SVM Prediction - Titanic Competition

Szymon Pawłowski

4 01 2021

Kaggle Titanic Competition

It is one of the first challenges every ML beginner should dive in. In this competition the main goal is to predict which passengers survived the Titanic shipwreck using given data and creating a ML model. Here the SVM is used for prediction. In another file on GitHub "titanic_kaggle_competition.R" a whole work containing many approaches can be found.

The data

The data has been split into two groups: training set and test set. Columns they contain: * Survival - did passenger survive - 0=no, 1=yes * Pclass - ticket class - 1=1st, 2=2nd, 3=3rd * Name * Sex - sex - m/f * Age - age in years * Sibsp - number of siblings / spouses aboard the Titanic * Parch - number of parents / children aboard the Titanic * Ticket - ticket number * Fare - passenger fare * Cabin - cabin number * Embarked - port of embarkation - C=Cherbourg, Q=Quennstown, S=Southampton

Libraries

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.0.3
```

```
library(ggplot2)
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.0.3

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(fitdistrplus)
```

```
## Loading required package: MASS

## Warning: package 'MASS' was built under R version 4.0.3

## Loading required package: survival
```

Loading data

```
df_t <- read.csv("test.csv")
df <- read.csv("train.csv")
```

First look at data

```
head(df)
```

```
##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      3
## 4           4        1      1
## 5           5        0      3
## 6           6        0      3
##                                                    Name    Sex Age SibSp Parch
## 1                            Braund, Mr. Owen Harris   male  22     1     0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1     0
## 3                             Heikkinen, Miss. Laina female  26     0     0
## 4       Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1     0
## 5                           Allen, Mr. William Henry   male  35     0     0
## 6                                   Moran, Mr. James   male  NA     0     0
##             Ticket    Fare Cabin Embarked
## 1        A/5 21171  7.2500              S
## 2         PC 17599 71.2833   C85        C
## 3 STON/O2. 3101282  7.9250              S
## 4           113803 53.1000  C123        S
## 5           373450  8.0500              S
## 6           330877  8.4583              Q
```

```
tail(df)
```

```
##     PassengerId Survived Pclass                                     Name    Sex
## 886         886        0      3     Rice, Mrs. William (Margaret Norton) female
## 887         887        0      2                    Montvila, Rev. Juozas   male
## 888         888        1      1             Graham, Miss. Margaret Edith female
## 889         889        0      3 Johnston, Miss. Catherine Helen "Carrie" female
## 890         890        1      1                    Behr, Mr. Karl Howell   male
## 891         891        0      3                      Dooley, Mr. Patrick   male
##     Age SibSp Parch     Ticket   Fare Cabin Embarked
## 886  39     0     5     382652 29.125              Q
## 887  27     0     0     211536 13.000              S
## 888  19     0     0     112053 30.000   B42        S
## 889  NA     1     2 W./C. 6607 23.450              S
## 890  26     0     0     111369 30.000  C148        C
## 891  32     0     0     370376  7.750              Q
```

```
summary(df)
```

```
##   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
##
```

- Survival - need to transfer into factor
- Pclass - seems okay
- Name - could extract title from it
- Sex - need to transfer into numeric factor
- Age - contains missing values, need to replace them
- Sibsp and Parch - can get information of family size from here
- Ticket and Cabin - hard to get information so we will drop it for now
- Fare - seems okay, some NA's
- Embarked - need to transfer into numeric factor

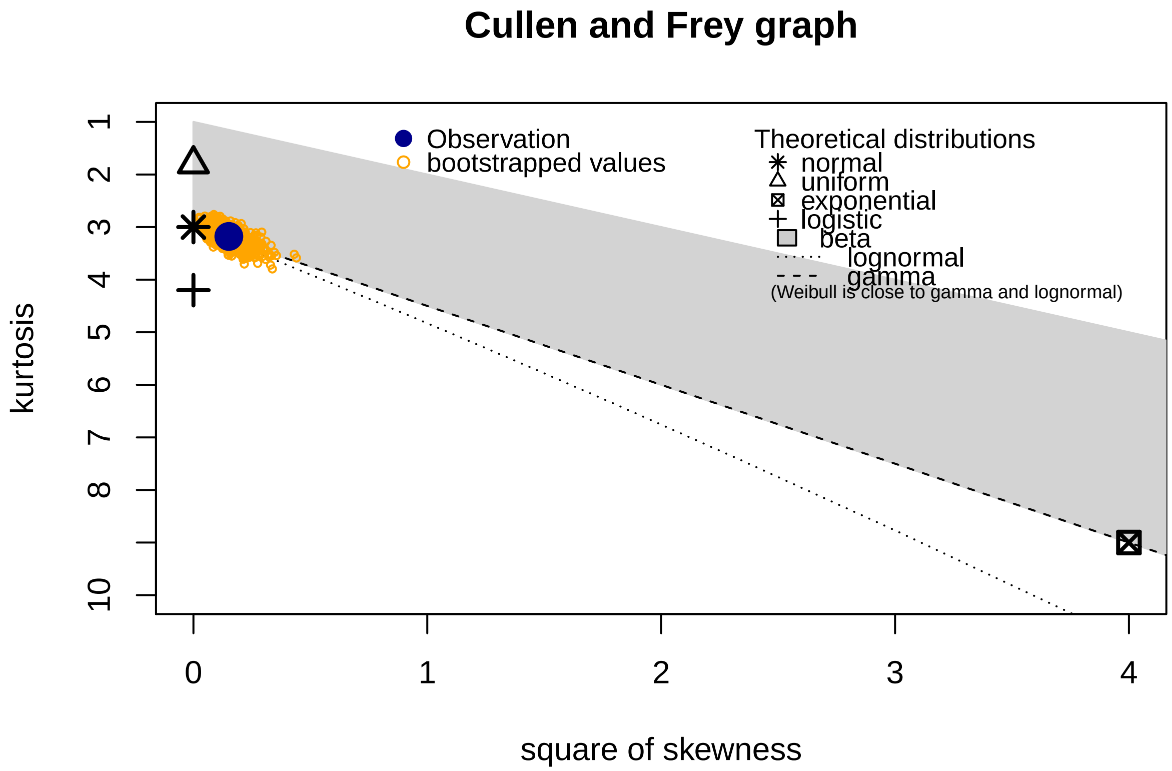
Feature Engineering

Age

Finding the proper distribution.

```
age<-df[is.na(df$Age)==FALSE, ]$Age
age_t<-df_t[is.na(df_t$Age)==FALSE, ]$Age

descdist(age,discrete=FALSE,boot=1000)
```

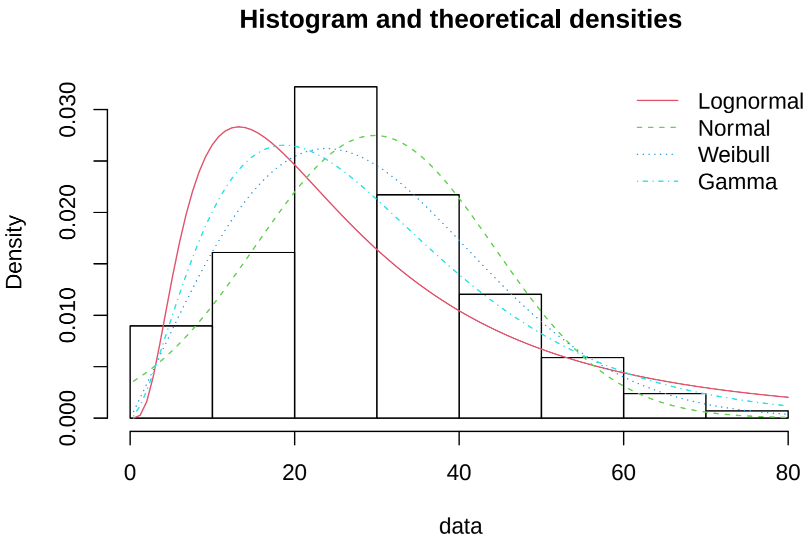


```
## summary statistics
## ------
## min:  0.42   max:  80
## median:  28
## mean:  29.69912
## estimated sd:  14.5265
## estimated skewness:  0.3891078
## estimated kurtosis:  3.178274
```

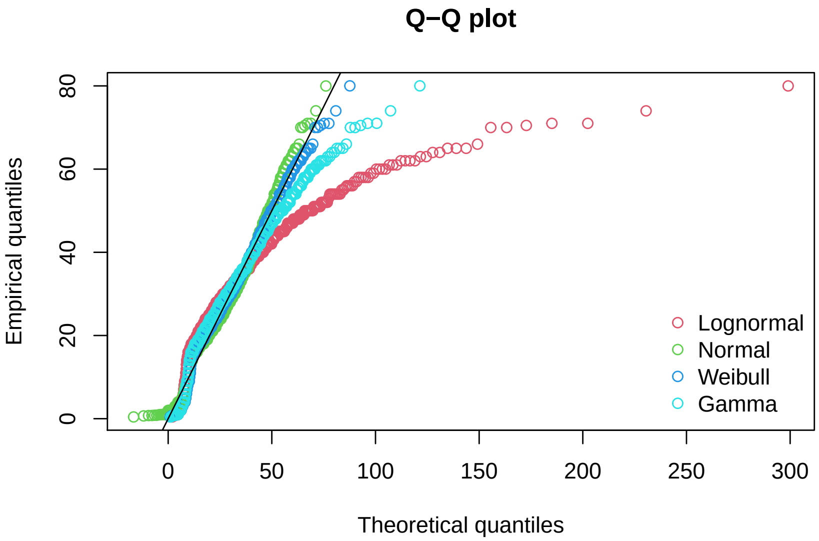
Looks like Normal, Lognormal or Gamma distribution but Weibull will also be checked.

```r
fln<-fitdist(age,"lnorm")
fn<-fitdist(age,"norm")
fw<-fitdist(age,"weibull")
fg<-fitdist(age,"gamma")
plot.legenda<-c("Lognormal","Normal","Weibull","Gamma")
```

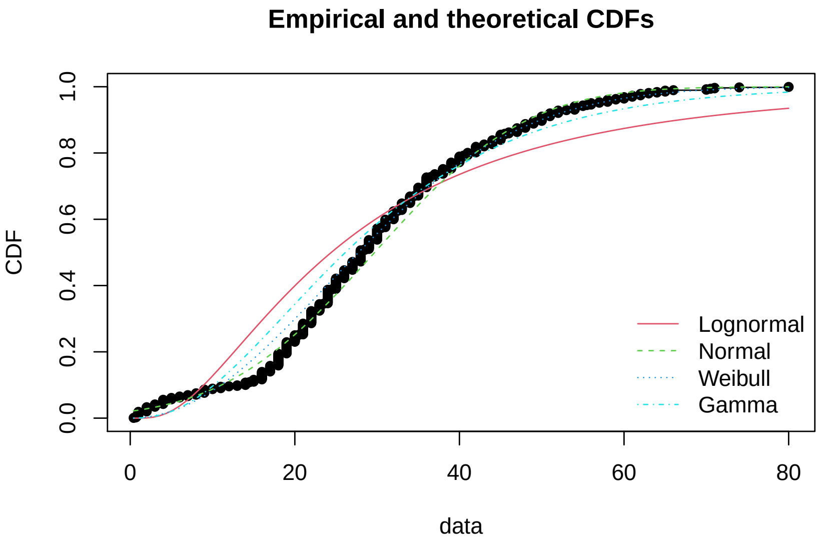
Plotting the density.



Plotting the Q-Q plot.



Plotting the cumulative distributant plot.



Looking at information criteria.

```
dists <- data.frame("Normal"=c(fn$loglik,fn$aic),"Gamma"=c(fg$loglik, fg$aic),
                    "Weibull"=c(fw$loglik,fw$aic), "Lognormal"=c(fln$loglik,fln$aic),
                    row.names = c("Loglikelihood","AIC"))
dists
```

```
##                  Normal     Gamma   Weibull Lognormal
## Loglikelihood -2923.267 -2981.790 -2932.530 -3121.256
## AIC            5850.535  5967.581  5869.059  6246.512
```

Depending on the results we assume that age is normally distributed Now, we will generate from this distribution random number in the place of NA's.

```
summary(df$Age) #177 NA's
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    0.42   20.12   28.00   29.70   38.00   80.00     177
```

```
set.seed(1)
df[is.na(df$Age)==TRUE, ]$Age <- round(rnorm(177, mean(age),sd(age)))

summary(df_t$Age) #86 NA's
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    0.17   21.00   27.00   30.27   39.00   76.00      86
```

```
df_t[is.na(df_t$Age)==TRUE, ]$Age <- round(rnorm(86, mean(age_t),sd(age_t)))
df_t$Age <- round(df_t$Age)
```

Sex

```
df[df$Sex=="male","Sex"]<-0
df[df$Sex=="female","Sex"]<-1
```

```
df$Sex<-as.numeric(df$Sex)

df_t[df_t$Sex=="male","Sex"]<-0
df_t[df_t$Sex=="female","Sex"]<-1
df_t$Sex<-as.numeric(df_t$Sex)
```

Ticket and Cabin

```
df$Cabin<-NULL
df$Ticket<-NULL

df_t$Cabin<-NULL
df_t$Ticket<-NULL
```

Name

```
df["Title"]<-sub("\\s.*","",sub(".*,\\s","",df$Name))
df["Title"]<-factor(df$Title)
meaning<-unique(df$Title)
df$Title<-as.numeric(df$Title)
numeric<-unique(df$Title)
meaning <- data.frame(numeric=numeric, meaning=meaning)
t(meaning)
```

```
##         [,1]  [,2]   [,3]    [,4]      [,5]   [,6]   [,7]  [,8]   [,9]
## numeric "12"  "13"   " 9"    " 8"      " 3"   "15"   " 4"  "11"   "14"
## meaning "Mr." "Mrs." "Miss." "Master." "Don." "Rev." "Dr." "Mme." "Ms."
##         [,10]    [,11]   [,12]  [,13]   [,14]  [,15]   [,16] [,17]
## numeric " 7"     " 6"    "16"   "10"    " 2"   " 1"    "17"  " 5"
## meaning "Major." "Lady." "Sir." "Mlle." "Col." "Capt." "the" "Jonkheer."
```

```
df[df$Title!=9 & df$Title!=12 & df$Title!=13 & df$Title!=14,]$Title <-4
df[df$Title==12,]$Title <- 1
df[df$Title==13,]$Title <- 2
df[df$Title==9 | df$Title==14,]$Title <- 3


df_t["Title"]<-sub("\\s.*","",sub(".*,\\s","",df_t$Name))
df_t["Title"]<-factor(df_t$Title)
meaning_t<-unique(df_t$Title)
df_t$Title<-as.numeric(df_t$Title)
numeric_t<-unique(df_t$Title)
meaning_t <- data.frame(numeric=numeric_t, meaning=meaning_t)
t(meaning_t)
```

```
##         [,1]  [,2]   [,3]    [,4]      [,5]  [,6]   [,7]   [,8]  [,9]
## numeric "6"   "7"    "5"     "4"       "8"   "1"    "9"    "3"   "2"
## meaning "Mr." "Mrs." "Miss." "Master." "Ms." "Col." "Rev." "Dr." "Dona."
```

```
df_t[df_t$Title!=6 & df_t$Title!=7& df_t$Title!=5& df_t$Title!=8,]$Title <-4
df_t[df_t$Title==6,]$Title <- 1
df_t[df_t$Title==7,]$Title <- 2
df_t[df_t$Title==5 | df_t$Title==8,]$Title <- 3
```

```r
df$Name <- NULL
df_t$Name <- NULL
```

Embarked

```r
df[df$Embarked=="C",]$Embarked <- 1
df[df$Embarked=="Q",]$Embarked <- 2
df[df$Embarked=="S",]$Embarked <- 3
df$Embarked<-as.numeric(df$Embarked)

df_t[df_t$Embarked=="C",]$Embarked <- 1
df_t[df_t$Embarked=="Q",]$Embarked <- 2
df_t[df_t$Embarked=="S",]$Embarked <- 3
df_t$Embarked<-as.numeric(df_t$Embarked)
```

We got 2 NA's, will replace them with random number

```r
df[is.na(df$Embarked)==TRUE,]$Embarked <- sample(c(1,2,3),2)
summary(df)
```

```
##   PassengerId       Survived          Pclass           Sex        
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Min.   :0.0000  
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   1st Qu.:0.0000  
##  Median :446.0   Median :0.0000   Median :3.000   Median :0.0000  
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309   Mean   :0.3524  
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000   3rd Qu.:1.0000  
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000   Max.   :1.0000  
##       Age           SibSp           Parch             Fare       
##  Min.   :-2.0   Min.   :0.000   Min.   :0.0000   Min.   :  0.00  
##  1st Qu.:21.0   1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:  7.91  
##  Median :28.5   Median :0.000   Median :0.0000   Median : 14.45  
##  Mean   :29.8   Mean   :0.523   Mean   :0.3816   Mean   : 32.20  
##  3rd Qu.:38.0   3rd Qu.:1.000   3rd Qu.:0.0000   3rd Qu.: 31.00  
##  Max.   :80.0   Max.   :8.000   Max.   :6.0000   Max.   :512.33  
##     Embarked         Title      
##  Min.   :1.000   Min.   :1.000  
##  1st Qu.:2.000   1st Qu.:1.000  
##  Median :3.000   Median :1.000  
##  Mean   :2.533   Mean   :1.773  
##  3rd Qu.:3.000   3rd Qu.:3.000  
##  Max.   :3.000   Max.   :4.000
```

Fare

Getting rid off NA in Fare test data

```r
summary(df_t)
```

```
##   PassengerId         Pclass           Sex              Age        
##  Min.   : 892.0   Min.   :1.000   Min.   :0.0000   Min.   :-11.00  
##  1st Qu.: 996.2   1st Qu.:1.000   1st Qu.:0.0000   1st Qu.: 21.00  
##  Median :1100.5   Median :3.000   Median :0.0000   Median : 28.00  
##  Mean   :1100.5   Mean   :2.266   Mean   :0.3636   Mean   : 30.25  
##  3rd Qu.:1204.8   3rd Qu.:3.000   3rd Qu.:1.0000   3rd Qu.: 39.00  
##  Max.   :1309.0   Max.   :3.000   Max.   :1.0000   Max.   : 76.00
```

```
##
##      SibSp            Parch             Fare             Embarked
##  Min.   :0.0000   Min.   :0.0000   Min.   :  0.000   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:  7.896   1st Qu.:2.000
##  Median :0.0000   Median :0.0000   Median : 14.454   Median :3.000
##  Mean   :0.4474   Mean   :0.3923   Mean   : 35.627   Mean   :2.402
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.: 31.500   3rd Qu.:3.000
##  Max.   :8.0000   Max.   :9.0000   Max.   :512.329   Max.   :3.000
##                                    NA's   :1
##      Title
##  Min.   :1.000
##  1st Qu.:1.000
##  Median :1.000
##  Mean   :1.744
##  3rd Qu.:3.000
##  Max.   :4.000
##
```

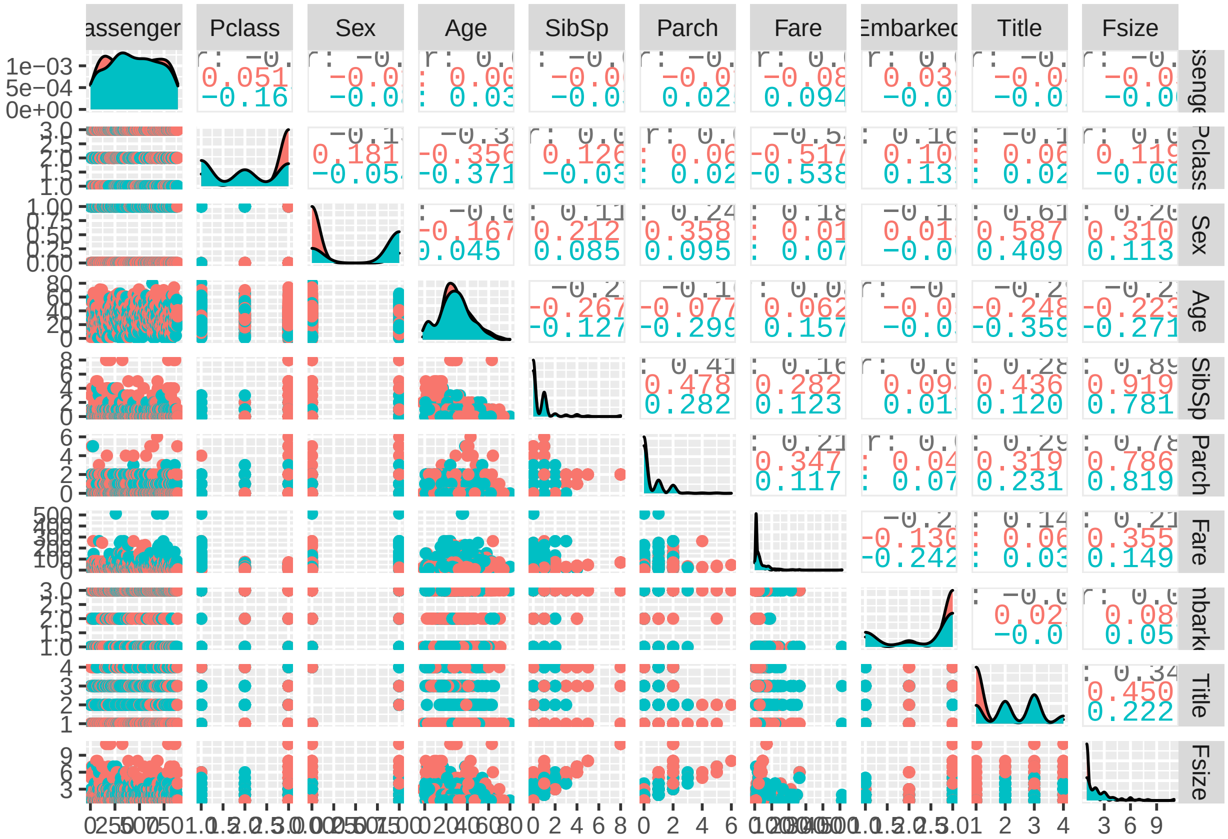
```
fare <- df_t[is.na(df_t$Fare)==FALSE,]$Fare
df_t[is.na(df_t$Fare)==TRUE,]$Fare<-rnorm(1,mean(fare),sd(fare))
```

Family Size

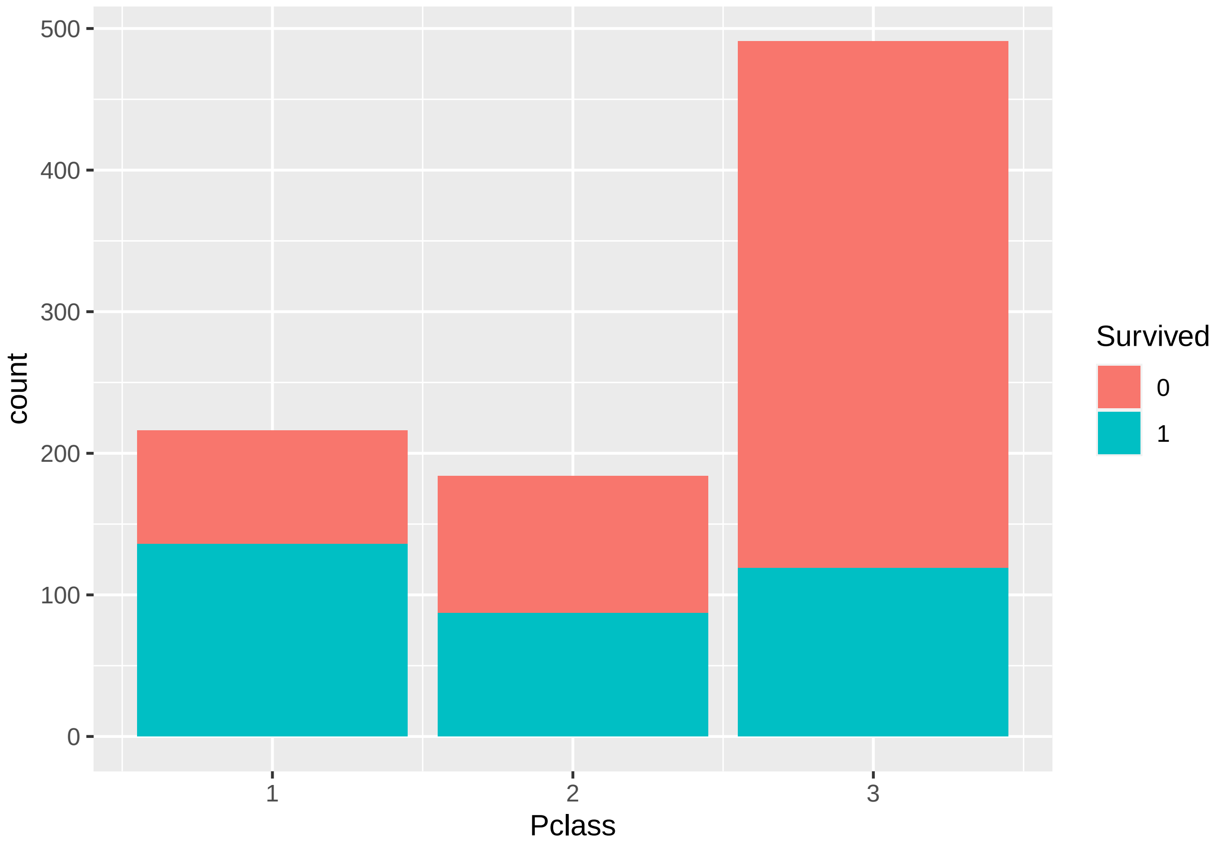
```
df["Fsize"]<-df$Parch + df$SibSp + 1
df_t["Fsize"] <- df_t$Parch + df_t$SibSp +1
```

Exploratory Data Analysis (EDA)

Creating correlogram to see relationships between features.

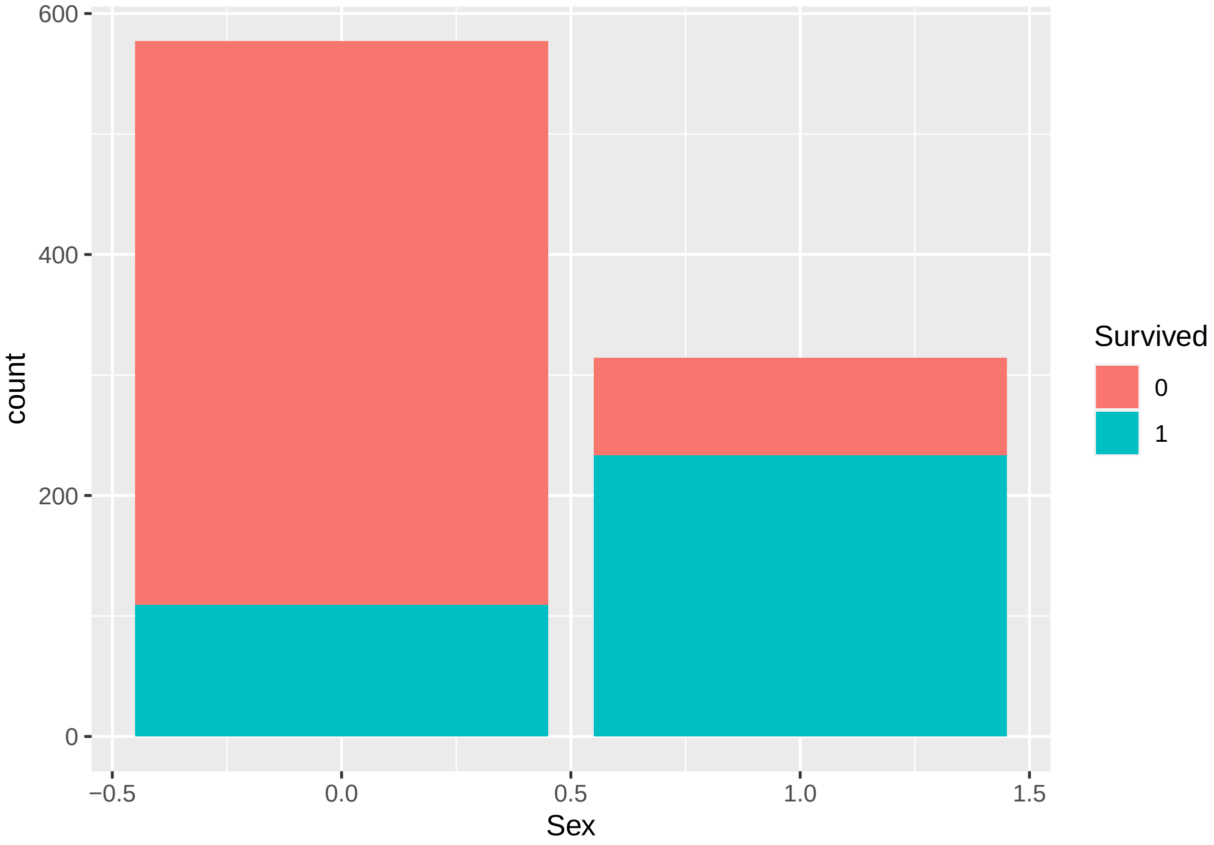


Pclass



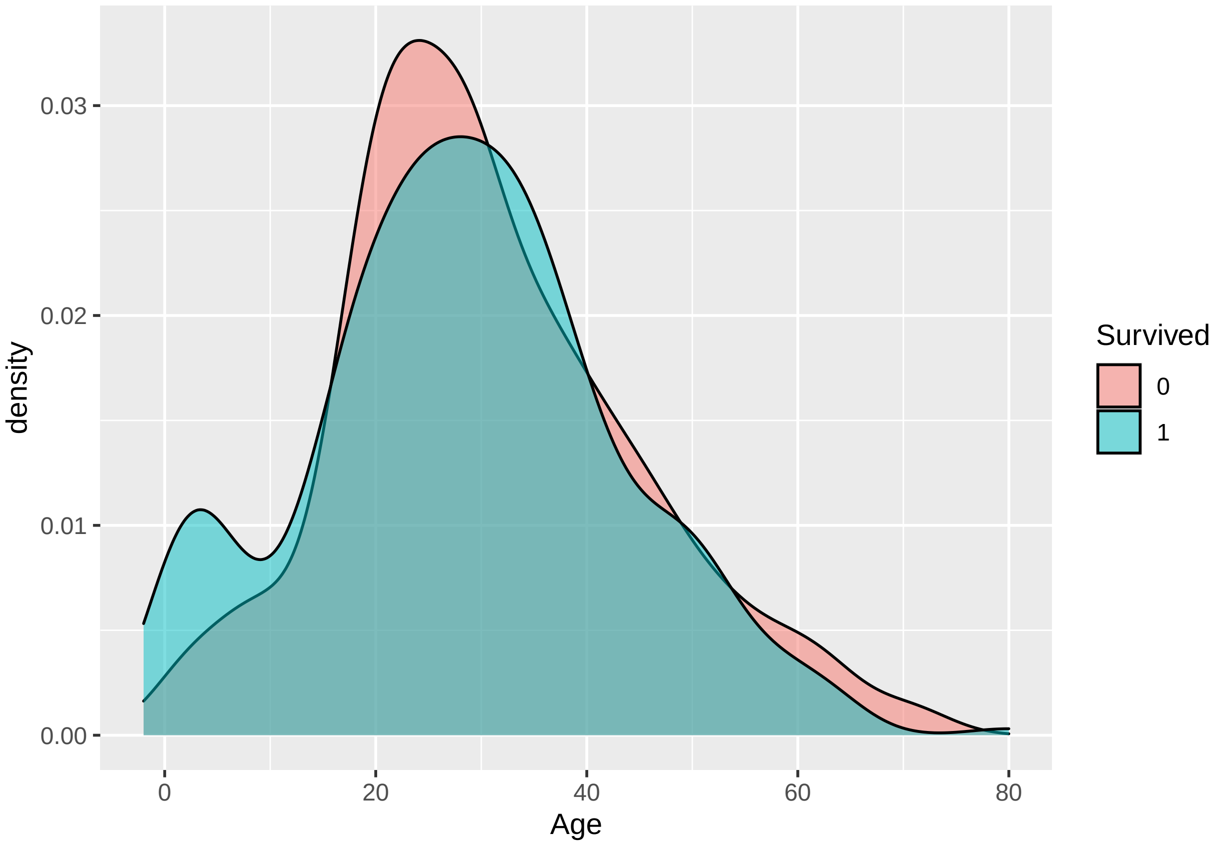
We see that in the 1st class there were more survivors and in the 3rd class the number of non-survivors is relatively high comparing to the number of survivors.

Sex



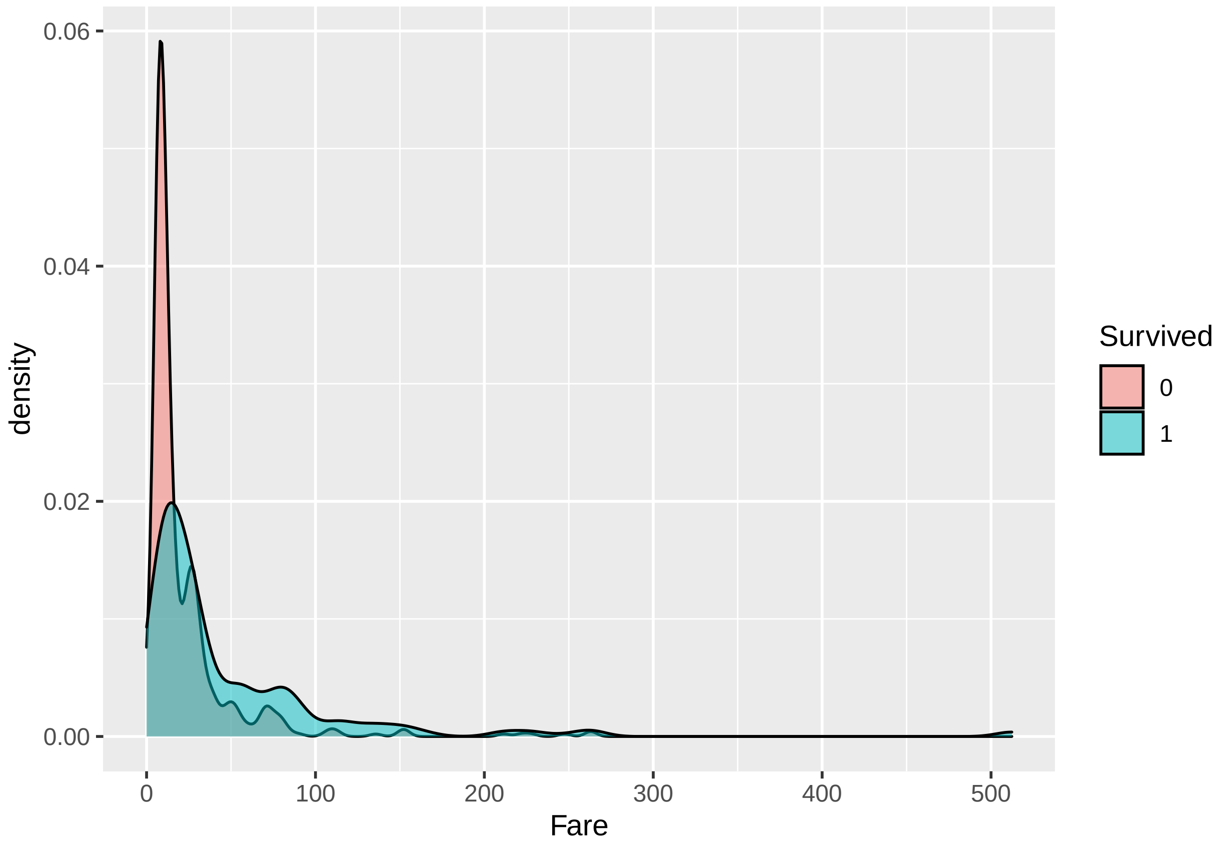
Most of the survivors were females. This is a well-known fact that women with children are in the first place to be saved.

Age



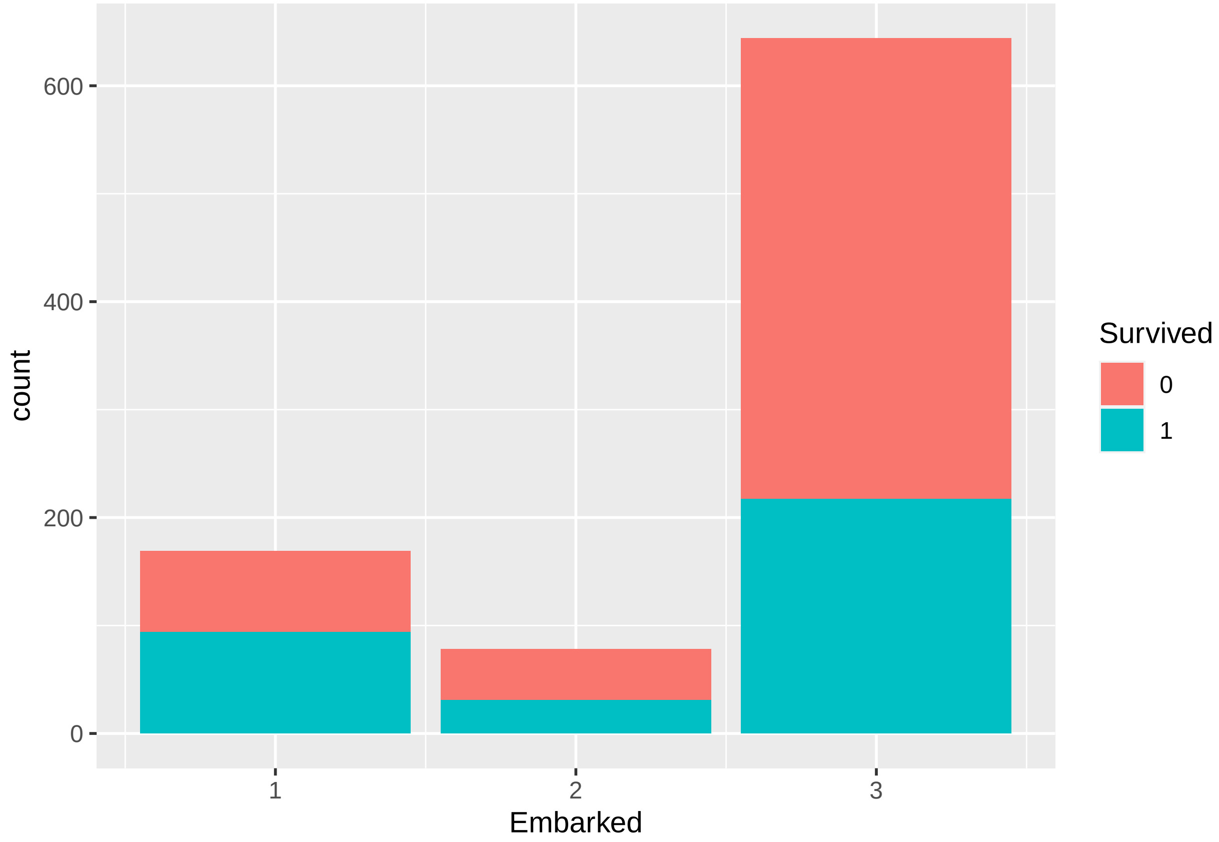
Most people who did survive were kids and adults, while the rest were mostly young adults and in the elderly age.

Fare

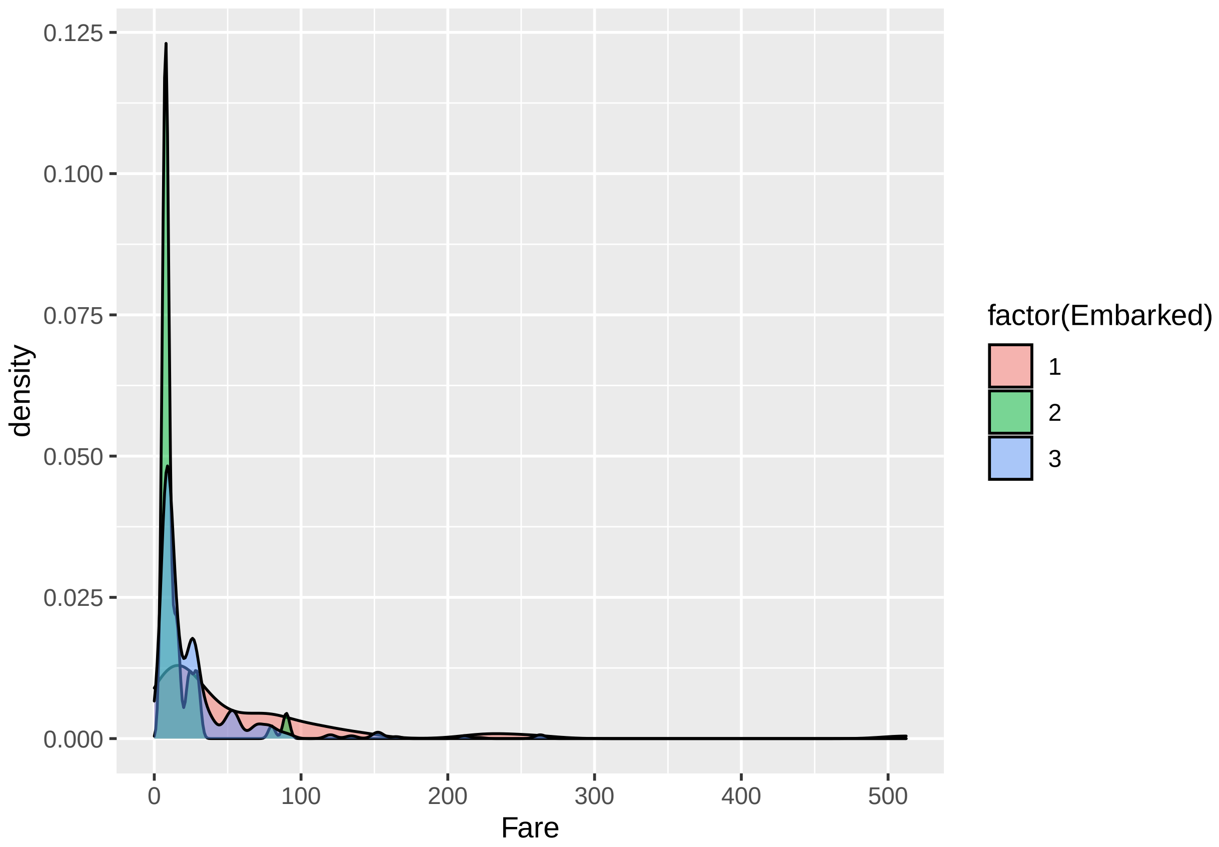


The less is the charge the less survivors. It might be caused with a fact that well paid cabins could be more durable or be placed closer to rescue boats.

Embarked

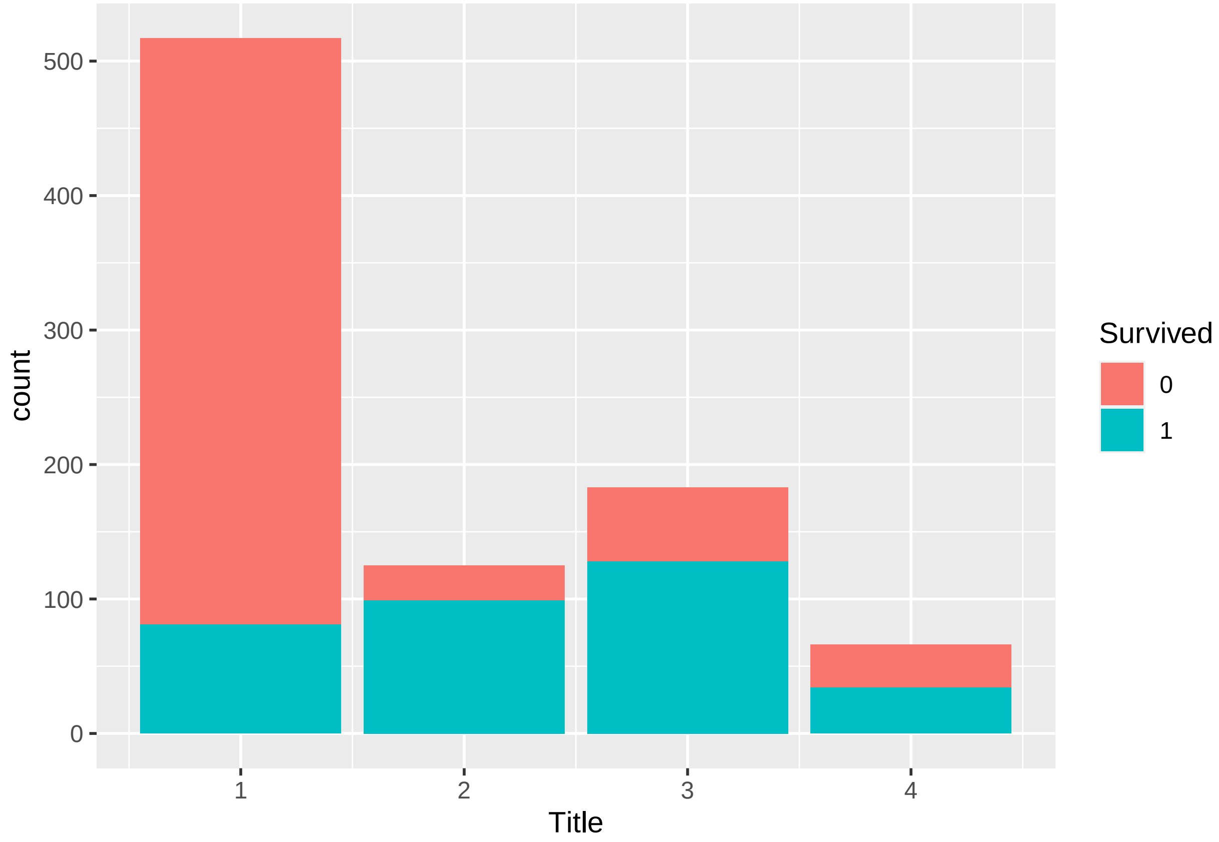


The are more survivors in Cherbourg but less in Queenstown or Southampton. Let's see if there is some relationship with Fares.



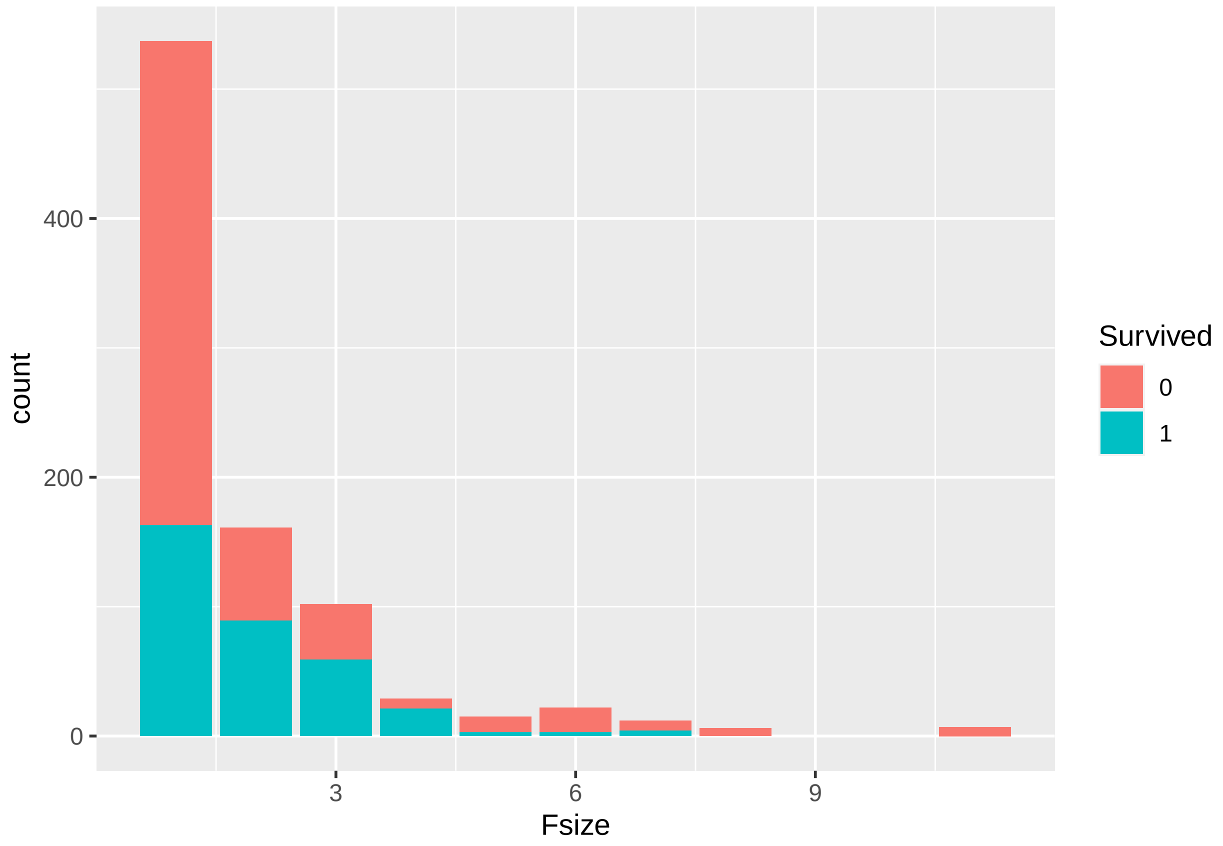
As we can see in Queenstown or Southampton most Fares are in the low interval while in the Cherbourg we may notice that Fares were aswell low as high so the reason standing behind more survivors in Cherbourg are probably better paid cabins.

Title



We see that most of the non-survivors were title Mr. while survivors - Ms. or Mrs..

Fsize



In the families of size 2-4 there were more survivors and in the families of size 1 or 5 and more there were more non-survivors. Most families of size 2-4 consists of married couples and children and as we said, they were a priority to be saved.

Creating SVM model

Now we will perform SVM using radial kernel and choosing best parameters with tune().

```
tune.out<-tune(svm, Survived~Pclass+Sex+Fare+Embarked+Title+Age+Fsize, data=df,
               kernel="radial", range = list(cost=c(0.1,1,10,100,1000)
                                             ,gamma=c(0.5,1,2,3,4)))
summary(tune.out)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost gamma
##   0.1   0.5
##
## - best performance: 0.1716979
##
## - Detailed performance results:
```

```
##     cost gamma     error dispersion
## 1  1e-01   0.5 0.1716979 0.02687508
## 2  1e+00   0.5 0.1773159 0.01723611
## 3  1e+01   0.5 0.1930212 0.02969924
## 4  1e+02   0.5 0.2019850 0.03794601
## 5  1e+03   0.5 0.2311486 0.03433152
## 6  1e-01   1.0 0.2143571 0.03973295
## 7  1e+00   1.0 0.1874282 0.01675743
## 8  1e+01   1.0 0.2020225 0.03178802
## 9  1e+02   1.0 0.2098252 0.03706938
## 10 1e+03   1.0 0.2524594 0.05743478
## 11 1e-01   2.0 0.3322347 0.03918875
## 12 1e+00   2.0 0.2020100 0.02038289
## 13 1e+01   2.0 0.2109738 0.02664771
## 14 1e+02   2.0 0.2334207 0.04804155
## 15 1e+03   2.0 0.2682272 0.05274897
## 16 1e-01   3.0 0.3838577 0.04180705
## 17 1e+00   3.0 0.2109988 0.03032451
## 18 1e+01   3.0 0.2188390 0.01904191
## 19 1e+02   3.0 0.2424095 0.05076851
## 20 1e+03   3.0 0.2805493 0.04403266
## 21 1e-01   4.0 0.3849938 0.04100214
## 22 1e+00   4.0 0.2199625 0.03173495
## 23 1e+01   4.0 0.2278277 0.02241639
## 24 1e+02   4.0 0.2547191 0.05019300
## 25 1e+03   4.0 0.2895256 0.04297028
```

```r
svm_fit<-tune.out$best.model
```

Checking training error rates.

```r
table(svm_fit$fitted,df$Survived)
```

```
## 
##       0   1
##   0 492  85
##   1  57 257
```

Making final prediction and saving result into data frame.

```r
prediction=predict(svm_fit,df_t)
svm_p<-data.frame(PassengerId=892:1309, Survived=prediction)
```

Writing into .csv file ready for a submission on Kaggle.

```r
write.csv(svm_p,file="svm_prediction.csv", row.names=F)
```

We ended up with a 77.51% prediction accuracy which is quite good but could be better. There might be some improvements of the model in the future.