113803 53.1
C123
5           5        0      3 Allen,… male     35     0     0 373450
8.05 <NA>
6           6        0      3 Moran,… male     NA     0     0 330877
8.46 <NA>
# ℹ 1 more variable: Embarked <chr>
```

Gerando Visualizações e Transfromando os Dados

Conforme vemos a base de dados consiste em 891 observações e 12 variávies, agora vamos gerar visualizações com o ggplot para entender melhor o comportamento dos dados.

```
summary(dados)
```

```
  PassengerId       Survived          Pclass          Name
 Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891
 1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character
 Median :446.0   Median :0.0000   Median :3.000   Mode  :character
 Mean   :446.0   Mean   :0.3838   Mean   :2.309
 3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
 Max.   :891.0   Max.   :1.0000   Max.   :3.000

     Sex                 Age            SibSp           Parch
 Length:891         Min.   : 0.42   Min.   :0.000   Min.   :0.0000
 Class :character   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000
 Mode  :character   Median :28.00   Median :0.000   Median :0.0000
                    Mean   :29.70   Mean   :0.523   Mean   :0.3816
                    3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
                    Max.   :80.00   Max.   :8.000   Max.   :6.0000
                    NA's   :177
    Ticket               Fare           Cabin              Embarked
 Length:891         Min.   :  0.00   Length:891         Length:891
 Class :character   1st Qu.:  7.91   Class :character   Class :character
 Mode  :character   Median : 14.45   Mode  :character   Mode  :character
                    Mean   : 32.20
                    3rd Qu.: 31.00
                    Max.   :512.33
```

```
sapply(dados,class)
```

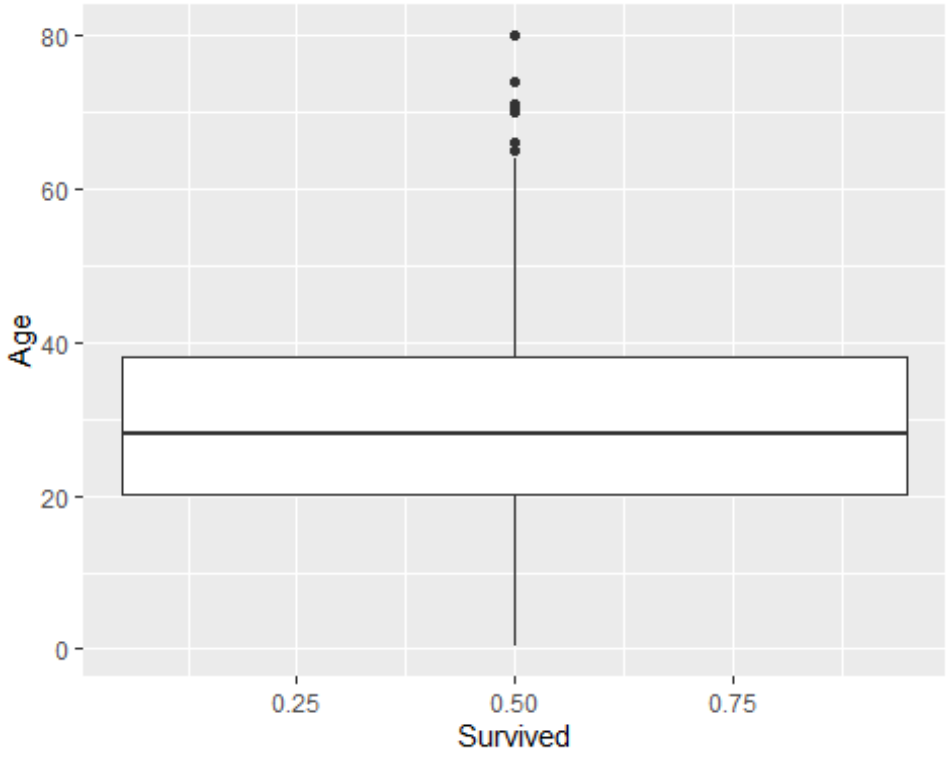
```
PassengerId    Survived      Pclass        Name         Sex         Age
  "numeric"   "numeric"   "numeric" "character" "character"   "numeric"
      SibSp       Parch      Ticket        Fare       Cabin    Embarked
  "numeric"   "numeric" "character"   "numeric" "character" "character"
```

Observando a classe de cada variável vemos que algumas não estão com a classificação correta, isso pode gerar problemas tanto na parte de visualização quanto na parte de rodar o modelo. Alguns exemplos de variavies q devem ser recategorizadas como fator: Survided, Pclass, Sex, Embarked.

Caso isso não seja feito teremos problemas como o gráfico mostrado a seguir.

```
ggplot(data = dados, mapping = aes(x = Survived, y = Age))+
geom_boxplot()
```

```
Warning: Continuous x aesthetic
ℹ did you forget `aes(group = ...)`?

Warning: Removed 177 rows containing non-finite values
(`stat_boxplot()`).
```

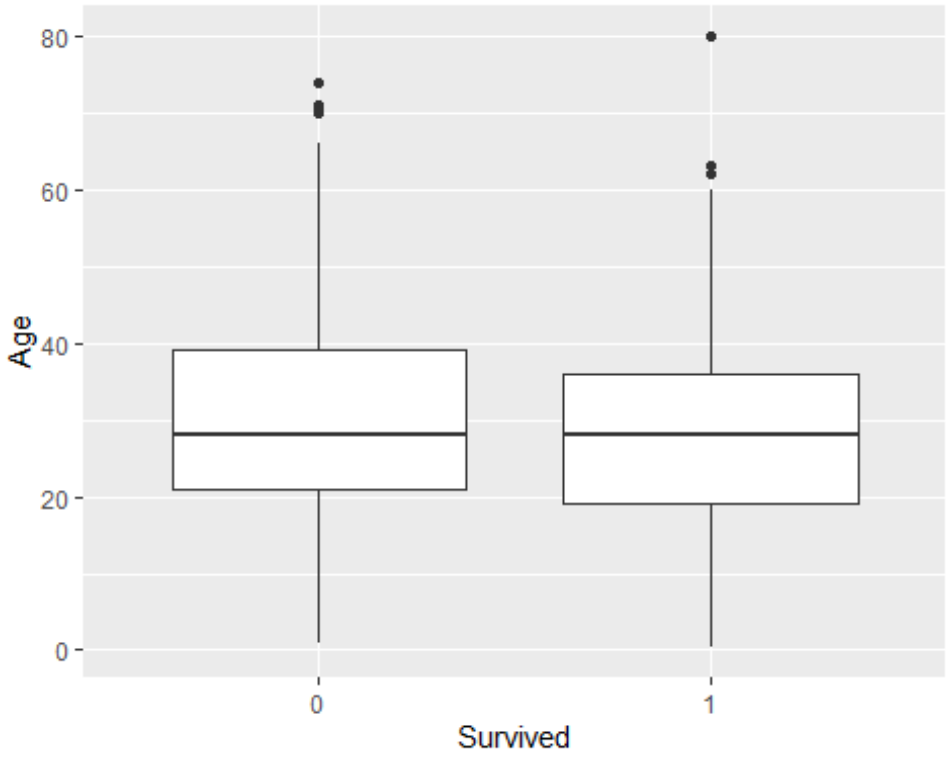


Corrigindo o problema por meio da recategorização das variáveis.

```
dados$Survived <- as.factor(dados$Survived)
dados$Sex <- as.factor(dados$Sex)
dados$Pclass <- as.factor(dados$Pclass)
dados$Embarked <- as.factor(dados$Embarked)
attach(dados)
```

```
ggplot(data = dados, mapping = aes(x = Survived, y = Age))+
geom_boxplot()
```

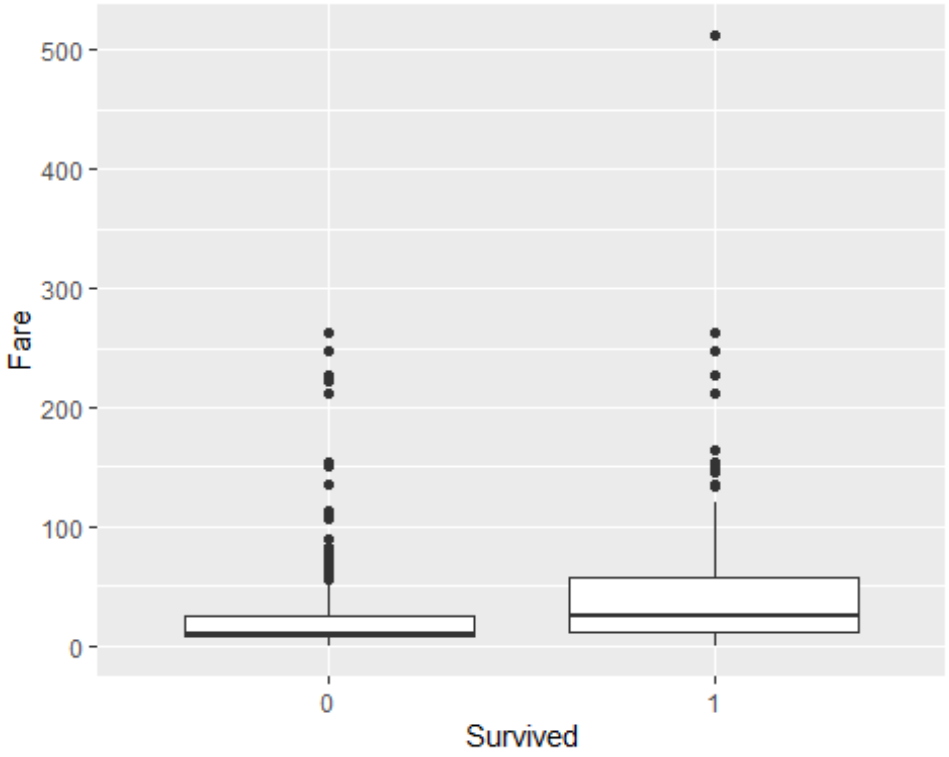
```
Warning: Removed 177 rows containing non-finite values
(`stat_boxplot()`).
```



No boxplot vemos uma sutil diferença entre a idade das pessoas q sobreviveram (S=1) e das pessoas que morreram (S=0)

Agora vamos tentar ver alguma relação com a tarifa dos passageiros.

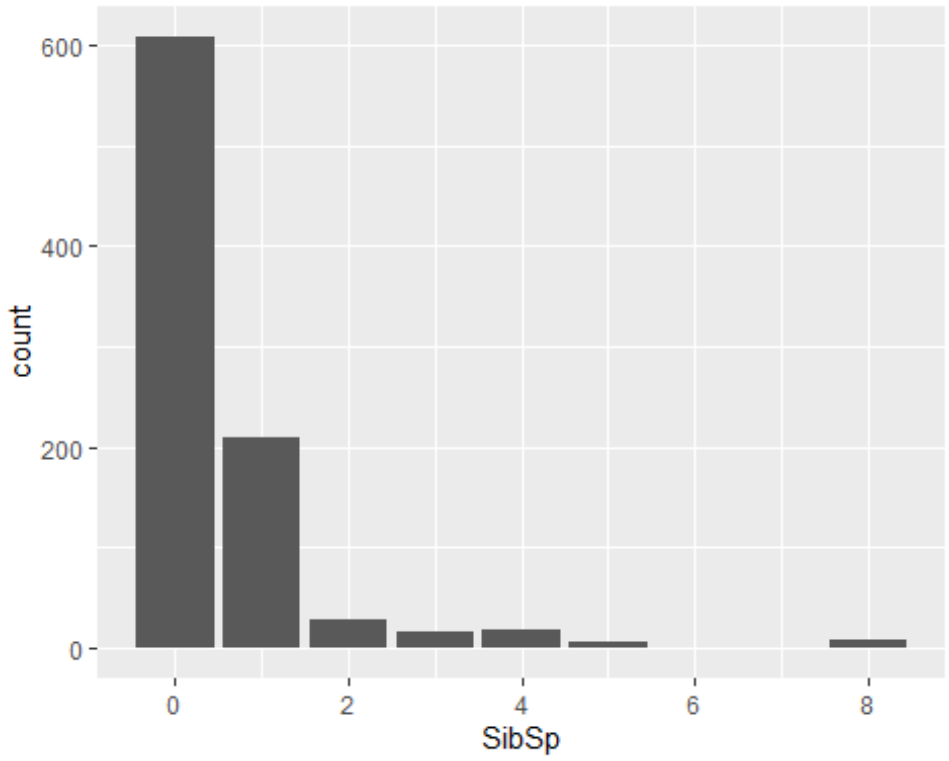
```
ggplot(data = dados, mapping = aes(x = Survived, y = Fare))+
geom_boxplot()
```



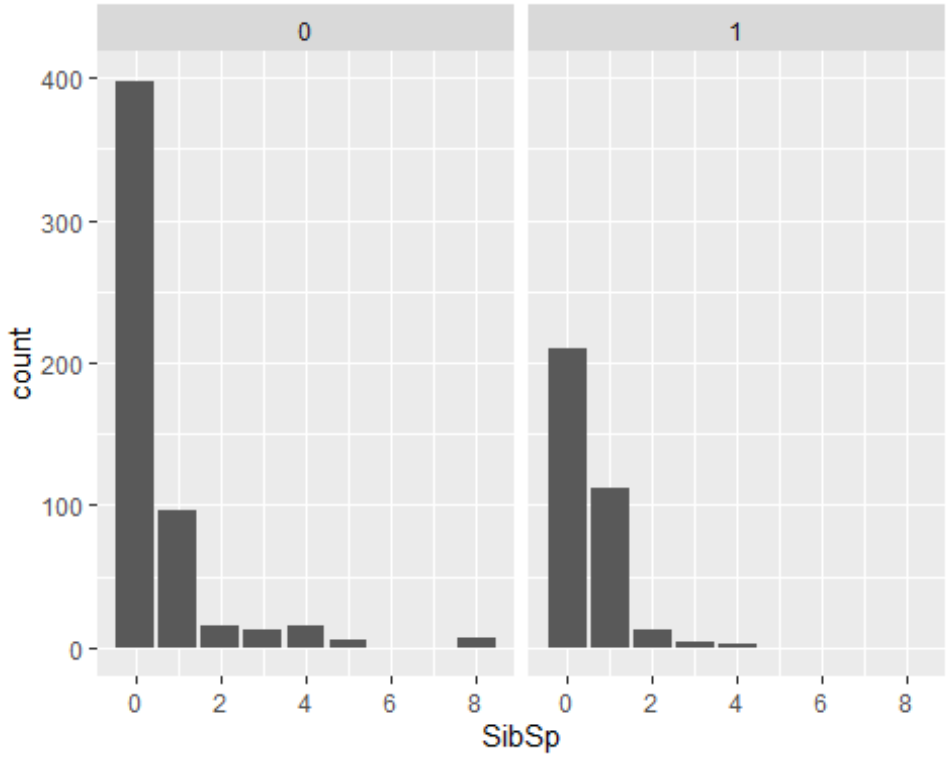
É possível notar uma diferença mais significativa nos boxplots quando se analisdo a tarifa paga.

Análise do impacto do número irmãos/cônjuges a bordo do Titanic e a sobrevivência

```
ggplot(data = dados, mapping = aes(x = SibSp))+ geom_bar()
```



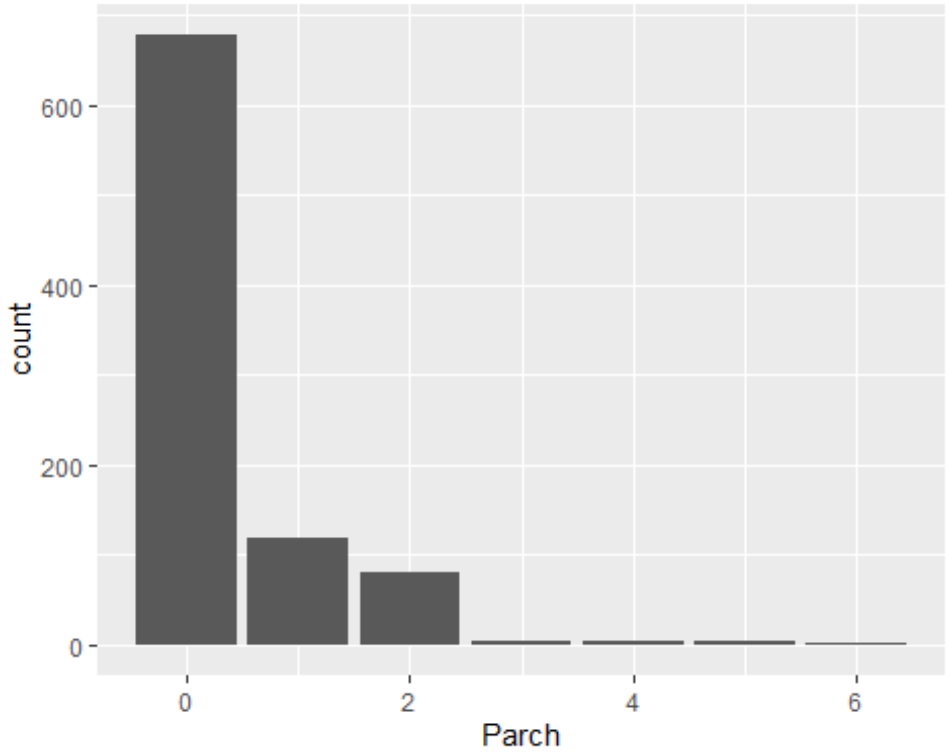

```
ggplot(data = dados, mapping = aes(x = SibSp))+
geom_bar()+facet_wrap(~Survived)
```



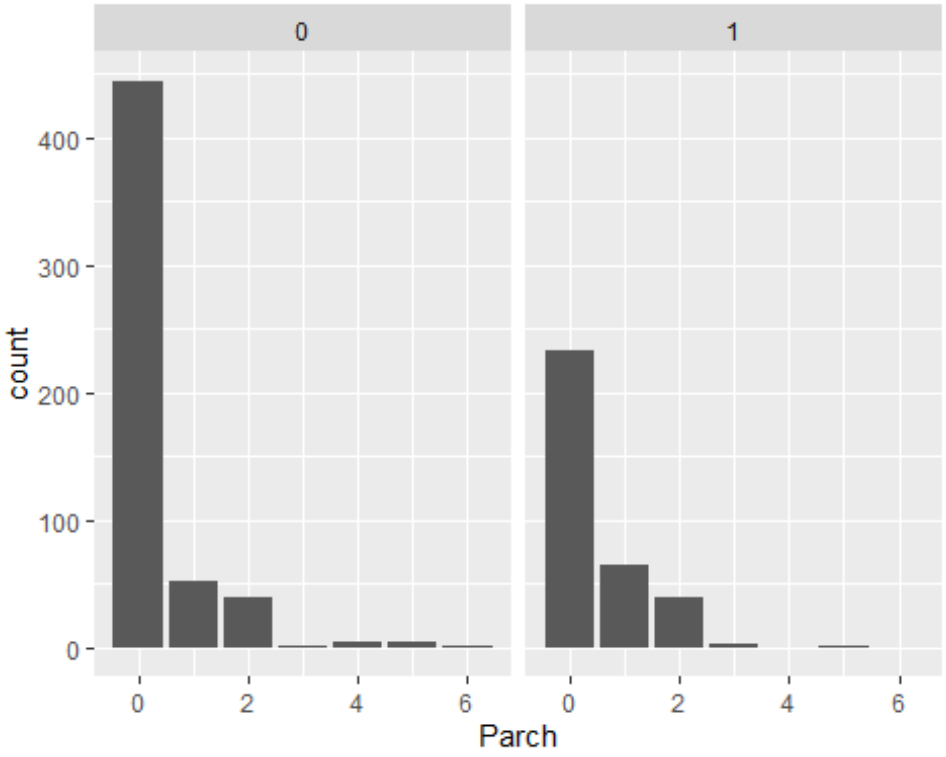
Observa-se que nenhuma pessoa com mais que 4 irmãos/cônjuges a bordo conseguiu sobreviver.

Analisando o impacto do número de de pais/filhos a bordo do Titanic na sobrevivência

```
ggplot(data = dados, mapping = aes(x = Parch))+ geom_bar()
```




```
ggplot(data = dados, mapping = aes(x = Parch))+
geom_bar()+facet_wrap(~Survived)
```

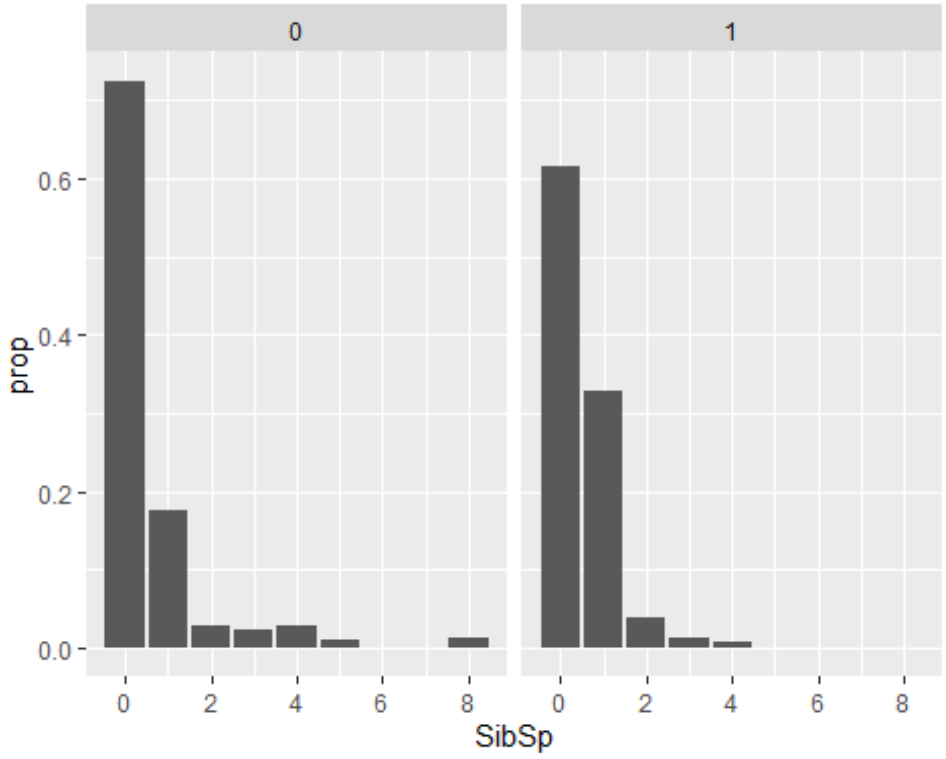


Esses gráficos não são tão informativos porque o número de pessoas que sobreviveram e não sobreviveram são muito diferentes, dessa maneira estamos mais interresados em ver o percentual de de pessoas dentre as que sobreviveram ou não que tinham filhos abordo etc...

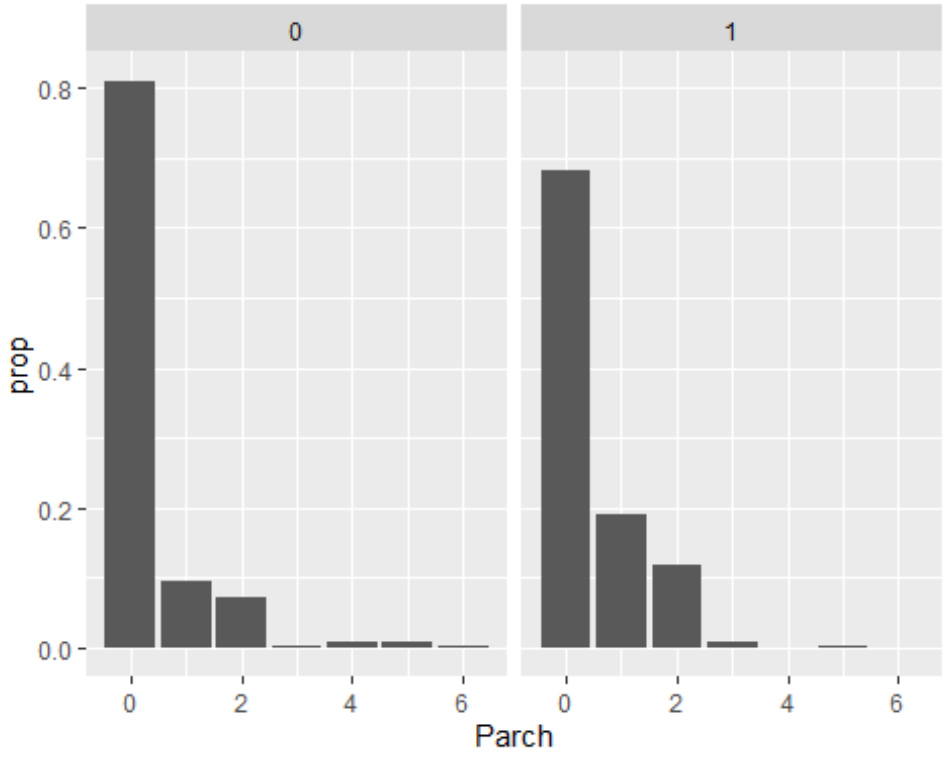
Repetindo os gráficos com o percentual por classe

```
ggplot(data = dados, mapping = aes(x = SibSp, y= ..prop..))+
geom_bar()+facet_wrap(~Survived)
```

```
Warning: The dot-dot notation (`..prop..`) was deprecated in ggplot2
3.4.0.
ℹ Please use `after_stat(prop)` instead.
```



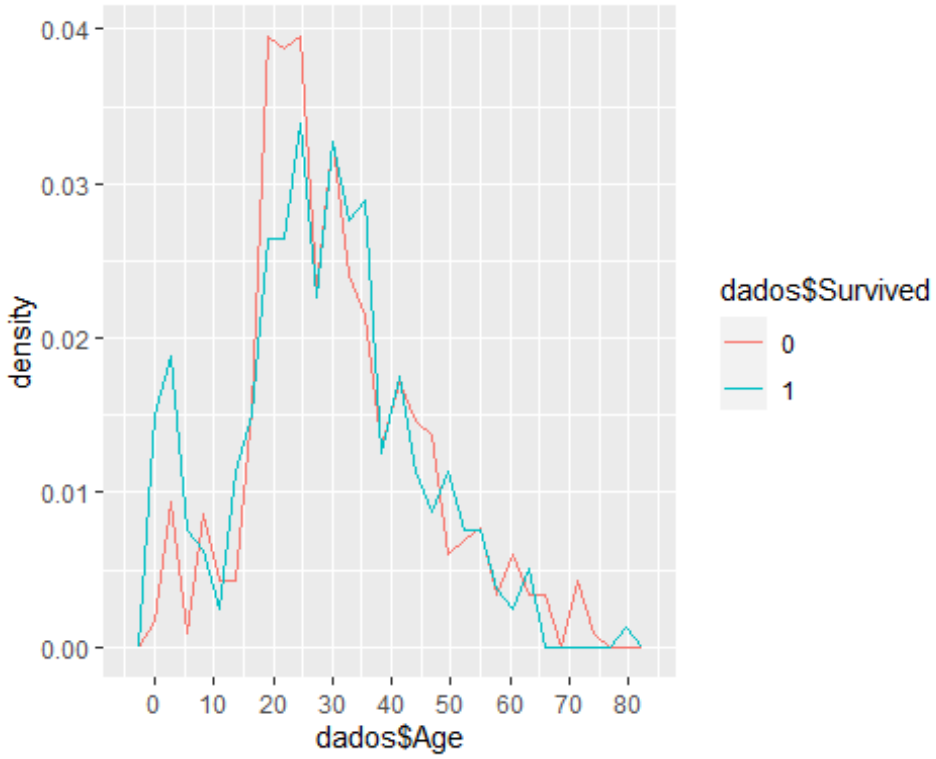

```
ggplot(data = dados, mapping = aes(x = Parch, y= ..prop..))+
geom_bar()+facet_wrap(~Survived)
```



Analisando a Idade

```
ggplot(data= dados, aes(x = dados$Age, y = ..density.. , col=
dados$Survived)) + geom_freqpoly()+  xlim(0,80) +
scale_x_continuous(n.breaks=10)
```

```
Scale for x is already present.
Adding another scale for x, which will replace the existing scale.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning: Removed 177 rows containing non-finite values (`stat_bin()`).
```

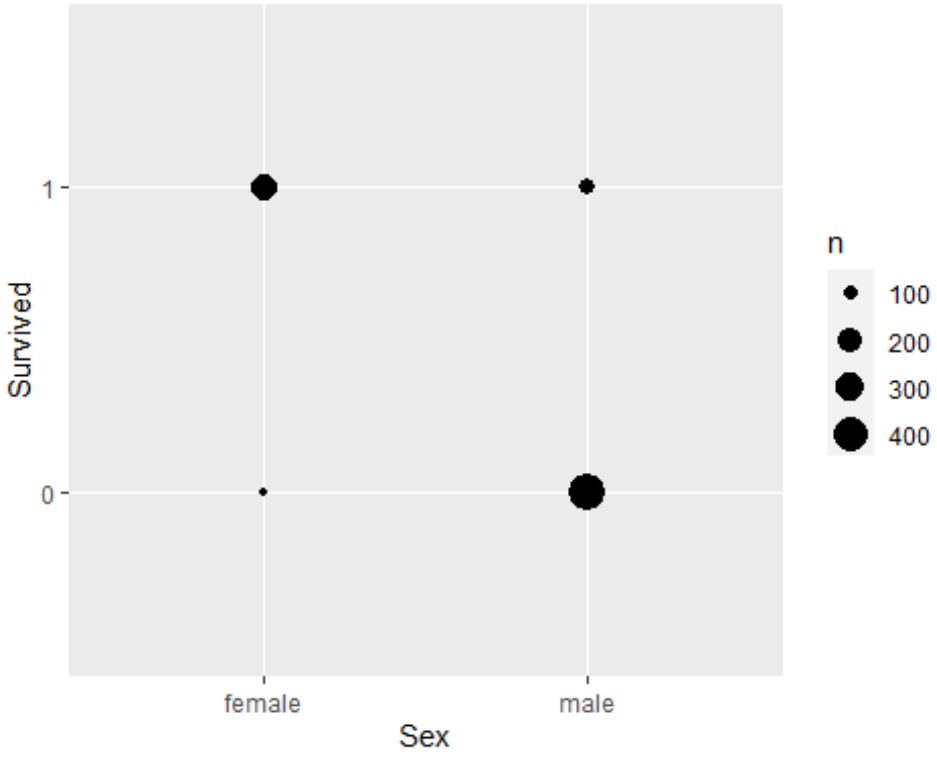



```
min(dados$Age, na.rm = T)
```

```
[1] 0.42
```

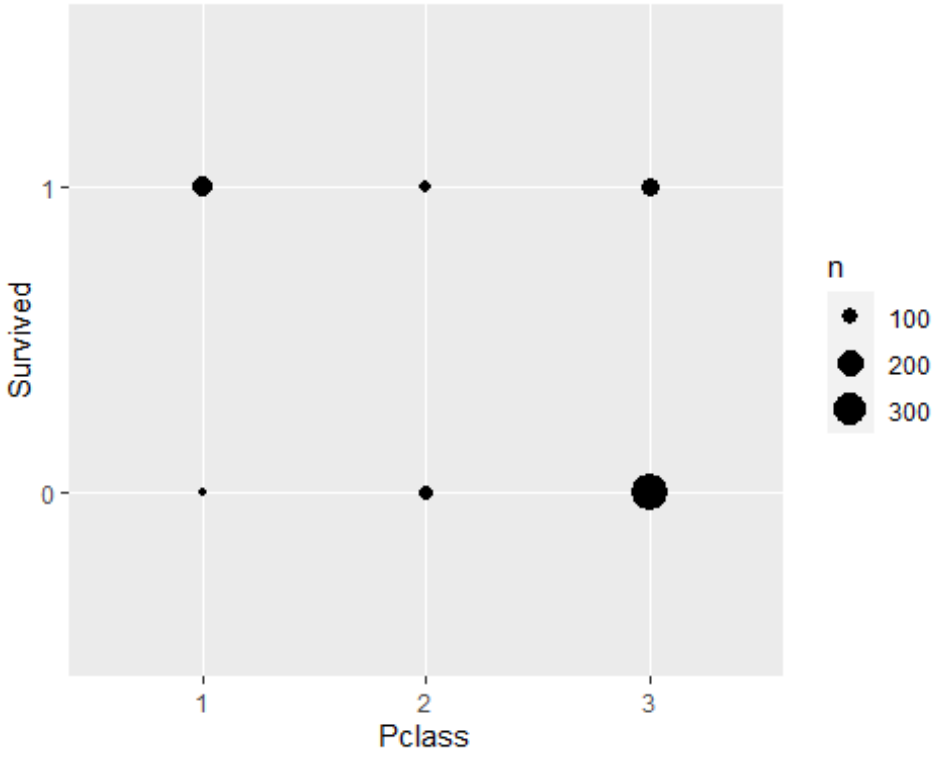
Analisando variável categórica com variável categórica

```
ggplot(data = dados) +
  geom_count(mapping = aes(x = Sex, y = Survived))
```



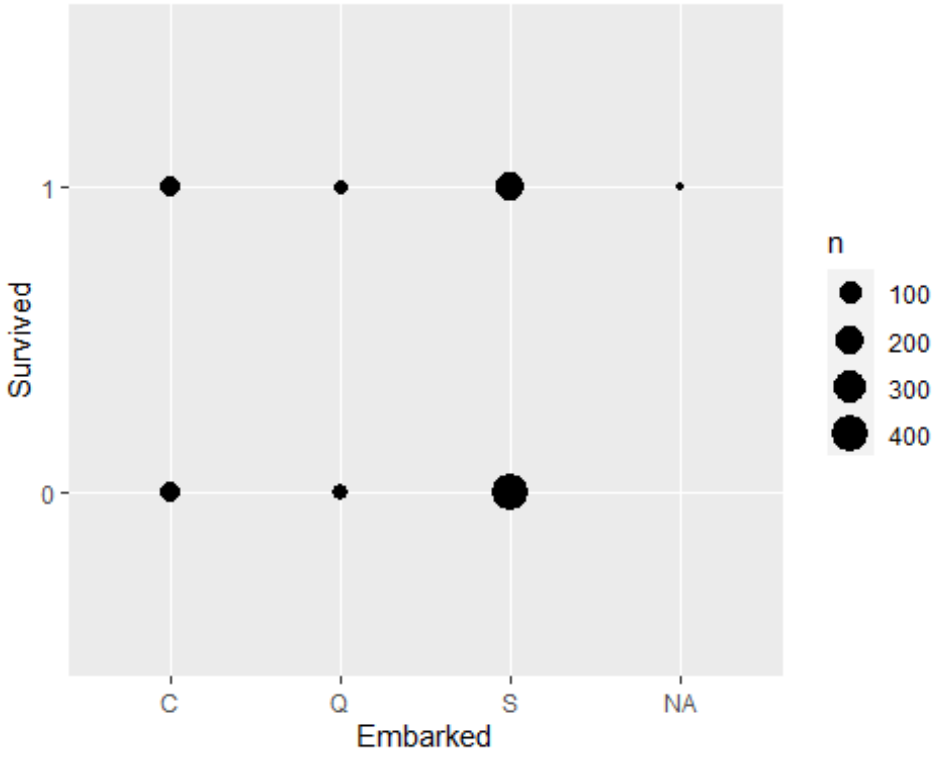
Observa-se que o sexo é um fator muito importante para determinar a sobrevivência, a grande maioria da mulheres sobrevivem e a grande maioria dos homens morrem.

```
ggplot(data = dados) +
  geom_count(mapping = aes(x = Pclass, y = Survived))
```



Nota-se que a classe do ticket também é relevnte para determinar a sobrevivência.

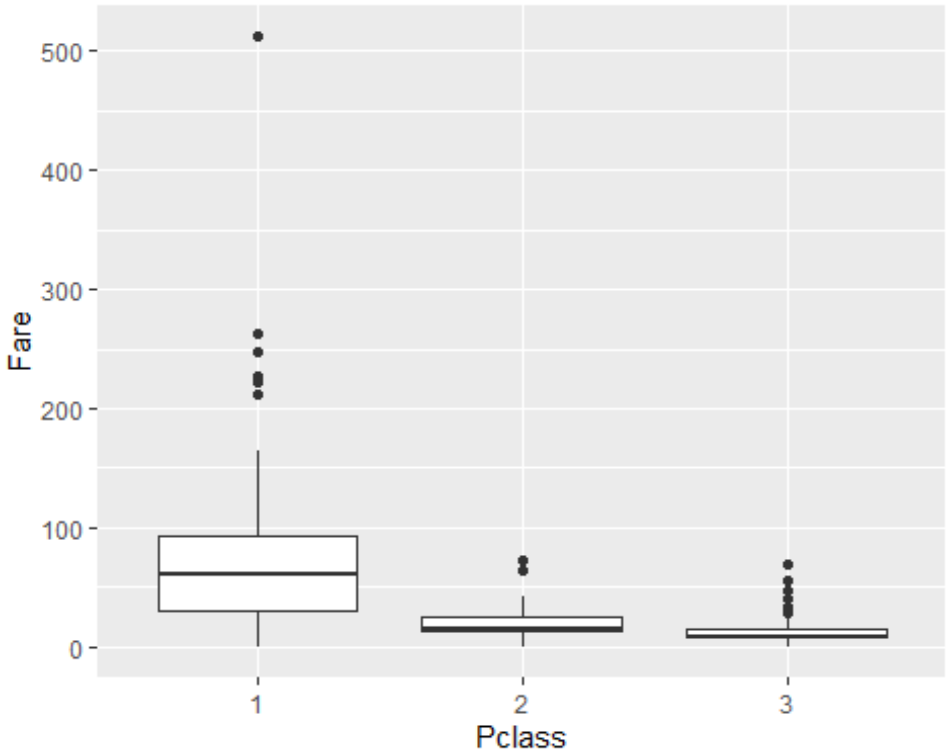
```
ggplot(data = dados) +
  geom_count(mapping = aes(x = Embarked, y = Survived))
```



Observa-se que o porto de embarque tem um impacto menor na sobrevivência.

Verificando multicolinearidade entre Pclass e Fare

```
ggplot(data = dados, mapping = aes(x = Pclass, y = Fare))+ geom_boxplot()
```



As duas variáveis aparentam estar fortemente correlacionadas e explicam a mesma coisa.

Verificar FIV = 1/(1-R^2)

```
ajuste <- lm(Fare ~ Pclass)
summary(ajuste)


Call:
lm(formula = Fare ~ Pclass)

Residuals:
   Min     1Q Median     3Q    Max
-84.15  -6.93  -5.75   5.03 428.17

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   84.155      2.723   30.91   <2e-16 ***
Pclass2      -63.493      4.014  -15.82   <2e-16 ***
Pclass3      -70.479      3.267  -21.57   <2e-16 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 40.01 on 888 degrees of freedom
Multiple R-squared:  0.3531,    Adjusted R-squared:  0.3516
F-statistic: 242.3 on 2 and 888 DF,  p-value: < 2.2e-16
```

O R^2 ajustado foi de 0,3516

```
# Calculando FIV
FIV <- 1/(1- 0.3516 )
FIV

[1] 1.542258

FIV < 5

[1] TRUE

# A multicolinearidade entre as variáveis não é severa.
```

Modelagem

A princípio abordaremos o problema usando técnicas de modelos lineares generalizados para dados Bernoulli/ Binários.

Regressão Logística

Considera a função de ligação o parâmetro canônico b(p)= ln(p/p-1).

Dessa forma teremos: ln(p/1-p)= xi^t * B = Ni (preditor linear)

p = 1 / (1 + e^-Ni)

Ex: ln(p/1-p) = b0 + b1*x1

Separando a base em Treino e Teste (falta validação)

O banco de dados de 891 observações será particionado em 700 observações para o banco treino e 191 observações para o banco teste, por meio de um sorteio.

```
set.seed(1)
index = sort(sample(nrow(dados), 700, replace=F))
table(dados$Survived[index])


  0   1
427 273

train.db <- dados[index,]
valid.db <- dados[-index,]
```

Feito essa separação e evitado o data leakage vamos fazer o 1° modelo

```
detach(dados)
attach(train.db)
```

Modelo 1

```
modelo1 <- glm(Survived ~ Pclass + Sex , family = binomial(link =
'logit'))

summary(modelo1)
```

```
Call:
glm(formula = Survived ~ Pclass + Sex, family = binomial(link = "logit"))

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   2.3183     0.2488   9.316  < 2e-16 ***
Pclass2      -0.9082     0.2759  -3.291 0.000997 ***
Pclass3      -1.9588     0.2449  -7.998 1.27e-15 ***
Sexmale      -2.6073     0.2056 -12.680  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 936.25  on 699  degrees of freedom
Residual deviance: 653.59  on 696  degrees of freedom
AIC: 661.59

Number of Fisher Scoring iterations: 4
```

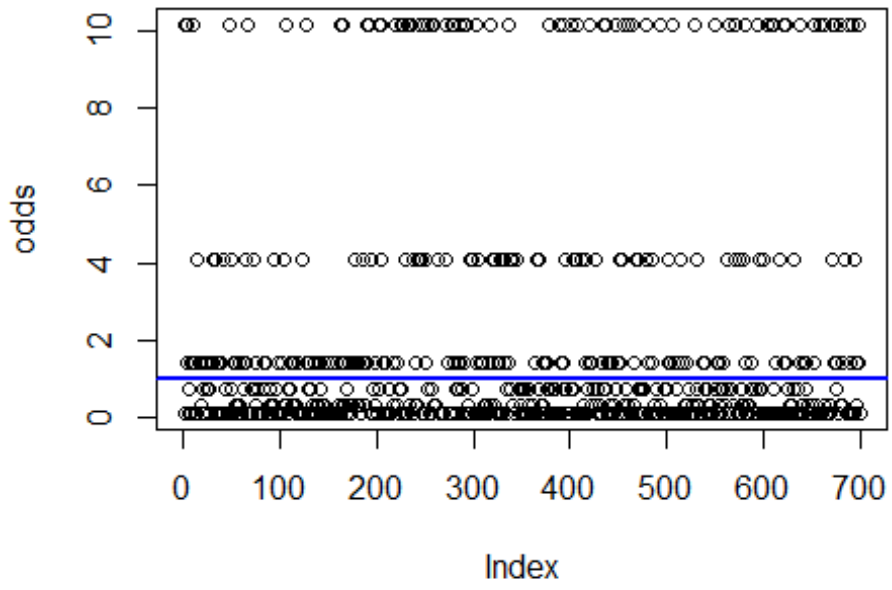
Todas as variáveis são significativas para o modelo

Fazendo teste da Anova

```
anova(modelo1, test="LR")
```

```
Analysis of Deviance Table

Model: binomial, link: logit

Response: Survived

Terms added sequentially (first to last)

       Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                     699     936.25
Pclass  2   83.318       697     852.93 < 2.2e-16 ***
Sex     1  199.336       696     653.59 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
eta = predict(modelo1)
odds = exp(eta)
plot(odds)+ abline(h=1, col='blue', lwd = 2)
```




```
integer(0)
```

Se oddsi = 1, as chances de morte ou sobrevivência são iguais para o tripulante i. Se oddsi > 1, o tripulante i tem chance de sobrevivência maior que a de morte. Se oddsi < 1, o tripulante i tem chance de sobrevivência menor que a de morte.

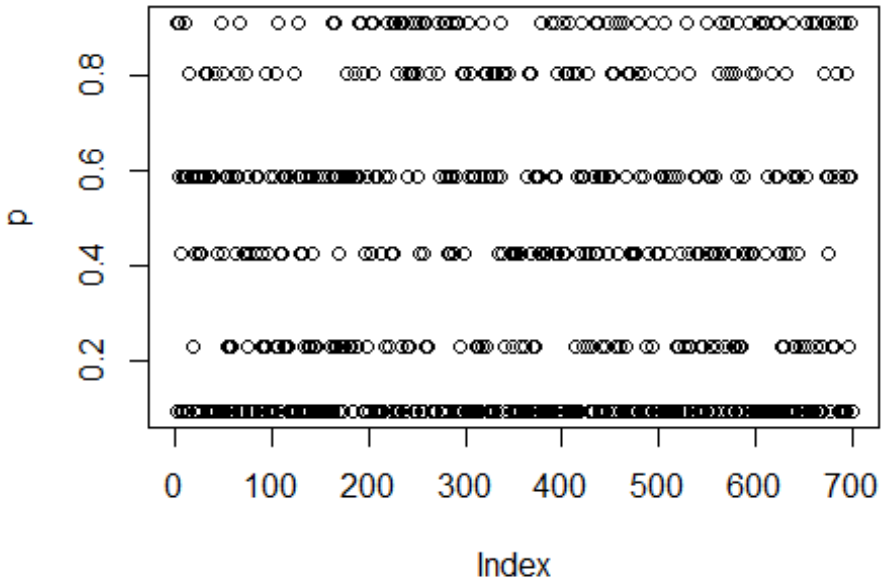
Interpretação

```
100* (exp(modelo1$coefficients['Sexmale'])-1)

  Sexmale
-92.62688
```

Ser homen causa uma redução de 92,5% na razão de chance de sobreviver.

Achando a Probabilidade

```
eta = predict(modelo1)
odds = exp(eta)
p = 1/(1+exp(-eta))
plot(p)
```

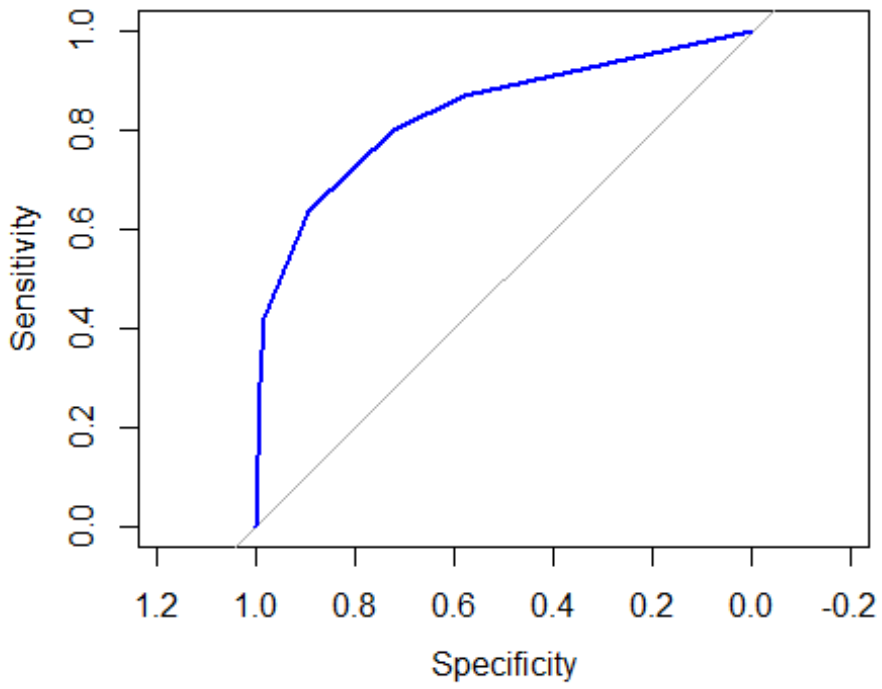


Calculando as probabilidades nos dados de validação

```
probabilidade =  1/(1+ exp(-predict(object = modelo1, newdata =
valid.db)))
```

Curva Roc

```
curva = roc(valid.db$Survived ~ probabilidade)

Setting levels: control = 0, case = 1

Setting direction: controls < cases

plot(curva, col = 'blue')
```

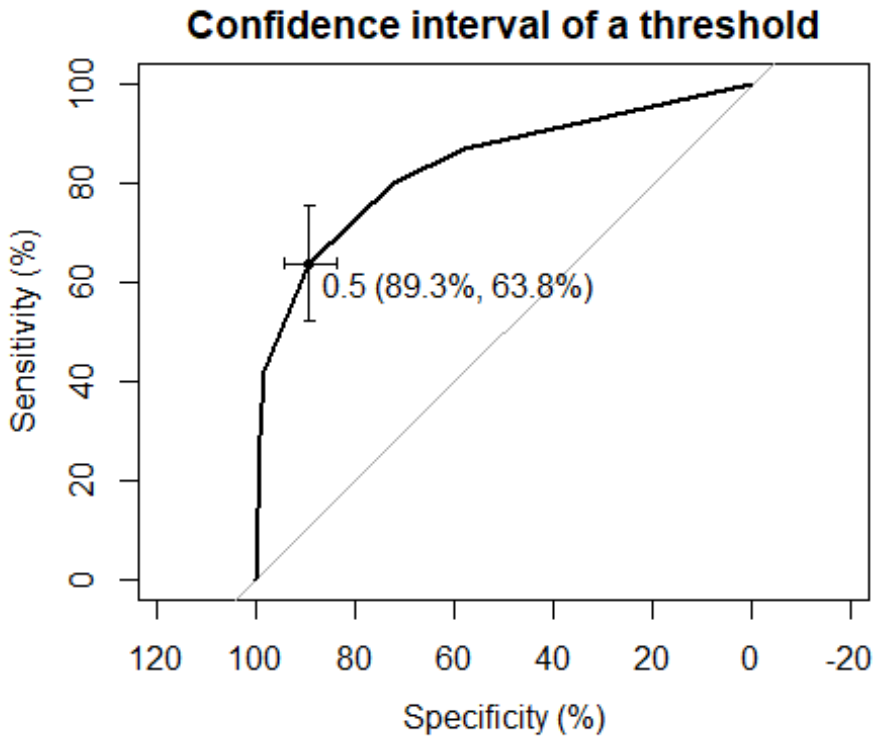


Escolher um critério de classificação

```
# não sei extamente como fazer
# acredito q tenha q fazer curva roc/ olhar AUC para diferentes criterios
plot.roc(valid.db$Survived , probabilidade,
         main="Confidence interval of a threshold", percent=TRUE,
         ci=TRUE, of="thresholds", # compute AUC (of threshold)
         thresholds="best", # select the (best) threshold
         print.thres="best") # also highlight this threshold on the plot
```

```
Setting levels: control = 0, case = 1
```

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



Fazendo Previsão

```
# Calculando para a base teste do kaggle

teste_kaggle <- read_csv("test.csv")

Rows: 418 Columns: 11
── Column specification
────────────────────────────────────────────────────────
Delimiter: ","
chr (5): Name, Sex, Ticket, Cabin, Embarked
dbl (6): PassengerId, Pclass, Age, SibSp, Parch, Fare

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.

teste_kaggle$Pclass <- as.factor(teste_kaggle$Pclass)

prob_prevista2 =  1/(1+ exp(-predict(object = modelo1, newdata =
teste_kaggle)))

teste_kaggle$prob_prevista2 <- prob_prevista2

previsao <- ifelse(teste_kaggle$prob_prevista2 > 0.5, 1, 0)

teste_kaggle$previsao <- previsao
```

```r
submissao_reg_log <- data.frame(
  PassengerId = teste_kaggle$PassengerId,
  Survived = teste_kaggle$previsao
)
head(submissao_reg_log, 10)
```

```
   PassengerId Survived
1          892        0
2          893        1
3          894        0
4          895        0
5          896        1
6          897        0
7          898        1
8          899        0
9          900        1
10         901        0
```

```r
# write_csv(submissao_reg_log, 'submissao_reg_log.csv')
```

Modelo 2

Criando Variável

```r
train.db$Age_No_NA <- ifelse(is.na(train.db$Age),mean(train.db$Age, na.rm = T),train.db$Age)
train.db$crianca <- ifelse(train.db$Age_No_NA < 6,1,0)
```

Removendo NA Embarked

```r
train.db_sem_na <- train.db
train.db_sem_na <- train.db_sem_na[-is.na(train.db$Embarked),]
```

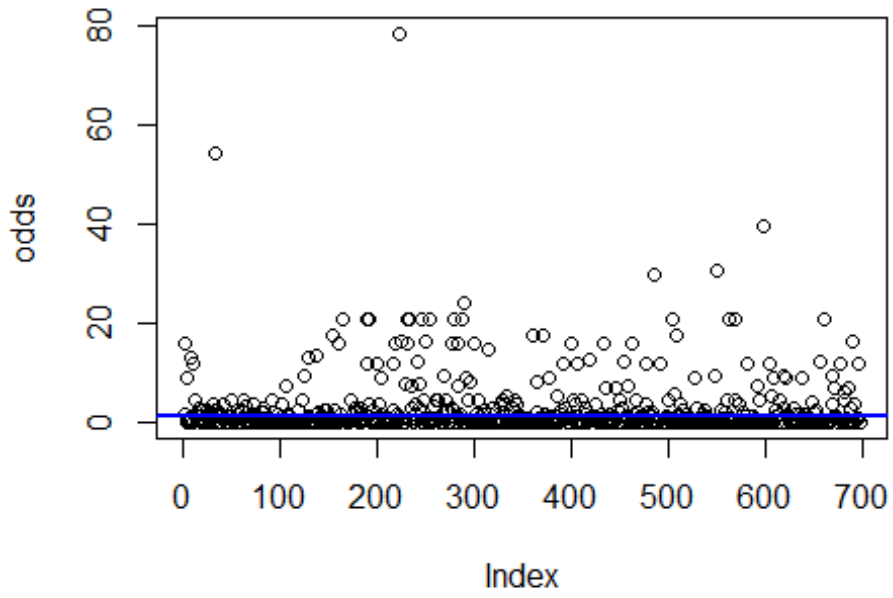
Ajustando Modelo

```r
detach(train.db)
attach(train.db_sem_na)

modelo2 <- glm(Survived ~ Pclass + Sex + crianca + SibSp + Parch + Embarked , family = binomial(link = 'logit'))
summary(modelo2)
```

```

Call:
glm(formula = Survived ~ Pclass + Sex + crianca + SibSp + Parch + 
    Embarked, family = binomial(link = "logit"))

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   3.0380     0.3330   9.123  < 2e-16 ***
Pclass2      -0.9312     0.2960  -3.146  0.00165 **
Pclass3      -2.0378     0.2675  -7.619 2.56e-14 ***
Sexmale      -2.8545     0.2320 -12.302  < 2e-16 ***
crianca       2.6753     0.5655   4.730 2.24e-06 ***
```

```
SibSp        -0.2815     0.1271  -2.215  0.02674 *
Parch        -0.2505     0.1318  -1.901  0.05736 .
EmbarkedQ    -0.0604     0.4214  -0.143  0.88604
EmbarkedS    -0.5663     0.2649  -2.138  0.03250 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 931.49  on 696  degrees of freedom
Residual deviance: 612.65  on 688  degrees of freedom
  (2 observations deleted due to missingness)
AIC: 630.65

Number of Fisher Scoring iterations: 5
```

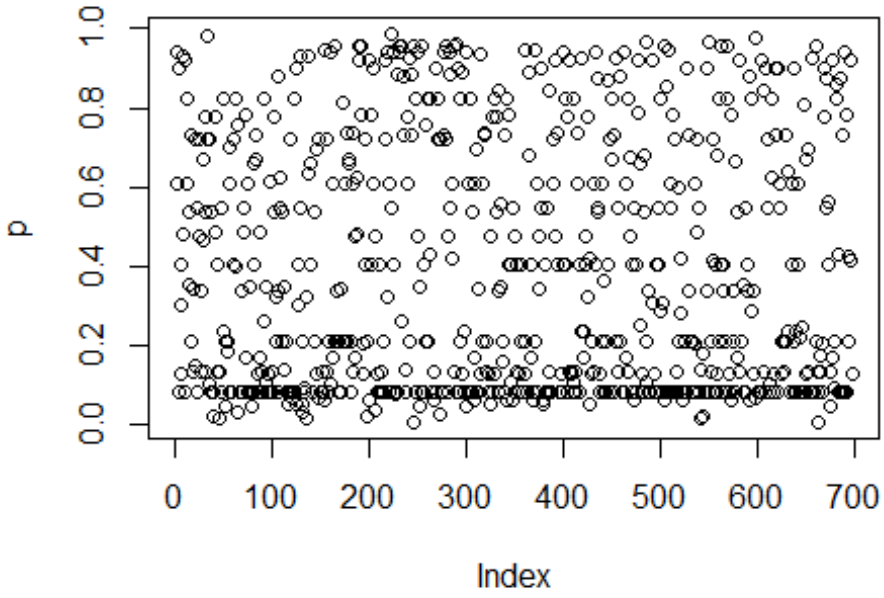
```
eta = predict(modelo2)
odds = exp(eta)
plot(odds)+ abline(h=1, col='blue', lwd = 2)
```




```
integer(0)
```

Achando a Probabilidade

```
eta = predict(modelo2)
odds = exp(eta)
p = 1/(1+exp(-eta))
plot(p)
```



Primeiro tratar os dados

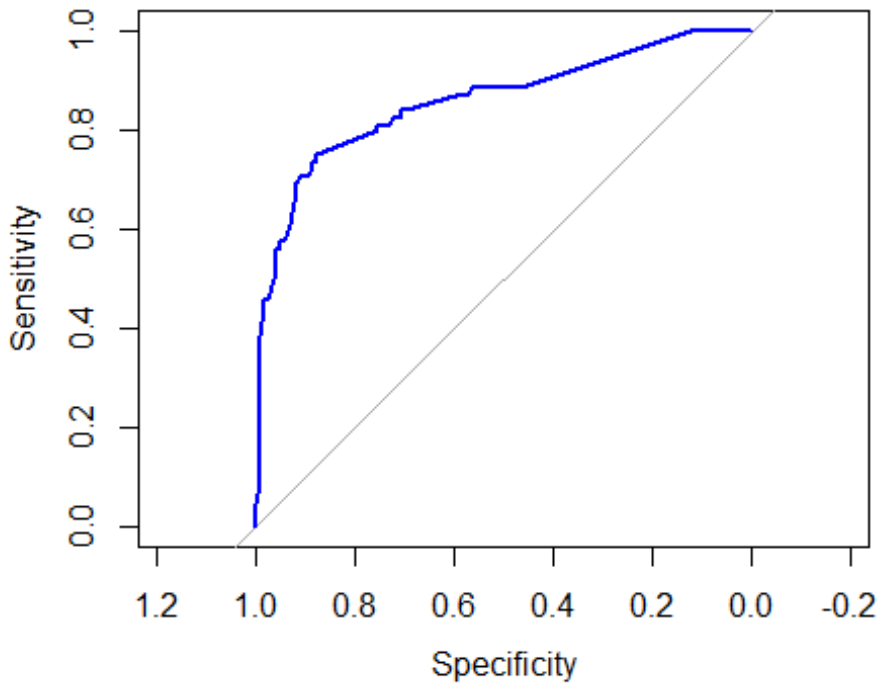
```
# Primeiro tratar os dados
valid.db$Age_No_NA <- ifelse(is.na(valid.db$Age),mean(valid.db$Age, na.rm
= T),valid.db$Age)
valid.db$crianca <- ifelse(valid.db$Age_No_NA < 6,1,0)
valid.db_sem_na <- valid.db
valid.db_sem_na <- valid.db_sem_na[-is.na(valid.db$Age),]
```

Calculando as probabilidades nos dados de validação

```
probabilidade =  1/(1+ exp(-predict(object = modelo2, newdata =
valid.db_sem_na)))
```

Curva Roc

```
curva = roc(valid.db_sem_na$Survived ~ probabilidade)

Setting levels: control = 0, case = 1

Setting direction: controls < cases

plot(curva, col = 'blue')
```

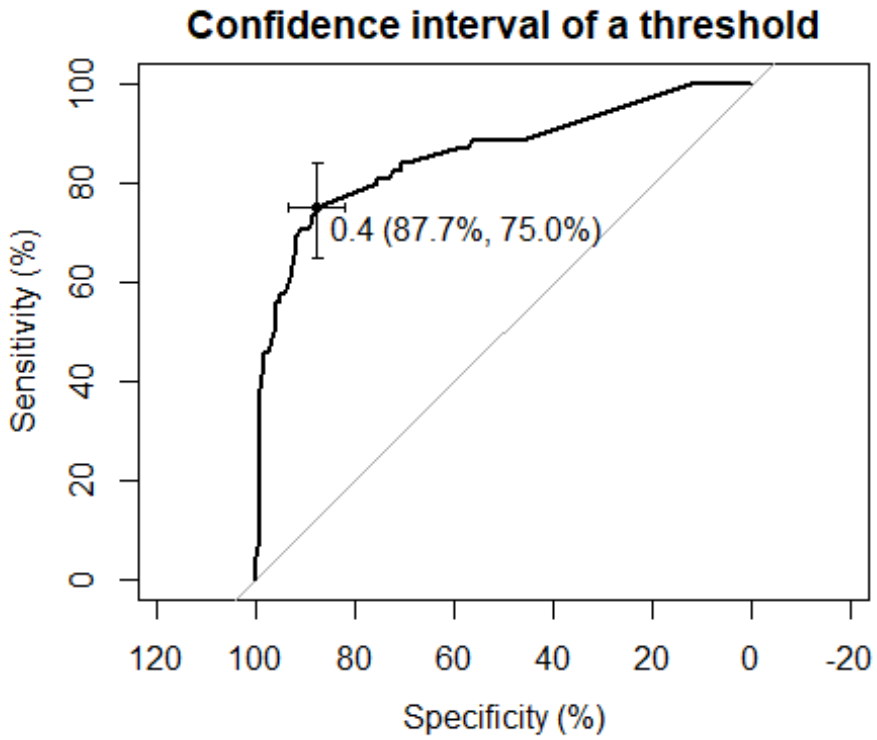


Escolher um critério de classificação

```
plot.roc(valid.db_sem_na$Survived , probabilidade,
         main="Confidence interval of a threshold", percent=TRUE,
         ci=TRUE, of="thresholds", # compute AUC (of threshold)
         thresholds="best", # select the (best) threshold
         print.thres="best") # also highlight this threshold on the plot
```

```
Setting levels: control = 0, case = 1
```

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



Fazendo Previsão

```
# Calculando para a base teste do kaggle

teste_kaggle <- read_csv("test.csv")

Rows: 418 Columns: 11
── Column specification
────────────────────────────────────────────────────────
Delimiter: ","
chr (5): Name, Sex, Ticket, Cabin, Embarked
dbl (6): PassengerId, Pclass, Age, SibSp, Parch, Fare

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.
```

Primeiro tratar os dados

```
teste_kaggle$Age_No_NA <-
ifelse(is.na(teste_kaggle$Age),mean(teste_kaggle$Age, na.rm =
T),teste_kaggle$Age)
teste_kaggle$crianca <- ifelse(teste_kaggle$Age_No_NA < 6,1,0)
teste_kaggle_sem_na <- teste_kaggle
#teste_kaggle_sem_na <- teste_kaggle_sem_na[-
is.na(teste_kaggle$Embarked),]

teste_kaggle_sem_na$Pclass <- as.factor(teste_kaggle_sem_na$Pclass)
```

```r
prob_prevista2 =  1/(1+ exp(-predict(object = modelo2, newdata =
teste_kaggle_sem_na)))

teste_kaggle_sem_na$prob_prevista2 <- prob_prevista2

# tinha esquecido de mudar aq para 0.4
# mudei e piorou :(
previsao <- ifelse(teste_kaggle_sem_na$prob_prevista2 > 0.4, 1, 0)

teste_kaggle_sem_na$previsao <- previsao

submissao_reg_log_full <- data.frame(
  PassengerId = teste_kaggle_sem_na$PassengerId,
  Survived = teste_kaggle_sem_na$previsao
)
head(submissao_reg_log_full, 10)
```

```
   PassengerId Survived
1          892        0
2          893        1
3          894        0
4          895        0
5          896        1
6          897        0
7          898        1
8          899        0
9          900        1
10         901        0
```

```r
write_csv(submissao_reg_log_full, 'submissao_reg_log_full.csv')

probabilidade1 =  1/(1+ exp(-predict(object = modelo1, newdata =
valid.db)))
curva1 = roc(valid.db$Survived ~ probabilidade1)
```

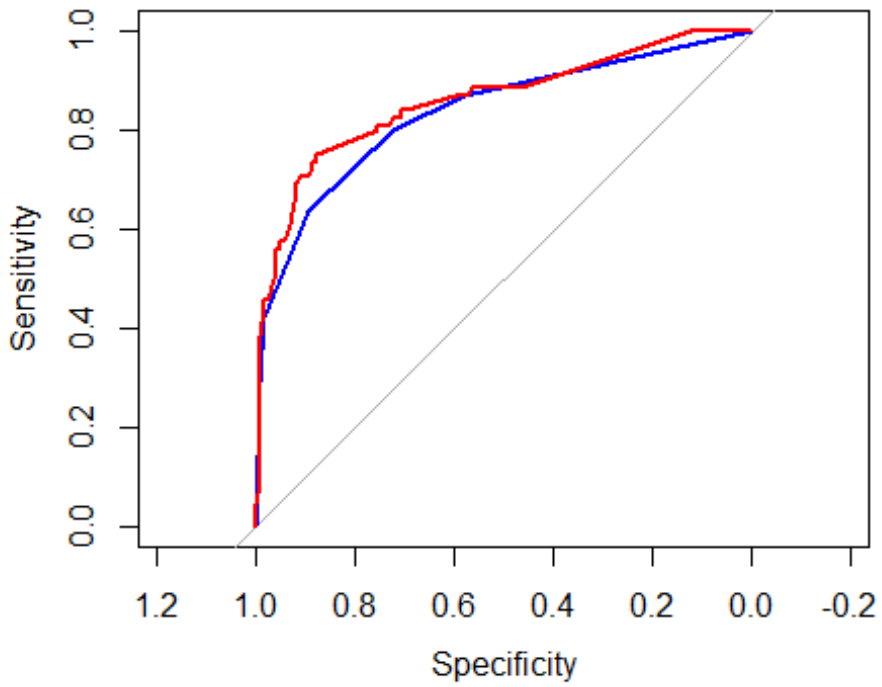
```
Setting levels: control = 0, case = 1
```

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

```r
curva2 = roc(valid.db_sem_na$Survived ~ probabilidade)
```

```
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```

```r
plot(curva1, col = 'blue')
lines(curva2, col = 'red')
```



Vemos que o ajuste com mais variáveis foi ligeiramente melhor.