Introduction to Machine Learning with R

Darshna Shah, Laura da Silva, Bianca Furtuna

The Scenario

One of the most famous tragedies in history, the sinking of the Titanic is associated with the limited numbers of lifeboats, not enough for all passengers. There are a lot of hypotheses covering which groups of people were more likely to survive and you are going to explore some of those in this workshop. After investigating and analysing the dataset, a machine learning model will be built to predict which passengers survived the Titanic.

The data

The dataset used throughout this workshop is very similar to the Titanic dataset available on Kaggle:

https://www.kaggle.com/c/titanic/data

The dataset has 9 columns:

- **Survived** - Yes(1) and NO(0)
- **Pclass** - Ticket Class: 1 - Upper , 2- Middle , 3 - Lower
- **Sex** - male or female
- **Age** - age in years
- **SibSp** - number of siblings / spouses aboard the Titanic
- **Parch** - number of parents / children aboard the Titanic
- **Fare** - passenger fare
- **Embarked** - port of embarkation: C - Cherbourg, Q - Queenstown, S - Southampton
- **FamilySize** - number of family members

Setup

In order to follow through this workshop, R and Rstudio needs to be installed on your machine. If you have not installed them, please use the following links:

- R
- Rstudio

Once you have installed RStudio, take some time to get familiar with it. Try to create an R script, find the interactive console and figure out how to access the help documentation. The RStudio cheatsheet might be an useful way to get started quicker.

There are three libraries which are going to be used in this workshop: **caret**, **ggplot2** and **e1071**. Install and load the R libraries:

```r
#install the R libraries needed for this workshop
install.packages("caret")
install.packages("ggplot2")
install.packages("e1071")
```

```r
#load the libraries
library(caret)
library(ggplot2)
library(e1071)
```

View the current working directory in RStudio and then set it to the folder which contains the Titanic dataset:

```
path <- "insert your folder path here"

getwd()
setwd(path)
```

Load and explore the data

One of the most important parts of the Machine Learning process is data understanding, so it is essential to make sure you understand the features of your dataset and what their meaning is.

The first step is to read in the titanic dataset from the csv file:

```
TitanicData <- read.csv("train.csv")
```

Examine the dimensions of the data set:

```
dim(TitanicData)
```

A dataset can contain features of different types. Features can be:

- numerical (e.g. 124, 53, 4.5)
- integer(e.g. 1, 5, 7, 8)
- character(e.g. “data”, “set”, “one”)
- logical (TRUE/FALSE)
- factor or categorical variables

Examine the data types of variables in the Titanic data set:

```
str(TitanicData)
```

```
## 'data.frame':    891 obs. of  9 variables:
##  $ Survived  : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass    : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ 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 ...
##  $ Fare      : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Embarked  : Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
##  $ FamilySize: int  2 2 1 2 1 1 1 5 3 2 ...
```

Some of the features in this dataset should be categorical rather than integer or numerical.

Convert relevant variables to factors:

```
TitanicData$Survived <- as.factor(TitanicData$Survived)
TitanicData$Pclass <- as.factor(TitanicData$Pclass)
TitanicData$Age <- as.integer(TitanicData$Age)
```

Examine the the summary statistics for each variable in the data frame:

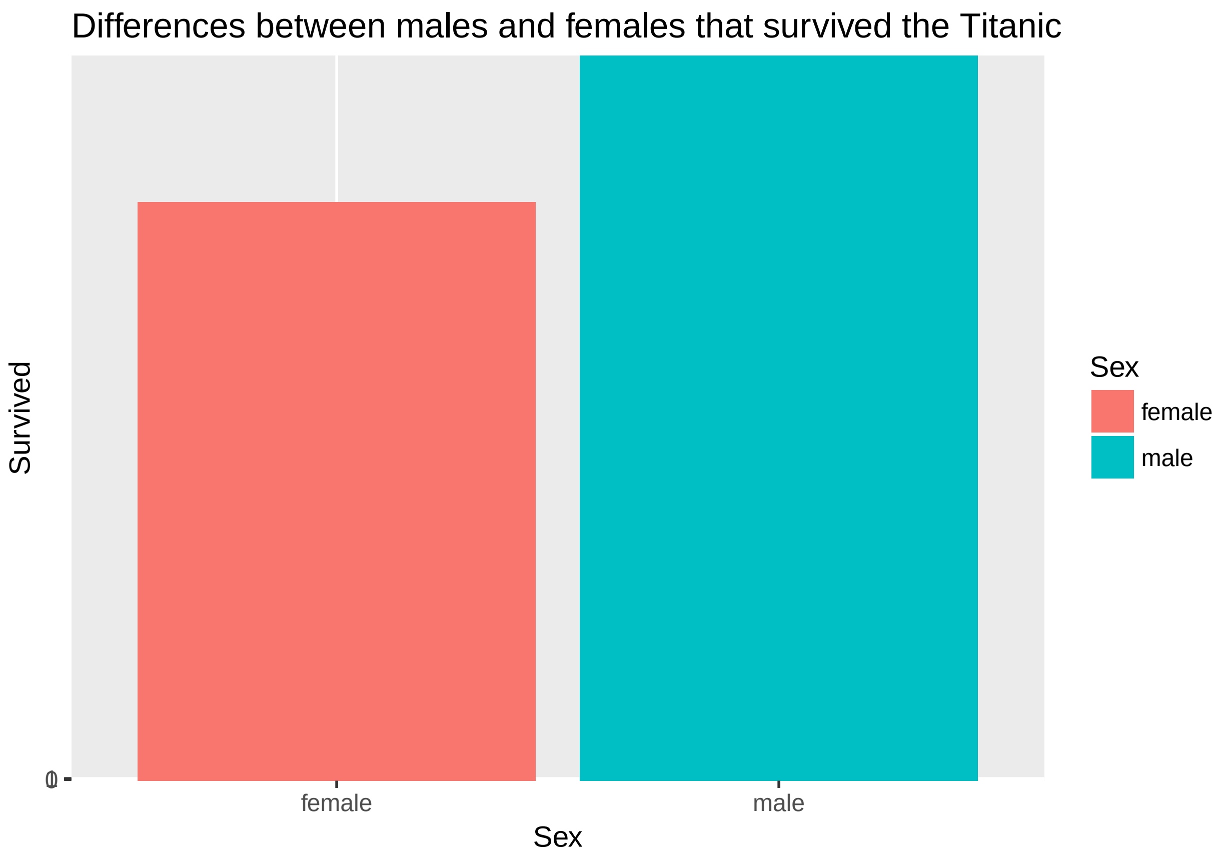
```
summary(TitanicData)
```

```
##  Survived Pclass      Sex            Age             SibSp
##  0:549    1:216   female:314   Min.   : 0.00   Min.   :0.000
##  1:342    2:184   male  :577   1st Qu.:22.00   1st Qu.:0.000
##           3:491                Median :28.00   Median :0.000
##                                 Mean   :29.44   Mean   :0.523  
##                                 3rd Qu.:36.00   3rd Qu.:1.000  
##                                 Max.   :80.00   Max.   :8.000  
##      Parch             Fare        Embarked   FamilySize    
##  Min.   :0.0000   Min.   :  0.00   C:168    Min.   : 1.000  
##  1st Qu.:0.0000   1st Qu.:  7.91   Q: 77    1st Qu.: 1.000  
##  Median :0.0000   Median : 14.45   S:646    Median : 1.000  
##  Mean   :0.3816   Mean   : 32.20            Mean   : 1.905  
##  3rd Qu.:0.0000   3rd Qu.: 31.00            3rd Qu.: 2.000  
##  Max.   :6.0000   Max.   :512.33            Max.   :11.000
```

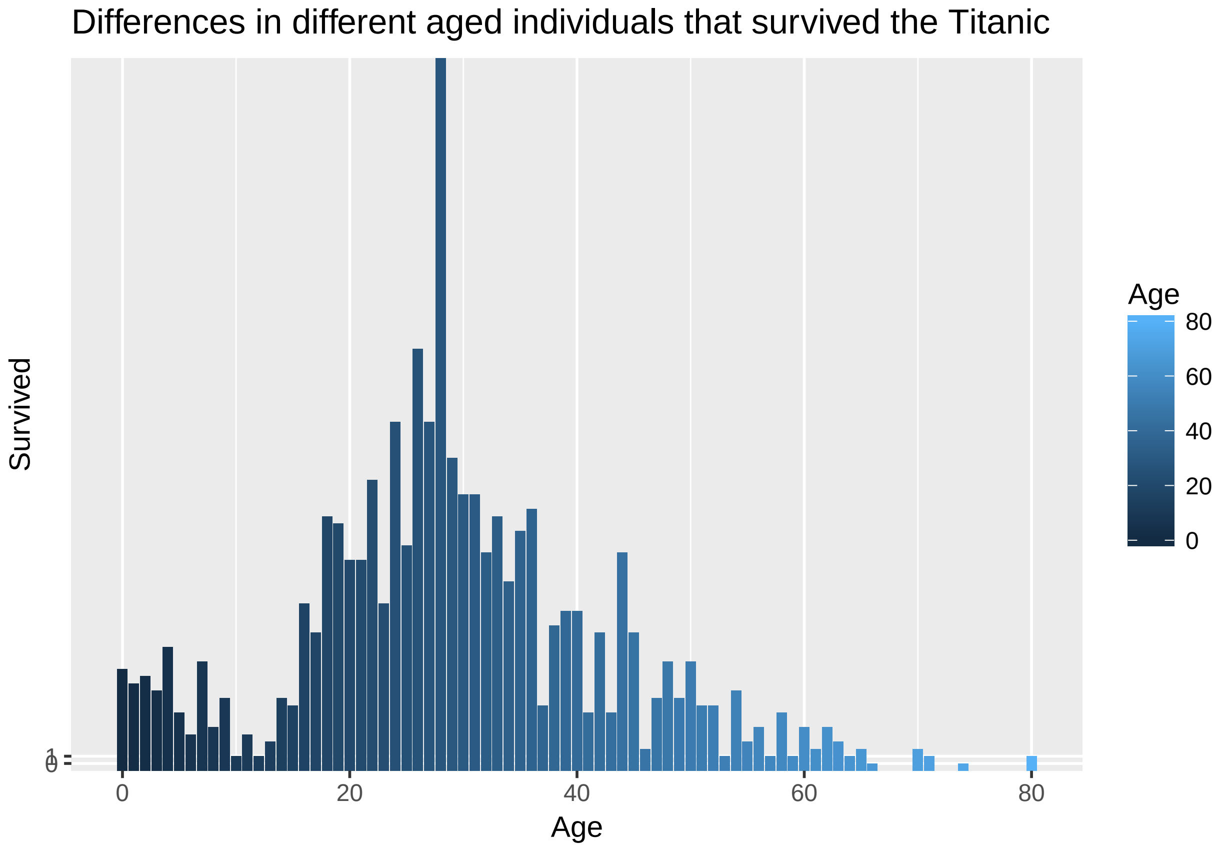
Visualisations between variables of interest can assist you to understand your data and develop hypotheses.

Ggplot2 is a great plotting system for R which will help you create great visuals very quickly. Explore relationships between variables with some basic plots in ggplot2:

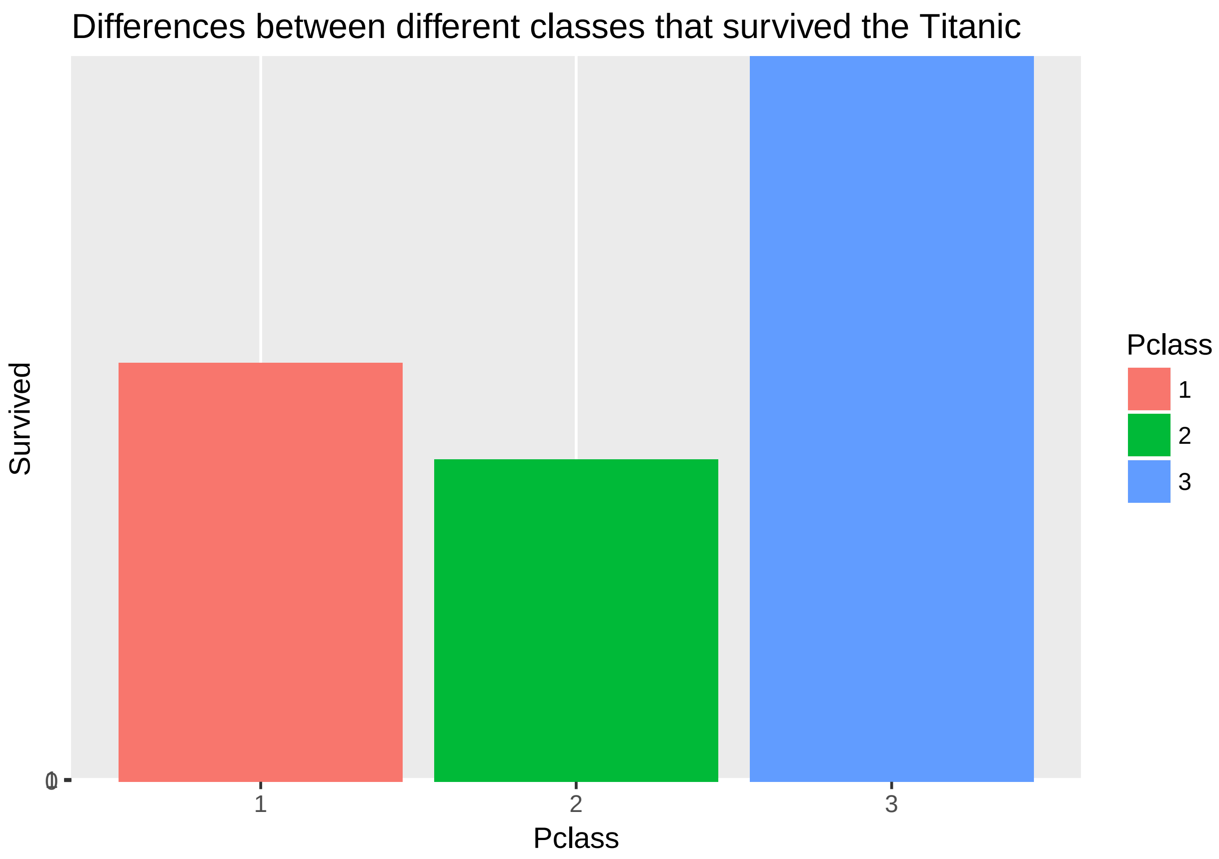
```
ggplot(data=TitanicData, aes(x=Sex, y=Survived, fill=Sex)) +
  geom_bar(stat="identity")+
  ggtitle("Differences between males and females that survived the Titanic")
```



```
ggplot(data=TitanicData, aes(x=Age, y=Survived, fill=Age)) +
  geom_bar(stat="identity")+
  ggtitle("Differences in different aged individuals that survived the Titanic")
```



```
ggplot(data=TitanicData, aes(x=Pclass, y=Survived, fill=Pclass)) +
  geom_bar(stat="identity")+
  ggtitle("Differences between different classes that survived the Titanic")
```



Train a machine learning model

Once you are familiar with your dataset, you can start experimenting with different machine learning techniques. You are going to build a machine learning model to predict which passengers would survive the Titanic. The variable **Survived** will be used to teach the model which patterns to look for in a passenger in order to classify them as a survivor or not a survivor. In Machine Learning talk, this variable is normally called a label.

As there is a label and this label contains only two classes: yes for survived and no for not survived, the model that will be built is going to be a **Binary classification model**.

A common practice in the machine learning process is to split your initial dataset into a training and testing dataset. The training dataset is used to train the model, hence to teach how to distinguish between survivors and non-survivors. The testing dataset is used to evaluate how well your trained model is performing on unseen data.

Use the **caret** package to split the dataset into training and testing dataset. Create a 70%/30% split and keep the proportions of the Survived class label the same across splits:

```
set.seed(54321)
indexes <- createDataPartition(TitanicData$Survived,
                               times = 1,
                               p = 0.7,
                               list = FALSE)
titanic.train <- TitanicData[indexes,]
titanic.test <- TitanicData[-indexes,]
```

```
# Examine the proportions of the Survived class lable across the datasets.
prop.table(table(TitanicData$Survived))
```

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

```
prop.table(table(titanic.train$Survived))
```

```
##
##     0     1
## 0.616 0.384
```

```
prop.table(table(titanic.test$Survived))
```

```
##
##         0         1
## 0.6165414 0.3834586
```

Use the **train** function from the **caret** package to build a linear classification model and explore the characteristics of your train model:

```
model <- train(Survived~., data = titanic.train, method = "glm")

model
```

```
## Generalized Linear Model
##
## 625 samples
##   8 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 625, 625, 625, 625, 625, 625, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.7959739  0.5613112
```

```
model$finalModel
```

```
##
## Call:  NULL
##
## Coefficients:
## (Intercept)      Pclass2      Pclass3      Sexmale          Age
##    4.448760    -0.910731    -2.331294    -2.642270    -0.042840
##       SibSp        Parch         Fare    EmbarkedQ    EmbarkedS
##   -0.332563    -0.165494     0.001357    -0.275704    -0.586190
##  FamilySize
##          NA
##
## Degrees of Freedom: 624 Total (i.e. Null);  615 Residual
## Null Deviance:       832.5
## Residual Deviance: 556.1     AIC: 576.1
```

Score and evaluate the model

Once you have trained your model, you need to fit it to your test dataset and evaluate how well the model is performing.

Use the **predict** function to fit the model to the test data set and predict survival:

```
preds <- predict(model, titanic.test)

preds
```

```
##   [1] 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 0 0 0 0 1 0 1 0 1 0 1 0
##  [36] 1 0 0 1 0 0 0 1 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 1 1 0 0
##  [71] 1 0 1 0 1 1 0 0 0 1 0 0 0 0 0 0 1 1 1 1 1 0 1 0 1 0 1 0 0 0 1 0 0 1 0
## [106] 0 0 0 1 0 1 0 0 0 0 1 0 0 0 1 0 0 1 1 1 0 0 0 0 0 1 1 0 1 0 0 0 0 1 0
## [141] 0 0 1 0 0 0 0 1 0 0 1 1 0 0 1 0 0 1 0 1 1 1 1 1 0 0 0 0 0 1 0 1 1 0 1
## [176] 0 0 1 0 1 1 0 1 0 1 0 0 0 0 0 1 1 0 1 0 1 1 1 0 0 0 0 0 0 1 0 0 1 0 1
## [211] 0 0 0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 1 0 0 0 0 1 0 1
## [246] 0 1 0 0 0 1 0 0 0 0 1 0 1 0 1 1 0 0 0 0 1
## Levels: 0 1
```

Evaluate the model performance with a confusion matrix:

```
confusionMatrix(preds, titanic.test$Survived)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0 141  28
##          1  23  74
## 
##                Accuracy : 0.8083
##                  95% CI : (0.7557, 0.8538)
##     No Information Rate : 0.6165
##     P-Value [Acc > NIR] : 1.164e-11
## 
##                   Kappa : 0.5907
##  Mcnemar's Test P-Value : 0.5754
## 
##             Sensitivity : 0.8598
##             Specificity : 0.7255
##          Pos Pred Value : 0.8343
##          Neg Pred Value : 0.7629
##              Prevalence : 0.6165
##          Detection Rate : 0.5301
##    Detection Prevalence : 0.6353
##       Balanced Accuracy : 0.7926
## 
##        'Positive' Class : 0
## 
```