

# ML Desastre do Titanic

Carol

19/02/2023

## Contents

Introduction	1
Pacotes e base	1
Primeiras impressões	2
Valores faltantes	3
Visualizações	6

## Introduction

Estudo de caso com a base famosinha Titanic: Machine Learning from Disaster

Análise descritiva, imputação de valores, modelo preditivo

## Pacotes e base

```
library(dplyr) # Data Munging
library(ggplot2) # Data Visualization
library(ggthemes) #Themes
library(corrplot) #Correlation
library(rpart) #Data modelling
library(randomForest) #Data modelling
library(pscl)
#library(Deducer) # Modelling
library(Amelia) #Missing values
library(forcats)
library(rpart.plot) # Random forest
library(Hmisc)
library(VIM)
train=read.csv("C:/Users/cselis/OneDrive - SEBRAE/Área de Trabalho/Pessoal/Scripts/train.csv",header=TRUE)
test=read.csv("C:/Users/cselis/OneDrive - SEBRAE/Área de Trabalho/Pessoal/Scripts/test.csv",header=TRUE)
```

# Primeiras impressões

##Dimensões

```
dim(train)
```

```
## [1] 891 12
```

```
dim(test)
```

```
## [1] 418 11
```

##Resumo (Summary)

```
summary(train) #Age:177 NA
```

```
## 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
##
```

```
summary(test) #Age:86NA
```

```
## PassengerId      Pclass      Name      Sex
## Min.   : 892.0    Min.   :1.000  Length:418  Length:418
## 1st Qu.: 996.2    1st Qu.:1.000  Class :character  Class :character
## Median :1100.5    Median :3.000  Mode  :character  Mode  :character
## Mean   :1100.5    Mean   :2.266
## 3rd Qu.:1204.8    3rd Qu.:3.000
## Max.   :1309.0    Max.   :3.000
##
```

```
##      Age      SibSp      Parch      Ticket
## Min.   : 0.17   Min.   :0.0000   Min.   :0.0000   Length:418
## 1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.0000   Class :character
## Median :27.00   Median :0.0000   Median :0.0000   Mode  :character
## Mean   :30.27   Mean   :0.4474   Mean   :0.3923
## 3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.0000
## Max.   :76.00   Max.   :8.0000   Max.   :9.0000
## NA's    :86
##      Fare      Cabin      Embarked
## Min.   : 0.000   Length:418   Length:418
## 1st Qu.: 7.896   Class :character   Class :character
## Median :14.454   Mode  :character   Mode  :character
## Mean   :35.627
## 3rd Qu.:31.500
## Max.   :512.329
## NA's    :1
```

Analisando os dados faltantes, observamos que a idade é a característica com mais registros incompletos no conjunto de treinamento. No conjunto de teste, além da idade, há um registro faltante na tarifa. Para aprofundar nossa análise, vamos combinar os dois conjuntos e realizar uma EDA.

```
titanic=full_join(train,test)
summary(titanic)
```

```
## PassengerId      Survived      Pclass      Name
## Min.   : 1      Min.   :0.0000   Min.   :1.000   Length:1309
## 1st Qu.:328      1st Qu.:0.0000   1st Qu.:2.000   Class :character
## Median :655      Median :0.0000   Median :3.000   Mode  :character
## Mean   :655      Mean   :0.3838   Mean   :2.295
## 3rd Qu.:982      3rd Qu.:1.0000   3rd Qu.:3.000
## Max.   :1309     Max.   :1.0000   Max.   :3.000
## NA's    :418
##      Sex      Age      SibSp      Parch
## Length:1309   Min.   : 0.17   Min.   :0.0000   Min.   :0.000
## Class :character 1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.000
## Mode  :character Median :28.00   Median :0.0000   Median :0.000
## Mean   :29.88   Mean   :0.4989   Mean   :0.385
## 3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.000
## Max.   :80.00   Max.   :8.0000   Max.   :9.000
## NA's    :263
##      Ticket      Fare      Cabin      Embarked
## Length:1309     Min.   : 0.000   Length:1309   Length:1309
## Class :character 1st Qu.: 7.896   Class :character   Class :character
## Mode  :character Median :14.454   Mode  :character   Mode  :character
## Mean   :33.295
## 3rd Qu.:31.275
## Max.   :512.329
## NA's    :1
```

## Valores faltantes

Vamos primeiro focar na imputação dos valores faltantes e fazer uma exploração visual depois disso.

##Age

```
prop.table(table(is.na(titanic$Age)))
```

```
##  
##      FALSE      TRUE  
## 0.7990833 0.2009167
```

Como 20% dos dados da idade estão ausentes, utilizaremos a técnica de partição recursiva (rpart) para prever esses valores.

```
age=rpart(Age ~Pclass+Sex+SibSp+Parch+Fare+Embarked,data=titanic[!(is.na(titanic$Age)),],method="anova",  
titanic$Age[is.na(titanic$Age)]=predict(age,titanic[is.na(titanic$Age),])
```

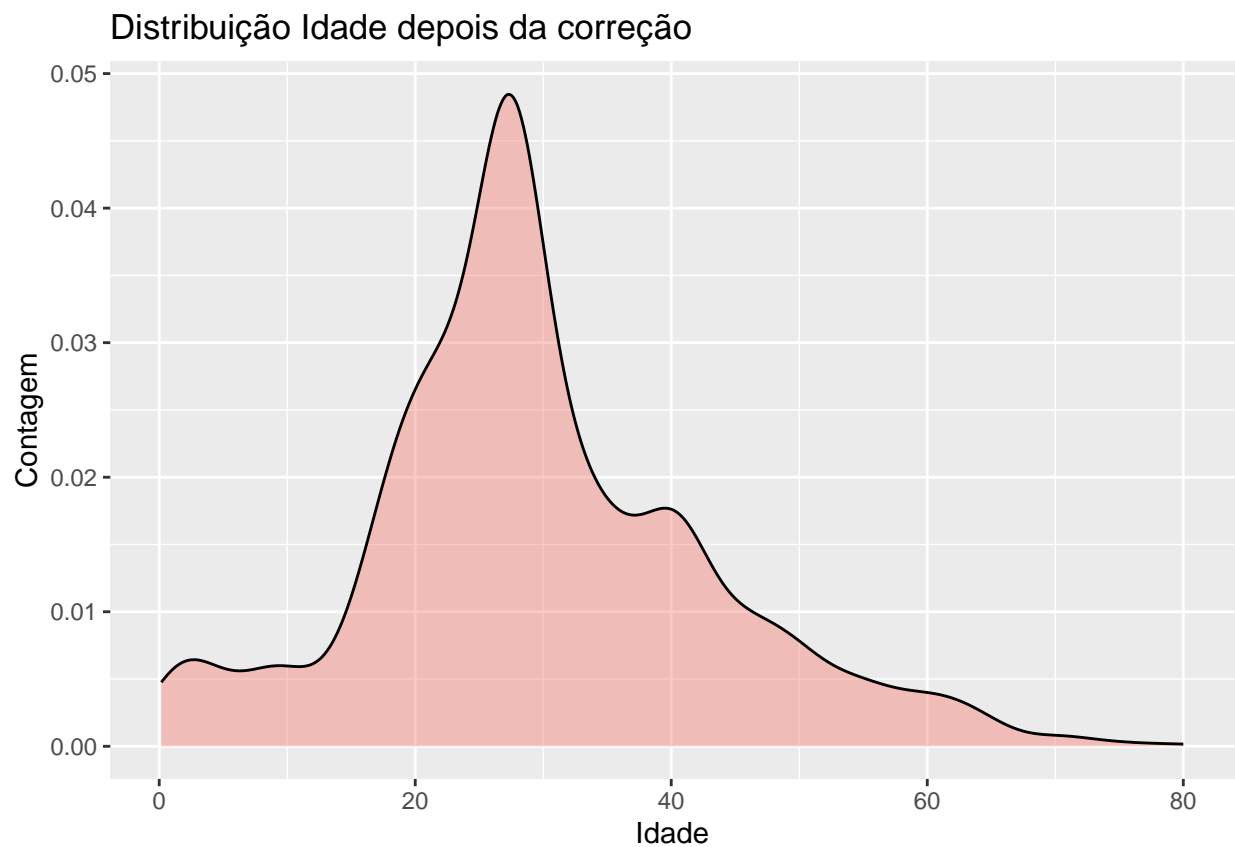
Verificar

```
prop.table(table(is.na(titanic$Age)))
```

```
##  
## FALSE  
##      1
```

A coluna foi corrigida com sucesso :)

```
ggplot(titanic,aes(Age,fill="green"))+geom_density(alpha=0.4)+labs(x="Idade",y="Contagem",title="Distribuição Idade depois da correção")
```



##Taxa de embarque:

```
sum(is.na(titanic$Fare))
```

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

Uma informação faltante, qual?

```
which(is.na(titanic$Fare))
```

```
## [1] 1044
```

```
titanic[1044,]
```

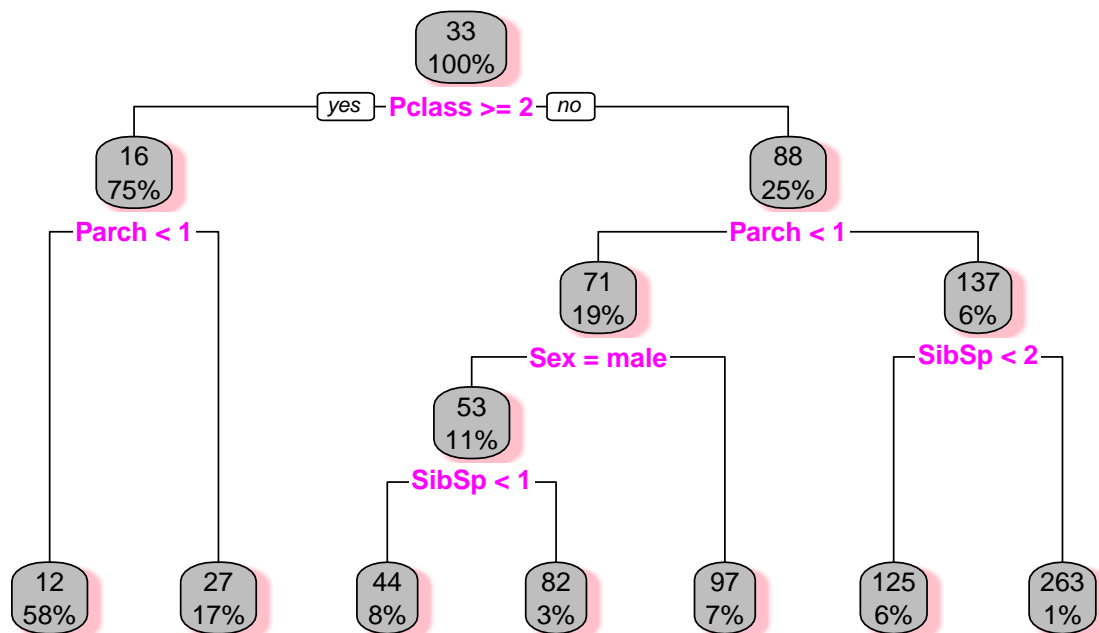
```
##      PassengerId Survived Pclass      Name Sex  Age SibSp Parch
## 1044          1044       NA      3 Storey, Mr. Thomas male 60.5    0    0
##      Ticket Fare Cabin Embarked
## 1044   3701   NA      S
```

O passageiro é do sexo masculino, pertence à 3ª classe e embarcou no S.

Utilizaremos a função rpart novamente para imputação.

```
fare=rpart(Fare ~Parch+SibSp+Sex+Pclass,data=titanic[!(is.na(titanic$Fare)),],method="anova")
titanic$Fare[(is.na(titanic$Fare))]=predict(fare,data=titanic[is.na(titanic$Fare),])
rpart.plot(fare,shadow.col="pink",box.col="gray",split.col="magenta",main="Árvore de Decisão para imputar")
```

## Árvore de Decisão para imputar



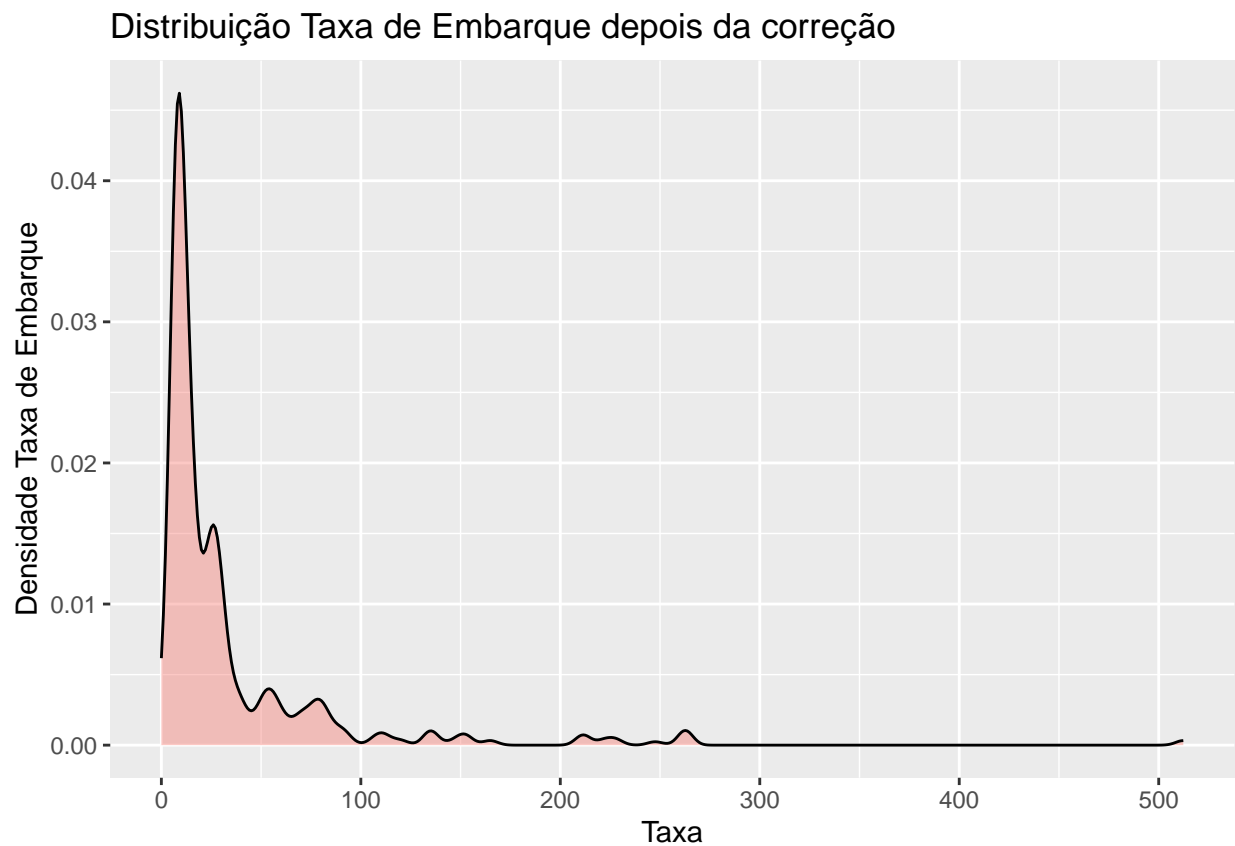
De acordo com a árvore de decisão, os passageiros da 2ª ou 3ª classe pagaram menos do que os da 1ª classe, enquanto aqueles que viajavam com pais ou filhos pagaram mais do que os que estavam sozinhos.

```
prop.table(table(is.na(titanic$Fare)))
```

```
##  
## FALSE  
##      1
```

Gráfico de Densidade

```
ggplot(titanic,aes(Fare,fill="green"))+geom_density(alpha=0.4)+labs(x="Taxa",y="Densidade Taxa de Embarque")
```



Dados estão fortemente deslocados para a direita

## Visualizações

Agora, vamos nos concentrar em nome, sexo, SibSp, Parch e Pclass para realizar algumas tarefas de organização e visualização de dados. Começaremos com o nome.

```
str(titanic$Name)
```

```
## chr [1:1309] "Braund, Mr. Owen Harris" ...
```

Tirar título dos nomes

```
titanic$Title=gsub('(.*, )|(\\.*)', '',titanic$Name)
head(titanic$Title)
```

```
## [1] "Mr"    "Mrs"   "Miss"  "Mrs"   "Mr"    "Mr"
```

```
table(titanic$Title,titanic$Sex)
```

```
##
##           female male
## Capt           0    1
## Col            0    4
## Don            0    1
## Dona           1    0
## Dr             1    7
## Jonkheer       0    1
## Lady           1    0
## Major          0    2
## Master         0   61
## Miss          260    0
## Mlle           2    0
## Mme            1    0
## Mr             0  757
## Mrs           197    0
## Ms             2    0
## Rev            0    8
## Sir            0    1
## the Countess   1    0
```

Convertemos a variável em fator e, usando a função da biblioteca forcats, colapsamos alguns desses níveis.

```
titanic <- titanic %>% mutate(Title = factor(Title)) %>% mutate(Title = fct_collapse(Title, "Miss" = c(
str(titanic$Title)
```

```
## Factor w/ 6 levels "Ranked","Royalty",...: 6 5 4 5 6 6 6 3 5 5 ...
```

Criaremos uma nova coluna chamada Família para identificar quem viajou sozinho e quem viajou com a família. Isso será feito usando uma declaração ifelse simples. A condição será verdadeira se a pessoa tiver viajado com pais/filhos ou irmãos/cônjuge, e falsa caso contrário.

```
titanic$Families= factor(ifelse(titanic$SibSp + titanic$Parch + 1> 1,"Yes","No"))
prop.table(table(titanic$Families))
```

```
##
##           No           Yes
## 0.6035141 0.3964859
```

Quase 40% das pessoas viajaram com famílias.

Vamos agora examinar a classe do passageiro.”

```
prop.table(table(titanic$Pclass))
```

```
##  
##           1           2           3  
## 0.2467532 0.2116119 0.5416348
```

54% deles viajaram na terceira classe, enquanto 25% viajaram na primeira classe

##Sobreviventes

```
titanic=titanic %>% mutate(Survived=factor(Survived)) %>% mutate(Survived=fct_recode(Survived,"No"="0",  
# train=titanic[1:891,]  
# test=titanic[1:1309,]  
prop.table(table(train$Survived))
```

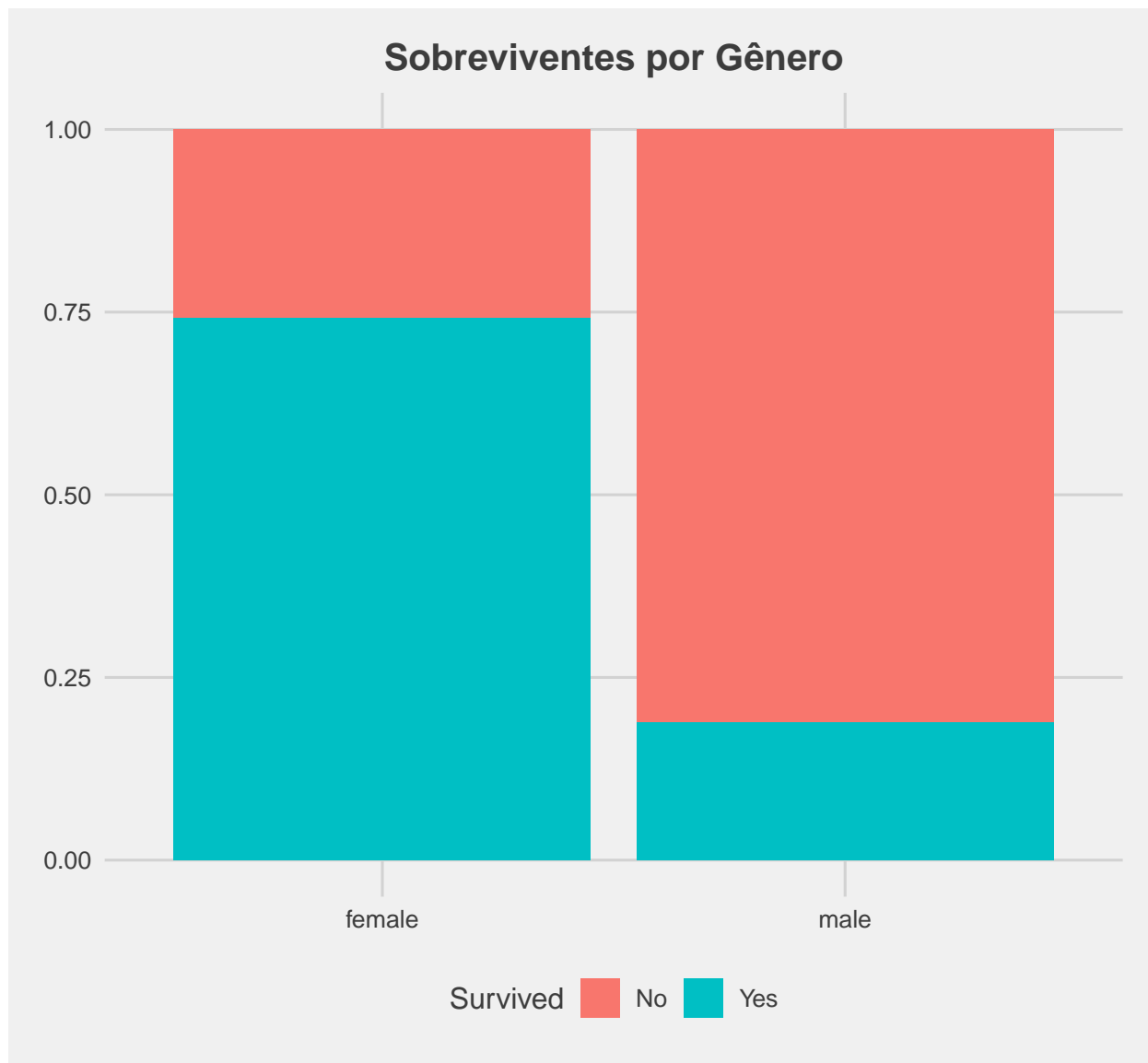
```
##  
##           0           1  
## 0.6161616 0.3838384
```

Nos dados de treino, apenas 38% sobreviveram, enquanto 61% pereceram.

**Por gênero**

```
ggplot(titanic[1:891,],aes(Sex,fill=Survived))+geom_bar(position="fill")+theme_fivethirtyeight()+theme(
```





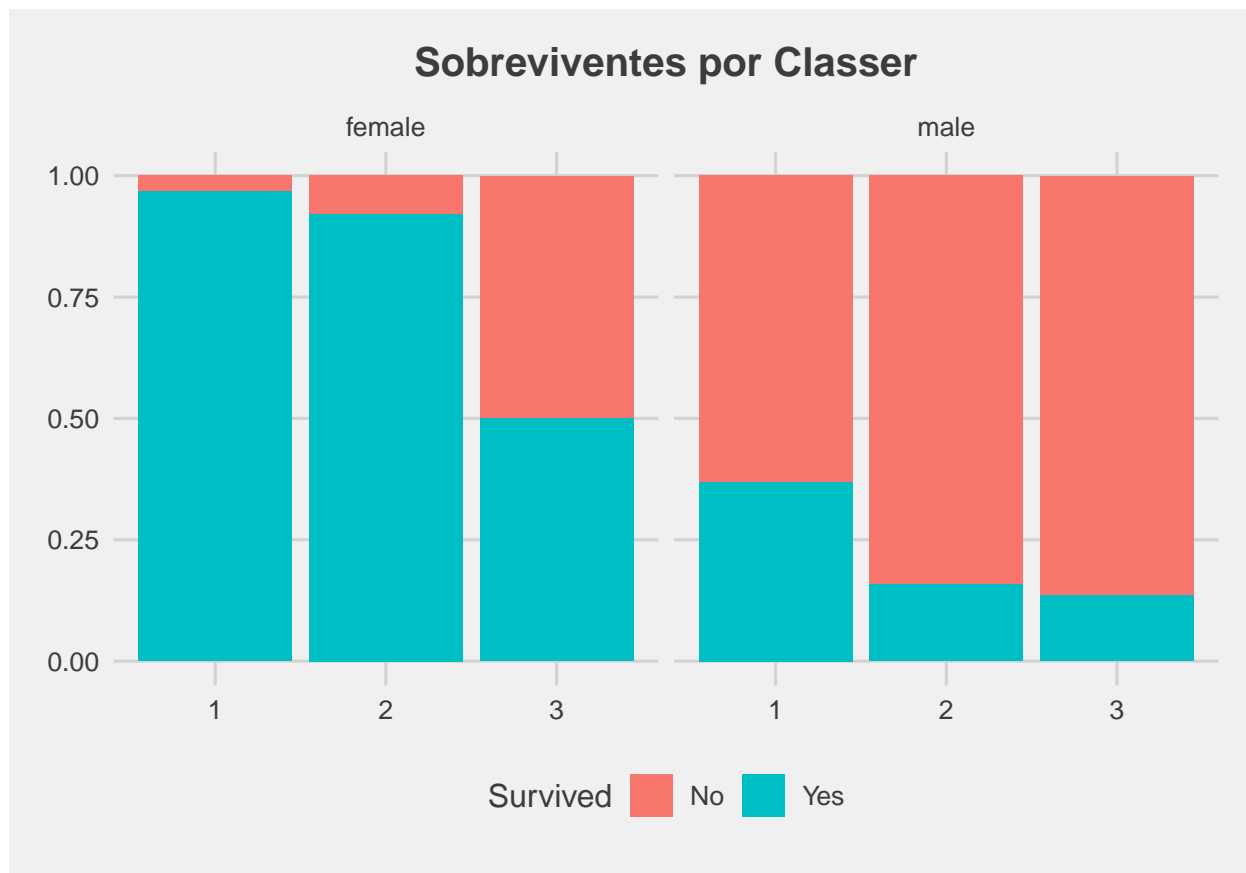
A bordo do Titanic, as chances de sobrevivência eram muito menores para homens do que para mulheres.

**Por classe**

```
str(titanic$Pclass)
```

```
## int [1:1309] 3 1 3 1 3 3 1 3 3 2 ...
```

```
ggplot(titanic[1:891,],aes(Pclass,fill=Survived))+geom_bar(position="fill")+facet_wrap(~Sex)+theme_five
```

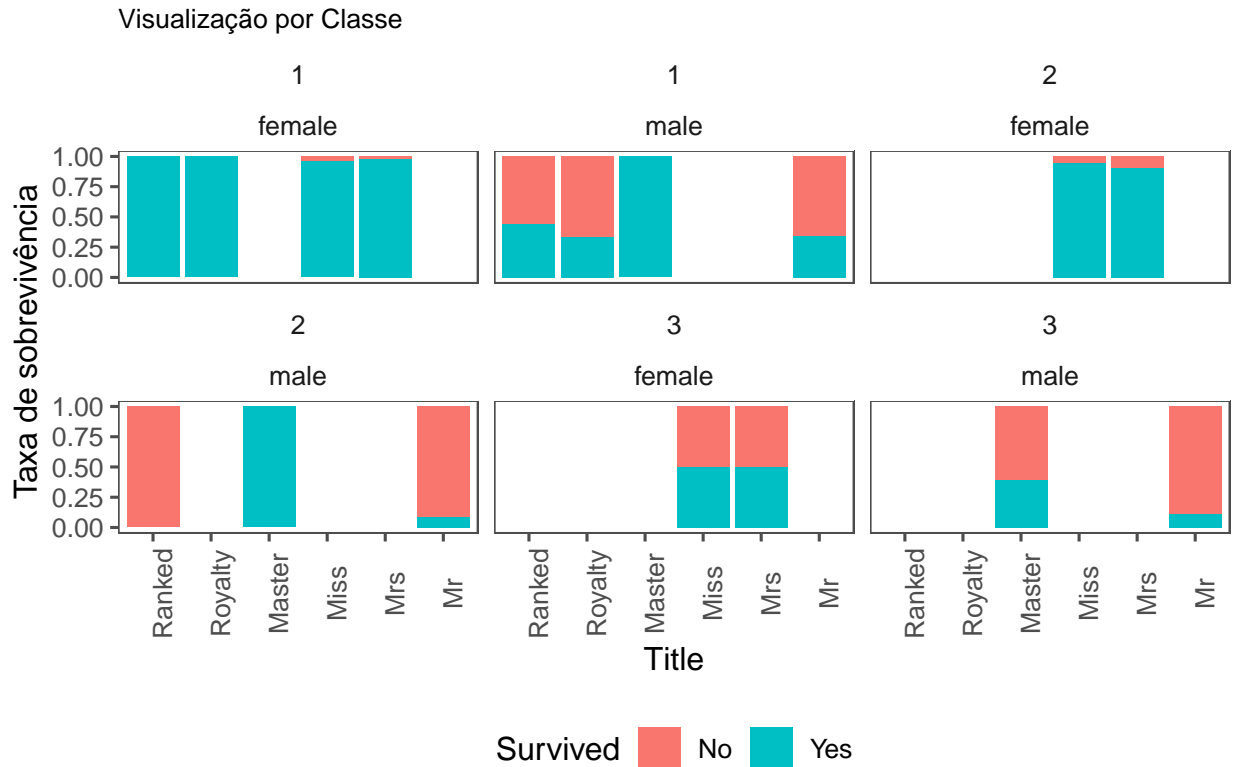


Analisando os dois gráficos, podemos entender que, independentemente do gênero, há uma maior chance de sobrevivência se você fosse da 1ª classe. No entanto, se você for uma passageira feminina da 1ª classe, as chances de sobrevivência aumentam significativamente. Os azarados são os homens da 2ª e 3ª classe.

##Análise por classe

```
ggplot(titanic[1:891,],aes(Title,fill=Survived))+geom_bar(position="fill")+facet_wrap(Pclass~Sex)+theme.
```

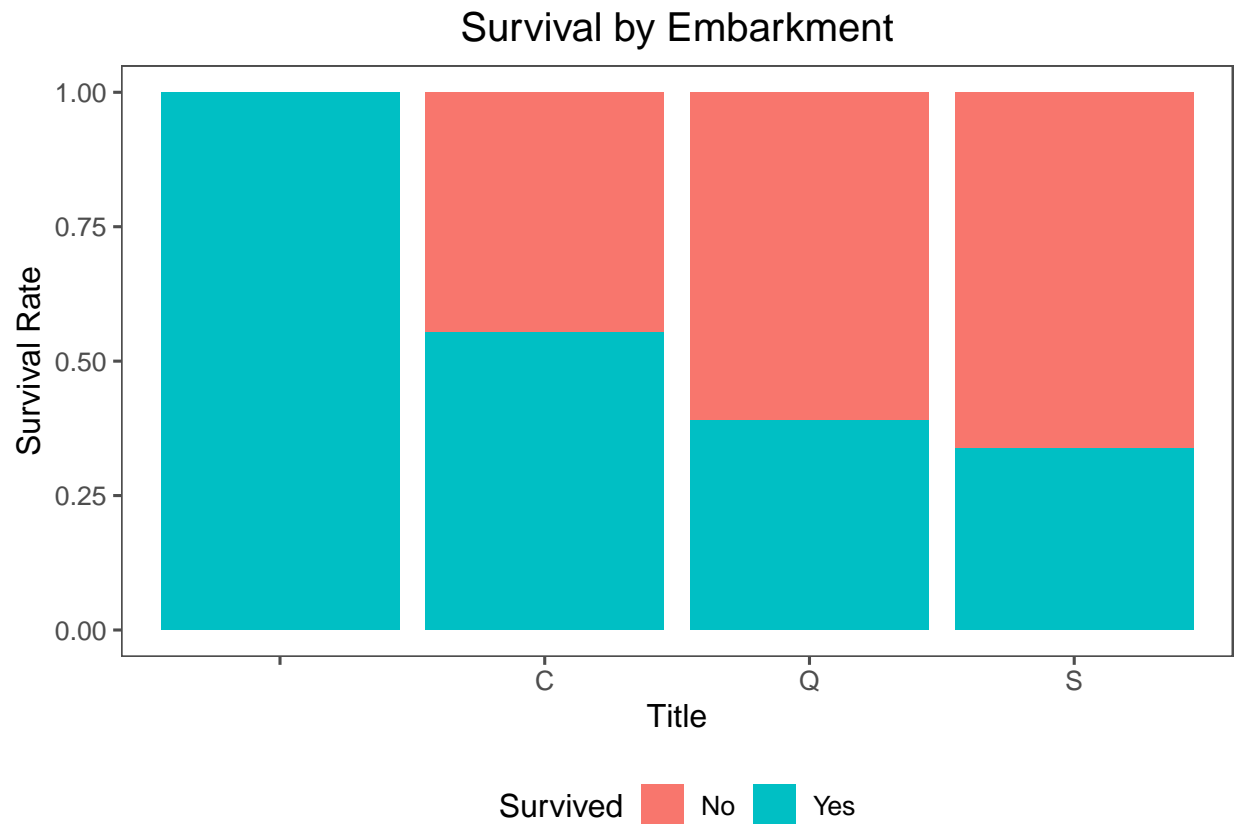
## Sobreviventes por gênero e Título



- From the plot, we understand that the survival rate is highly influenced by Passenger class and gender.
- As seen earlier, the survival rate is usually higher for female which is reinstated here.
- 1st class and 2nd class, female passengers had almost 100 % chance of survival compared to their male counterpart.
- Chances of survival is 50% for female travelling in 3rd class.
- For male, the only way a male could have survived is that he should have been a boy (as indicated by master). For 1st and 2nd class, the survival is almost 100 % whereas it is 50 % in 3rd class.
- The probability of survival is worse for an adult male in 2nd and 3rd class whereas in 1st class it is around 50 %.

###By Embarkment:

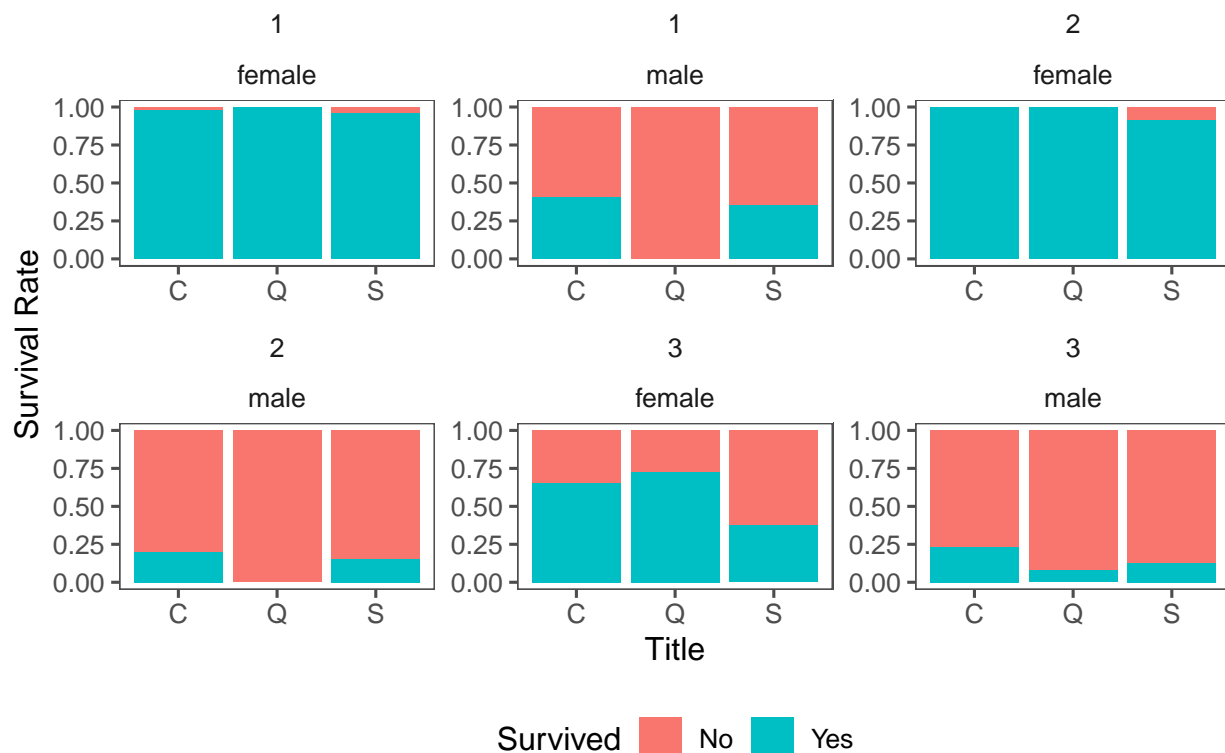
```
ggplot(titanic[1:891,], aes(Embarked, fill=Survived)) + geom_bar(position="fill") + theme_few() + theme(legend.position="right")
```



There seems to be one row with no value for embarkment.let us add this to the majority class.

```
titanic =titanic %>% mutate(Embarked=ifelse(Embarked=="",names(which.max(table(titanic$Embarked))),Embarked)
ggplot(titanic[1:891,],aes(Embarked,fill=Survived))+geom_bar(position="fill")+theme_few()+theme(legend.position="bottom")
```

## Survival by Embarkment



Analisando o gráfico, entendemos que a taxa de sobrevivência é fortemente influenciada pela classe do passageiro e pelo gênero. Como visto anteriormente, a taxa de sobrevivência é geralmente maior para mulheres, o que é reafirmado aqui.

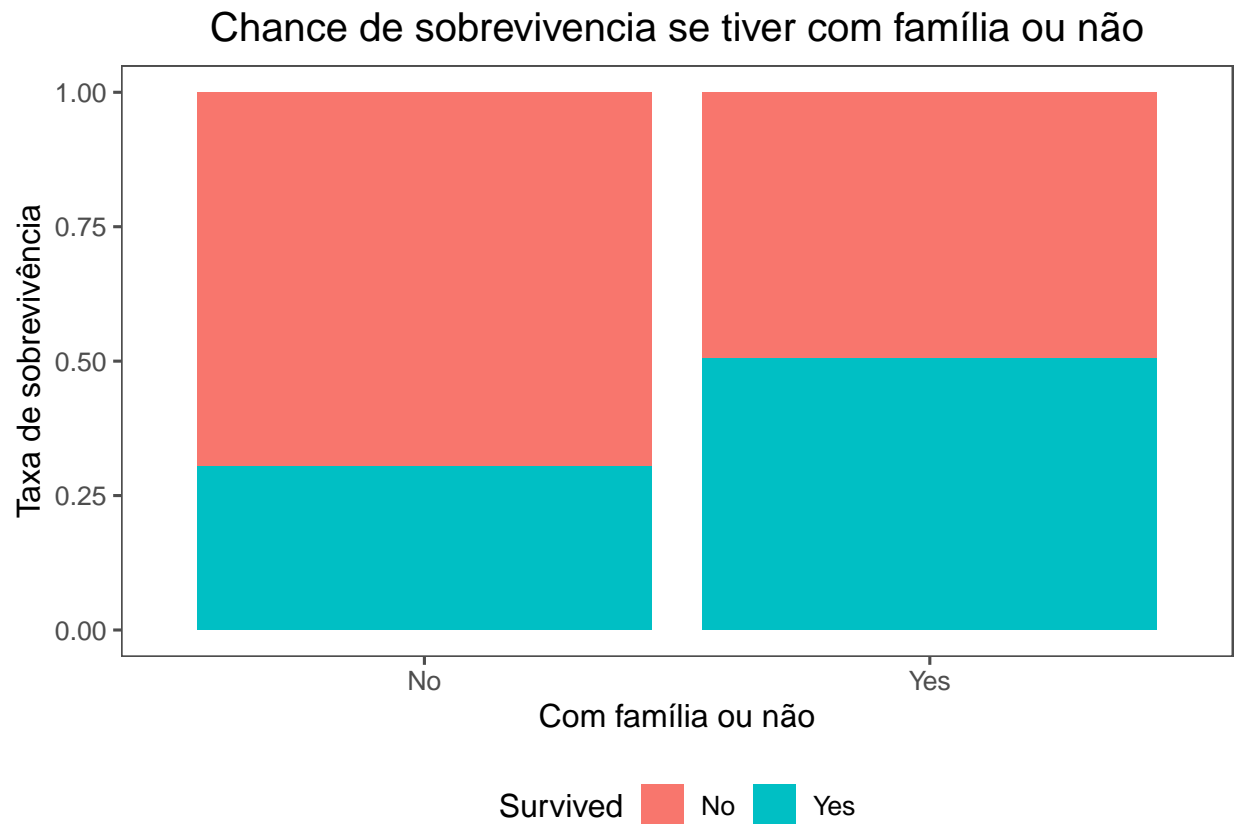
Passageiras da 1ª e 2ª classe tinham quase 100% de chance de sobrevivência em comparação com seus equivalentes masculinos. As chances de sobrevivência são de 50% para mulheres viajando na 3ª classe.

Para homens, a única maneira de ter sobrevivido era ser um menino (como indicado por 'master'). Na 1ª e 2ª classe, a sobrevivência é quase 100%, enquanto na 3ª classe é de 50%.

A probabilidade de sobrevivência é pior para um homem adulto na 2ª e 3ª classe, enquanto na 1ª classe é de cerca de 50%.

###Por Famílias:

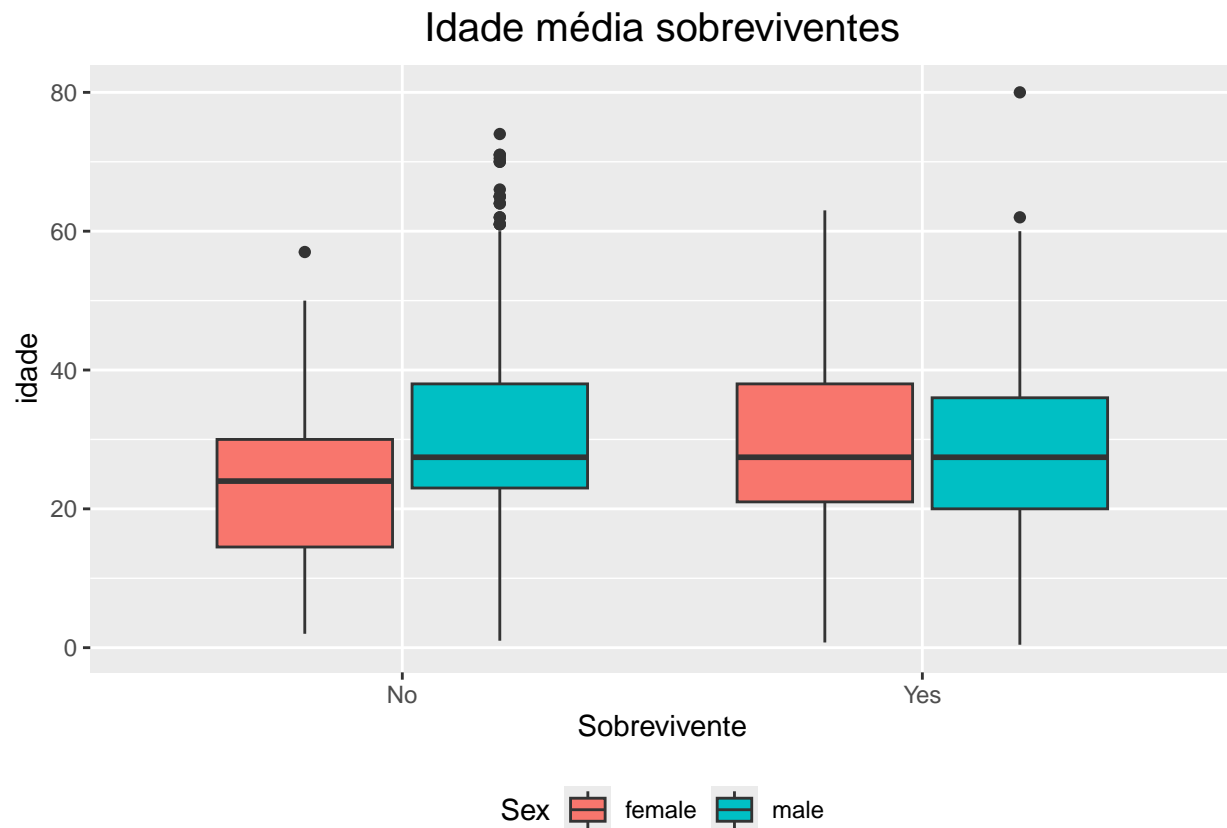
```
ggplot(titanic[1:891,],aes(Families,fill=Survived))+geom_bar(position="fill")+theme_few()+theme(legend.position="right")
```



As chances de sobrevivência parecem ser um pouco maiores para aqueles que viajam com suas famílias.

Vamos criar um boxplot para entender a idade mediana de sobrevivência em relação ao gênero.”

```
ggplot(titanic[1:891,],aes(Survived,Age,fill=Sex))+geom_boxplot()+theme(legend.position="bottom",plot.t
```



A idade mediana tem sido em torno de 30 para sobreviventes de ambos os sexos, enquanto a idade mediana das mulheres que não sobreviveram é de cerca de 25 e para homens é de cerca de 28

### Por cabine

```
str(titanic$Cabin)
```

```
## chr [1:1309] "" "C85" "" "C123" "" "" "E46" "" "" "" "G6" "C103" "" "" "" ...
```

Tentamos dividir o primeiro caractere da variável 'cabine' e visualizar a taxa de sobrevivência.

```
titanic$Deck=factor(sapply(titanic$Cabin, function(x) strsplit(x, NULL)[[1]][1]))
str(titanic$Deck)
```

```
## Factor w/ 8 levels "A","B","C","D",...: NA 3 NA 3 NA NA 5 NA NA NA ...
```

```
table(is.na(titanic$Deck)) #297 missing values
```

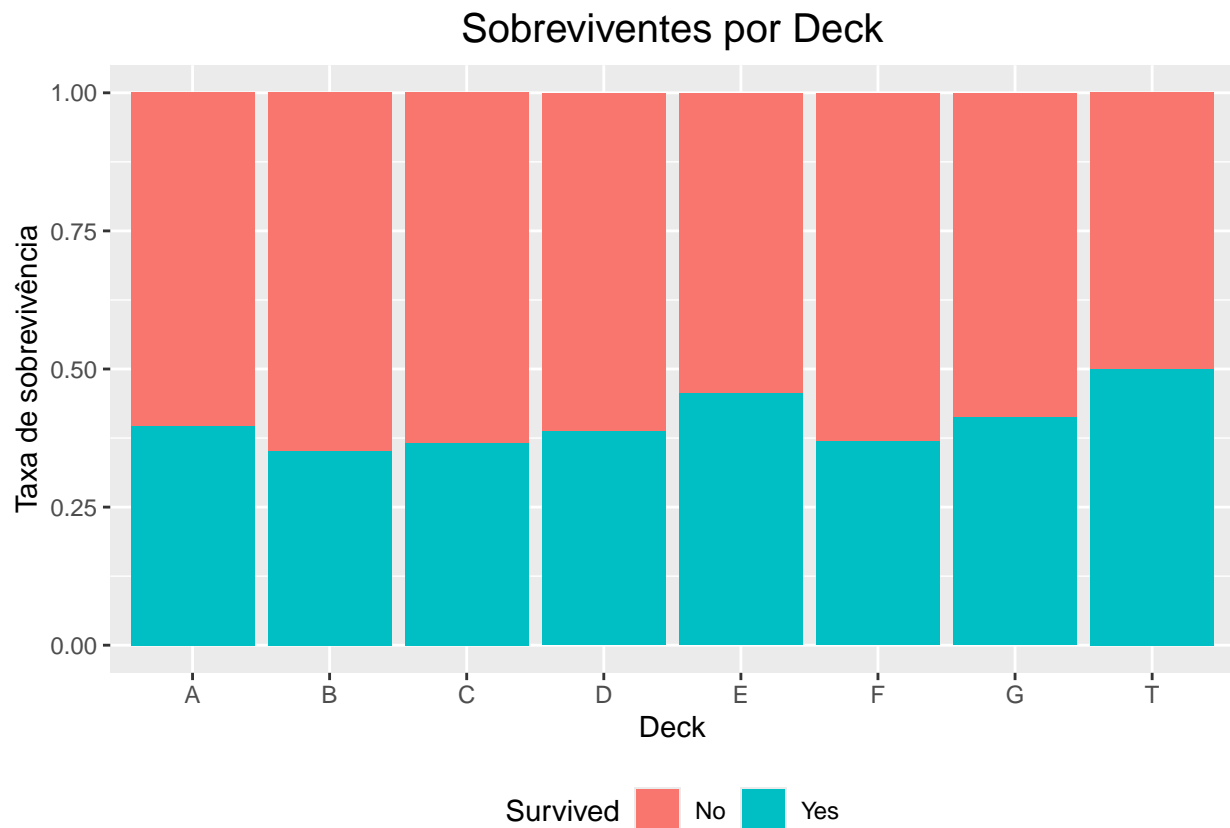
```
##
## FALSE TRUE
## 295 1014
```

```
round(prop.table(table(titanic$Deck,titanic$Survived))*100,2)
```

```
##
##      No   Yes
## A  3.92  3.43
## B  5.88 17.16
## C 11.76 17.16
## D  3.92 12.25
## E  3.92 11.76
## F  2.45  3.92
## G  0.98  0.98
## T  0.49  0.00
```

Os decks C e B apresentam taxas de sobrevivência mais altas, enquanto os decks F, G e T têm taxas mais baixas.

```
set.seed(100)
titanic$Deck=with(titanic,impute(Deck,'random'))
ggplot(titanic[1:891,],aes(Deck,fill=Survived))+geom_bar(position="fill")+theme(legend.position="bottom")
```

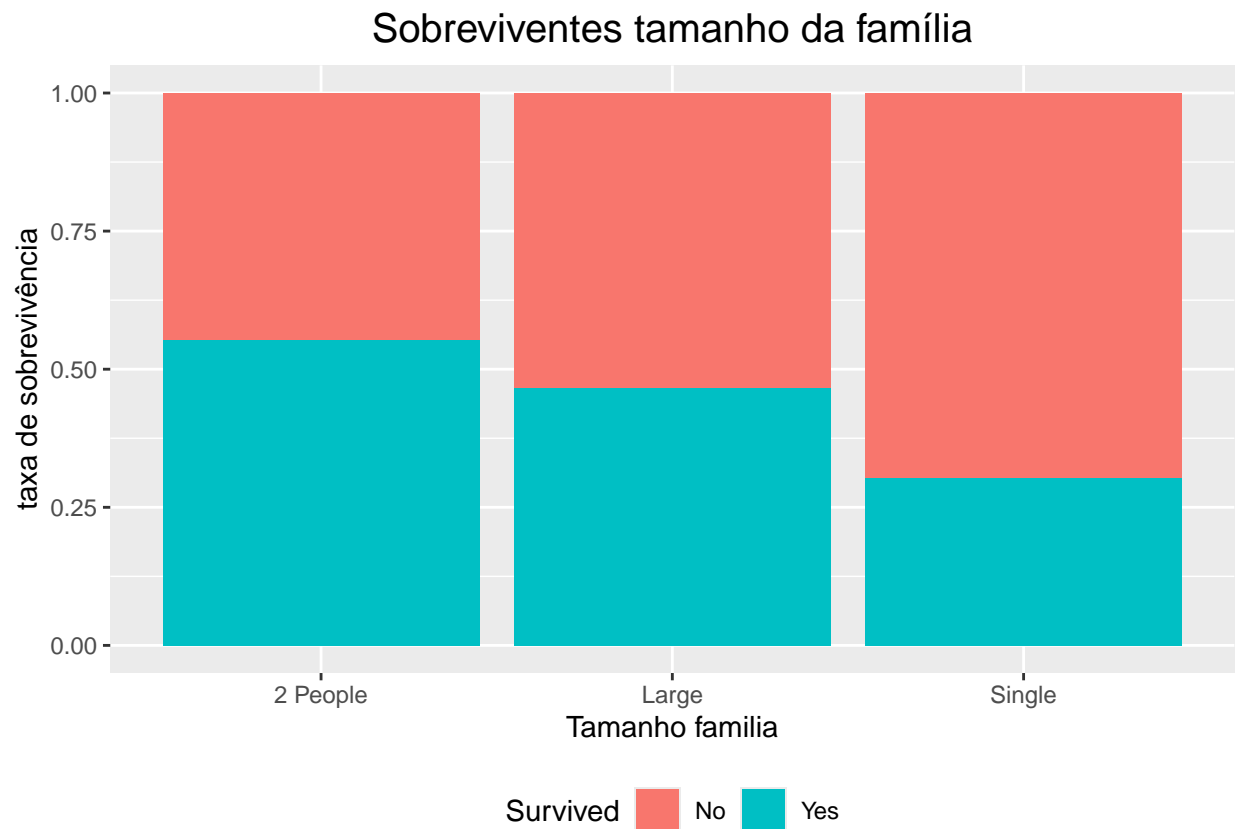


### Tamanho da Família:

Chances de sobrevivência são maiores se você estiver com a família?



```
titanic=titanic %>% mutate(FamilySize=SibSp+Parch+1) %>% mutate(Type=ifelse(FamilySize==1,"Single",ifelse(FamilySize==2,"2 People",ifelse(FamilySize>2,"Large",))))
ggplot(titanic[1:891,],aes(Type,fill=Survived))+geom_bar(position="fill")+theme(legend.position="bottom")
```

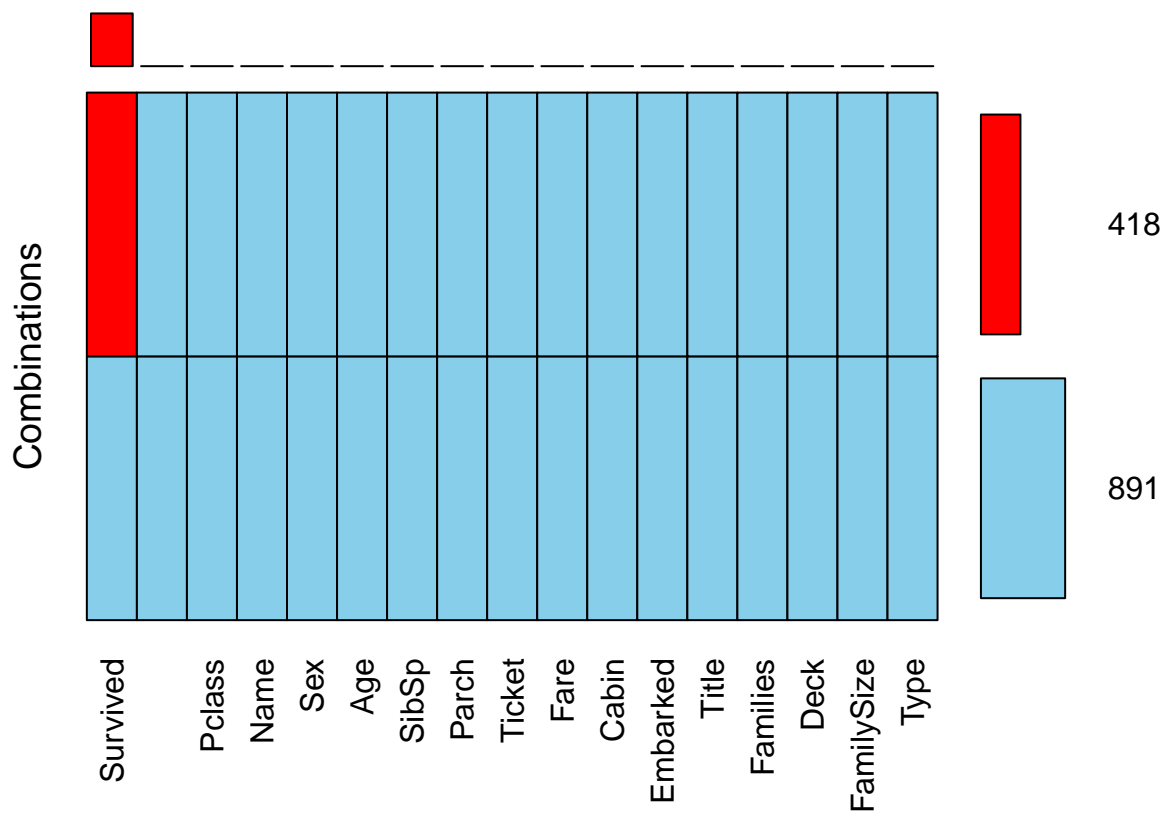


A taxa de sobrevivência é alta para famílias de 2 pessoas, enquanto é baixa para solteiros.

Vamos primeiro visualizar se os valores faltantes relevantes foram imputados corretamente.

### Modelo random forest

```
aggr(titanic,prop=FALSE,combined=TRUE,sortVars=TRUE,sortCombs=TRUE,numbers=TRUE)
```



```
##
## Variables sorted by number of missings:
## Variable Count
## Survived 418
## PassengerId 0
## Pclass 0
## Name 0
## Sex 0
## Age 0
## SibSp 0
## Parch 0
## Ticket 0
## Fare 0
## Cabin 0
## Embarked 0
## Title 0
## Families 0
## Deck 0
## FamilySize 0
## Type 0
```

A partir dos dados, entendemos que não há valores faltantes nos dados de treinamento. Os dados de teste têm 418 valores faltantes em 'Sobrevivente', que precisam ser previstos. Vamos dividir os conjuntos de dados de treinamento e teste. Convertamos as variáveis de caracteres em fatores.

```
titanic = titanic %>% mutate(Type=factor(Type)) %>% mutate(Embarked=factor(Embarked)) %>% mutate(Sex=factor(Sex))
```

```
train=titanic[1:891,]
test=titanic[892:1309,]
names(train)
```

```
## [1] "PassengerId" "Survived" "Pclass" "Name" "Sex"
## [6] "Age" "SibSp" "Parch" "Ticket" "Fare"
## [11] "Cabin" "Embarked" "Title" "Families" "Deck"
## [16] "FamilySize" "Type"
```

```
str(train)
```

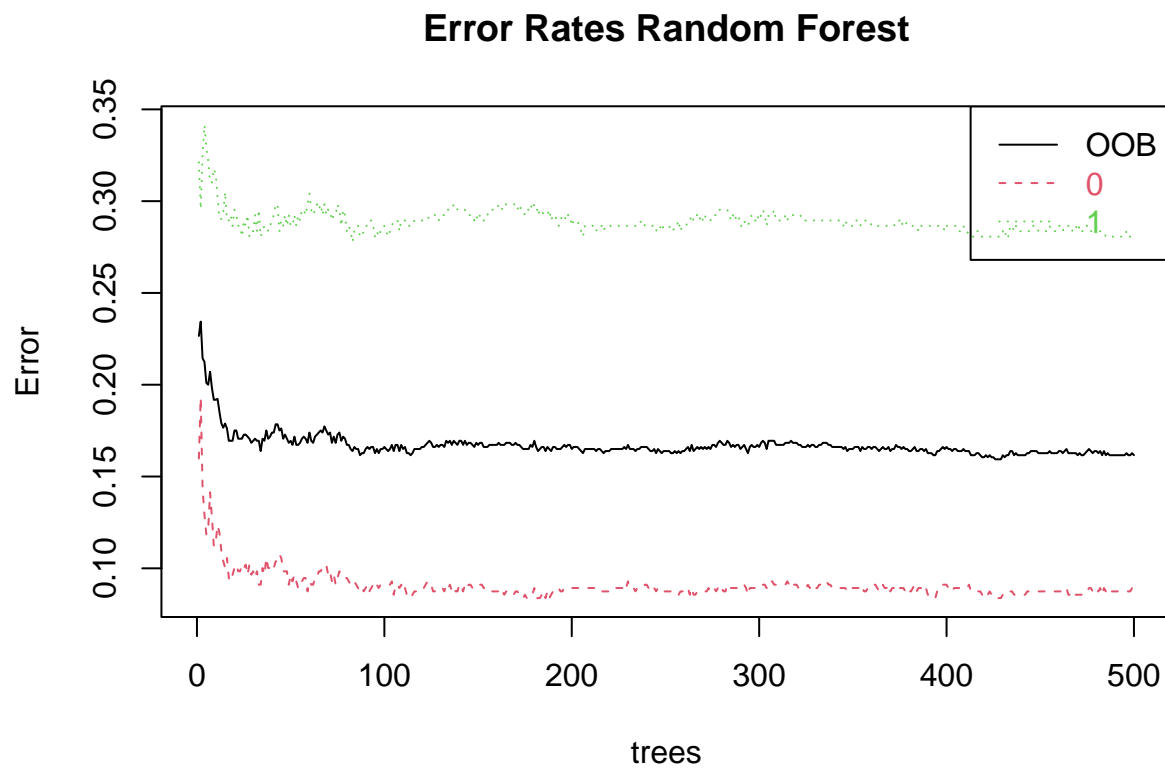
```
## 'data.frame': 891 obs. of 17 variables:
## $ PassengerId: int 1 2 3 4 5 6 7 8 9 10 ...
## $ Survived : Factor w/ 2 levels "No","Yes": 1 2 2 2 1 1 1 1 2 2 ...
## $ Pclass : int 3 1 3 1 3 1 3 1 3 3 2 ...
## $ Name : chr "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
## $ Sex : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
## $ Age : num 22 38 26 35 35 ...
## $ SibSp : int 1 1 0 1 0 0 0 3 0 1 ...
## $ Parch : int 0 0 0 0 0 0 0 1 2 0 ...
## $ Ticket : chr "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
## $ Fare : num 7.25 71.28 7.92 53.1 8.05 ...
## $ Cabin : chr "" "C85" "" "C123" ...
## $ Embarked : Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
## $ Title : Factor w/ 6 levels "Ranked","Royalty",...: 6 5 4 5 6 6 6 3 5 5 ...
## $ Families : Factor w/ 2 levels "No","Yes": 2 2 1 2 1 1 1 2 2 2 ...
## $ Deck : Factor w/ 8 levels "A","B","C","D",...: 3 3 4 3 5 7 5 1 1 4 ...
## ..- attr(*, "imputed")= int [1:687] 1 3 5 6 8 9 10 13 14 15 ...
## $ FamilySize : num 2 2 1 2 1 1 1 5 3 2 ...
## $ Type : Factor w/ 3 levels "2 People","Large",...: 1 1 3 1 3 3 3 2 2 1 ...
```

```
rfmodel=randomForest(factor(Survived) ~ Pclass+Sex+Age+Fare+Embarked+Title+Deck+FamilySize+Type+SibSp+Parch, data=train)
print(rfmodel)
```

```
##
## Call:
## randomForest(formula = factor(Survived) ~ Pclass + Sex + Age + Fare + Embarked + Title + Deck + FamilySize + Type + SibSp + Parch, data = train)
## Type of random forest: classification
## Number of trees: 500
## No. of variables tried at each split: 3
##
## OOB estimate of error rate: 16.16%
## Confusion matrix:
## No Yes class.error
## No 501 48 0.08743169
## Yes 96 246 0.28070175
```

A matriz de confusão mostra que o erro de classificação é de 30%.

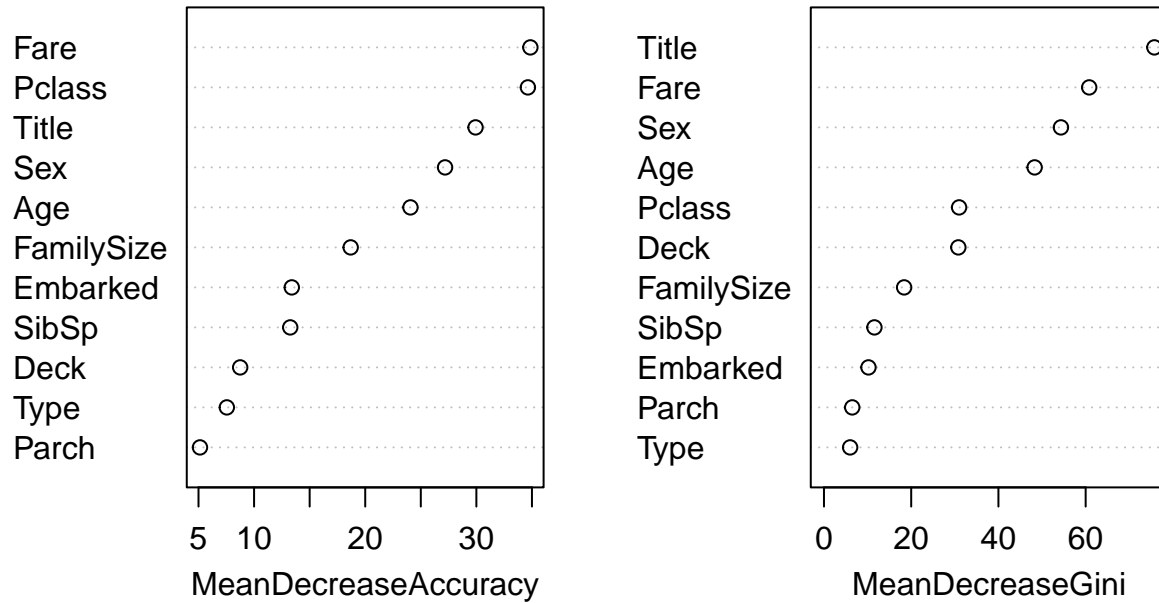
```
plot(rfmodel, main="")
legend("topright", c("OOB", "0", "1"), text.col=1:6, lty=1:3, col=1:3)
title(main="Error Rates Random Forest")
```



O gráfico mostra que em algum momento entre 0 e 100 árvores, o ótimo é alcançado e depois disso o erro OOB se torna plano. Vamos verificar a importância das variáveis.

```
varImpPlot(rfmodel)
```

## rfmodel



A diminuição média da precisão é de 100% para a classe p, o que significa que se fizermos uma permutação aleatória na variável, a diminuição será de 100%.

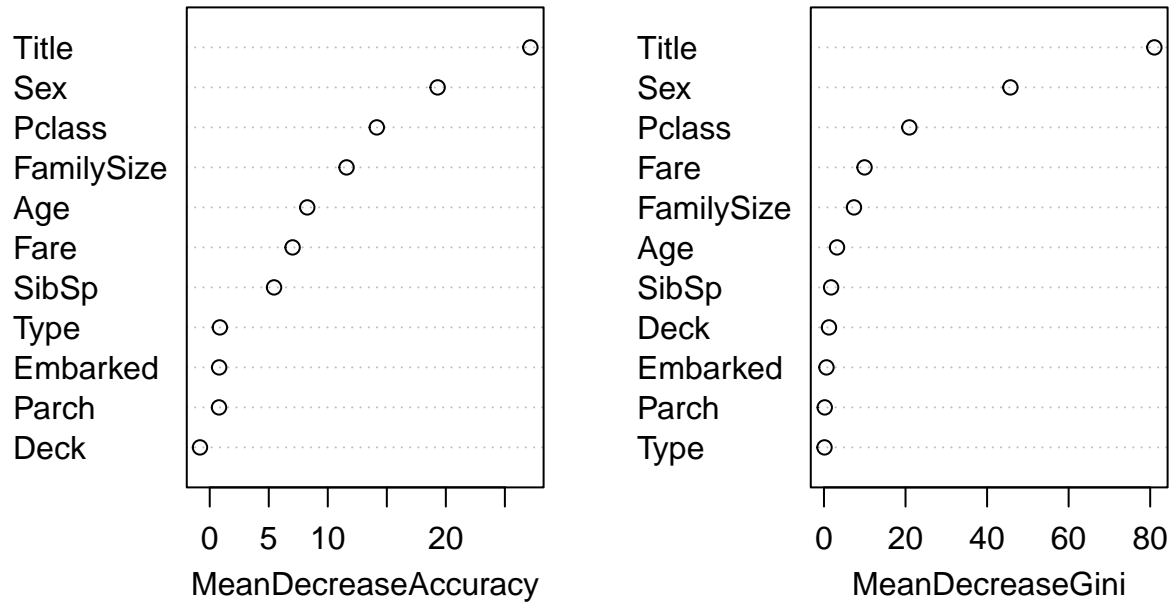
Vamos ajustar nossa floresta aleatória.

```
variable=c("Pclass","Sex","Age","Fare","Embarked","Title","Deck","FamilySize","Type","SibSp","Parch")
tunedrfmodel=tuneRF(x=train[,variable],y=as.factor(train$Survived),mtryStart = 3,ntreeTry = 100,stepFac
```

```
## -0.01156069 0.001
## 0.02890173 0.001
## -0.1071429 0.001
```

```
varImpPlot(tunedrfmodel)
```

## tunedrfmodel



A partir da análise de ajuste, podemos observar que os títulos e o sexo são as variáveis mais importantes para nossa previsão.

Vamos fazer a previsão usando os dados de treinamento.

```
trainpredict=table(predict(tunedrfmodel),train$Survived)
caret::confusionMatrix(trainpredict)
```

```
## Confusion Matrix and Statistics
##
##
##      No Yes
## No  485  92
## Yes   64 250
##
##              Accuracy : 0.8249
##              95% CI   : (0.7983, 0.8493)
##    No Information Rate : 0.6162
##    P-Value [Acc > NIR] : < 2e-16
##
##              Kappa : 0.6241
##
##  Mcnemar's Test P-Value : 0.03064
##
##              Sensitivity : 0.8834
##              Specificity : 0.7310
```

```
##          Pos Pred Value : 0.8406
##          Neg Pred Value : 0.7962
##          Prevalence : 0.6162
##          Detection Rate : 0.5443
##          Detection Prevalence : 0.6476
##          Balanced Accuracy : 0.8072
##
##          'Positive' Class : No
##
```

A precisão é de 80%.

Vamos usar os dados de teste.

```
test$Survived=NULL
titanicpred=predict(tunedrfmodel,test,OOB=TRUE,type="response")
titanicpred=ifelse(titanicpred=="No",0,1)
solution=data.frame(PassengerID=test$PassengerId,Survived=titanicpred)
write.csv(solution,file="submission.csv",row.names=F)
```

##Conclusão:

Este conjunto de dados, um dos mais populares na comunidade Kaggle, proporcionou uma excelente oportunidade para explorar diversas técnicas de análise de dados. A partir dele, pudemos aprofundar nossos conhecimentos em visualização de dados, tratamento de valores ausentes, modelagem utilizando diferentes pacotes e avaliação da precisão dos modelos.

Ao analisar os resultados, identificamos que as variáveis “título” e “sexo” exercem um papel fundamental nas nossas previsões. Essa descoberta demonstra a importância de uma análise cuidadosa dos dados para identificar os fatores que mais influenciam o resultado desejado.

Embora tenhamos obtido resultados promissores, há ainda espaço para aprimorar o modelo. A implementação de regressão logística, por exemplo, pode trazer novas insights e melhorar a precisão das nossas previsões. Essa será uma das próximas etapas deste estudo.

Em resumo, este trabalho nos permitiu aplicar conceitos de aprendizado de máquina em um cenário real e compreender a importância da interpretação dos resultados para a tomada de decisões.