Titanic dataset analasys

Jetze Luyten, David Silva Troya, Axel Van Gestel

Setup the enviroment

The first thing to setup for the analysts is the environment with the required packages and settings.

Install required packages and load required libraries

During this analysis we used the `tidyverse` package for reading, cleaning and plotting the data and the `ggcorrplot` package to visualize the correlation matrix into a heat map.

```r
# install.packages("tidyverse")
# install.packages("ggcorrplot")

library(tidyverse) # Contains all tidyverse packages (ggplot2, dplyr, ...)
library(ggcorrplot) # Used for generating correlation heatmaps (uses ggplot2)
```

Setup enviroment settings

In the following code block we set the language R uses for it's messages to English, clear all the global variables so that we always start with a clean slate and setup ggplot to center the plot titles by default.

```r
Sys.setenv(LANG = "en") # Set language to English
rm(list = ls()) # Clears the Global Env
theme_update(plot.title = element_text(hjust = 0.5)) # Center all plot titles
```

Read and import the data set

Read the data set (uses readr)

```r
column_types <- cols(
  Survived = col_factor(),
  Pclass = col_factor(include_na = TRUE, ordered = TRUE),
  Name = col_character(),
  Sex = col_factor(),
  Age = col_double(),
  SibSp = col_integer(),
  Parch = col_integer(),
  Ticket = col_character(),
  Fare = col_double(),
  Cabin = col_character(),
  Embarked = col_factor(include_na = TRUE, ordered = TRUE)
)
train <- read_csv("./kaggle/titanic/train.csv",
                  col_types = column_types,
                  col_select = -c(PassengerId))
```

Rename the factors to be human readable (uses dplyr)

```
train$Survived <- recode_factor(train$Survived,
                                "0" = "No",
                                "1" = "Yes",)

train$Pclass <- recode_factor(train$Pclass,
                              "1" = "1st",
                              "2" = "2nd",
                              "3" = "3rd",
                              .default = "Unknown", # NA -> Unknown
                              .ordered = TRUE)

train$Embarked <- recode_factor(train$Embarked,
                                "S" = "Southampton (England)",
                                "C" = "Cherbourg (France)",
                                "Q" = "Queenstown (Ireland)",
                                .default = "Unknown", # NA -> Unknown
                                .ordered = TRUE)

# Clear not needed variables
rm(column_types)
```

Filtering and cleaning

Check for the number of NA's in each column

```
sanity_check <- function(my_df) {
  for (j in 1:ncol(my_df)) {
    print(paste(names(my_df[j]), ":", sum(is.na(my_df[, j]))))
  }
}

sanity_check(train)
```

```
## [1] "Survived : 0"
## [1] "Pclass : 0"
## [1] "Name : 0"
## [1] "Sex : 0"
## [1] "Age : 177"
## [1] "SibSp : 0"
## [1] "Parch : 0"
## [1] "Ticket : 0"
## [1] "Fare : 0"
## [1] "Cabin : 687"
## [1] "Embarked : 0"
```

View 'train' tibble

```
train
```

```
## # A tibble: 891 x 11
##    Survived Pclass Name       Sex     Age SibSp Parch Ticket  Fare Cabin Embar~1
##    <fct>    <ord>  <chr>      <fct> <dbl> <int> <int> <chr>  <dbl> <chr> <ord>
##  1 No       3rd    Braund, M~ male     22     1     0 A/5 2~  7.25 <NA>  Southa~
##  2 Yes      1st    Cumings, ~ fema~    38     1     0 PC 17~ 71.3  C85   Cherbo~
##  3 Yes      3rd    Heikkinen~ fema~    26     0     0 STON/~  7.92 <NA>  Southa~
##  4 Yes      1st    Futrelle,~ fema~    35     1     0 113803 53.1  C123  Southa~
##  5 No       3rd    Allen, Mr~ male     35     0     0 373450  8.05 <NA>  Southa~
##  6 No       3rd    Moran, Mr~ male     NA     0     0 330877  8.46 <NA>  Queens~
##  7 No       1st    McCarthy,~ male     54     0     0 17463  51.9  E46   Southa~
##  8 No       3rd    Palsson, ~ male      2     3     1 349909 21.1  <NA>  Southa~
##  9 Yes      3rd    Johnson, ~ fema~    27     0     2 347742 11.1  <NA>  Southa~
## 10 Yes      2nd    Nasser, M~ fema~    14     1     0 237736 30.1  <NA>  Cherbo~
## # ... with 881 more rows, and abbreviated variable name 1: Embarked
```

Adding useful columns

Add a total Family size column

```
train <- mutate(train, FamilySize = SibSp + Parch)
```

Group the cabin label into has cabin and has no cabin

```
train <- mutate(train, CabinGroups = ifelse(is.na(train$Cabin),
                                            "No cabin",
                                            "Cabin"))
```

Add Married column, only works for female passengers

```
train <- mutate(train,
                Married = ifelse(Sex == "female",
                                 stringr::str_detect(Name, "^[Mm]rs"), NA))
```

Quick sanity check of the 'train' tibble

```
tail(train)
```

```
## # A tibble: 6 x 14
##   Survived Pclass Name        Sex     Age SibSp Parch Ticket  Fare Cabin Embar~1
##   <fct>    <ord>  <chr>       <fct> <dbl> <int> <int> <chr>  <dbl> <chr> <ord>
## 1 No       3rd    "Rice, Mrs~ fema~    39     0     5 382652 29.1  <NA>  Queens~
## 2 No       2nd    "Montvila,~ male     27     0     0 211536 13    <NA>  Southa~
## 3 Yes      1st    "Graham, M~ fema~    19     0     0 112053 30    B42   Southa~
## 4 No       3rd    "Johnston,~ fema~    NA     1     2 W./C.~ 23.4  <NA>  Southa~
## 5 Yes      1st    "Behr, Mr.~ male     26     0     0 111369 30    C148  Cherbo~
## 6 No       3rd    "Dooley, M~ male     32     0     0 370376  7.75 <NA>  Queens~
## # ... with 3 more variables: FamilySize <int>, CabinGroups <chr>,
## #   Married <lgl>, and abbreviated variable name 1: Embarked
```

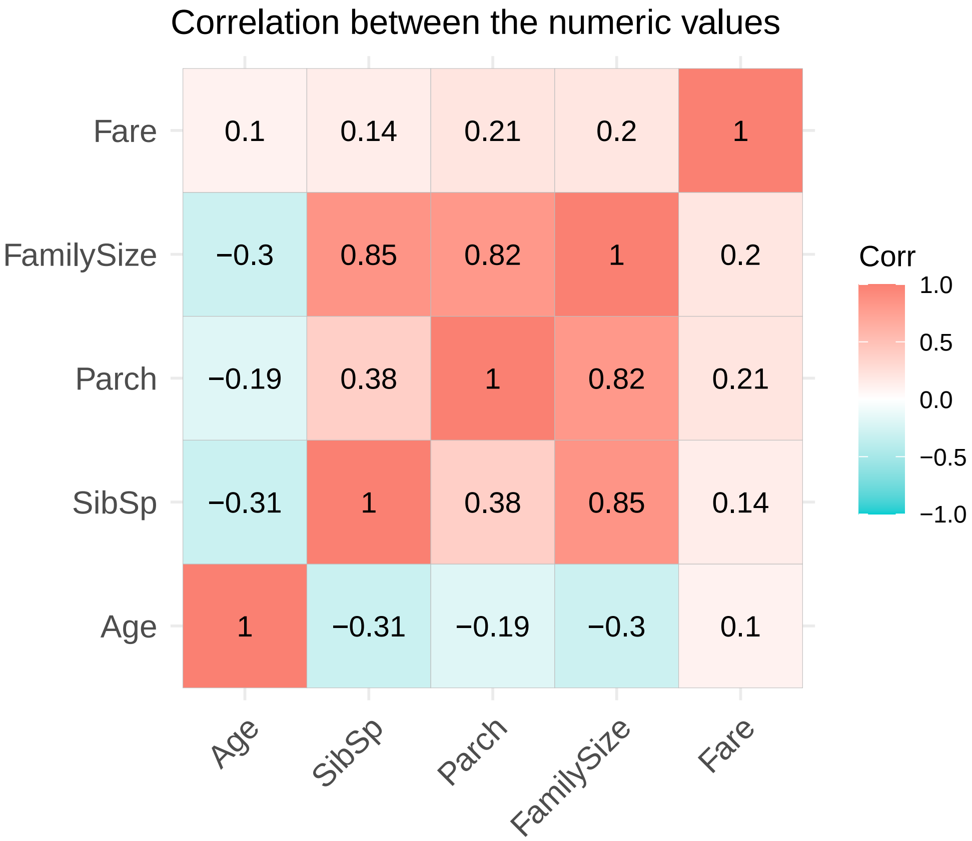
Correlation heatmap (uses ggcorrplot)

Generate a correlation heatmap of the numeric values

```
train_numeric <- select(train, Age, SibSp, Parch, FamilySize, Fare)

train_numeric_corr <- cor(train_numeric, use = "complete.obs") # Use only non NA
```

```
ggcorrplot::ggcorrplot(train_numeric_corr,
                       lab = TRUE, # Show correlation coefficients
                       colors = c("darkturquoise", "white", "salmon"),
                       title = "Correlation between the numeric values")
```

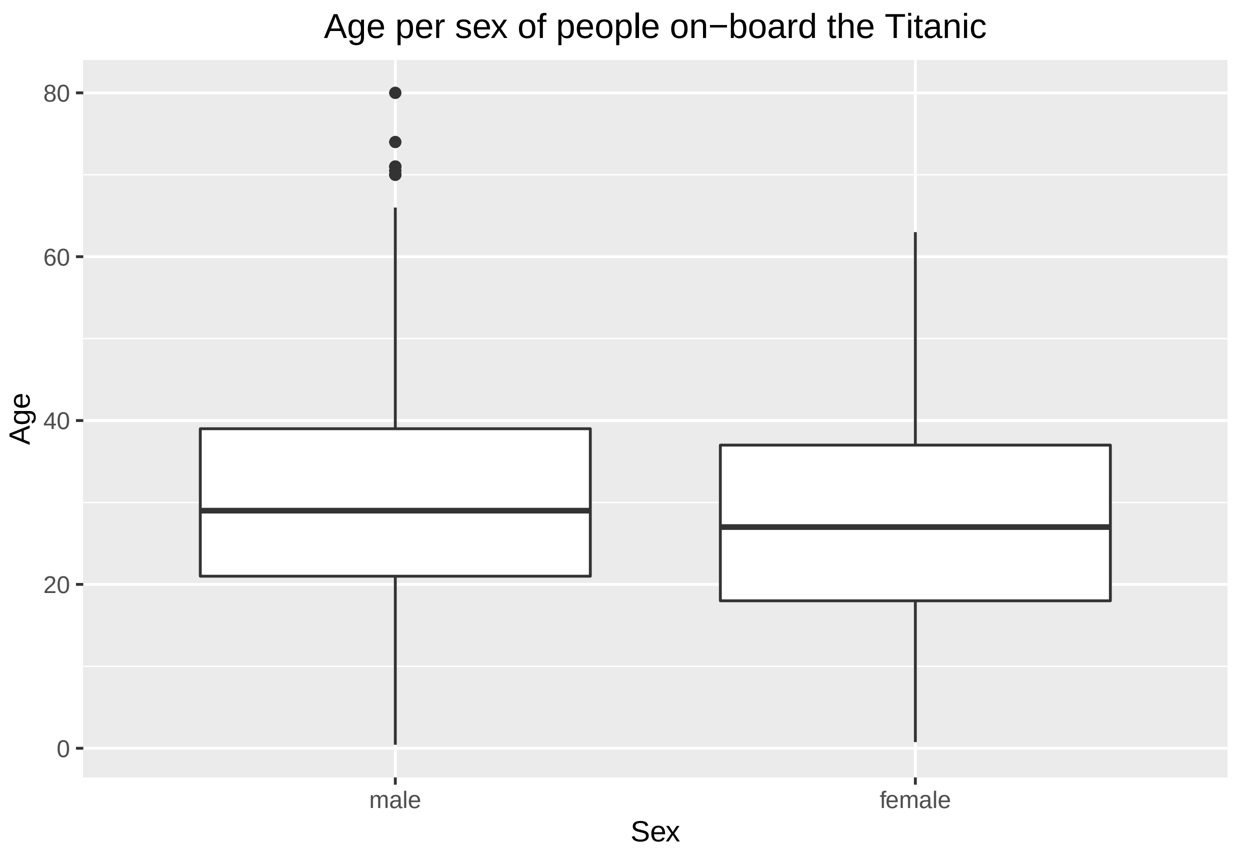


```
# Clear not needed variables
rm(train_numeric, train_numeric_corr)
```

Plots and stuff (uses ggplot2)

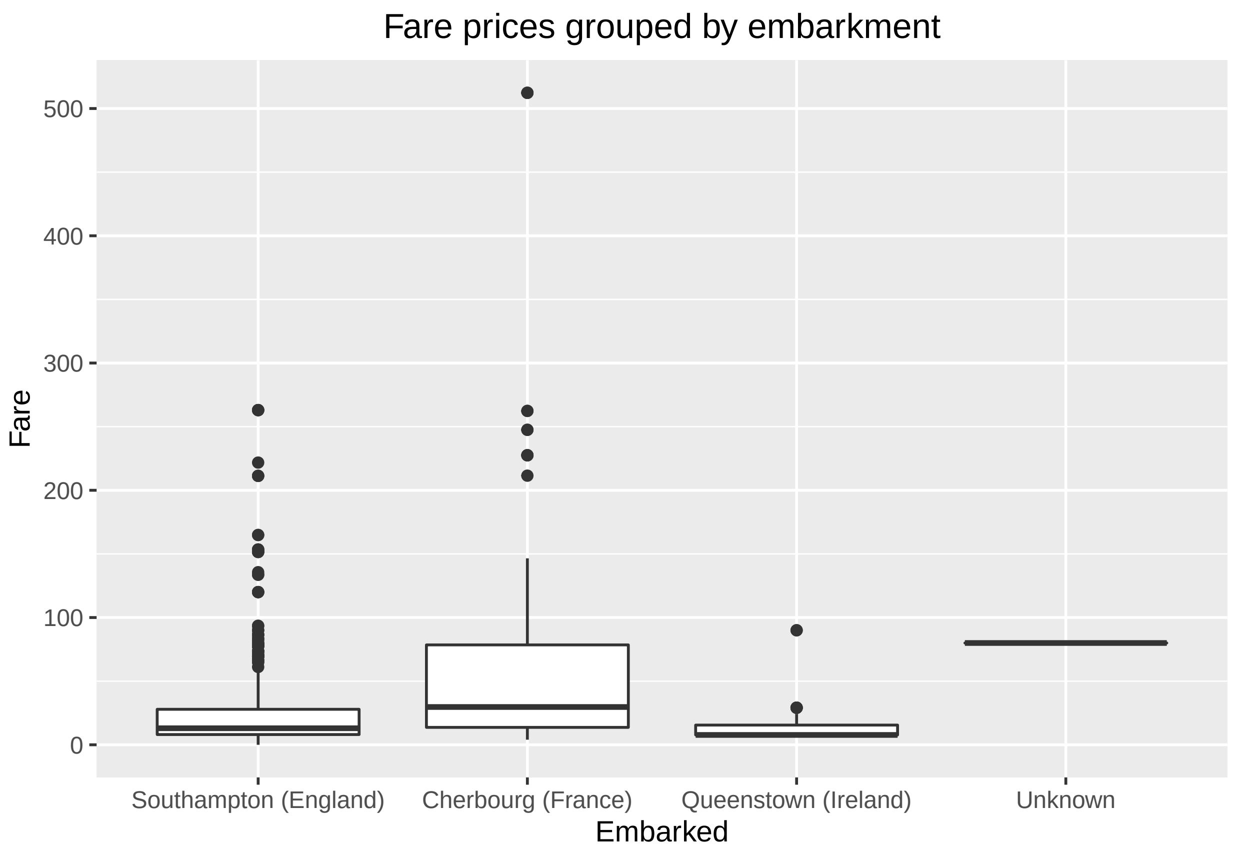
Age per sex of people on-board the Titanic

```
ggplot(data = train, mapping = aes(x = Sex, y = Age)) +
  geom_boxplot() +
  ggtitle("Age per sex of people on-board the Titanic")
```



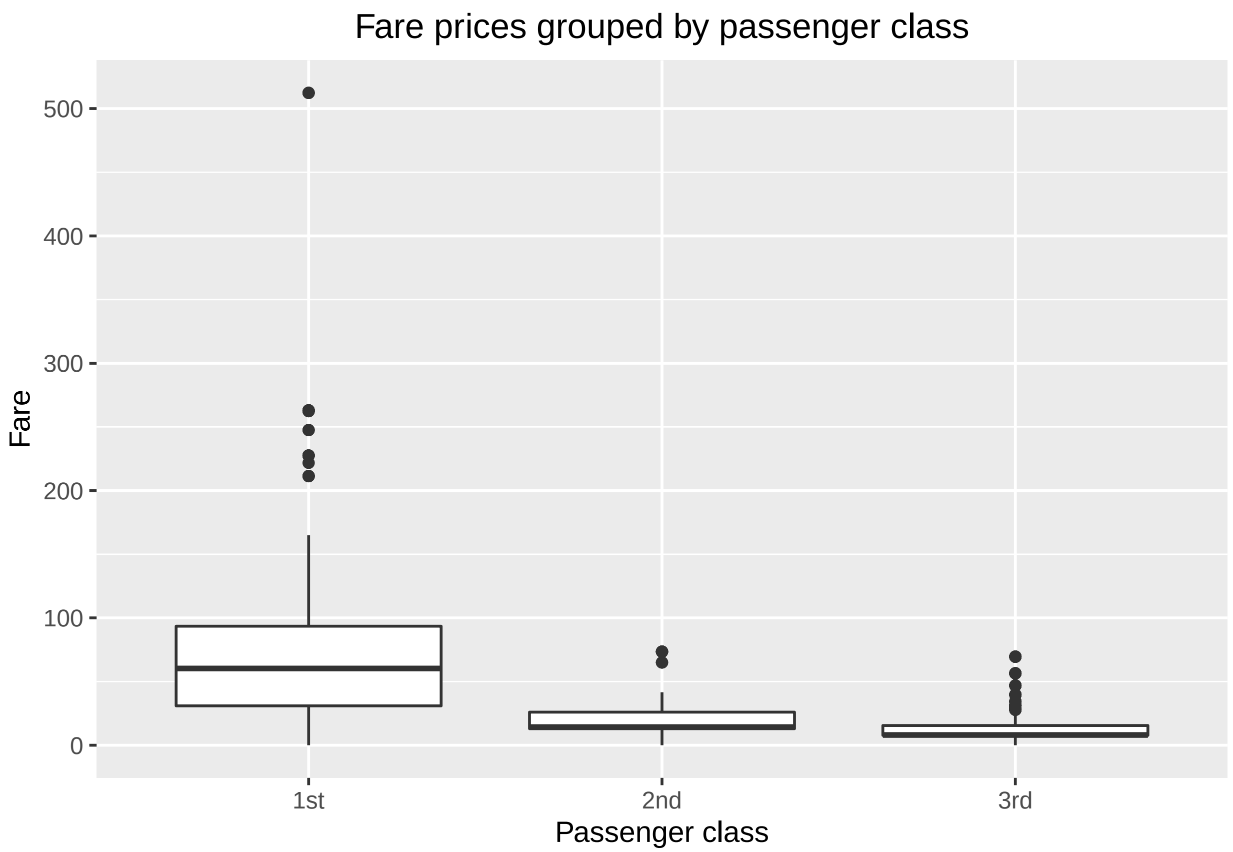
Fare prices grouped by embarkment

```
ggplot(data = train, mapping = aes(x = Embarked, y = Fare)) +
  geom_boxplot() +
  ggtitle("Fare prices grouped by embarkment")
```



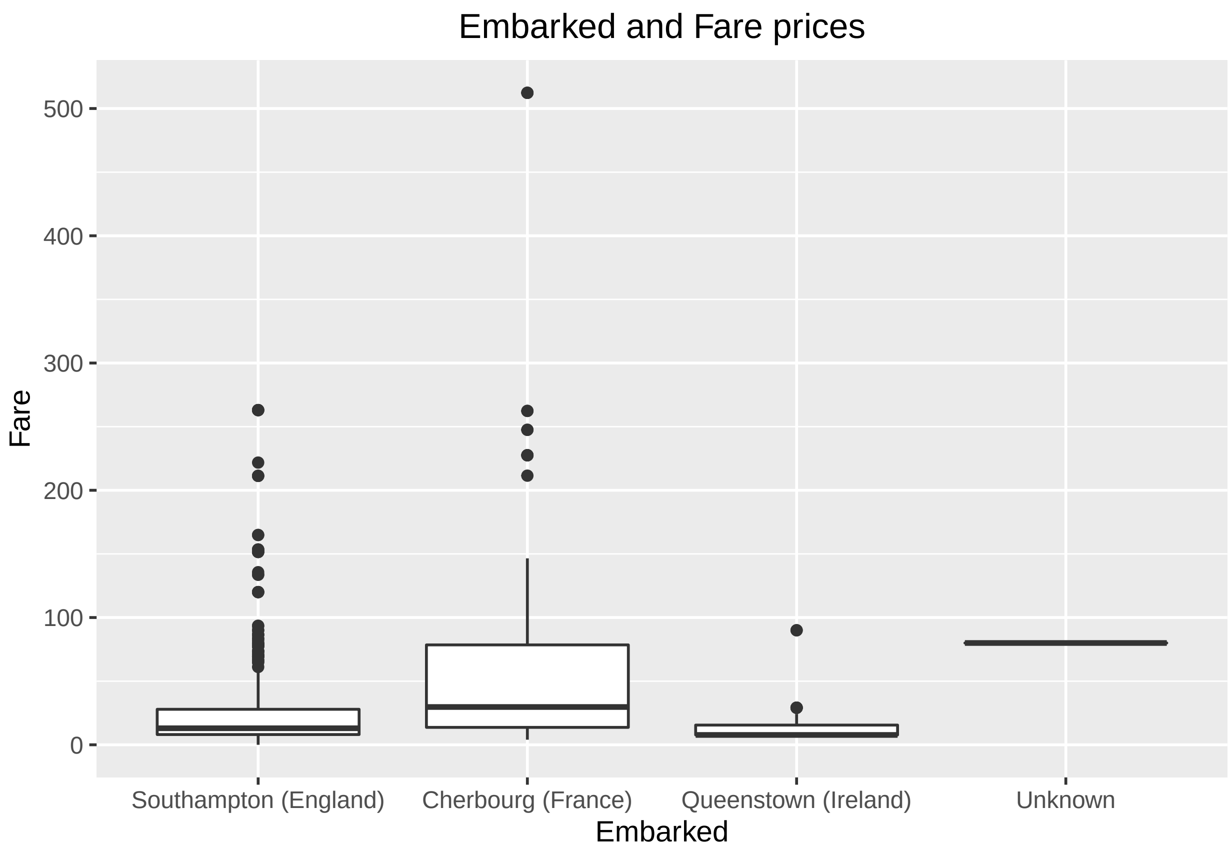
Fare prices grouped by passenger class

```
ggplot(data = train, mapping = aes(x = Pclass, y = Fare)) +
  geom_boxplot() +
  xlab("Passenger class") +
  ggtitle("Fare prices grouped by passenger class")
```



Embarked and Fare prices

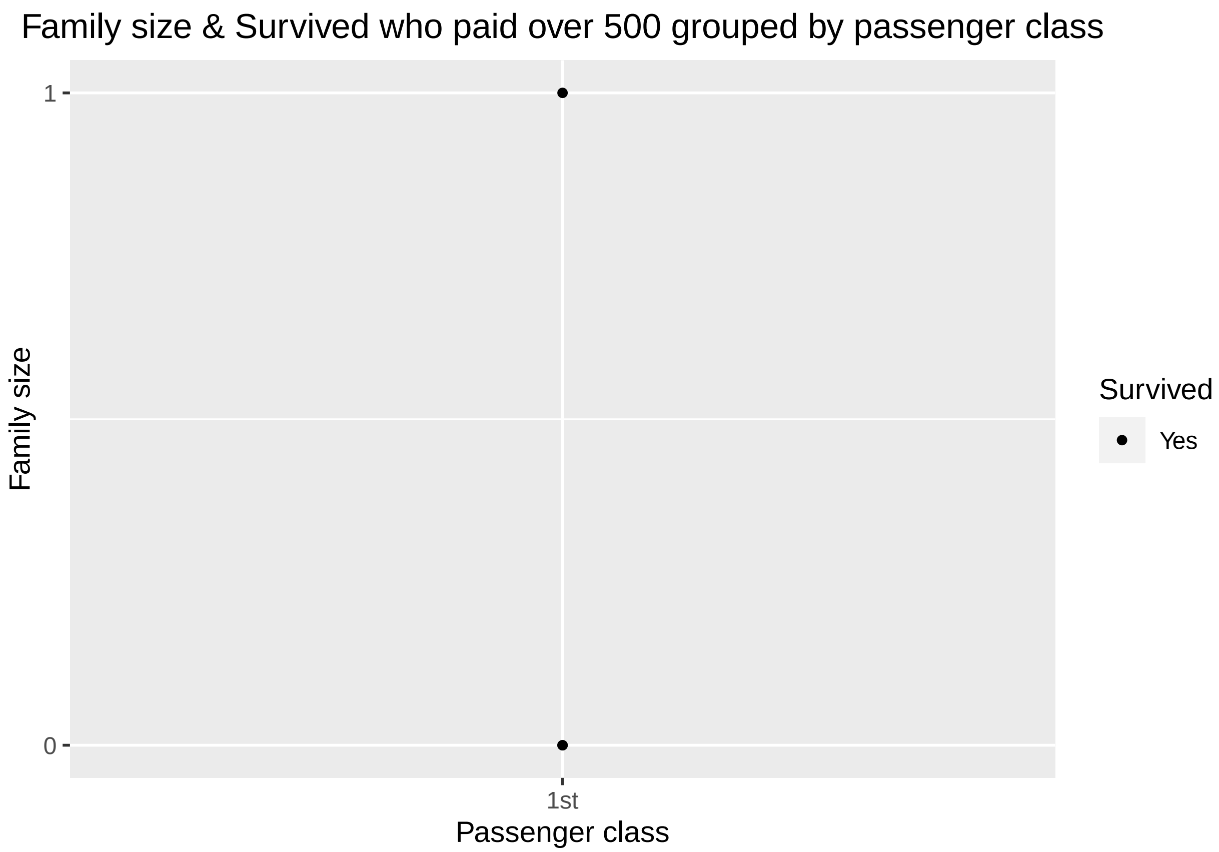
```
ggplot(data = train, mapping = aes(x = Embarked, y = Fare)) +
  geom_boxplot() +
  ggtitle("Embarked and Fare prices")
```



Family size & Survived who paid over 500 grouped by passenger class

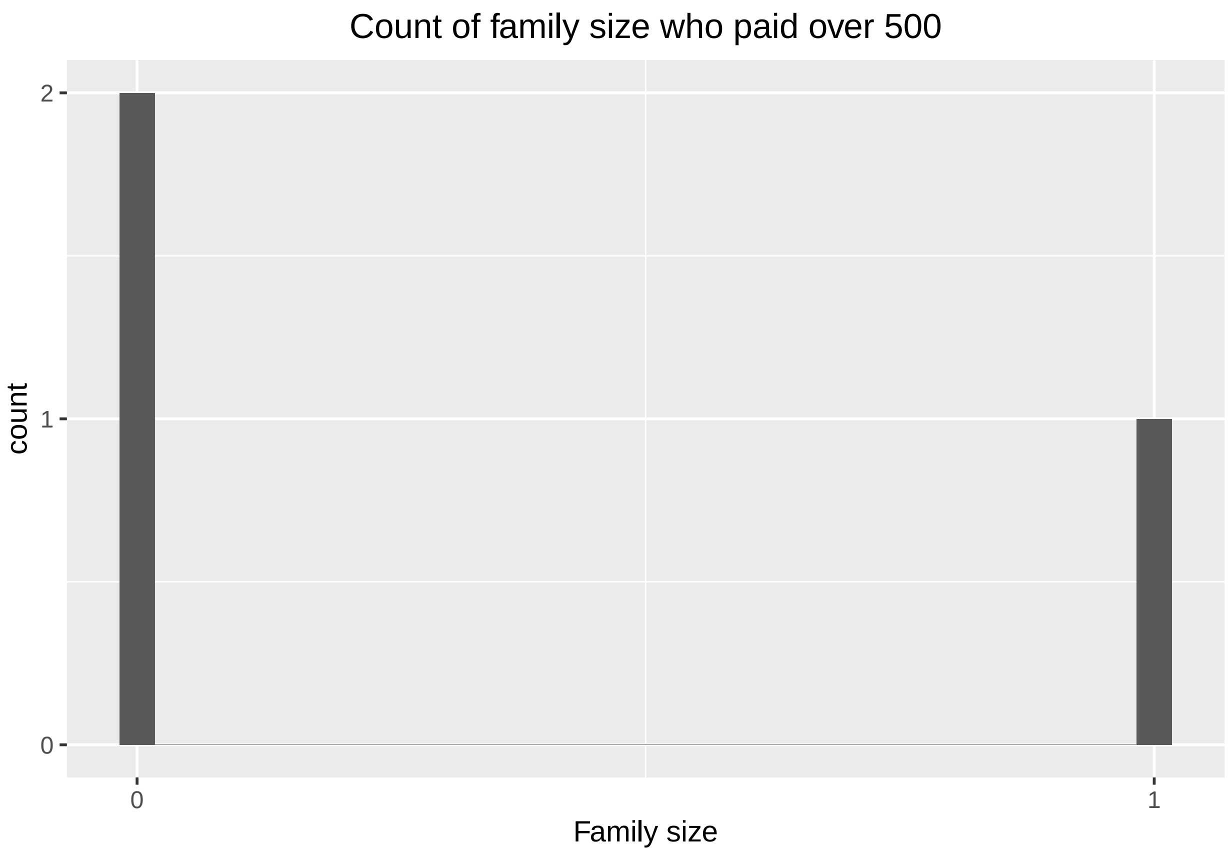
```
FareEnough <- filter(train, Fare > 500) # Fare bigger than 500

ggplot(data = FareEnough, mapping = aes(x = Pclass, y = FamilySize)) +
  geom_point(aes(shape=Survived)) +
  xlab("Passenger class") +
  ylab("Family size") +
  scale_y_continuous(breaks = scales::breaks_width(1)) +
  ggtitle("Family size & Survived who paid over 500 grouped by passenger class")
```



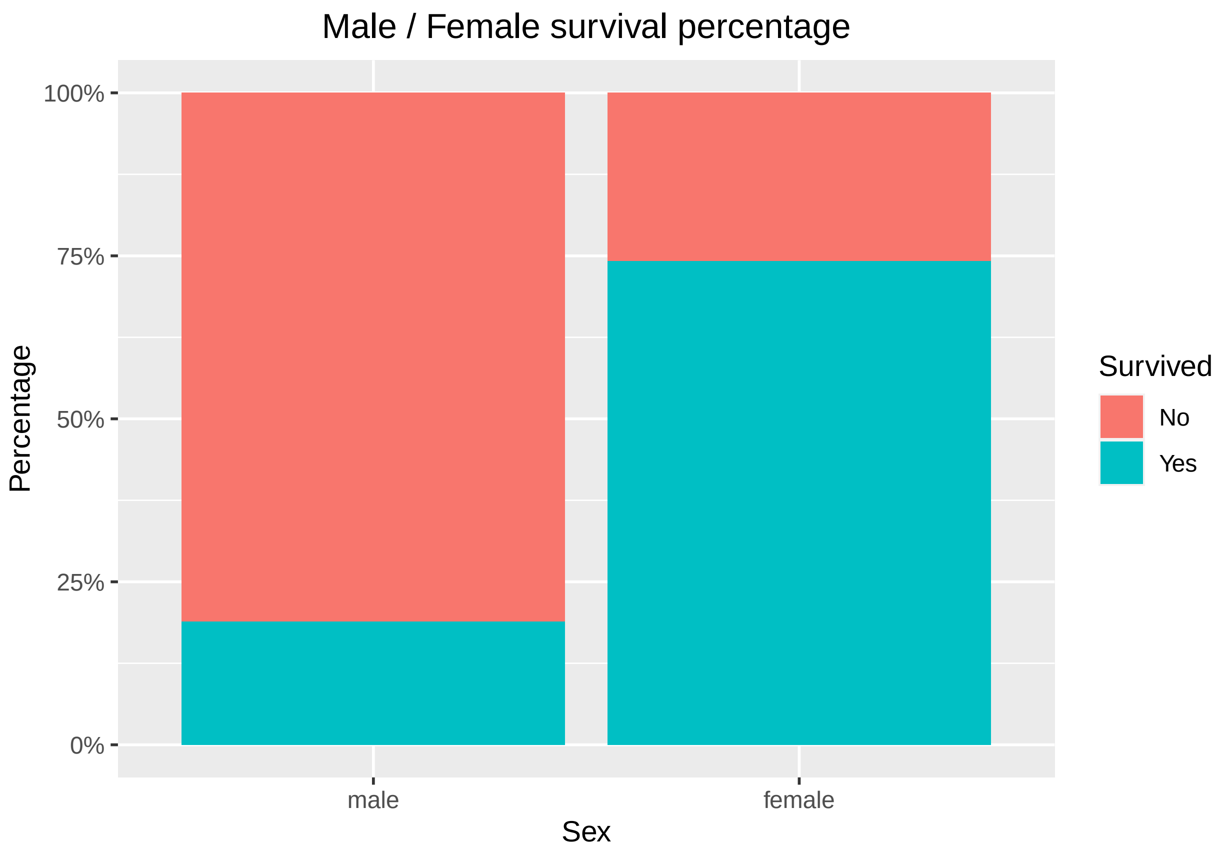
Count of family size who paid over 500

```r
ggplot(data = FareEnough, mapping = aes(x = FamilySize)) +
  geom_histogram() +
  xlab("Family size") +
  scale_y_continuous(breaks = scales::breaks_width(1)) +
  scale_x_continuous(breaks = scales::breaks_width(1)) +
  ggtitle("Count of family size who paid over 500")
```



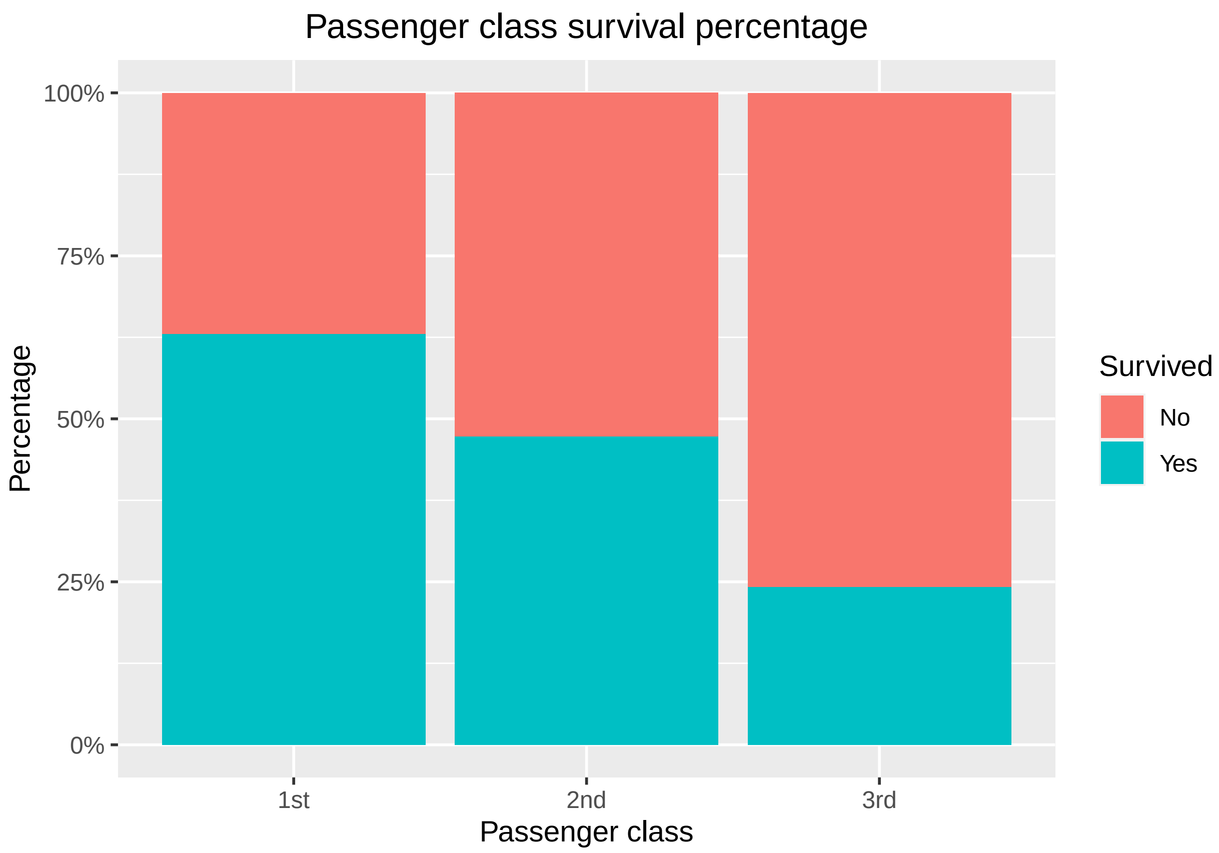
Male / Female survival percentage

```
ggplot(data = train, mapping = aes(x = Sex, fill = Survived)) +
  geom_bar(position = "fill") +
  ylab("Percentage") +
  scale_y_continuous(labels = scales::percent) +
  ggtitle("Male / Female survival percentage")
```



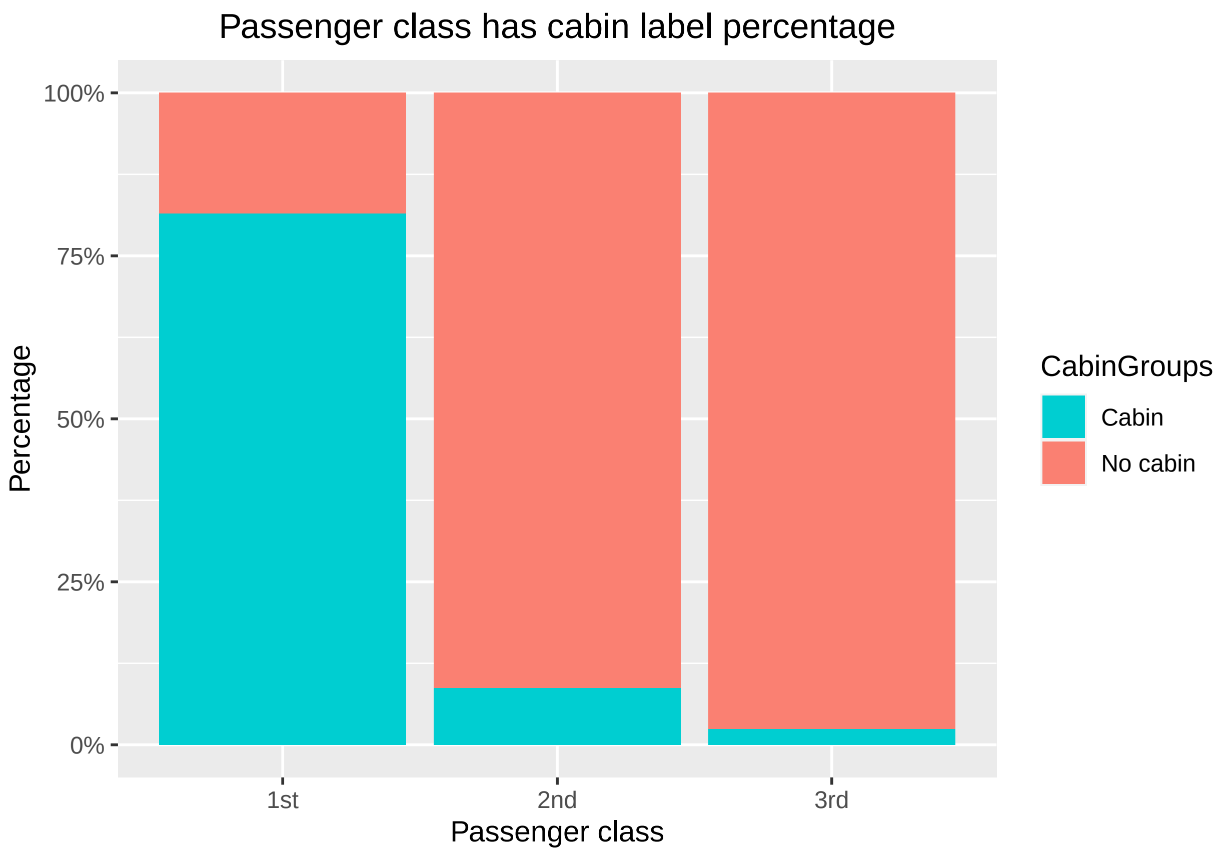
Passenger class survival percentage

```
ggplot(data = train, mapping = aes(x = Pclass, fill = Survived)) +
  geom_bar(position = "fill") +
  xlab("Passenger class") +
  ylab("Percentage") +
  scale_y_continuous(labels = scales::percent) +
  ggtitle("Passenger class survival percentage")
```



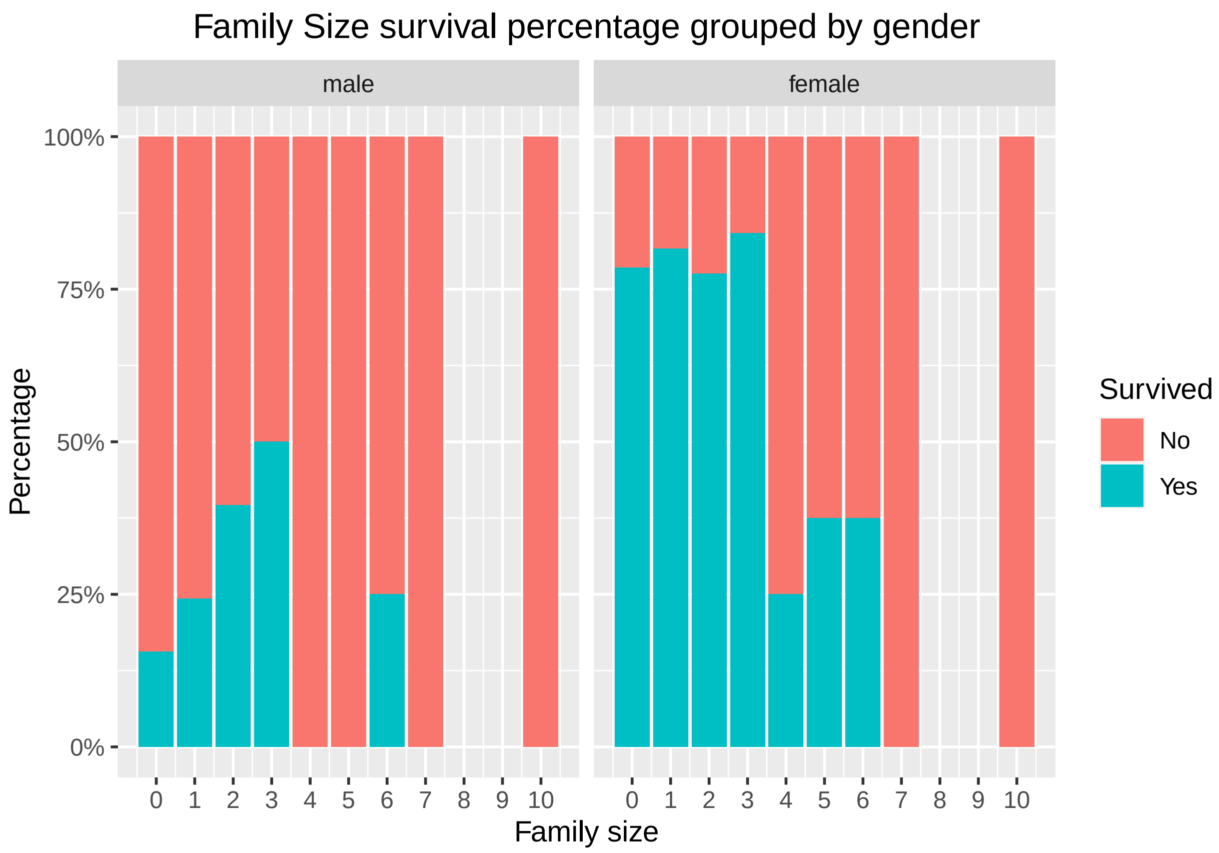
Passenger class has cabin label percentage

```
ggplot(data = train, mapping = aes(x = Pclass, fill = CabinGroups)) +
  geom_bar(position = position_fill(reverse = TRUE)) +
  scale_fill_manual(values = c("darkturquoise",
                               "salmon")) +
  xlab("Passenger class") +
  ylab("Percentage") +
  scale_y_continuous(labels = scales::percent) +
  ggtitle("Passenger class has cabin label percentage")
```



Family Size survival percentage grouped by gender

```
ggplot(data = train, mapping = aes(x = FamilySize, fill = Survived)) +
  geom_bar(position = "fill") +
  facet_wrap(~ Sex) +
  scale_x_continuous(breaks = min(train$FamilySize):max(train$FamilySize)) +
  xlab("Family size") +
  ylab("Percentage") +
  scale_y_continuous(labels = scales::percent) +
  ggtitle("Family Size survival percentage grouped by gender")
```



References

- Correlation heatmap using ggplot2