```
#### Explore Titanic data set #####

#### General setups ####
# Packages managemement
library(dplyr)
```

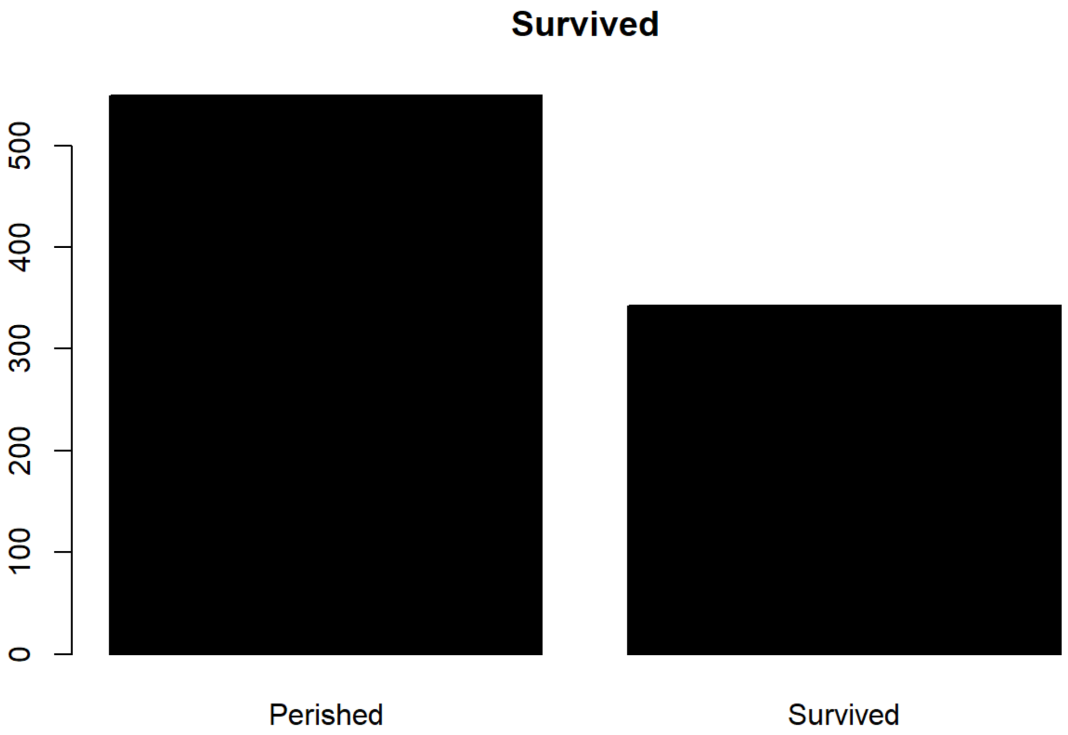
```
##
## Attaching package: 'dplyr'
## The following object is masked from 'package:seqinr':
##
##     count
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(reshape2)
library(vcd)
library('randomForest')
```

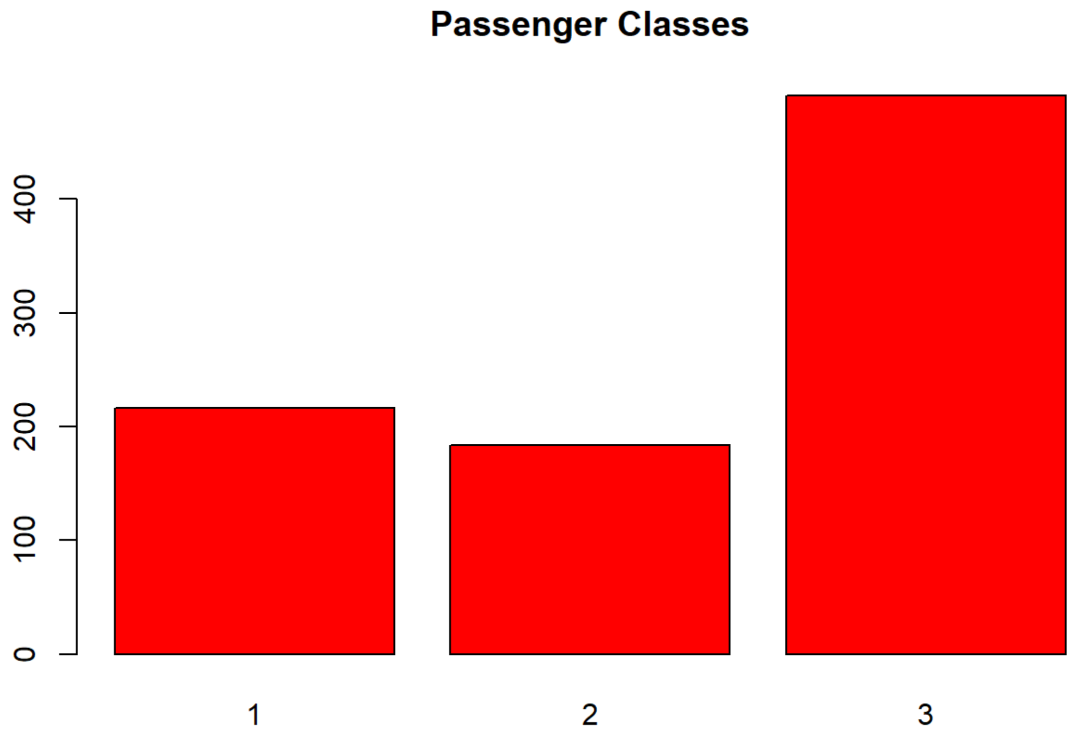
```
## randomForest 4.7-1
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:dplyr':
##
##     combine
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
# Set working directory

# Load data
missing.types <- c("NA", "")
```

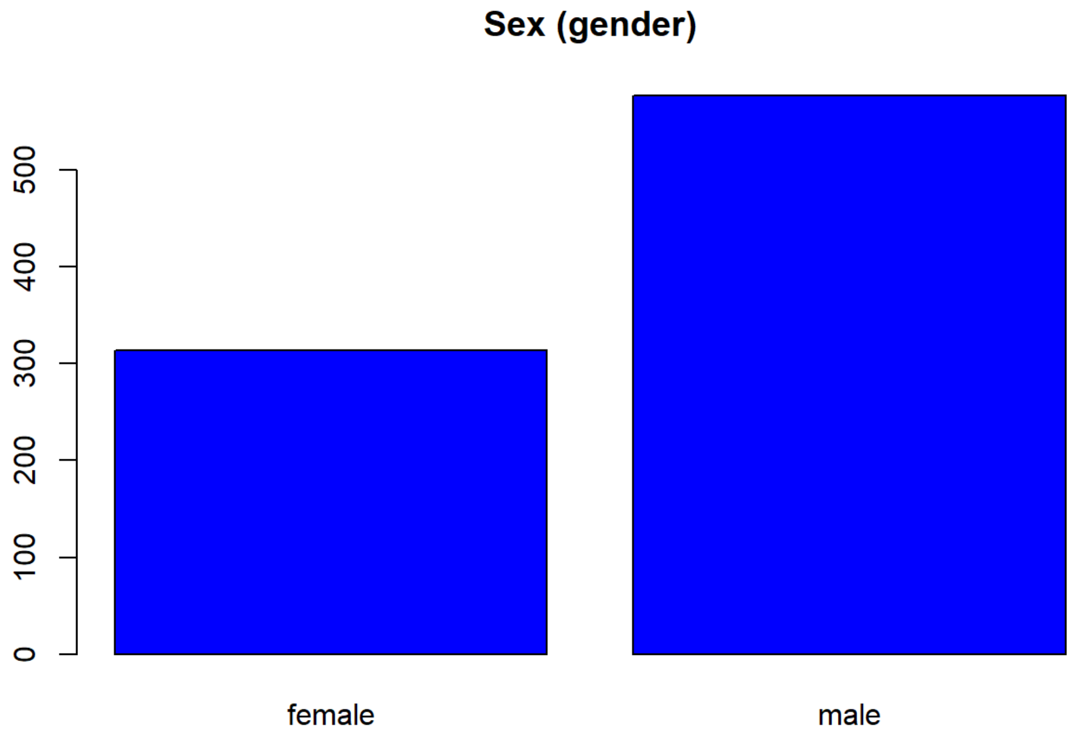
```r
train.data <- read.csv("train.csv", na.strings = missing.types, stringsAsFactors = F)

test.data  <- read.csv("test.csv",  na.strings = missing.types, stringsAsFactors = F)

total.data <- bind_rows(train.data, test.data)

# Explore survival rates

barplot(

  table(train.data$Survived),

  names.arg = c("Perished", "Survived"),

  main = "Survived",

  col = "black"

)
```



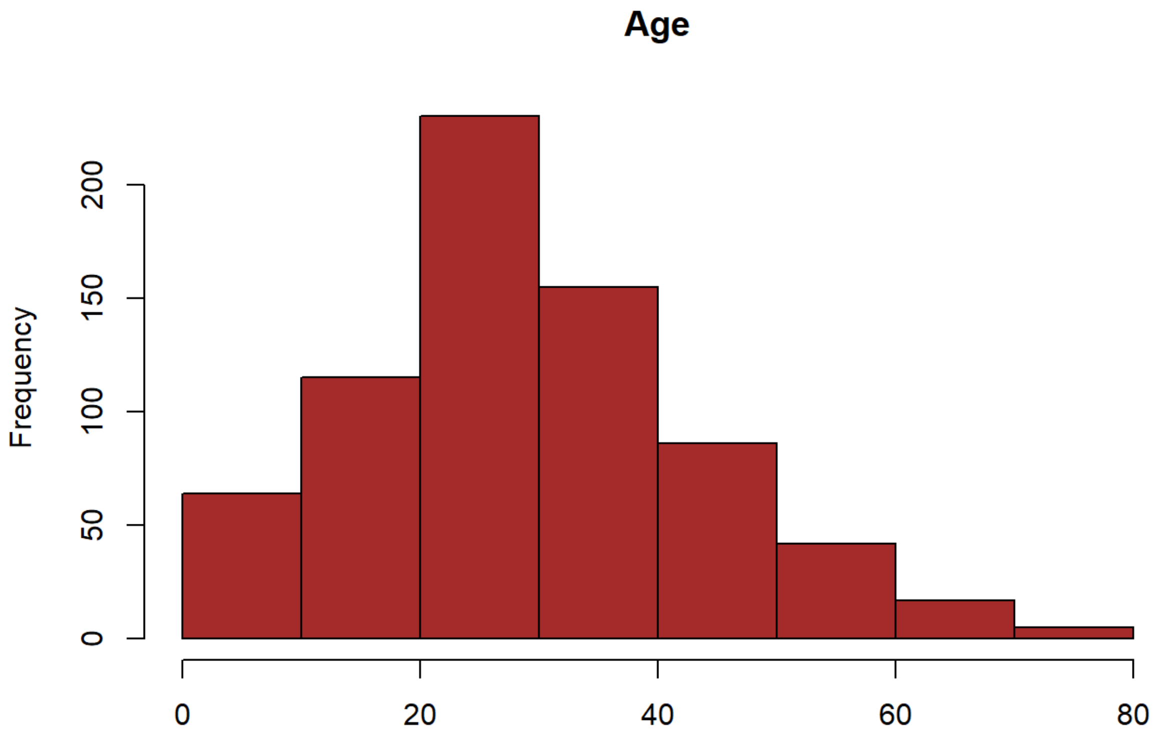
```
# Explore passenger classes
barplot(
  table(train.data$Pclass),
  main = "Passenger Classes",
  col = "red"
)
```



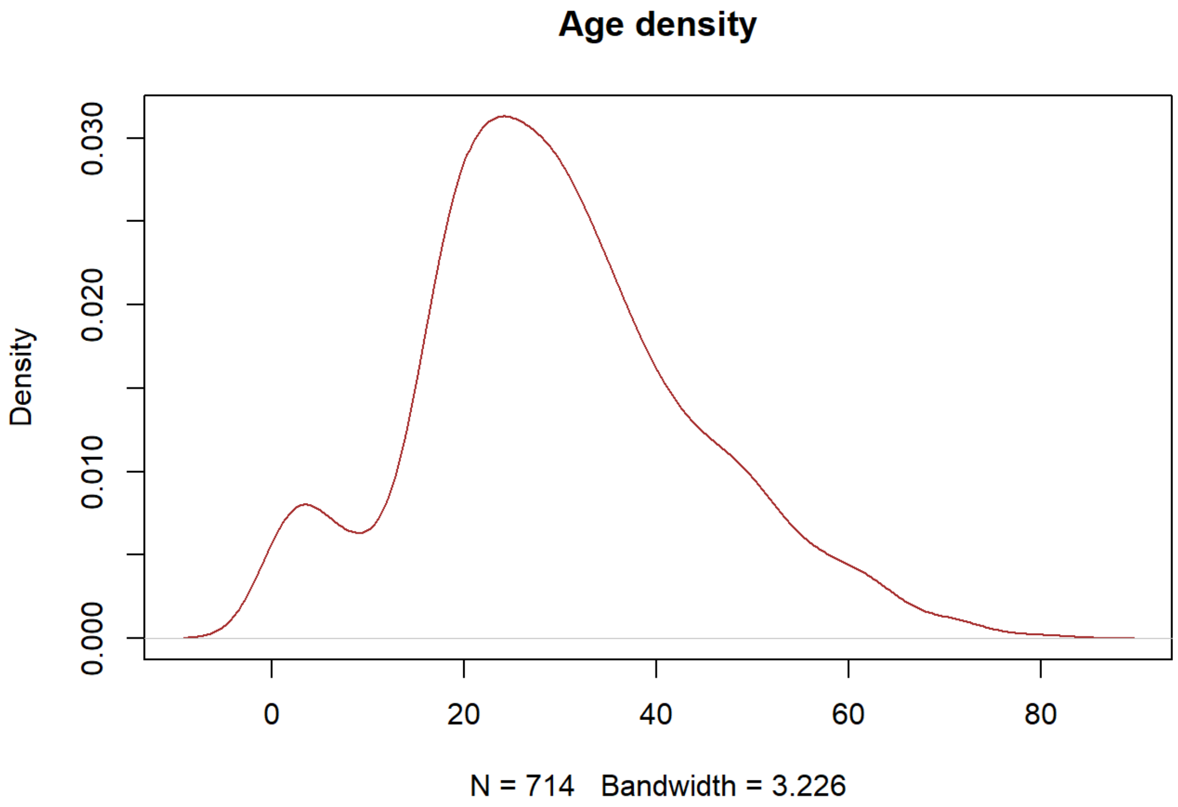
```
# Explore gender repartition
barplot(
  table(train.data$Sex),
  main = "Sex (gender)",
  col = "blue"
)
```



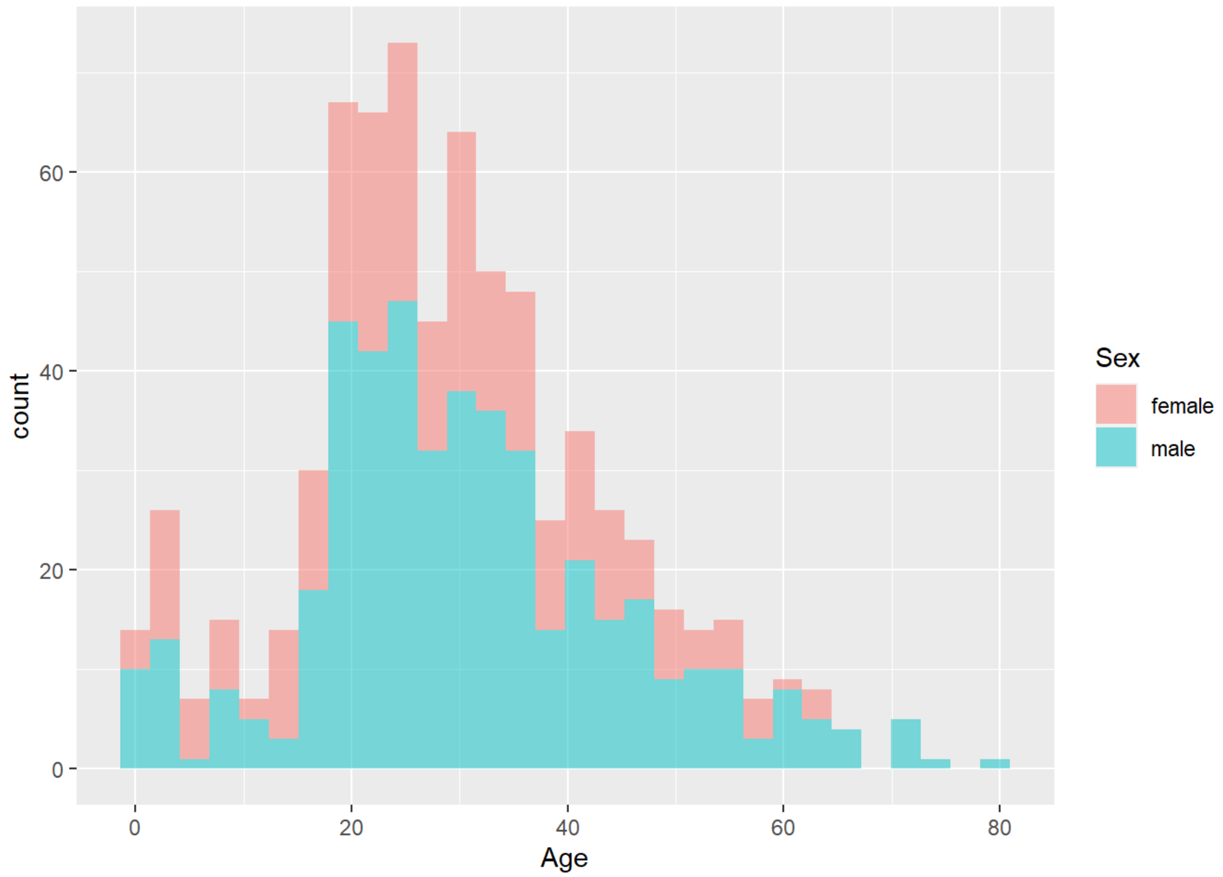
```
# Explore age repartition
hist(
  train.data$Age,
  main = "Age",
  xlab = NULL,
  col = "brown"
)
```



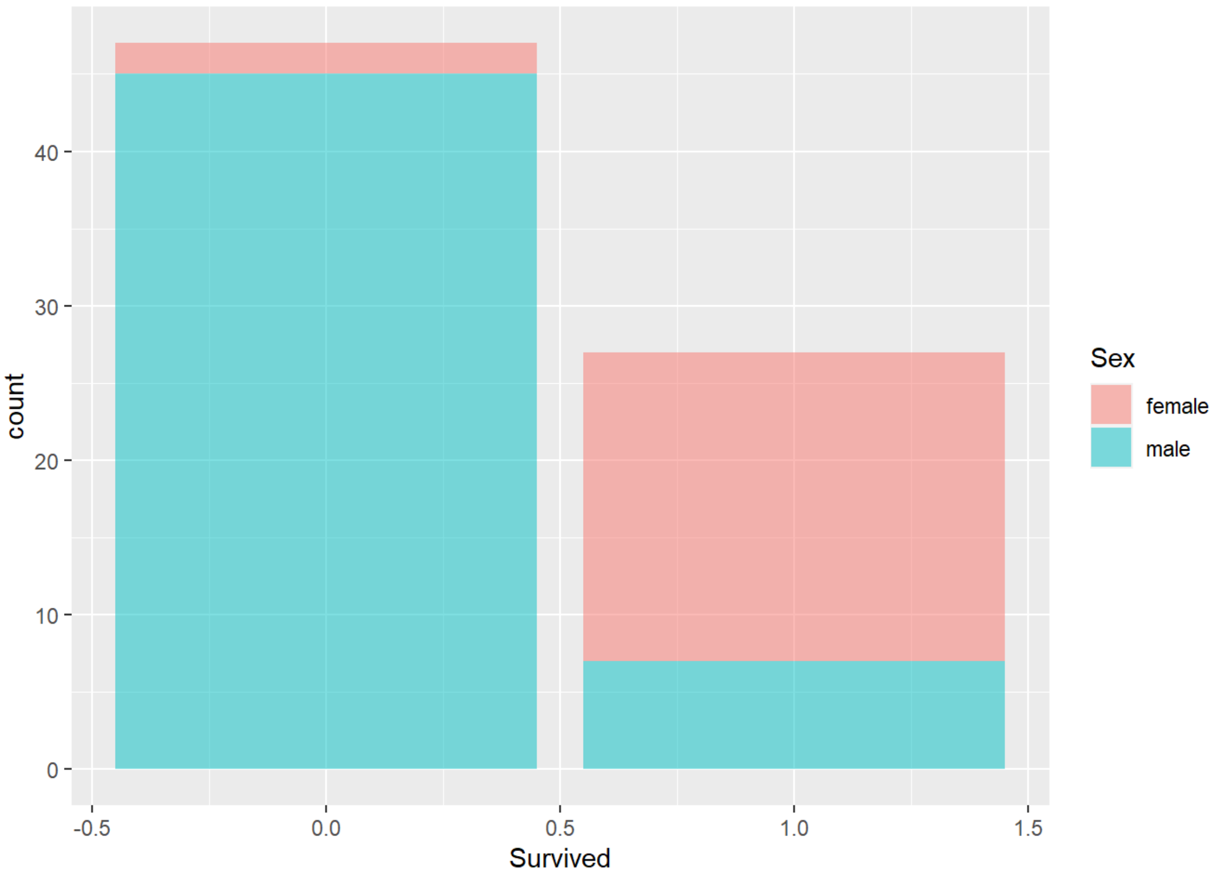
```
d <- density(train.data[!is.na(train.data$Age),]$Age)
plot(d, main = "Age density", xlab = NULL, col = "brown")
```



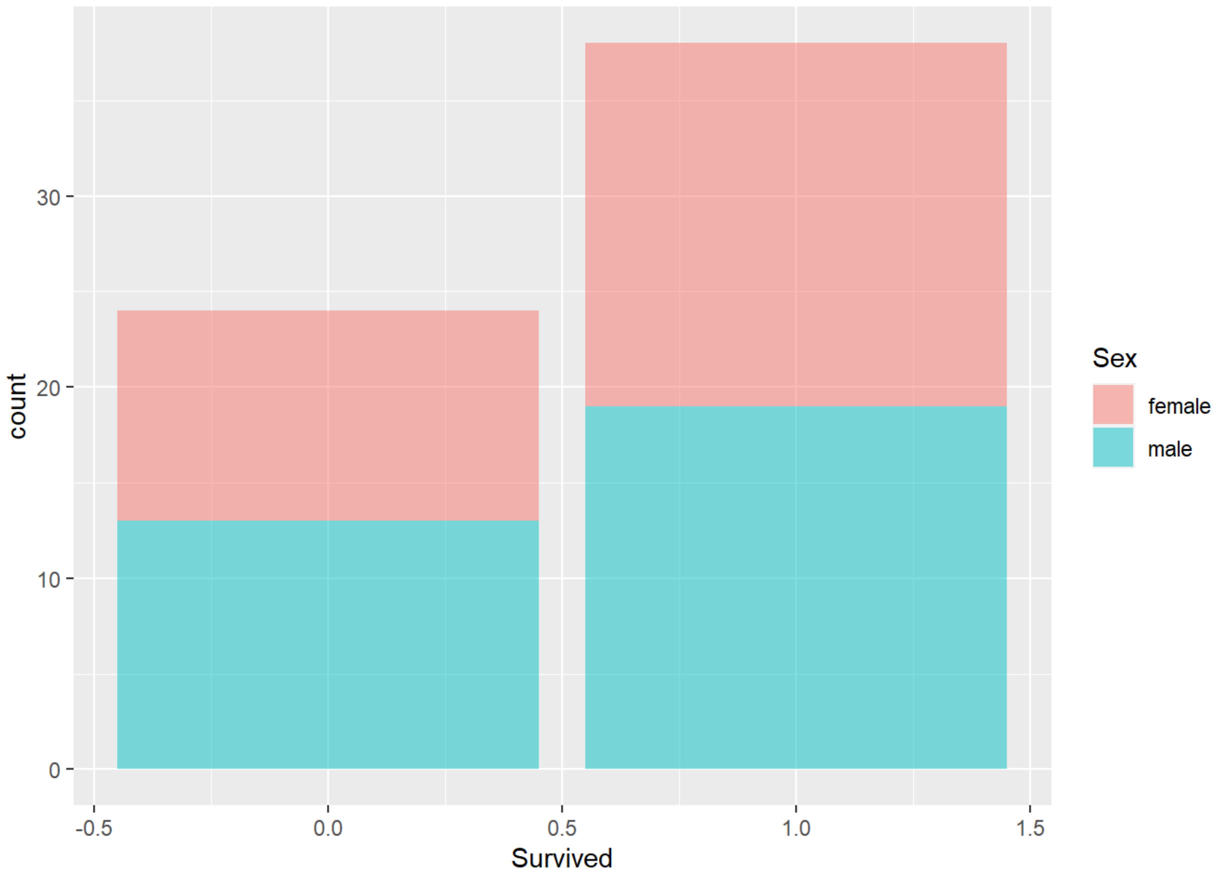
```
# Explore distribution of ages and sex
ggplot(train.data, aes(Age, fill = Sex)) +
  geom_histogram(alpha = 0.5, aes(y = ..count..))
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 177 rows containing non-finite values (stat_bin).
```



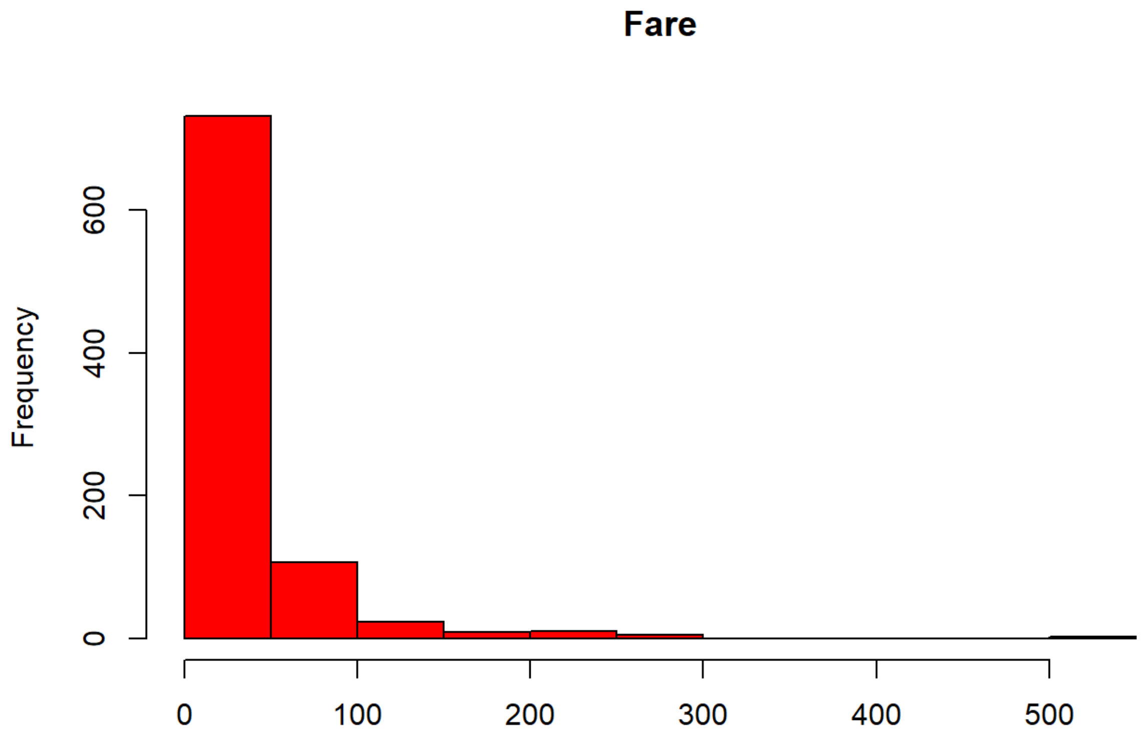
```
ggplot(train.data[train.data$Age >= 50,], aes(Survived, fill = Sex)) +
  geom_bar(alpha = 0.5, aes(y = ..count..))
## Warning: Removed 177 rows containing non-finite values (stat_count).
```



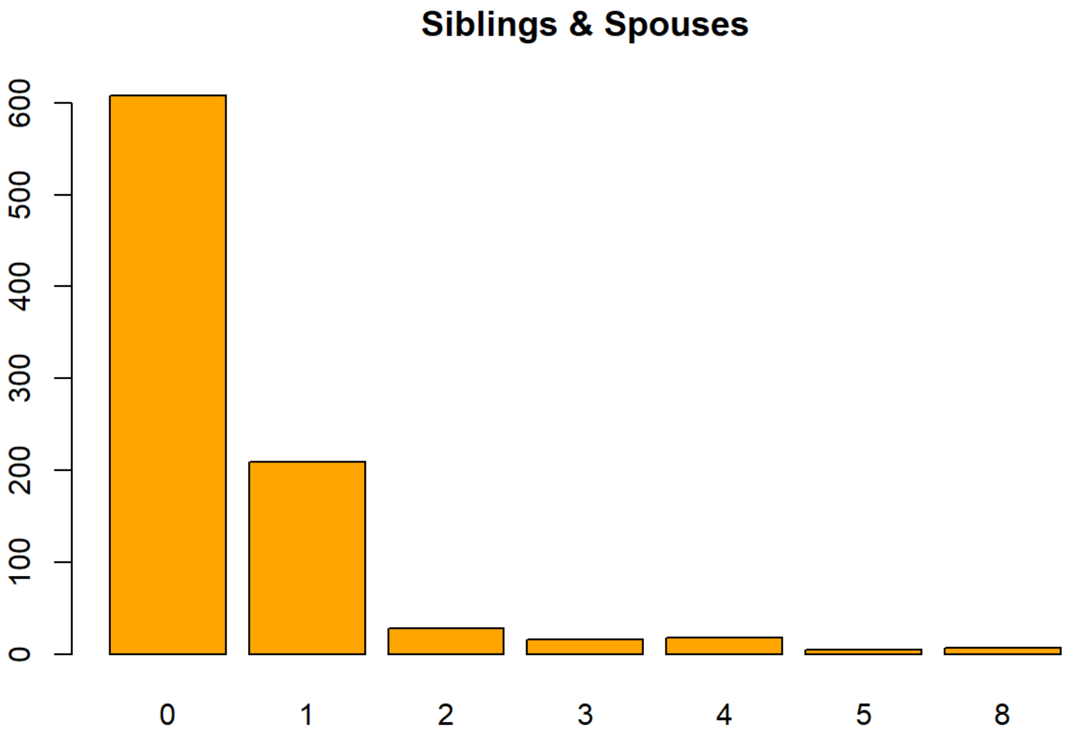
```
ggplot(train.data[train.data$Age < 10,], aes(Survived, fill = Sex)) +
  geom_bar(alpha = 0.5, aes(y = ..count..))
## Warning: Removed 177 rows containing non-finite values (stat_count).
```



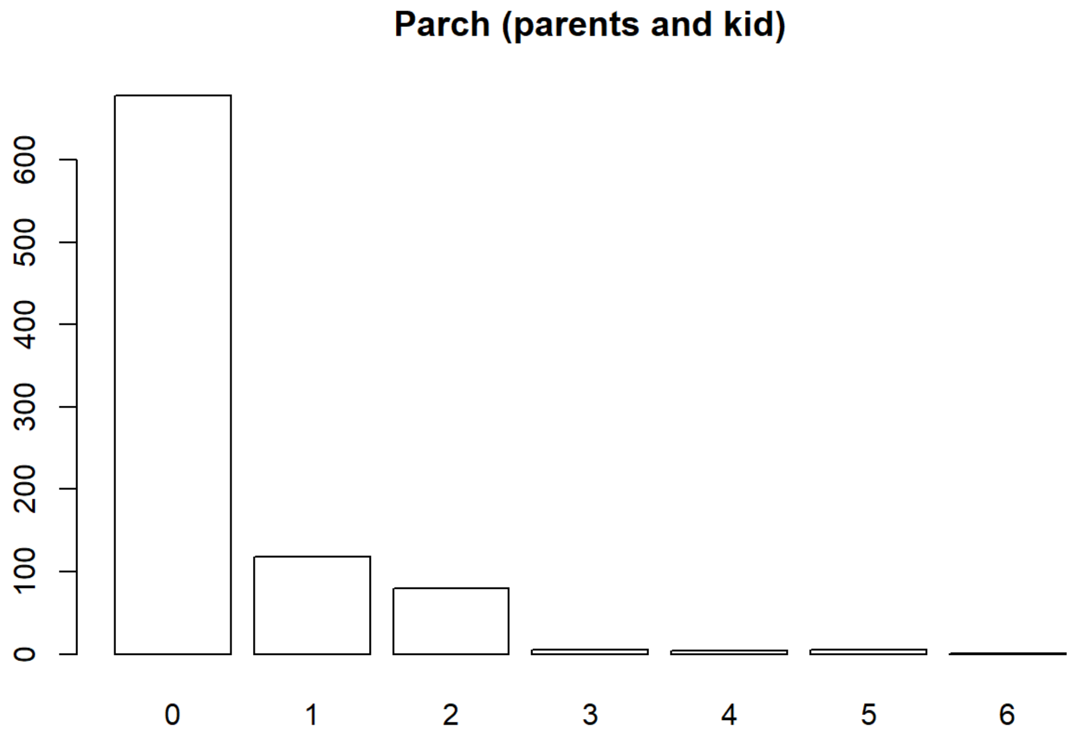
```
# Explore fare paid by passengers
hist(
  train.data$Fare,
  main = "Fare",
  xlab = NULL,
  col = "red"
)
```



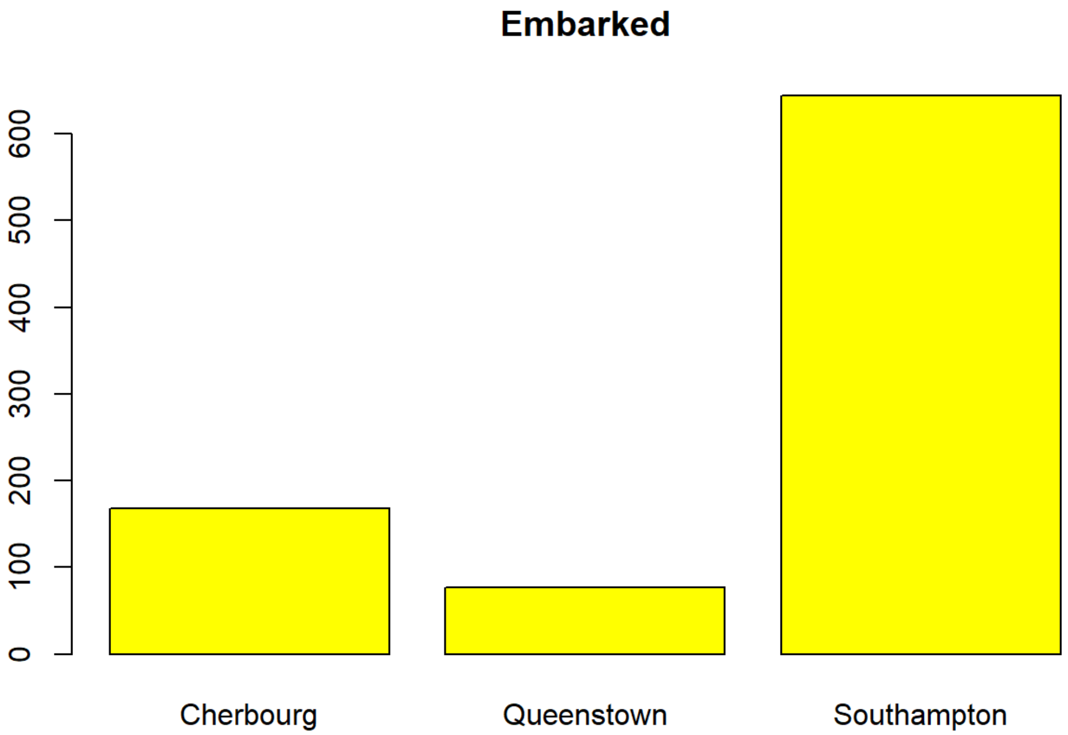
```
# Explore Siblings and spouses repartition
barplot(
  table(train.data$SibSp),
  main = "Siblings & Spouses",
  col = "orange"
)
```



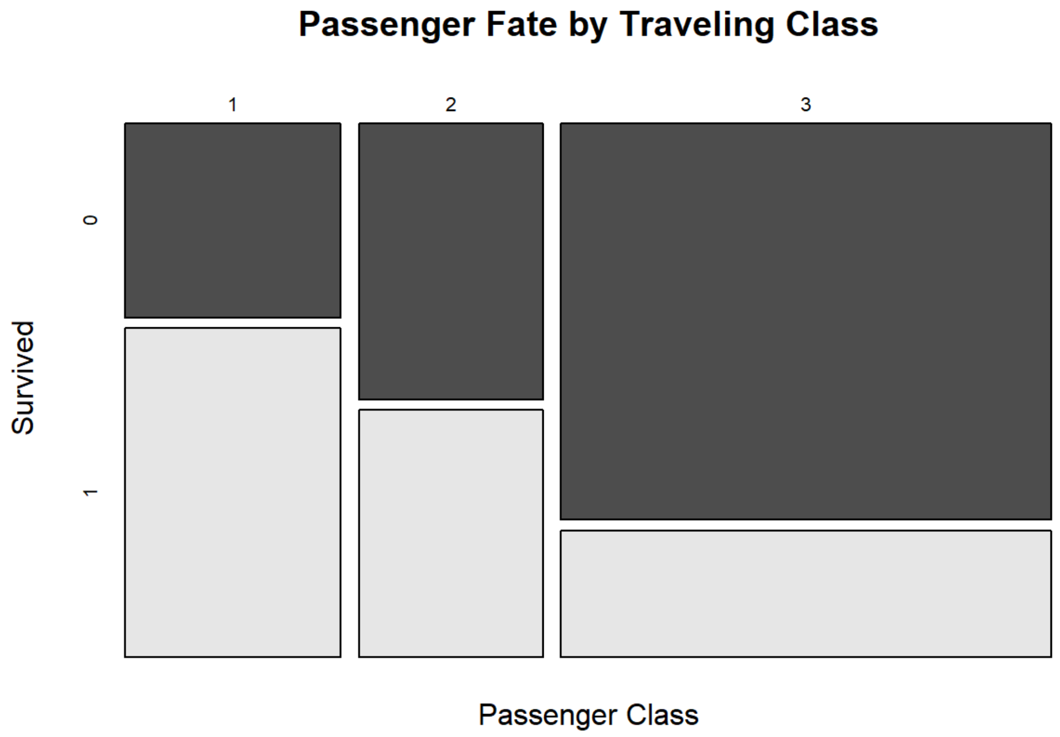
```
# Explore parents and kid repartition
barplot(
  table(train.data$Parch),
  main = "Parch (parents and kid)",
  col = "white"
)
```



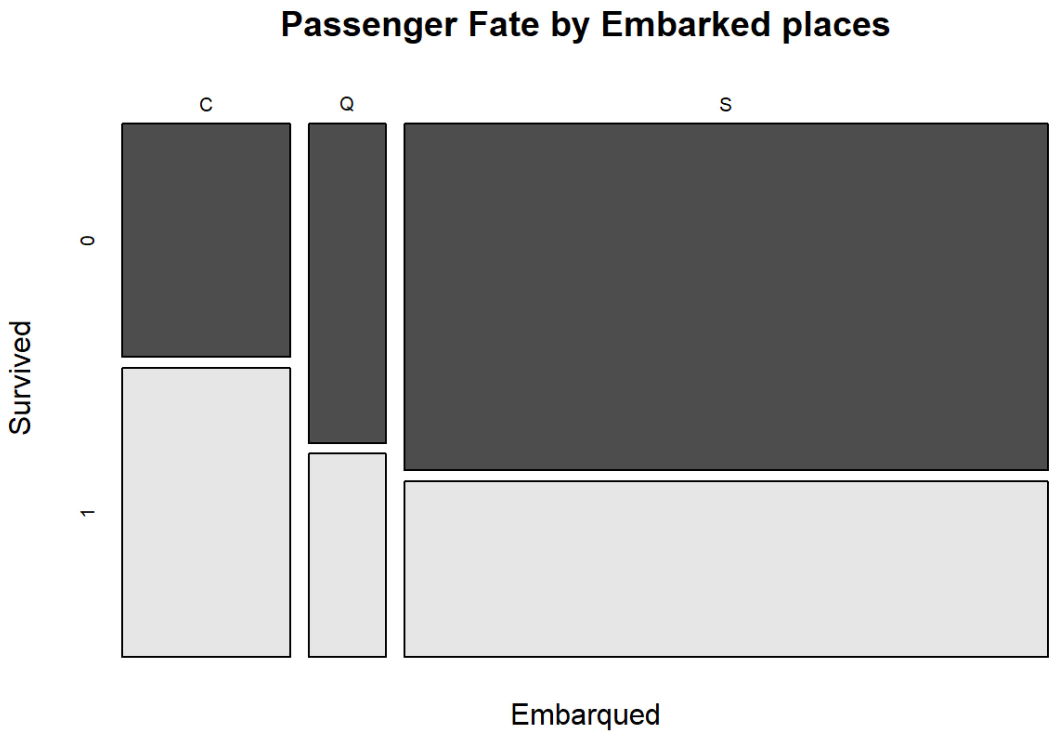
```
# Explore boarding location
barplot(
  table(train.data$Embarked),
  names.arg = c("Cherbourg", "Queenstown", "Southampton"),
  main = "Embarked",
  col = "yellow"
)
```



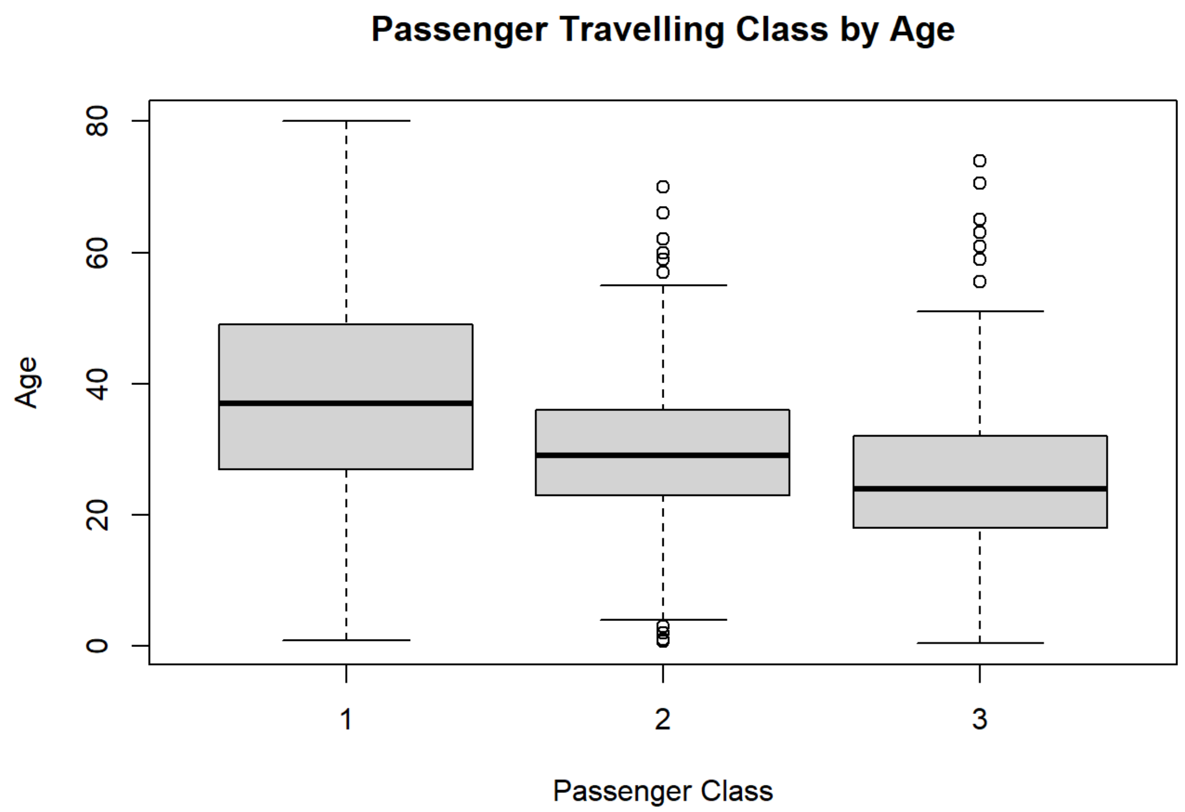
```r
# Explore passenger Fate by Traveling Class
mosaicplot(
  train.data$Pclass ~ train.data$Survived,
  main = "Passenger Fate by Traveling Class",
  shade = FALSE,
  color = TRUE,
  xlab = "Passenger Class",
  ylab = "Survived"
)
```



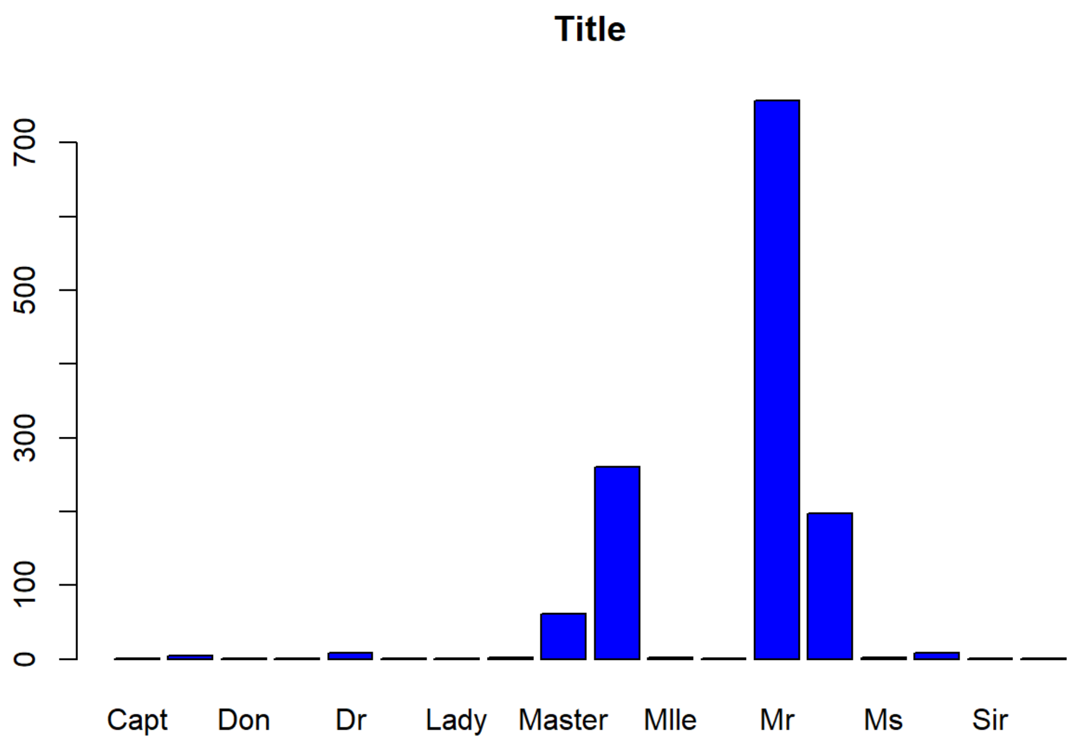
```r
# Explore passenger Fate by Embarked places
mosaicplot(
  train.data$Embarked ~ train.data$Survived,
  main = "Passenger Fate by Embarked places",
  shade = FALSE,
  color = TRUE,
  xlab = "Embarqued",
  ylab = "Survived"
)
```



```
# Explore passenger Travelling Class by Age
boxplot(
  Age ~ Pclass,
  data = train.data,
  main = "Passenger Travelling Class by Age",
  xlab = "Passenger Class",
  ylab = "Age"
)
```

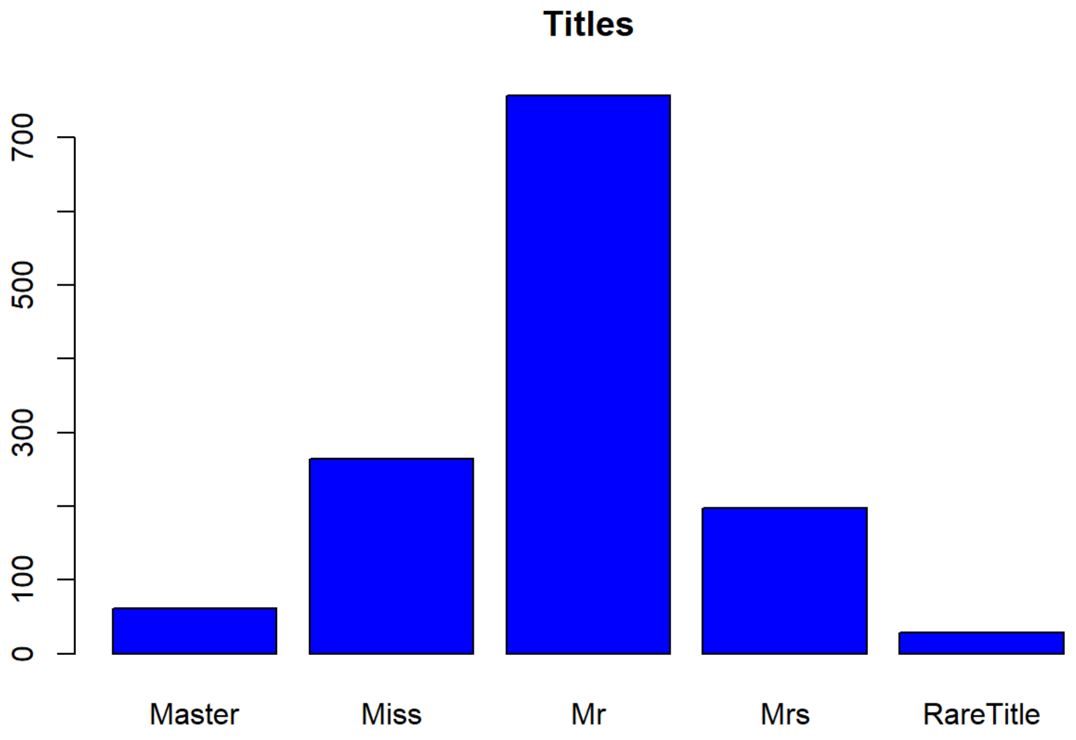


```
#### Work on data ####
### Extract Titles and create new categorical column ###
total.data$Title   <- gsub('(.*, )|(\\..*)', '', total.data$Name)
total.data$Name    <- gsub('(, [a-zA-Z]{,20}. )', ', ', total.data$Name)
total.data$Surname <- gsub('(.*,)', '', total.data$Name)
total.data$Name    <- gsub('(,.*)', '', total.data$Name)

# Display repartition of the titles
barplot(
  table(total.data$Title),
  main = "Title",
  col = "blue"
)
```



```r
# Rare titles management
rare.titles <-
  c(
    'Dona',
    'Lady',
    'the Countess',
    'Capt',
    'Col',
    'Don',
    'Dr',
    'Major',
    'Rev',
    'Sir',
    'Jonkheer'
  )
```

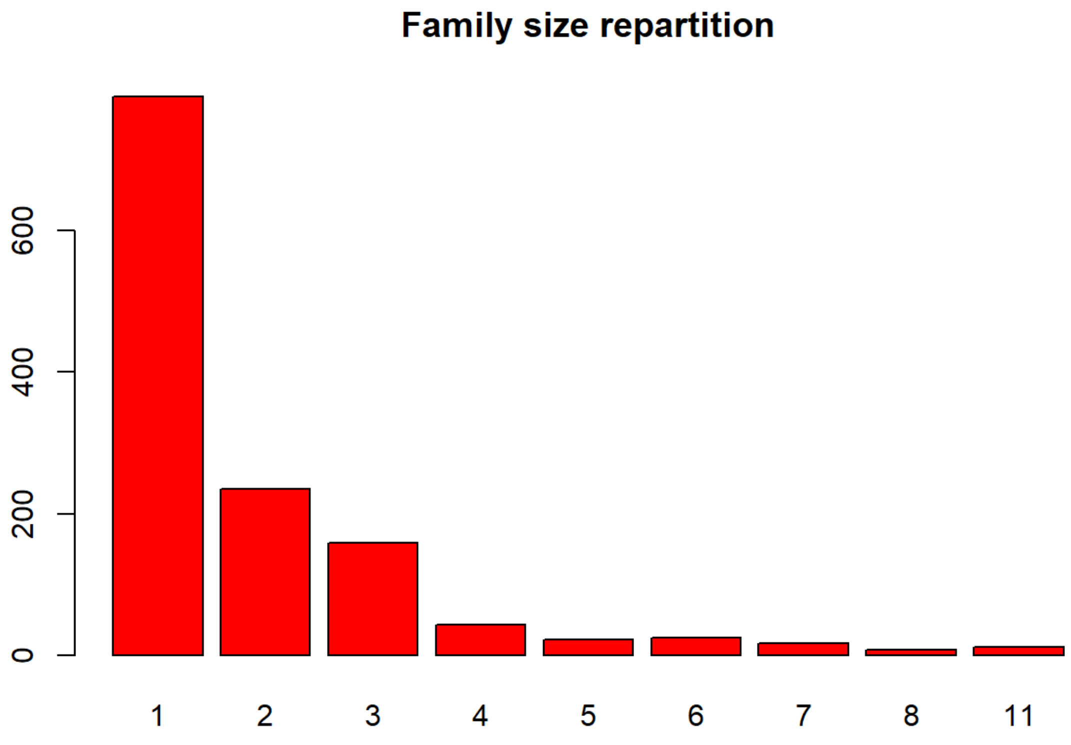
```
# Replace rare titles with correct values
total.data$Title[total.data$Title == 'Mlle']        <- 'Miss'
total.data$Title[total.data$Title == 'Ms']          <- 'Miss'
total.data$Title[total.data$Title == 'Mme']         <- 'Mrs'
total.data$Title[total.data$Title %in% rare.titles]  <- 'RareTitle'

# Display result again
barplot(
  table(total.data$Title),
  main = "Titles",
  col = "blue"
)
```



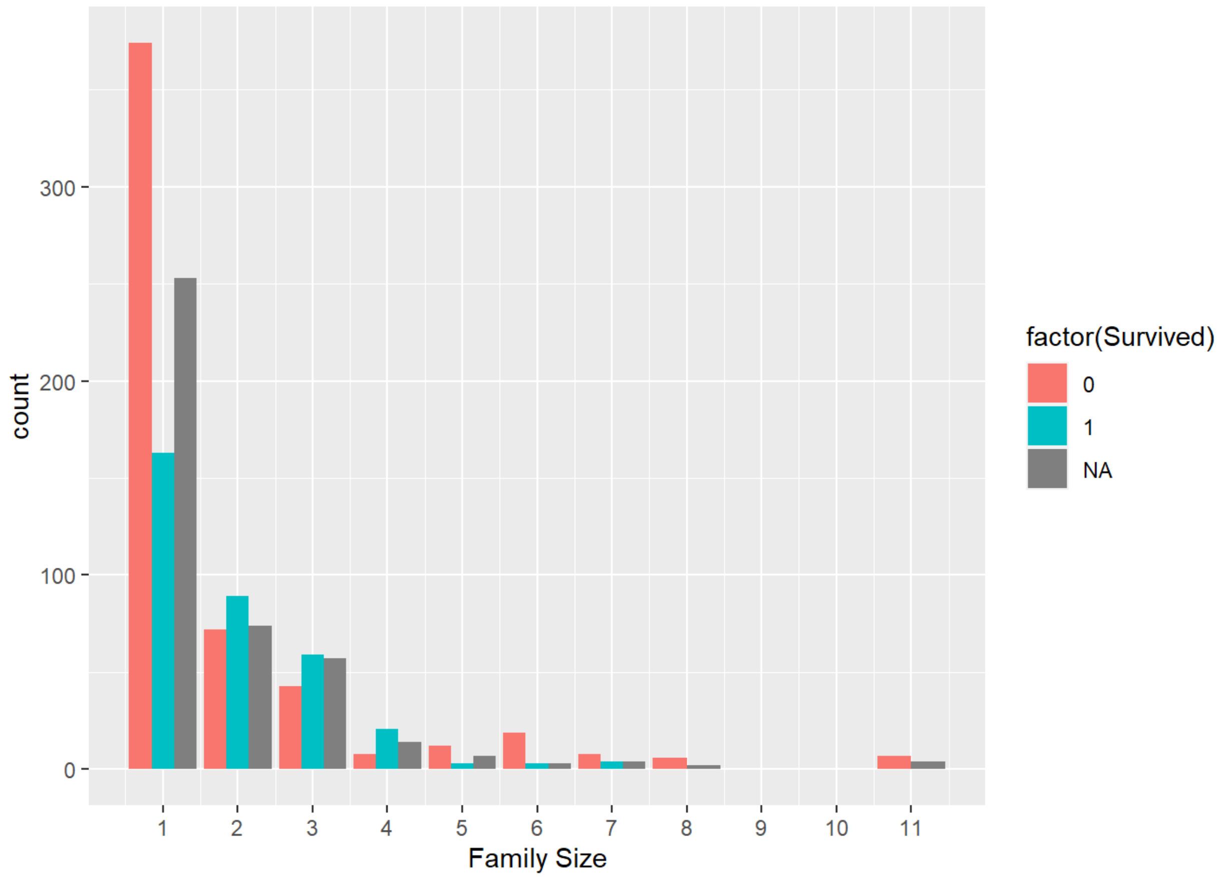
```
table(total.data$Sex, total.data$Title)
```

```
##
##            Master Miss  Mr Mrs RareTitle
##   female      0  264   0 198         4
##   male       61    0 757   0        25
```

```
### Create families description ###
# First, add family size
total.data$FamilySize = total.data$SibSp + total.data$Parch + 1

# Explore family size
barplot(table(total.data$FamilySize),
        main = "Family size repartition",
        col = "red")
```



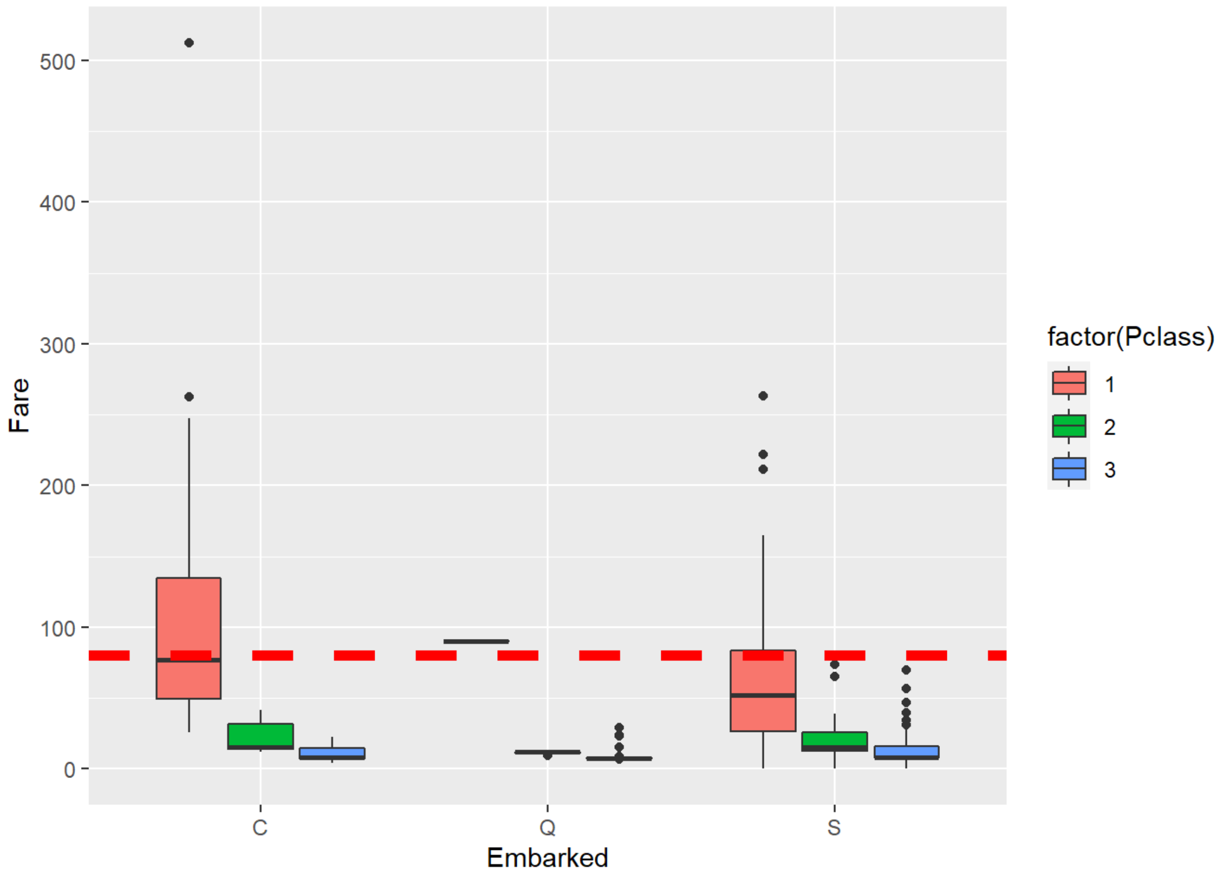
```
# Survival vs Size:
ggplot(total.data, aes(x = FamilySize, fill = factor(Survived))) +
  geom_bar(stat = 'count', position = 'dodge') +
  scale_x_continuous(breaks = c(1:11)) +
  labs(x = 'Family Size')
```



```
# Then, add family size categorical feature
total.data$FamilySizeD[total.data$FamilySize == 1] <-
  'singleton'
total.data$FamilySizeD[total.data$FamilySize > 1 & total.data$FamilySize < 5]
<-
  'small'
total.data$FamilySizeD[total.data$FamilySize >= 5] <-
  'big'

### Deal with missing boarding
View(total.data[is.na(total.data$Embarked),])

# Get rid of our missing passenger IDs
embark.fare <- total.data %>%
  filter(PassengerId != 62 & PassengerId != 830)
```

```
# Use ggplot2 to visualize embarkment, passenger class, & median fare
ggplot(embark.fare, aes(x = Embarked, y = Fare, fill = factor(Pclass))) +
  geom_boxplot() +
  geom_hline(aes(yintercept=80),
             colour='red', linetype='dashed', lwd=2) +
  scale_y_continuous()
```

```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```



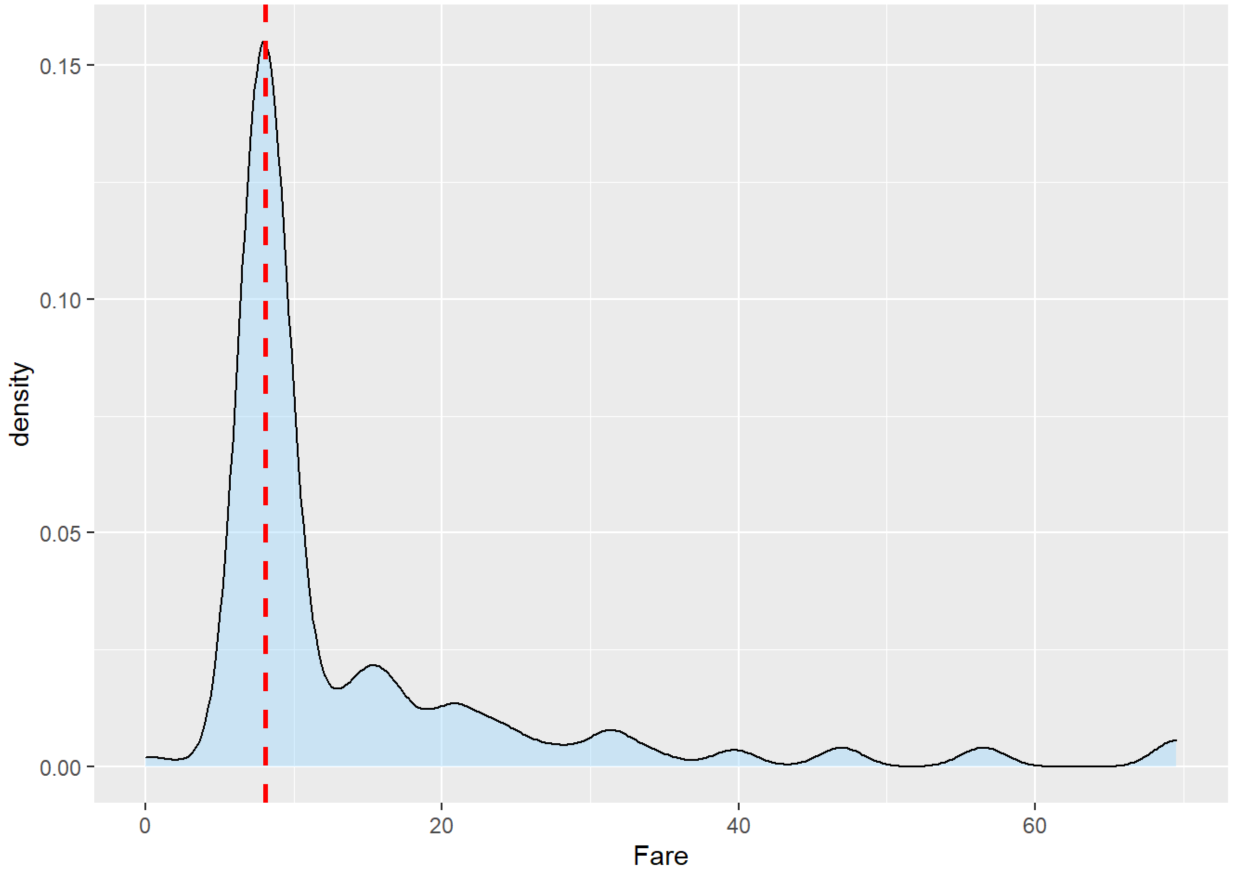
```
total.data[(total.data$PassengerId == 62 | total.data$PassengerId == 830),]$E
mbarked <- "C"


### Deal with guys with several cabins (family?)
total.data$SeveralCabins <- 0
total.data[grepl(" ", total.data$Cabin),]$SeveralCabins <- 1


### Deal with missing fares
```

```
View(total.data[train.data$Fare <= 10 | is.na(total.data$Fare),] %>% arrange(
Fare))
```

```
## Warning in train.data$Fare <= 10 | is.na(total.data$Fare): longer object l
ength

## is not a multiple of shorter object length
```

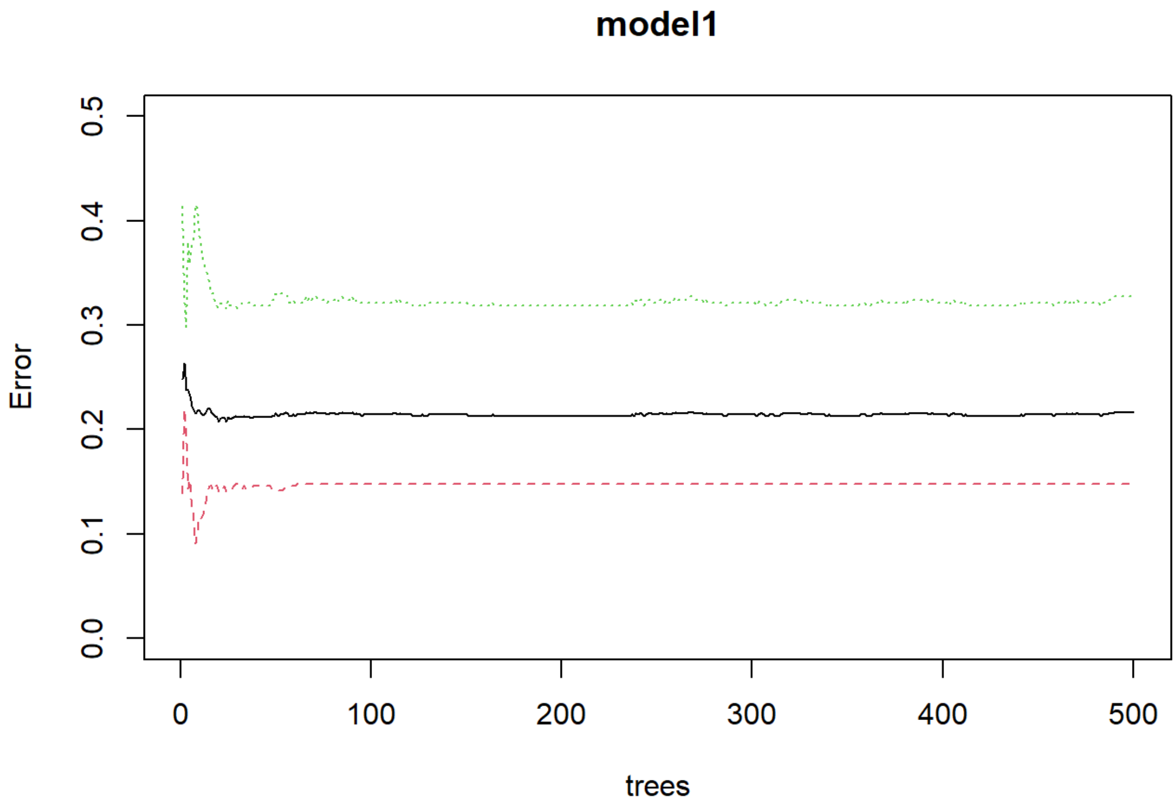
```
View(total.data[is.na(total.data$Fare),] %>% arrange(Fare))

ggplot(total.data[total.data$Pclass == '3' & total.data$Embarked == 'S', ],

       aes(x = Fare)) +

  geom_density(fill = '#99d6ff', alpha=0.4) +

  geom_vline(aes(xintercept=median(Fare, na.rm=T)),

             colour='red', linetype='dashed', lwd=1)
```

```
## Warning: Removed 1 rows containing non-finite values (stat_density).
```

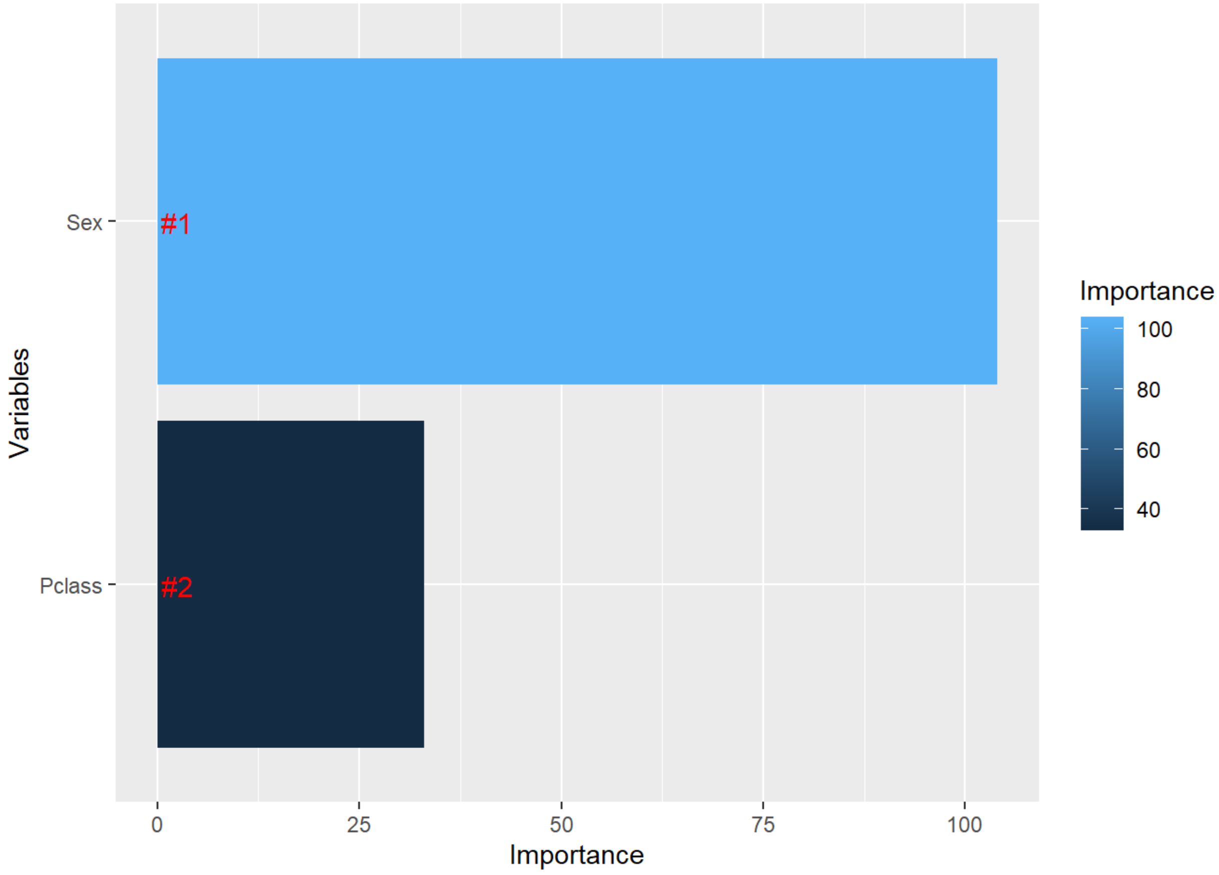


```
total.data$Fare[1044] <- median(total.data[total.data$Pclass == '3' & total.d
ata$Embarked == 'S', ]$Fare, na.rm = TRUE)


### Add some meaningfull features

# Women and children before
```

```r
total.data$WomOrChildren <- 0
total.data[which(total.data$Age < 18 | total.data$Sex == 'female'),]$WomOrChildren <- 1

# First char of cabin is the deck
total.data$Deck <- substring(total.data$Cabin, 1, 1)
total.data$Deck[which(is.na(total.data$Deck))] <- "NoSe"

### Missing ages management
# Method 1: mean age method
mean.age <- total.data[!is.na(total.data$Age),] %>%
  group_by(Sex) %>%
  summarise(Mean = mean(Age),
            Mediane = median(Age))



# Method 2: predictive imputation
# TODO

#### Prediction ####
### Split the train and test data
total.data <- as.data.frame(unclass(total.data))
train <- total.data[1:891,]
test  <- total.data[892:1309,]

### Building the model
# Random seed
set.seed(42)

model1 <- randomForest(factor(Survived) ~ Pclass + Sex, data = train)

plot(model1, ylim=c(0,0.5))
```



```
### Learning from the model created
# Get importance
importance    <- importance(model1)
varImportance <- data.frame(Variables = row.names(importance),
                            Importance = round(importance[ ,'MeanDecreaseGini
'],2))

# Create a rank variable based on importance
rankImportance <- varImportance %>%
  mutate(Rank = paste0('#',dense_rank(desc(Importance))))

# Use ggplot2 to visualize the relative importance of variables
ggplot(rankImportance, aes(x = reorder(Variables, Importance),
                           y = Importance, fill = Importance)) +
  geom_bar(stat='identity') +
  geom_text(aes(x = Variables, y = 0.5, label = Rank),
            hjust=0, vjust=0.55, size = 4, colour = 'red') +
  labs(x = 'Variables') +
  coord_flip()
```



```
### Prediction
# Predict...
prediction <- predict(model1, test)

# Save prediction
solution <- data.frame(PassengerID = test$PassengerId, Survived = prediction)

# Write prediction on disk
write.csv(solution, file = 'model1.csv', row.names = F)
```