Titanic Survival

Import Packages

```
library('dplyr') # data manipulation
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library('ggplot2')
```

Load in datasets

```
# load train.csv
train <- read.csv('~/Downloads/train.csv', stringsAsFactors = F)
# load test.csv
test  <- read.csv('~/Downloads/test.csv', stringsAsFactors = F)
# combine them as a whole
test$Survived <- NA
full <- rbind(train,test)
```

Show & Check the full data

```
head(full)
```

```
##   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
## 1                            Braund, Mr. Owen Harris   male  22     1
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1
## 3                             Heikkinen, Miss. Laina female  26     0
## 4       Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1
## 5                           Allen, Mr. William Henry   male  35     0
## 6                                   Moran, Mr. James   male  NA     0
##   Parch           Ticket    Fare Cabin Embarked
## 1     0        A/5 21171  7.2500              S
## 2     0         PC 17599 71.2833   C85        C
```

```
## 3     0 STON/O2. 3101282  7.9250              S
## 4     0           113803 53.1000  C123        S
## 5     0           373450  8.0500              S
## 6     0           330877  8.4583              Q
```

Data Cleaning

```
# Dump out Name Column
full$Name <- NA
# Process Age Column
age <- full$Age
n = length(age)
# replace missing value with a random sample from raw data
set.seed(123)
for(i in 1:n){
  if(is.na(age[i])){
    age[i] = sample(na.omit(full$Age),1)
  }
}
# Process Cabin Column
cabin <- full$Cabin
n = length(cabin)
for(i in 1:n){
  if(is.na(cabin[i])){
    cabin[i] = 0
  } else{
    s = strsplit(cabin[i]," ")
    cabin[i] = length(s[[1]])
  }
}
# Check fare missing values
full$PassengerId[is.na(full$Fare)]
```

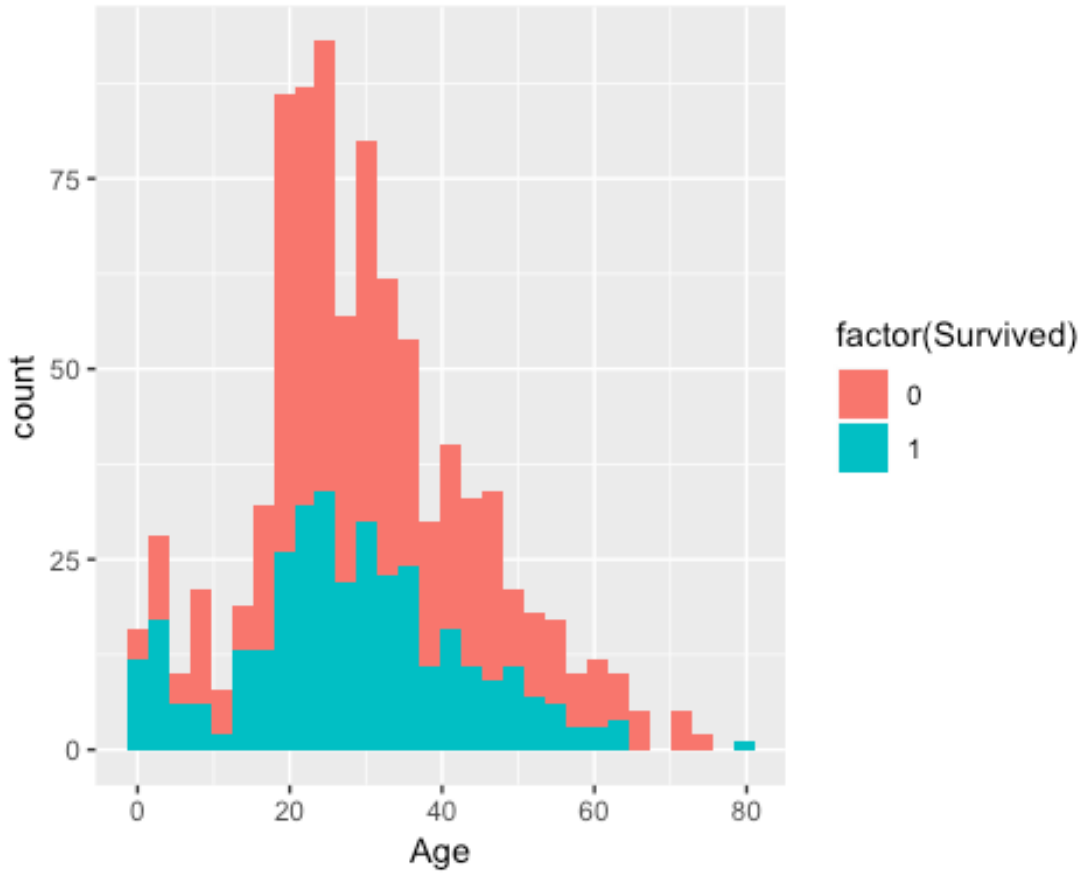
```
## [1] 1044
```

```
#full[1044,]
# Fill in fare missing values
full$Fare[1044] <- median(full[full$Pclass == '3' & full$Embarked == 'S',
]$Fare, na.rm = TRUE)
# Process Embarked Column
embarked <- full$Embarked
n = length(embarked)
for(i in 1:n){
  if(is.na(embarked[i])){
    embarked[i] = "S"
  }
}
```

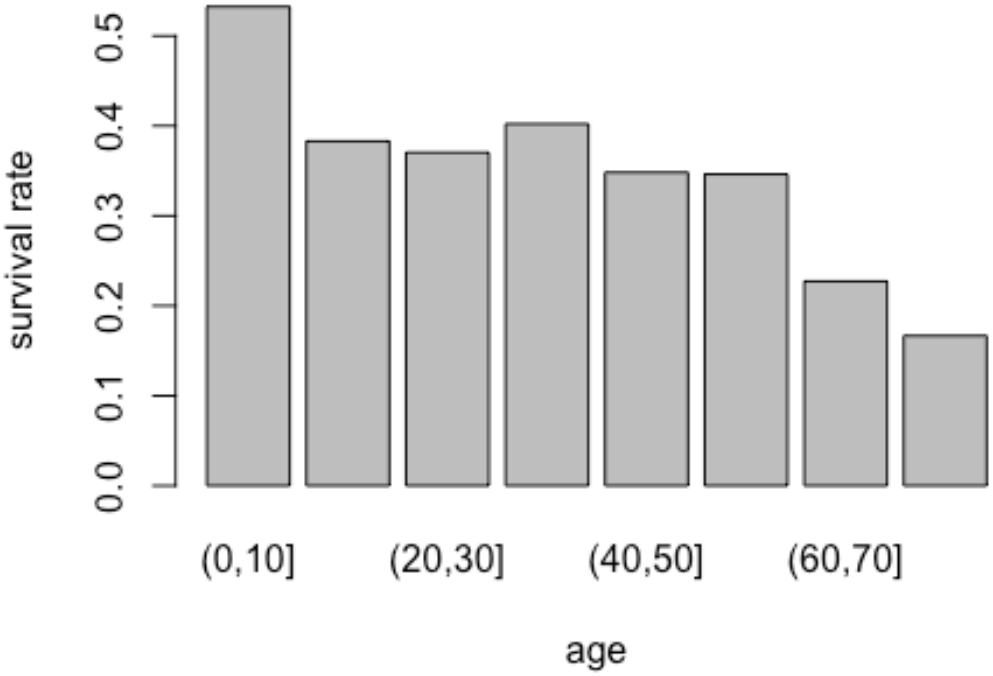
Exploratory Analysis & Data Processing

```r
# Survival vs Age
d <- data.frame(Age = age[0:891], Survived = train$Survived)
ggplot(d, aes(Age,fill = factor(Survived))) +
    geom_histogram()

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

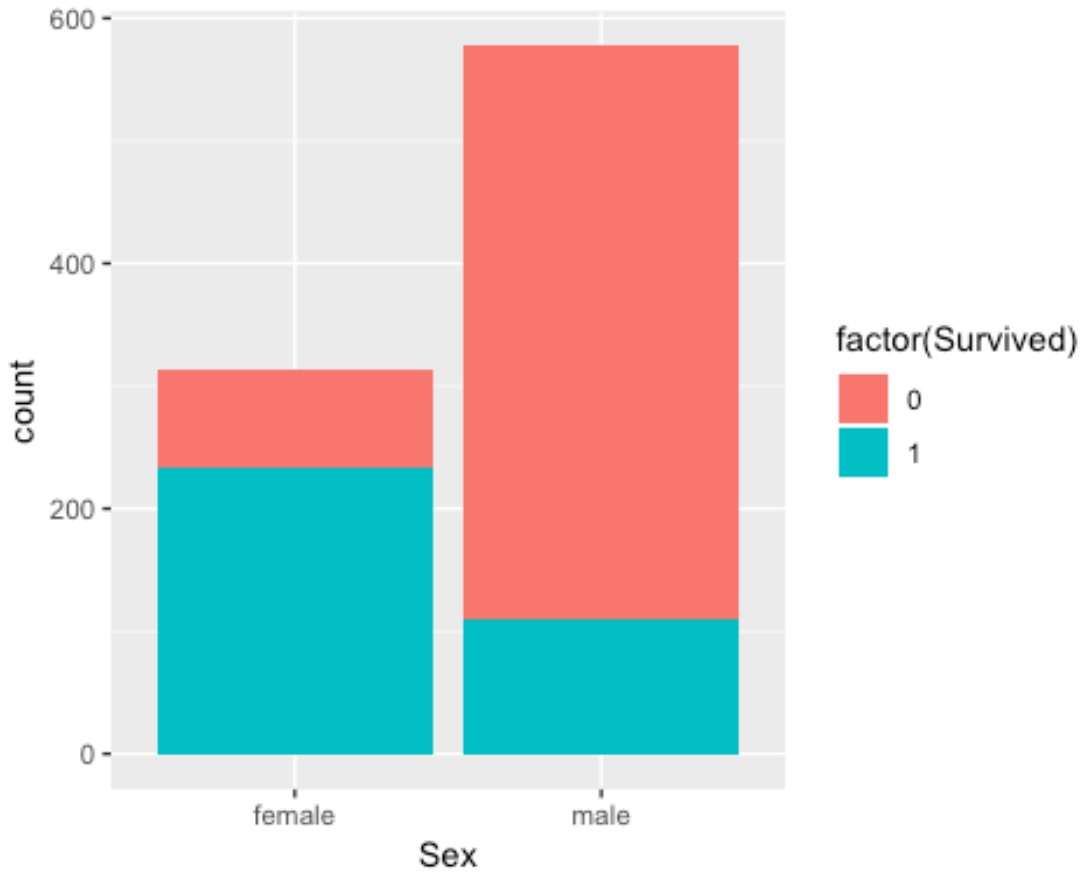



```r
# create bar chart to show relationship between survival rate and age
intervals
cuts <- cut(d$Age,hist(d$Age,10,plot = F)$breaks)
rate <- tapply(d$Survived,cuts,mean)
d2 <- data.frame(age = names(rate),rate)
barplot(d2$rate, xlab = "age",ylab = "survival rate")
```



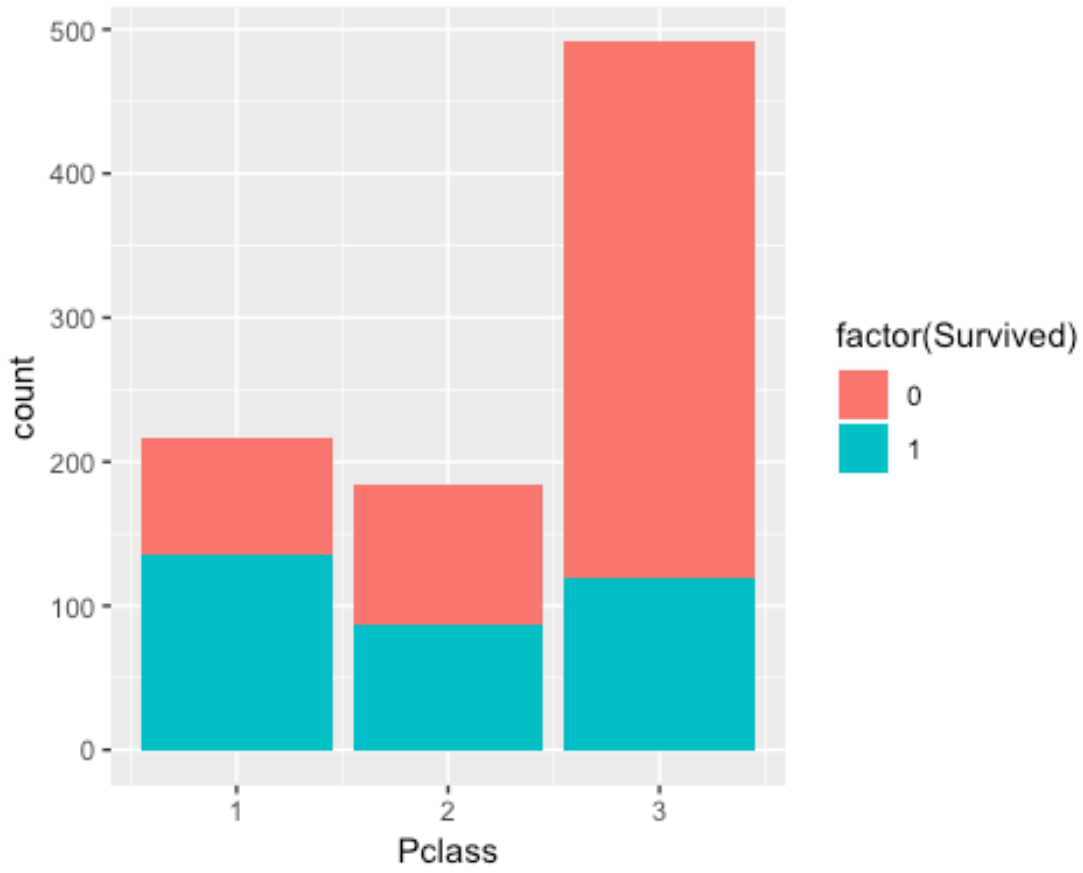

```
# create histgram to show effect of Sex on survival
ggplot(train, aes(Sex,fill = factor(Survived))) +
    geom_histogram(stat = "count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



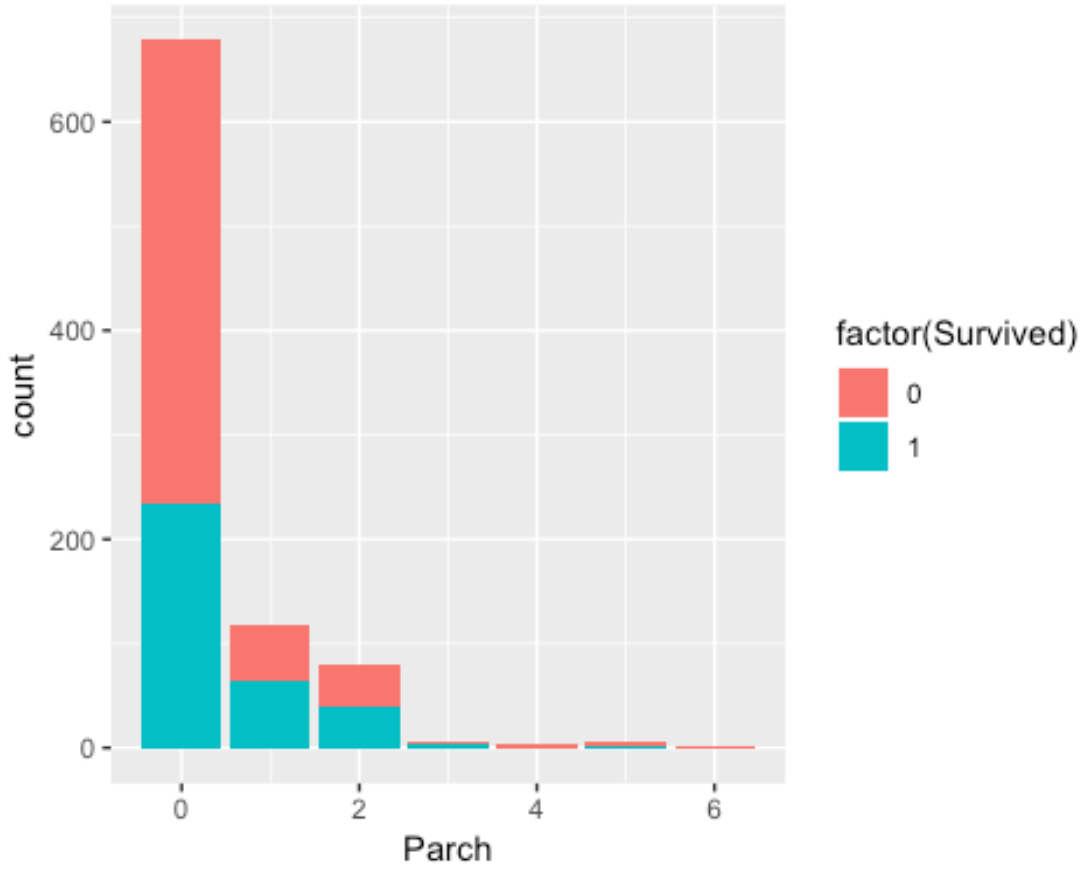

```
# Pclass v.s. Survival
ggplot(train, aes(Pclass,fill = factor(Survived))) +
    geom_histogram(stat = "count")

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



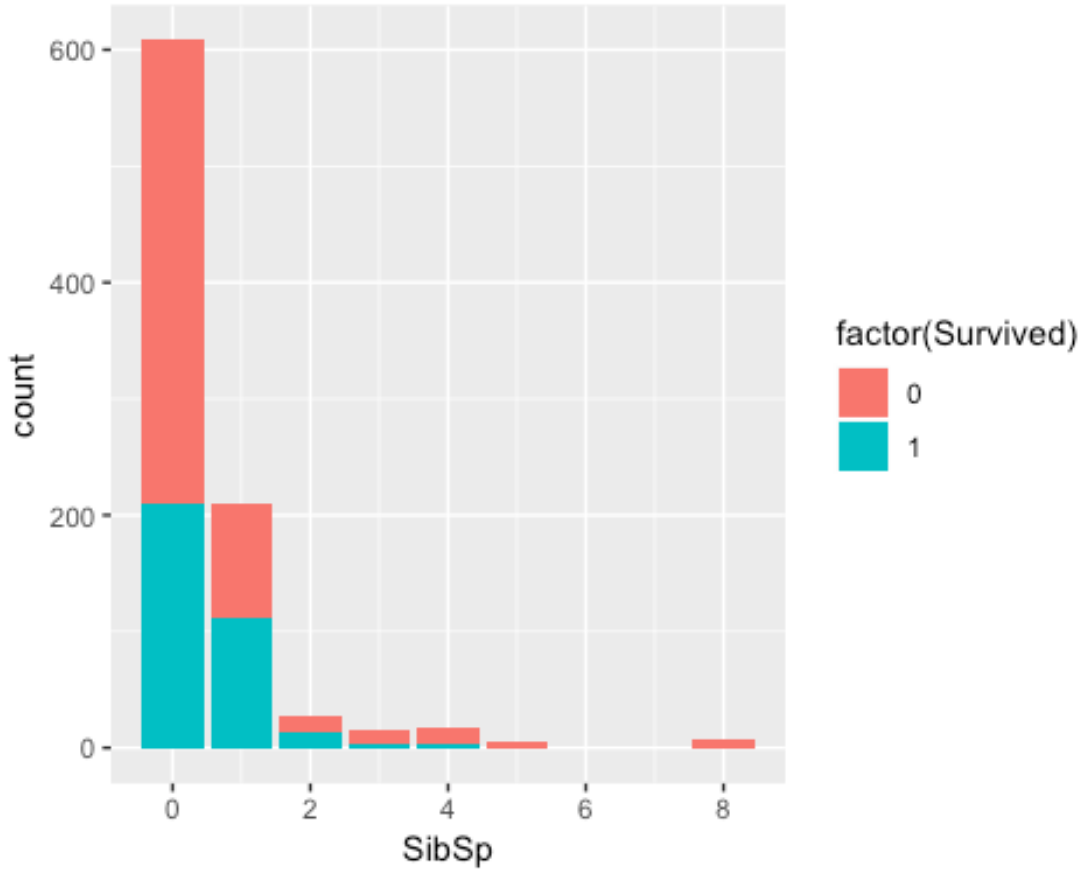

```
# Family Size v.s. Survival
ggplot(train, aes(Parch,fill = factor(Survived))) +
    geom_histogram(stat = "count")

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



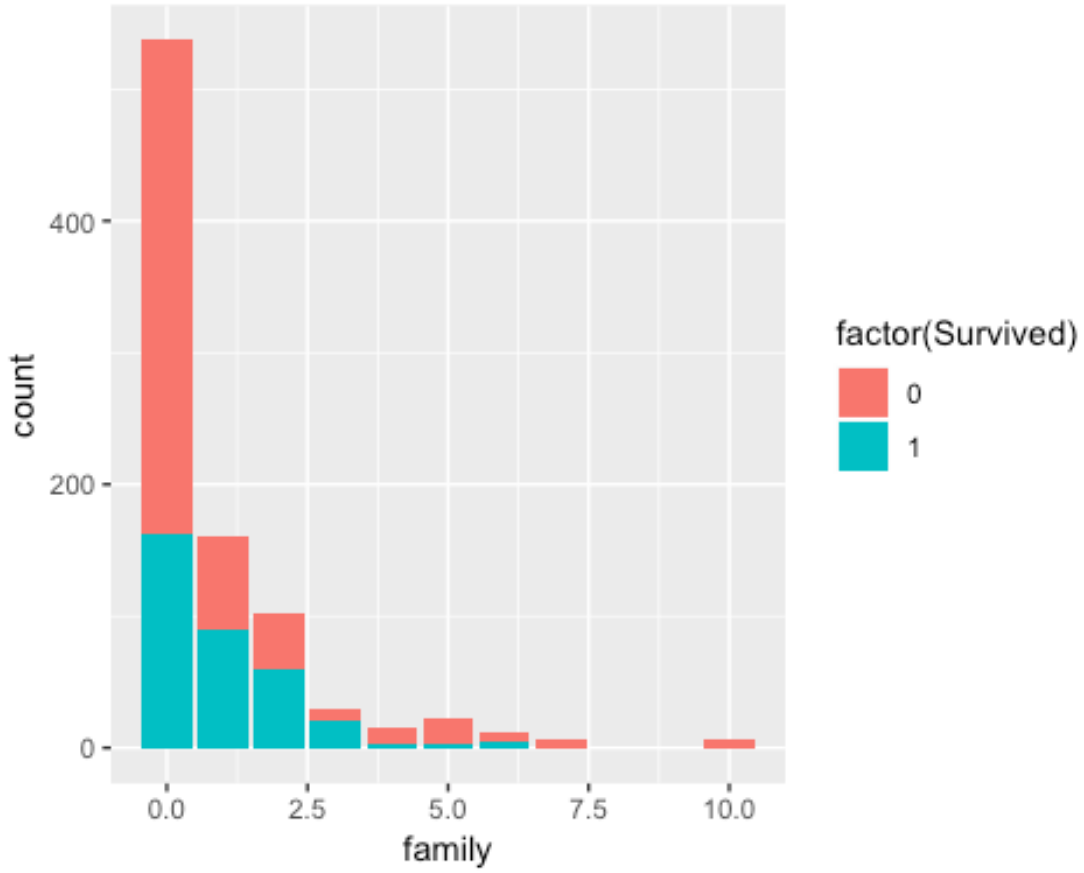

```
# Having siblings/spouse v.s. Survival
ggplot(train, aes(SibSp,fill = factor(Survived))) +
    geom_histogram(stat = "count")

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



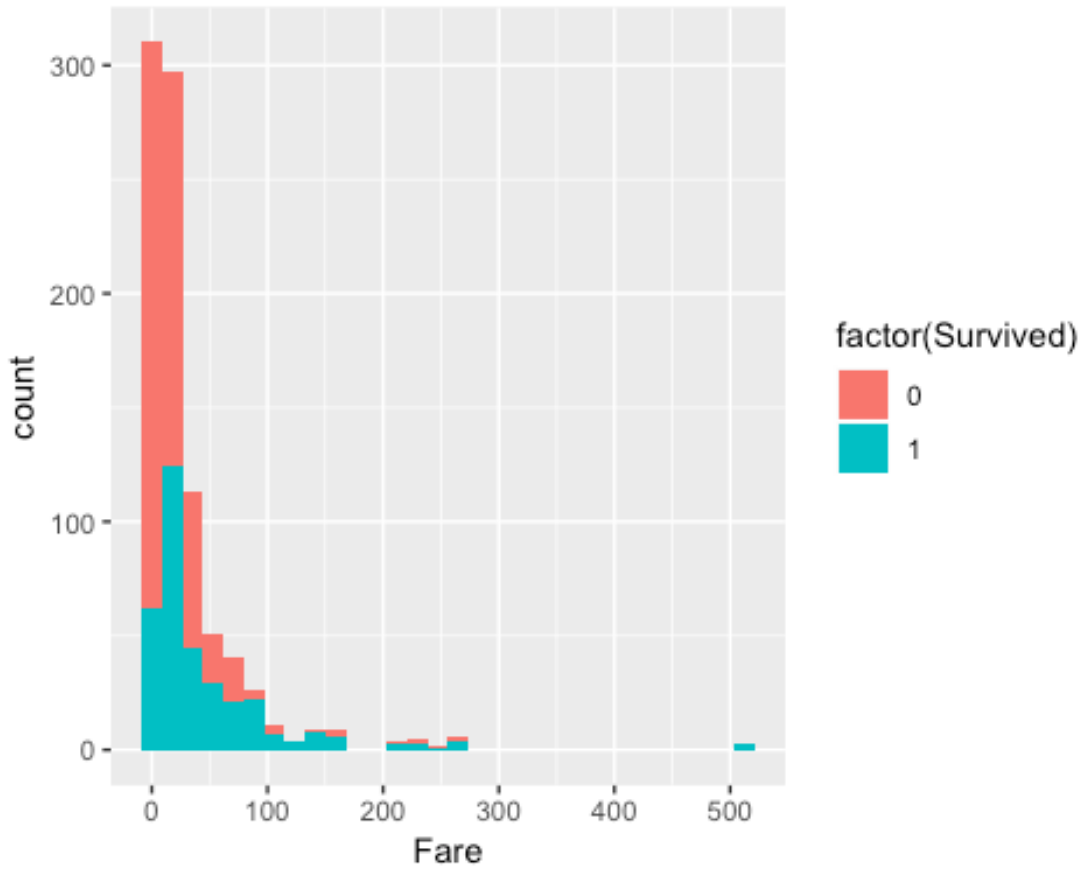

```
# Combine SibSp and Parch
family <- full$SibSp + full$Parch
d <- data.frame(family = family[1:891],Survived = train$Survived)
ggplot(d, aes(family,fill = factor(Survived))) +
    geom_histogram(stat = "count")

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

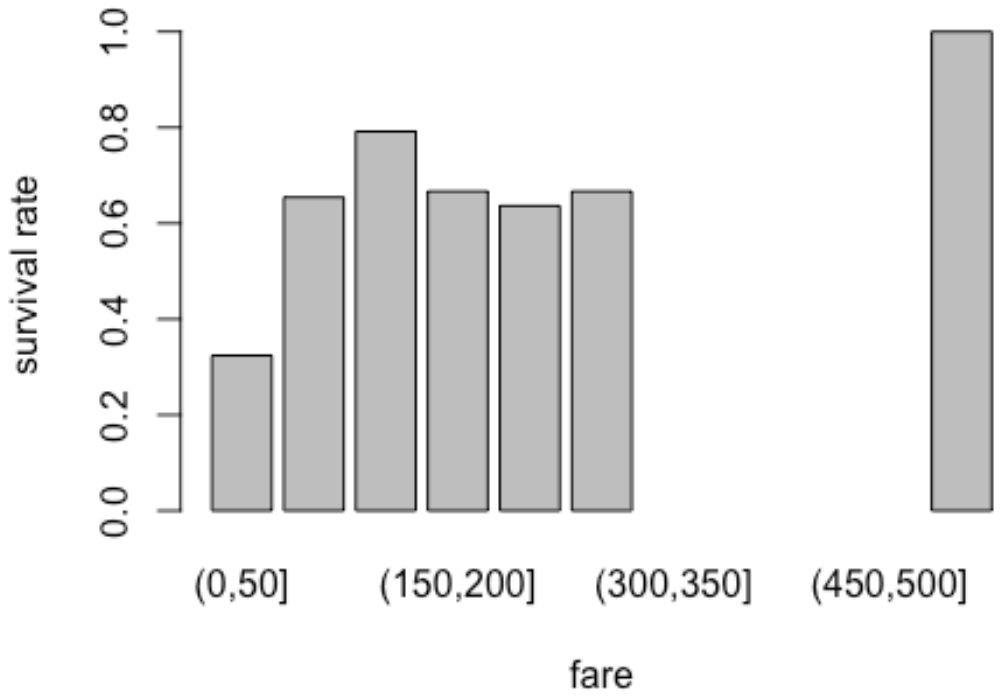



```
# Fare vs Survival
ggplot(train, aes(Fare,fill = factor(Survived))) +
    geom_histogram()

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

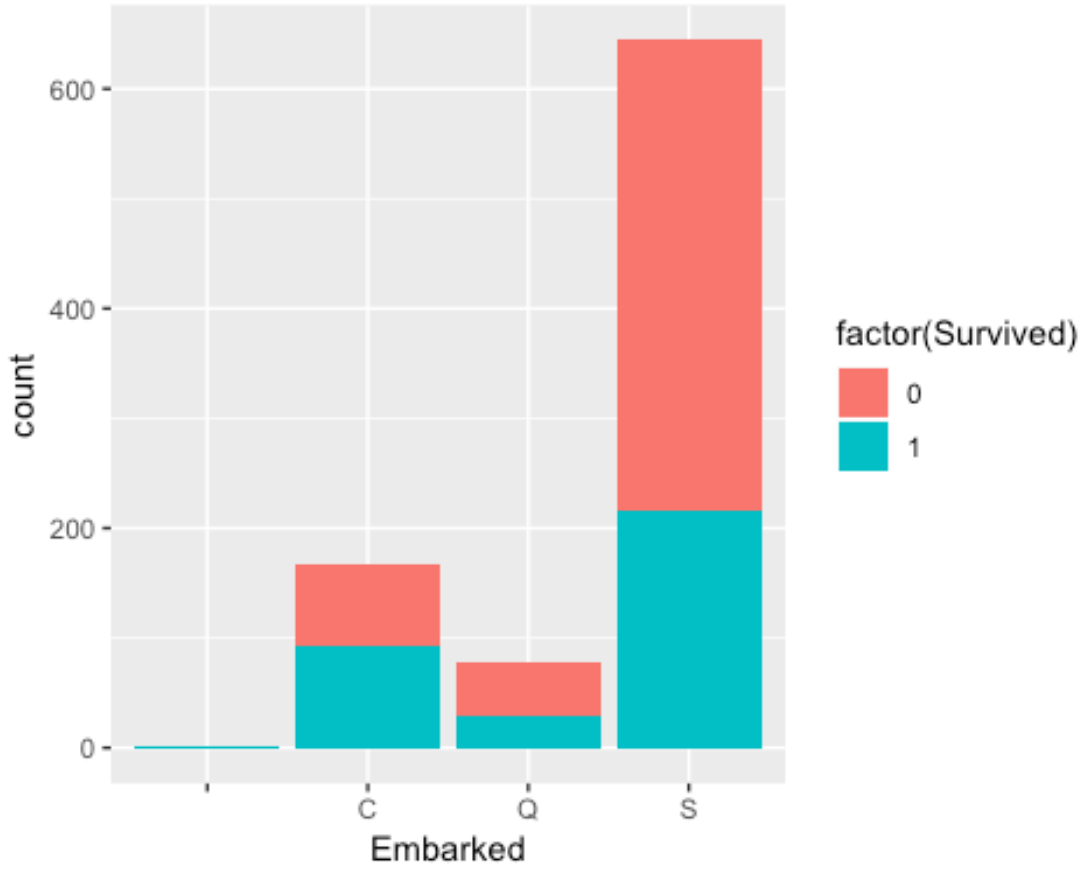



```
#Fare vs Survival
cuts <- cut(train$Fare,hist(train$Fare,10,plot = F)$breaks)
rate <- tapply(train$Survived,cuts,mean)
d <- data.frame(fare = names(rate),rate)
barplot(d$rate, xlab = "fare",ylab = "survival rate")
```




```
# Embarked v.s. Survival
d <- data.frame(Embarked = embarked[1:891], Survived = train$Survived)
ggplot(d, aes(Embarked,fill = factor(Survived))) +
    geom_histogram(stat = "count")

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



##

Feature Engineering

```
f.survived = train$Survived
# Train-test Split
f.age = age[1:891]
t.age = age[892:1309]
f.fare = full$Fare[1:891]
t.fare = full$Fare[892:1309]
f.cabin = cabin[1:891]
t.cabin = cabin[892:1309]
family <- full$SibSp + full$Parch
f.family = family[1:891]
t.family = family[892:1309]
f.pclass = train$Pclass
t.pclass = test$Pclass
f.sex = train$Sex
t.sex = test$Sex
f.embarked = embarked[1:891]
t.embarked = embarked[892:1309]
```

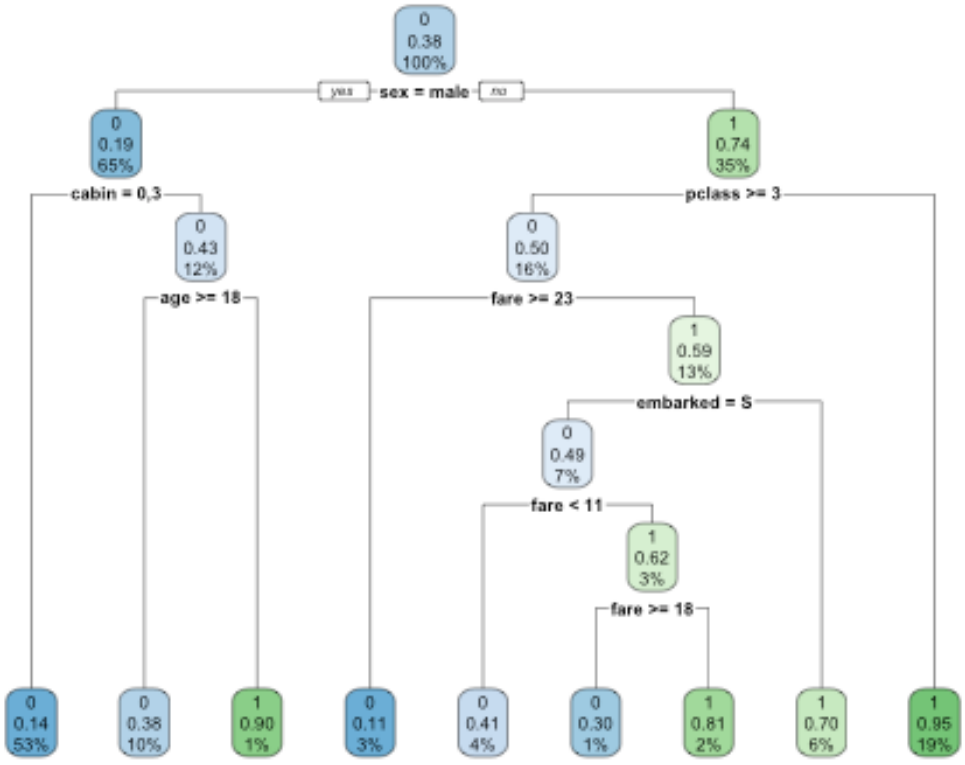
Modeling

```
## [1] 0.8002245
```

```r
# decision tree
library(rpart)
fit_dt <- rpart(factor(survived) ~ age + fare + sex + embarked + family
                + cabin + pclass,data = data)
# Prediction with Decision Tree
dt.fitted = predict(fit_dt)
pred1 = rep(NA,891)
for(i in 1:891){
  if(dt.fitted[i,1] >= dt.fitted[i,2] ){
    pred1[i] = 0
  } else{
    pred1[i] = 1
  }
}
# Check Accuracy
mean(pred1 == train$Survived)
```

```
## [1] 0.8316498
```

```r
# Plot Decision Tree
library(rpart.plot)
rpart.plot(fit_dt, extra = 106)
```



```r
# Random Forest
library('randomForest')

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

## The following object is masked from 'package:dplyr':
##
##     combine

set.seed(123)
fit_rf <- randomForest(factor(survived) ~ age + fare + sex + embarked +
family
                 + cabin + pclass,data = data)
# Prediction with Random Forest
rf.fitted = predict(fit_rf)
pred2 = rep(NA,891)
for(i in 1:891){
  pred2[i] = as.integer(rf.fitted[[i]]) - 1
}
# Check Accuracy
mean(pred2 == train$Survived)

## [1] 0.8125701

# svm
library(e1071)
fit_svm <- svm(factor(survived) ~ age + fare + sex + embarked + family
                 + cabin + pclass,data = data)
# Prediction with svm
svm.fitted = predict(fit_svm)
pred3 = rep(NA,891)
for(i in 1:891){
  pred3[i] = as.integer(svm.fitted[[i]]) - 1
}
# Check Accuracy
mean(pred3 == train$Survived)

## [1] 0.8249158
```

Prediction with decision tree model

```r
# Dataframe for training
test_data <- data.frame(age = t.age, fare = t.fare, sex = t.sex, embarked =
t.embarked,
```

```
                              family = t.family, cabin =  t.cabin, pclass =
t.pclass)
# make prediction
dt_predict = predict(fit_dt,newdata = test_data )
dt_predict1 = rep(NA,418)
for(i in 1:418){
  if(dt.fitted[i,1] >= dt.fitted[i,2] ){
    dt_predict1[i] = 0
  } else{
    dt_predict1[i] = 1
  }
}
table(dt_predict1)

## dt_predict1
##   0   1
## 294 124
```