Titanic Survival Prediction

Yinuo Song

5/15/2020

```
rm(list=ls())
```

```
getwd()
```

```
## [1] "/Users/yinuo/Desktop"
```

```
library(dplyr)
library(Amelia)
library(ggplot2)
library(scales)
library(caTools)
library(car)
library(ROCR)
library(e1071)
library(rpart)
library(rpart.plot)
library(randomForest)
library(caret)
```

Import data

```
train <- read.csv('train.csv', stringsAsFactors = F)
test  <- read.csv('test.csv', stringsAsFactors = F)
titanic  <- bind_rows(train, test)
str(titanic)
```

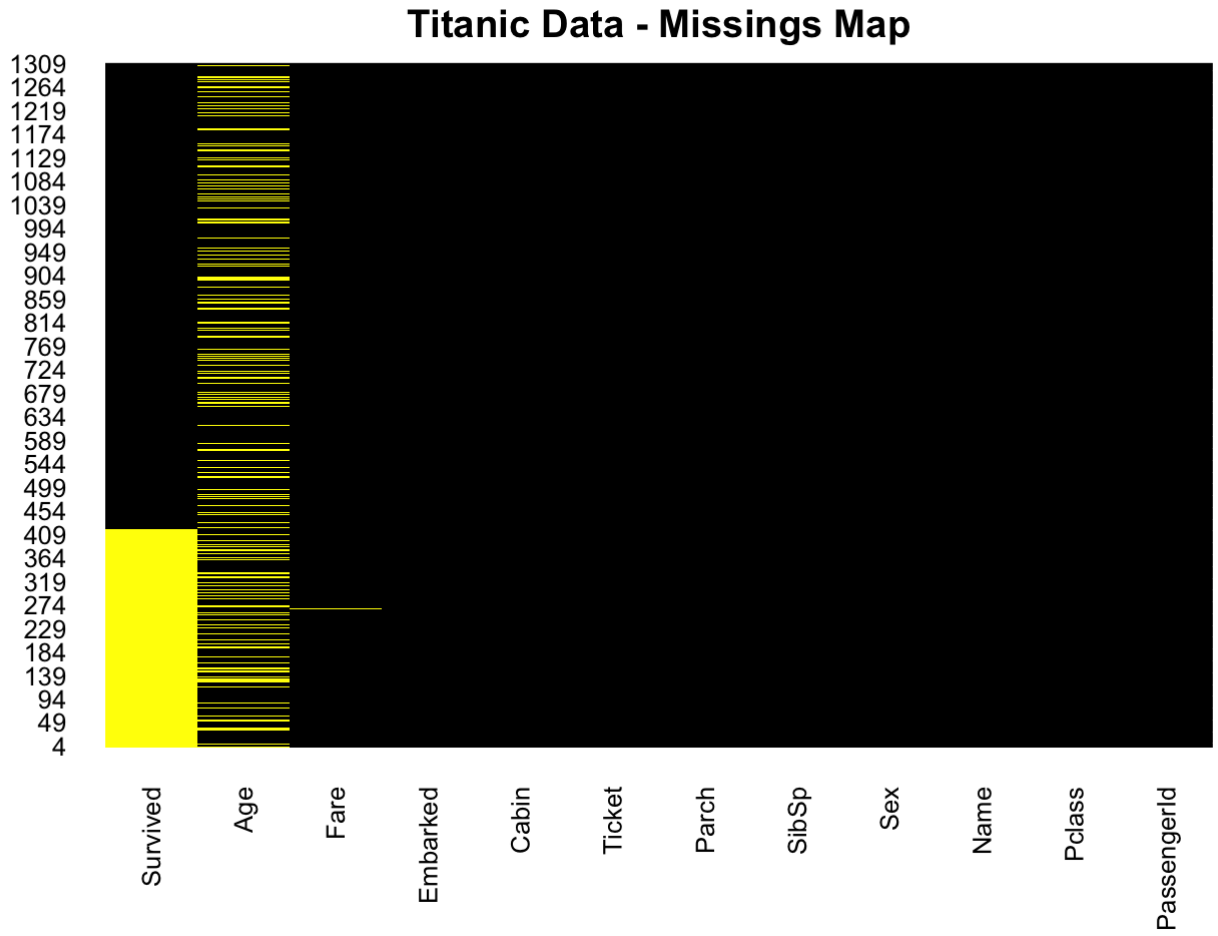
```
## 'data.frame':    1309 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence
Briggs Thayer)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques Heath (Lily May Peel)"
...
##  $ Sex        : chr  "male" "female" "female" "female" ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

Check missing data

```
colSums(is.na(titanic)|titanic=='')
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0         418           0           0           0         263
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           1        1014           2
```

```
missmap(titanic, main="Titanic Data - Missings Map",
        col=c("yellow", "black"), legend=FALSE)
```



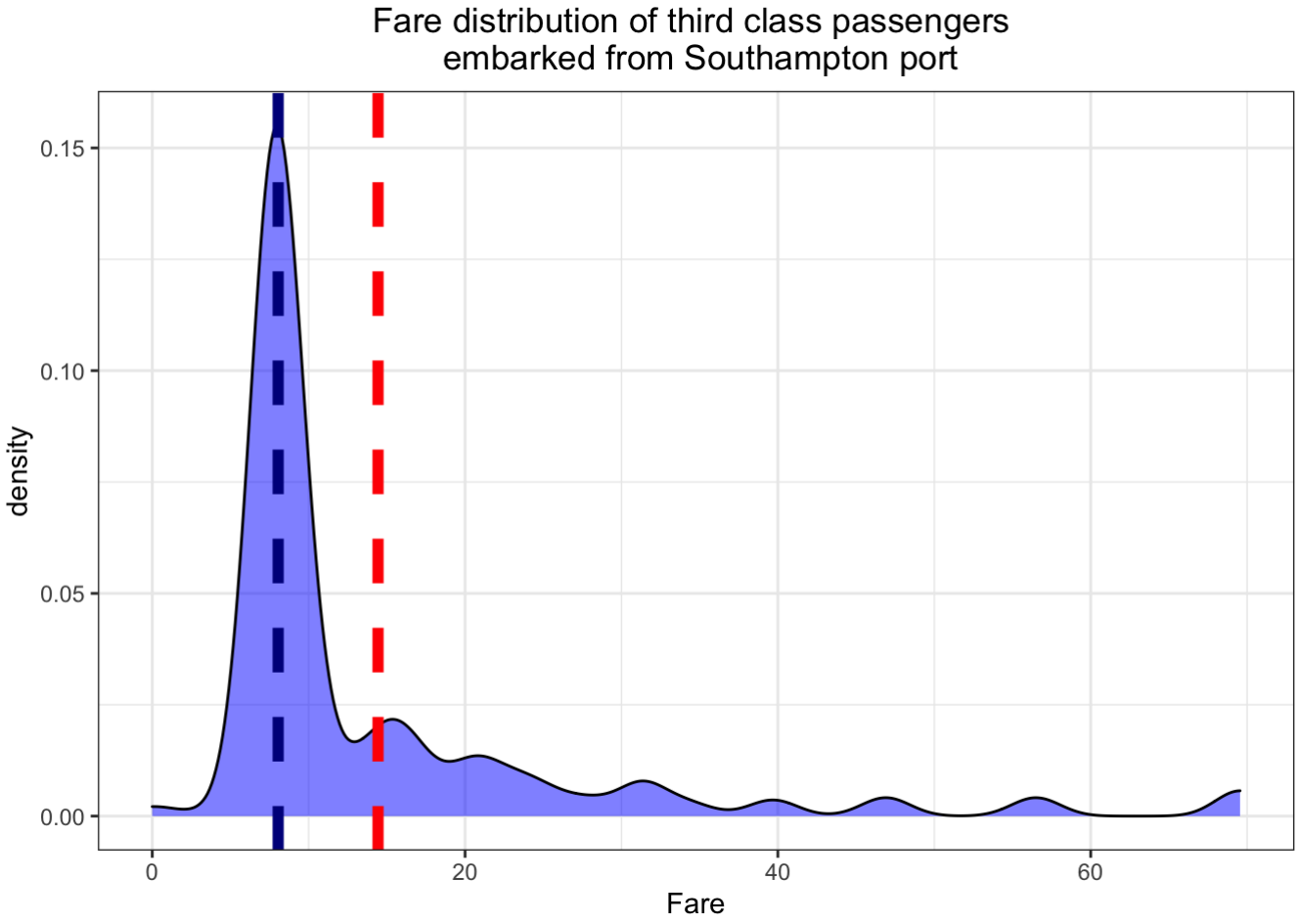
Missing fare data imputation

```
filter(titanic, is.na(Fare)==TRUE|Fare=='')
```

PassengerId	Survived	Pclass	Name	Sex	Age	Sib…	Par…	Ticket	F…
<int>	<int>	<int>	<chr>	<chr>	<dbl>	<int>	<int>	<chr>	<dbl>
1044	*NA*	3	Storey, Mr. Thomas	male	60.5	0	0	3701	*NA*

1 row | 1-10 of 12 columns

```
ggplot(filter(titanic, Pclass==3 & Embarked=="S"), aes(Fare)) +
  geom_density(fill="blue", alpha=0.5) +
  geom_vline(aes(xintercept=median(Fare, na.rm=T)), colour='darkblue', linetype='dashed'
, size=2) +
  geom_vline(aes(xintercept=mean(Fare, na.rm=T)), colour='red', linetype='dashed', size=
2) +
  ggtitle("Fare distribution of third class passengers \n embarked from Southampton por
t") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```




```
titanic$Fare[is.na(titanic$Fare)==TRUE] = median(filter(titanic, Pclass==3 & Embarked==
"S")$Fare, na.rm=TRUE)
colSums(is.na(titanic)|titanic=='')
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0         418           0           0           0         263
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0        1014           2
```

Missing embarked data imputation

```
filter(titanic, is.na(Embarked)==TRUE|Embarked=='')
```

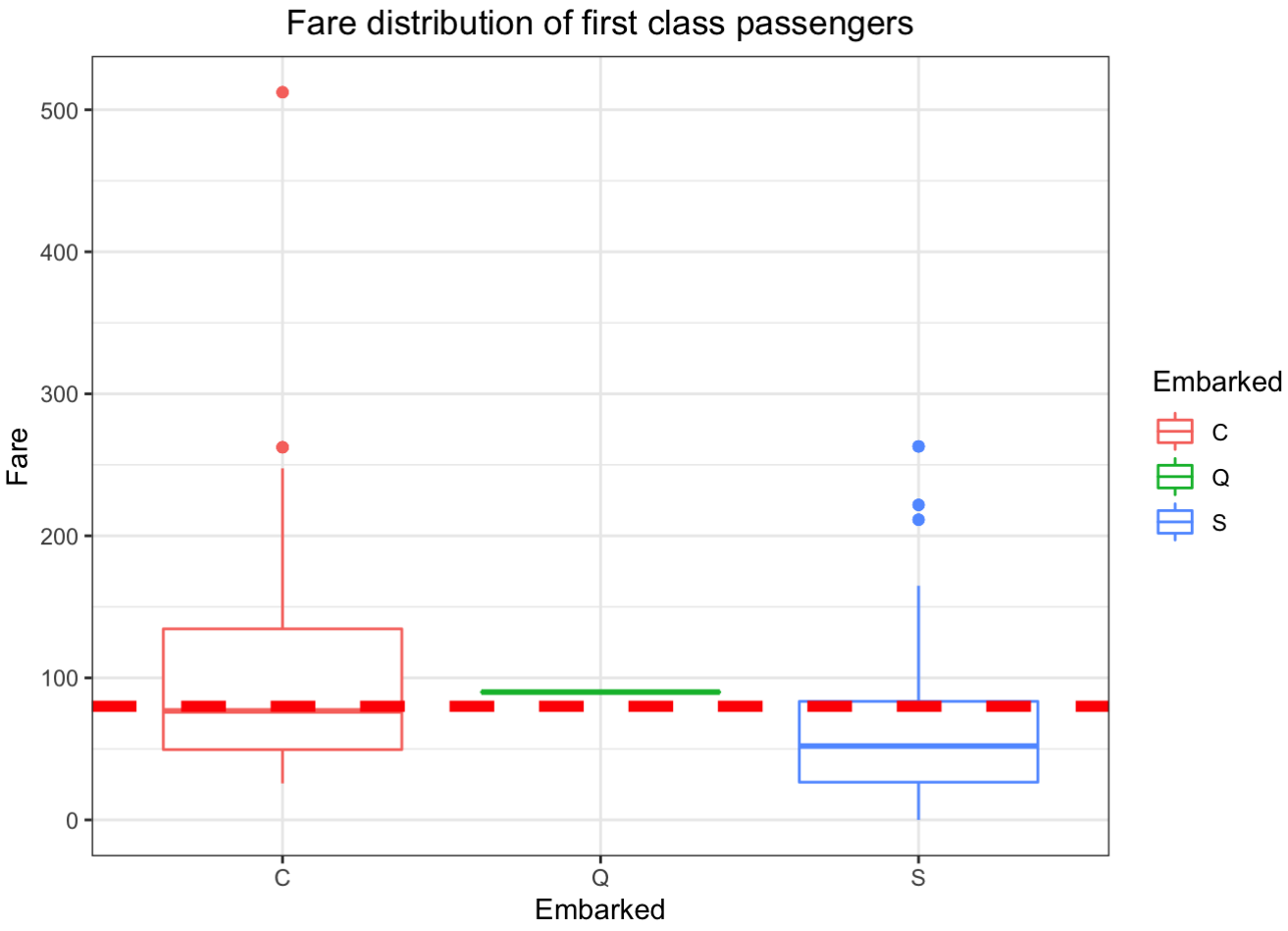
PassengerId	Survived	Pcla...	Name	Sex	...	Si
<int>	<int>	<int>	<chr>	<chr>	<dbl>	<in
62	1	1	Icard, Miss. Amelie	female	38	
830	1	1	Stone, Mrs. George Nelson (Martha Evelyn)	female	62	

2 rows | 1-8 of 12 columns

```
table(filter(titanic, Pclass==1)$Embarked)
```

```
##
##       C   Q   S
##   2 141   3 177
```

```
ggplot(filter(titanic, is.na(Embarked)==FALSE & Embarked!='' & Pclass==1),
       aes(Embarked, Fare)) +
  geom_boxplot(aes(colour = Embarked)) +
  geom_hline(aes(yintercept=80), colour='red', linetype='dashed', size=2) +
  ggtitle("Fare distribution of first class passengers") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```

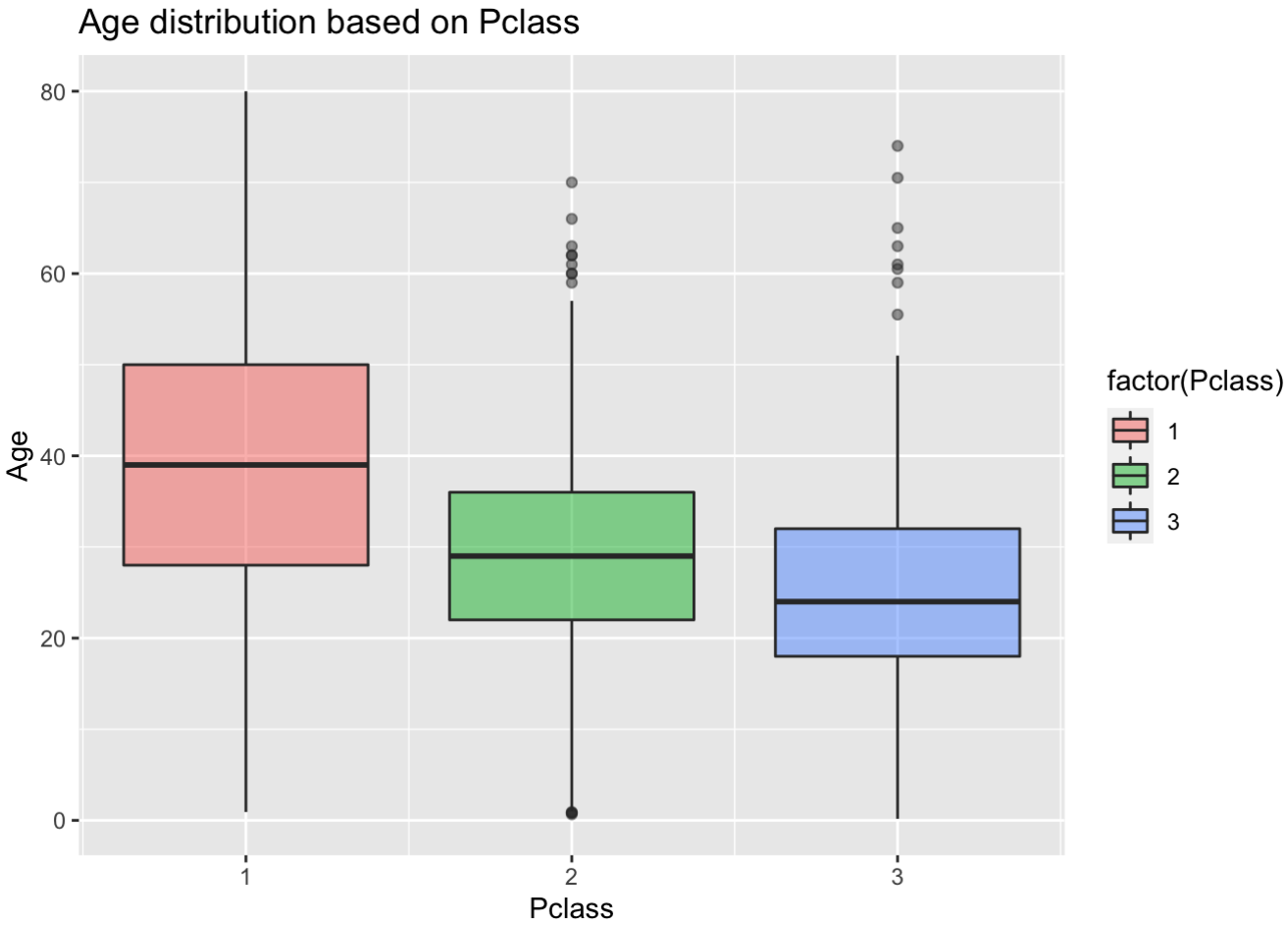


```
titanic$Embarked[titanic$Embarked==""] = "C"
colSums(is.na(titanic)|titanic=='')
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0         418           0           0           0         263
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0        1014           0
```

Missing age data imputation

```
ggplot(titanic,aes(Pclass,Age)) +
  geom_boxplot(aes(fill=factor(Pclass)),alpha=0.5) +
  ggtitle("Age distribution based on Pclass")
```



```
impute.age <- function(age,class){
  vector <- age
  for (i in 1:length(age)){
    if (is.na(age[i])){
      if (class[i] == 1){
        vector[i] <- round(mean(filter(titanic,Pclass==1)$Age, na.rm=TRUE),0)
      }else if (class[i] == 2){
        vector[i] <- round(mean(filter(titanic,Pclass==2)$Age, na.rm=TRUE),0)
      }else{
        vector[i] <- round(mean(filter(titanic,Pclass==3)$Age, na.rm=TRUE),0)
      }
    }else{
      vector[i]<-age[i]
    }
  }
  return(vector)
}
imputed.age <- impute.age(titanic$Age,titanic$Pclass)
titanic$Age <- imputed.age
```

```
colSums(is.na(titanic)|titanic=='')
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0         418           0           0           0           0
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0        1014           0
```

```
head(titanic$Name)
```

```
## [1] "Braund, Mr. Owen Harris"
## [2] "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
## [3] "Heikkinen, Miss. Laina"
## [4] "Futrelle, Mrs. Jacques Heath (Lily May Peel)"
## [5] "Allen, Mr. William Henry"
## [6] "Moran, Mr. James"
```

```
titanic$Title <- gsub("^.*, (.*?)\\..*$", "\\1", titanic$Name)
table(titanic$Sex, titanic$Title)
```

```
##
##          Capt Col Don Dona  Dr Jonkheer Lady Major Master Miss Mlle Mme  Mr Mrs
##   female    0   0   0    1   1        0    1     0      0  260    2   1   0 197
##   male      1   4   1    0   7        1    0     2     61    0    0   0 757   0
##
##           Ms Rev Sir the Countess
##   female   2   0   0            1
##   male     0   8   1            0
```

Title

```
titanic$Title[titanic$Title == 'Mlle' | titanic$Title == 'Ms'] <- 'Miss'
titanic$Title[titanic$Title == 'Mme']  <- 'Mrs'
Other <- c('Dona', 'Dr', 'Lady', 'the Countess','Capt', 'Col', 'Don', 'Jonkheer', 'Majo
r', 'Rev', 'Sir')
titanic$Title[titanic$Title %in% Other]  <- 'Other'
table(titanic$Sex, titanic$Title)
```

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

Family size

```
FamilySize <- titanic$SibSp + titanic$Parch + 1
table(FamilySize)
```

```
## FamilySize
##   1   2   3   4   5   6   7   8  11
## 790 235 159  43  22  25  16   8  11
```

```
titanic$FamilySize <- sapply(1:nrow(titanic), function(x)
                         ifelse(FamilySize[x]==1, "Single",
                         ifelse(FamilySize[x]>4, "Large", "Small")))

table(titanic$FamilySize)
```

```
##
##  Large Single  Small
##     82    790    437
```

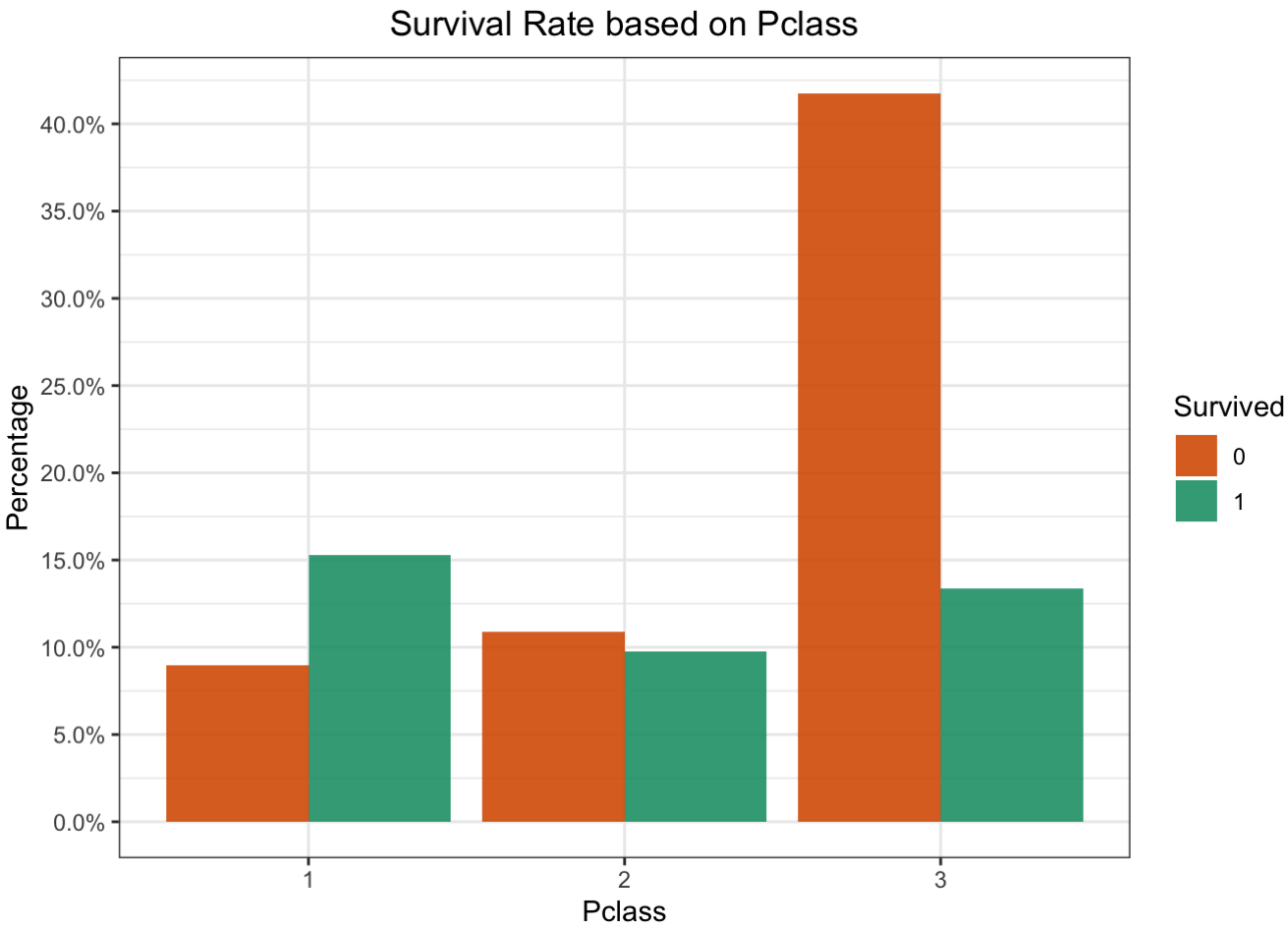
```
titanic$Survived = factor(titanic$Survived)
titanic$Pclass = factor(titanic$Pclass)
titanic$Sex = factor(titanic$Sex)
titanic$Embarked = factor(titanic$Embarked)
titanic$Title = factor(titanic$Title)
titanic$FamilySize = factor(titanic$FamilySize, levels=c("Single","Small","Large"))

str(titanic)
```

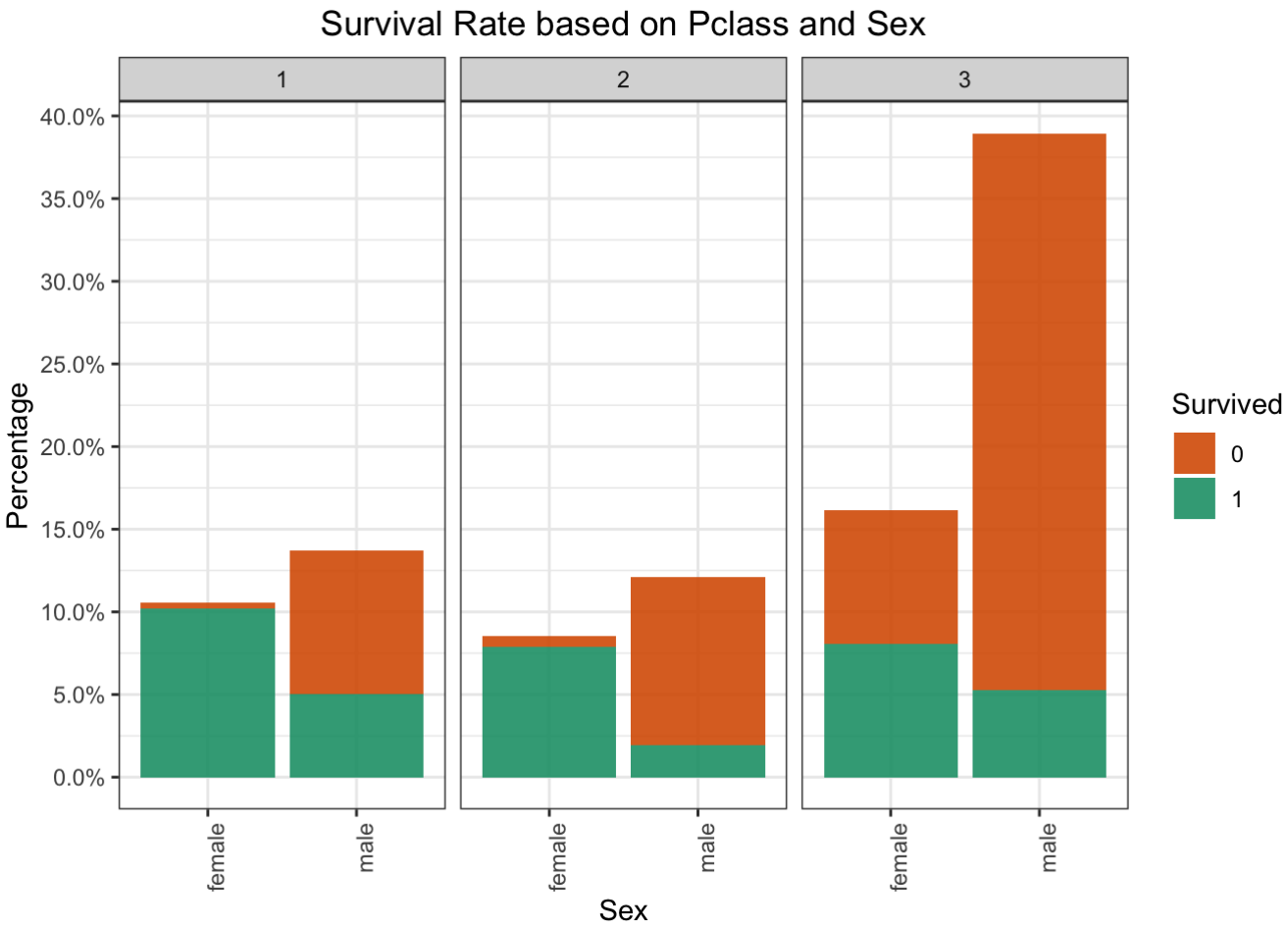
```
## 'data.frame':    1309 obs. of  14 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass     : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence
Briggs Thayer)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques Heath (Lily May Peel)"
...
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age        : num  22 38 26 35 35 25 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
##  $ Title      : Factor w/ 5 levels "Master","Miss",..: 3 4 2 4 3 3 3 1 4 4 ...
##  $ FamilySize : Factor w/ 3 levels "Single","Small",..: 2 2 1 2 1 1 1 3 2 2 ...
```

EDA

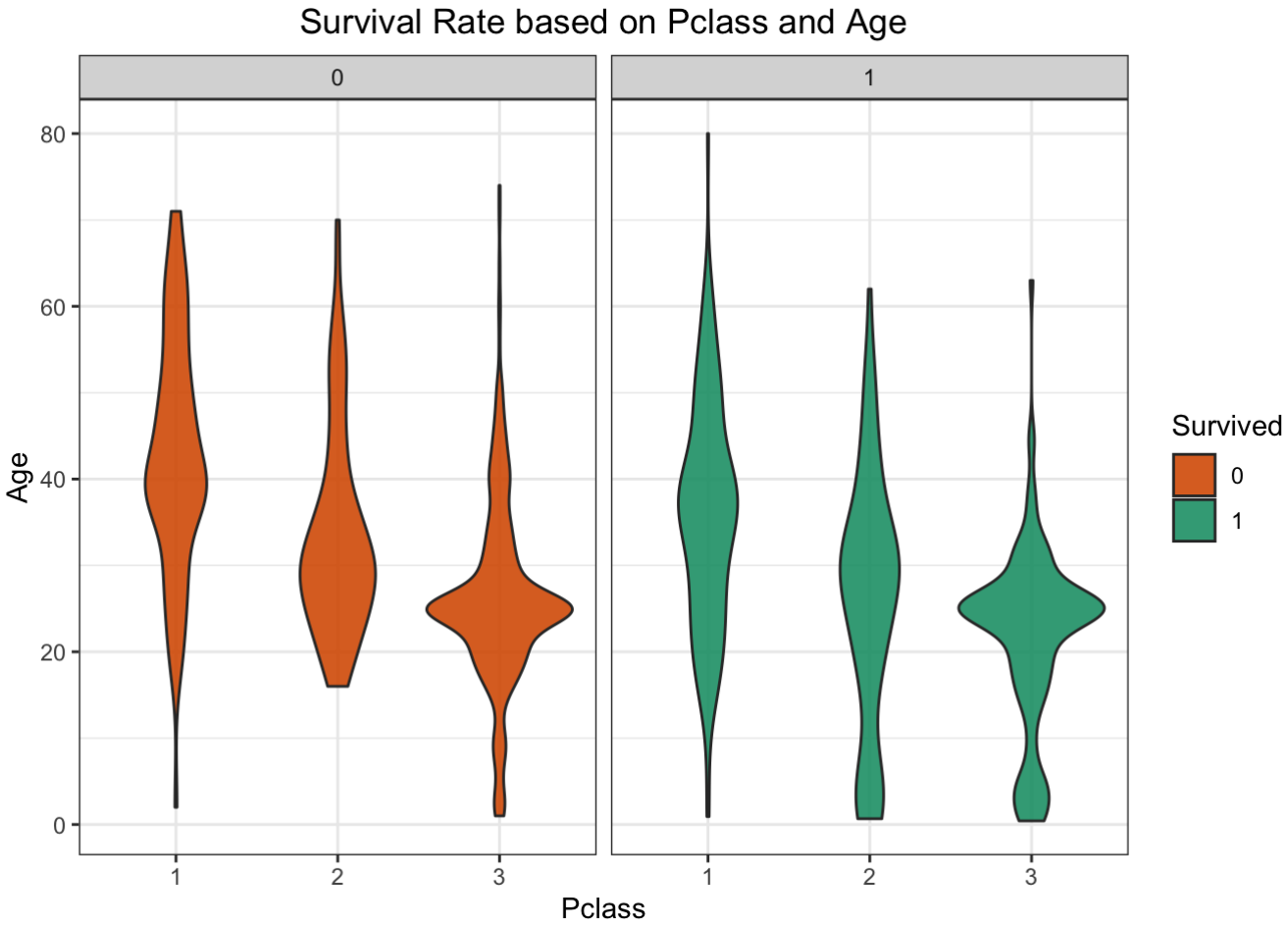
```
ggplot(filter(titanic, is.na(Survived)==FALSE), aes(Pclass, fill=Survived)) +
  geom_bar(aes(y = (..count..)/sum(..count..)), alpha=0.9, position="dodge") +
  scale_fill_brewer(palette = "Dark2", direction = -1) +
  scale_y_continuous(labels=percent, breaks=seq(0,0.6,0.05)) +
  ylab("Percentage") +
  ggtitle("Survival Rate based on Pclass") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



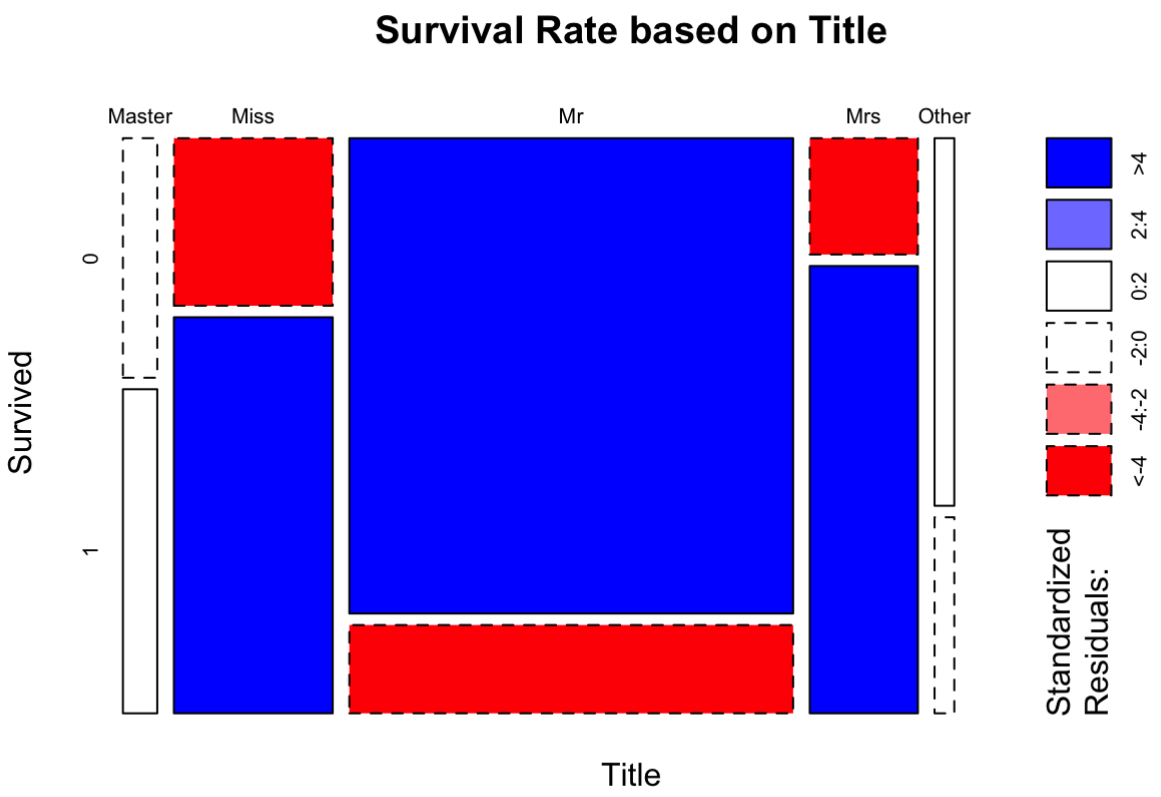

```
ggplot(filter(titanic, is.na(Survived)==FALSE), aes(Sex, fill=Survived)) +
  geom_bar(aes(y = (..count..)/sum(..count..)), alpha=0.9) +
  facet_wrap(~Pclass) +
  scale_fill_brewer(palette = "Dark2", direction = -1) +
  scale_y_continuous(labels=percent, breaks=seq(0,0.4,0.05)) +
  ylab("Percentage") +
  ggtitle("Survival Rate based on Pclass and Sex") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



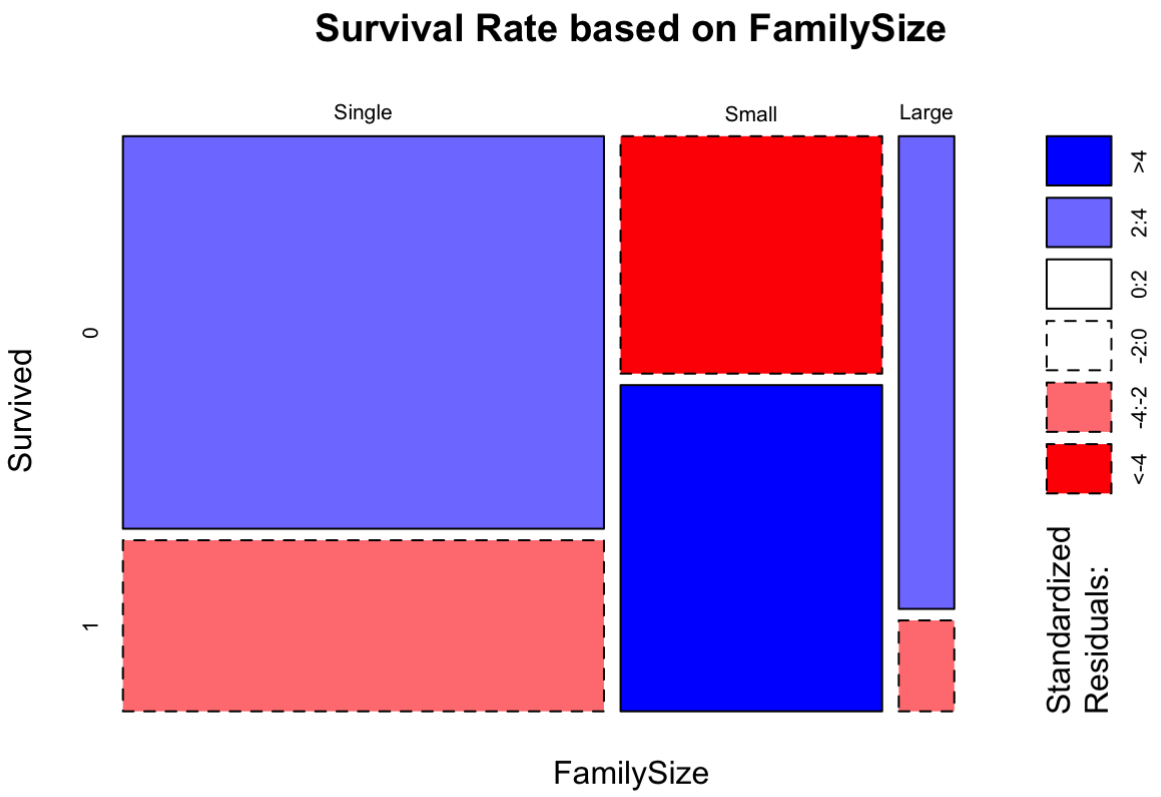

```
ggplot(filter(titanic, is.na(Survived)==FALSE), aes(Pclass, Age)) +
  geom_violin(aes(fill=Survived), alpha=0.9) +
  facet_wrap(~Survived) +
  scale_fill_brewer(palette = "Dark2", direction = -1) +
  ggtitle("Survival Rate based on Pclass and Age") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



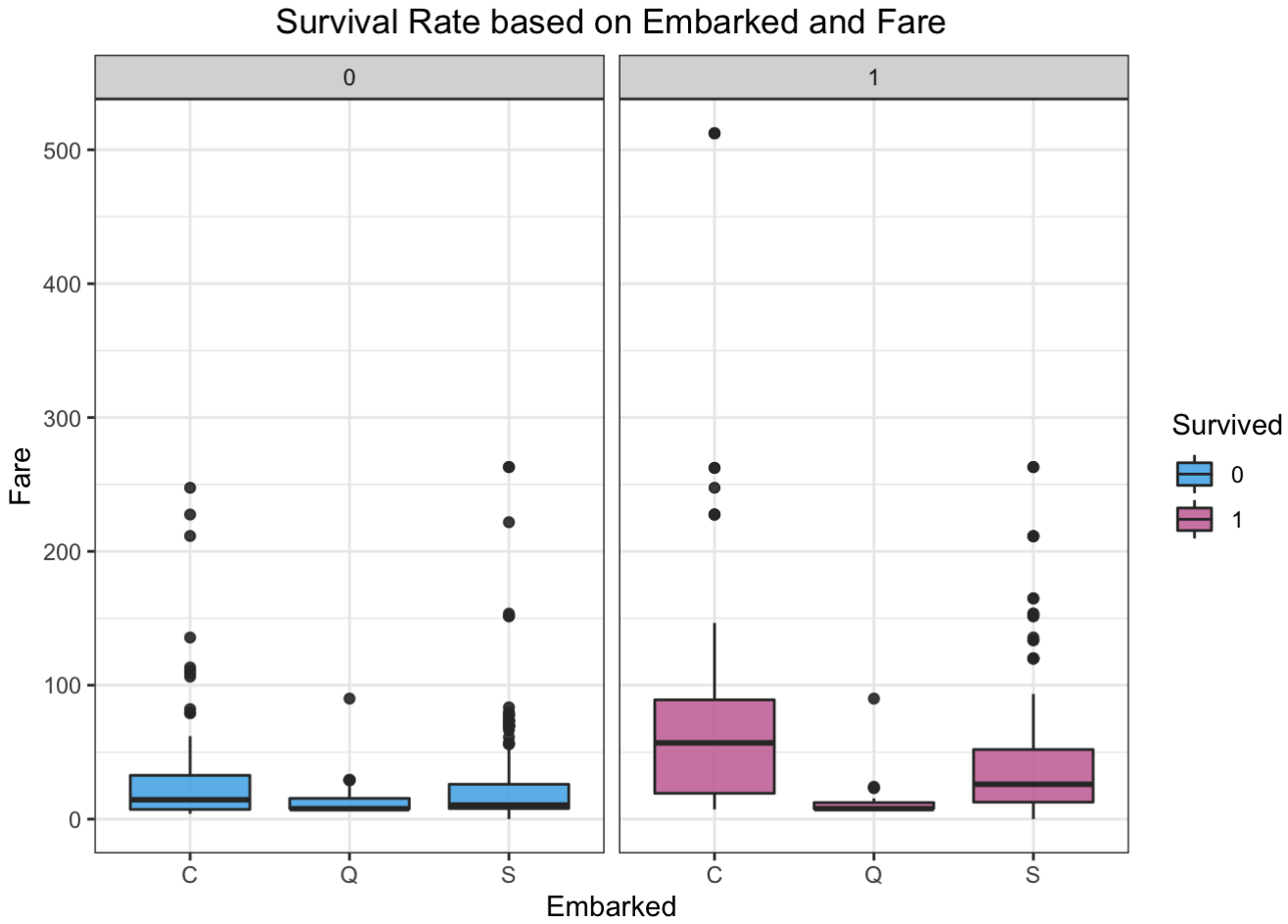

```
mosaicplot(~ Title + Survived, data=titanic, main='Survival Rate based on Title', shade=
TRUE)
```




```
mosaicplot(~ FamilySize + Survived, data=titanic, main='Survival Rate based on FamilySiz
e', shade=TRUE)
```




```
ggplot(filter(titanic, is.na(Survived)==FALSE), aes(Embarked, Fare)) +
  geom_boxplot(aes(fill=Survived), alpha=0.9) +
  facet_wrap(~Survived) +
  scale_fill_manual(values=c("#56B4E9", "#CC79A7")) +
  ggtitle("Survival Rate based on Embarked and Fare") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```




```
train_original <- titanic[1:891, c("Survived","Pclass","Sex","Age","SibSp","Parch","Far
e","Embarked","Title","FamilySize")]
test_original <- titanic[892:1309, c("Pclass","Sex","Age","SibSp","Parch","Fare","Embark
ed","Title","FamilySize")]
```

```
set.seed(789)
split = sample.split(train_original$Survived, SplitRatio = 0.8)
train = subset(train_original, split == TRUE)
test = subset(train_original, split == FALSE)
```

Logistic regression

```
cor(train[,unlist(lapply(train,is.numeric))])
```

```
##              Age      SibSp      Parch      Fare
## Age    1.0000000 -0.2758417 -0.2079948 0.1107712
## SibSp -0.2758417  1.0000000  0.4529568 0.1571153
## Parch -0.2079948  0.4529568  1.0000000 0.2361560
## Fare   0.1107712  0.1571153  0.2361560 1.0000000
```

```
ps = chisq.test(train$Pclass, train$Sex)$p.value
pe = chisq.test(train$Pclass, train$Embarked)$p.value
pt = chisq.test(train$Pclass, train$Title)$p.value
pf = chisq.test(train$Pclass, train$FamilySize)$p.value
se = chisq.test(train$Sex, train$Embarked)$p.value
st = chisq.test(train$Sex, train$Title)$p.value
sf = chisq.test(train$Sex, train$FamilySize)$p.value
et = chisq.test(train$Embarked, train$Title)$p.value
ef = chisq.test(train$Embarked, train$FamilySize)$p.value
tf = chisq.test(train$Title, train$FamilySize)$p.value
cormatrix = matrix(c(0, ps, pe, pt, pf,
                     ps, 0, se, st, sf,
                     pe, se, 0, et, ef,
                     pt, st, et, 0, tf,
                     pf, sf, ef, tf, 0),
                   5, 5, byrow = TRUE)
row.names(cormatrix) = colnames(cormatrix) = c("Pclass", "Sex", "Embarked", "Title", "Fa
milySize")
cormatrix
```

```
##                  Pclass           Sex     Embarked         Title   FamilySize
## Pclass     0.000000e+00  2.532566e-03 1.053100e-23  5.962301e-10 1.108964e-10
## Sex        2.532566e-03  0.000000e+00 1.321593e-02 1.116723e-150 4.591649e-15
## Embarked   1.053100e-23  1.321593e-02 0.000000e+00  1.383169e-04 2.490631e-06
## Title      5.962301e-10 1.116723e-150 1.383169e-04  0.000000e+00 2.204782e-51
## FamilySize 1.108964e-10  4.591649e-15 2.490631e-06  2.204782e-51 0.000000e+00
```

```
classifier = glm(Survived ~ ., family = binomial(link='logit'), data = train)
classifier <- step(classifier)
```

```
## Start:  AIC=612.29
## Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked +
##     Title + FamilySize
##
##              Df Deviance    AIC
## - SibSp       1   580.29 610.29
## - Embarked    2   582.69 610.69
## - Fare        1   580.81 610.81
## - Parch       1   581.59 611.59
## <none>            580.29 612.29
## - Sex         1   584.37 614.37
## - Age         1   585.69 615.69
## - FamilySize  2   590.68 618.68
## - Title       4   616.14 640.14
## - Pclass      2   624.73 652.73
##
## Step:  AIC=610.29
## Survived ~ Pclass + Sex + Age + Parch + Fare + Embarked + Title +
##     FamilySize
##
##              Df Deviance    AIC
## - Embarked    2   582.71 608.71
## - Fare        1   580.82 608.82
## - Parch       1   582.05 610.05
## <none>            580.29 610.29
## - Sex         1   584.37 612.37
## - Age         1   585.70 613.70
## - FamilySize  2   609.33 635.33
## - Title       4   616.76 638.76
## - Pclass      2   624.87 650.87
##
## Step:  AIC=608.71
## Survived ~ Pclass + Sex + Age + Parch + Fare + Title + FamilySize
##
##              Df Deviance    AIC
## - Fare        1   583.56 607.56
## - Parch       1   584.41 608.41
## <none>            582.71 608.71
## - Sex         1   586.56 610.56
## - Age         1   588.11 612.11
## - FamilySize  2   615.00 637.00
## - Title       4   619.32 637.32
## - Pclass      2   628.03 650.03
##
## Step:  AIC=607.56
## Survived ~ Pclass + Sex + Age + Parch + Title + FamilySize
##
##              Df Deviance    AIC
## - Parch       1   585.45 607.45
## <none>            583.56 607.56
## - Sex         1   587.31 609.31
## - Age         1   589.24 611.24
## - FamilySize  2   615.02 635.02
```

```
## - Title       4   619.43 635.43
## - Pclass      2   662.23 682.23
##
## Step:  AIC=607.45
## Survived ~ Pclass + Sex + Age + Title + FamilySize
##
##              Df Deviance    AIC
## <none>            585.45 607.45
## - Sex         1   589.15 609.15
## - Age         1   591.29 611.29
## - Title       4   623.47 637.47
## - FamilySize  2   622.80 640.80
## - Pclass      2   664.64 682.64
```

```
summary(classifier)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + Title + FamilySize,
##     family = binomial(link = "logit"), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6190  -0.5712  -0.3802   0.5462   2.4571
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      18.72233  506.79974   0.037 0.970531
## Pclass2          -1.42636    0.32191  -4.431 9.38e-06 ***
## Pclass3          -2.55761    0.31174  -8.204 2.32e-16 ***
## Sexmale         -14.63628  506.79929  -0.029 0.976960
## Age              -0.02518    0.01062  -2.370 0.017789 *
## TitleMiss       -15.04068  506.79960  -0.030 0.976324
## TitleMr          -3.36409    0.60123  -5.595 2.20e-08 ***
## TitleMrs        -14.59001  506.79969  -0.029 0.977033
## TitleOther       -2.96720    0.85828  -3.457 0.000546 ***
## FamilySizeSmall  -0.23457    0.25791  -0.909 0.363087
## FamilySizeLarge  -2.65324    0.50371  -5.267 1.38e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 949.90  on 712  degrees of freedom
## Residual deviance: 585.45  on 702  degrees of freedom
## AIC: 607.45
##
## Number of Fisher Scoring iterations: 13
```

```
vif(classifier)
```

```
##                    GVIF Df GVIF^(1/(2*Df))
## Pclass     1.667655e+00  2        1.136388
## Sex        5.751701e+06  1     2398.270329
## Age        1.894599e+00  1        1.376444
## Title      1.224494e+07  4        7.691208
## FamilySize 1.812345e+00  2        1.160273
```

```
classifier = glm(Survived ~ . -Sex, family = binomial(link='logit'), data = train)
```

```
classifier <- step(classifier)
```

```
## Start:  AIC=614.37
## Survived ~ (Pclass + Sex + Age + SibSp + Parch + Fare + Embarked +
##     Title + FamilySize) - Sex
##
##              Df Deviance    AIC
## - SibSp       1   584.37 612.37
## - Embarked    2   586.52 612.52
## - Fare        1   584.83 612.83
## - Parch       1   585.58 613.58
## <none>            584.37 614.37
## - Age         1   589.95 617.95
## - FamilySize  2   594.60 620.60
## - Pclass      2   629.59 655.59
## - Title       4   772.00 794.00
##
## Step:  AIC=612.37
## Survived ~ Pclass + Age + Parch + Fare + Embarked + Title + FamilySize
##
##              Df Deviance    AIC
## - Embarked    2   586.56 610.56
## - Fare        1   584.84 610.84
## - Parch       1   586.10 612.10
## <none>            584.37 612.37
## - Age         1   589.95 615.95
## - FamilySize  2   613.53 637.53
## - Pclass      2   629.78 653.78
## - Title       4   772.02 792.02
##
## Step:  AIC=610.56
## Survived ~ Pclass + Age + Parch + Fare + Title + FamilySize
##
##              Df Deviance    AIC
## - Fare        1   587.31 609.31
## - Parch       1   588.22 610.22
## <none>            586.56 610.56
## - Age         1   592.09 614.09
## - FamilySize  2   618.78 638.78
## - Pclass      2   632.93 652.93
## - Title       4   783.98 799.98
##
## Step:  AIC=609.31
## Survived ~ Pclass + Age + Parch + Title + FamilySize
##
##              Df Deviance    AIC
## - Parch       1   589.15 609.15
## <none>            587.31 609.31
## - Age         1   593.14 613.14
## - FamilySize  2   618.78 636.78
## - Pclass      2   667.53 685.53
## - Title       4   785.52 799.52
##
## Step:  AIC=609.15
## Survived ~ Pclass + Age + Title + FamilySize
```

```
##
##              Df Deviance    AIC
## <none>            589.15 609.15
## - Age         1   595.17 613.17
## - FamilySize  2   626.64 642.64
## - Pclass      2   669.92 685.92
## - Title       4   793.80 805.80
```

```
summary(classifier)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Age + Title + FamilySize, family = binomial(link =
"logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6304  -0.5750  -0.3792   0.5637   2.4607
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      4.11423    0.69575   5.913 3.35e-09 ***
## Pclass2         -1.46793    0.32032  -4.583 4.59e-06 ***
## Pclass3         -2.58021    0.31125  -8.290  < 2e-16 ***
## Age             -0.02543    0.01059  -2.403  0.01627 *
## TitleMiss       -0.40382    0.55940  -0.722  0.47037
## TitleMr         -3.36730    0.60132  -5.600 2.15e-08 ***
## TitleMrs         0.05208    0.63190   0.082  0.93431
## TitleOther      -2.56210    0.81122  -3.158  0.00159 **
## FamilySizeSmall -0.23202    0.25621  -0.906  0.36516
## FamilySizeLarge -2.65769    0.50374  -5.276 1.32e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 949.90  on 712  degrees of freedom
## Residual deviance: 589.15  on 703  degrees of freedom
## AIC: 609.15
##
## Number of Fisher Scoring iterations: 5
```

```
vif(classifier)
```

```
##                GVIF Df GVIF^(1/(2*Df))
## Pclass     1.688094  2        1.139854
## Age        1.909003  1        1.381667
## Title      2.618396  4        1.127858
## FamilySize 1.805038  2        1.159102
```

```
durbinWatsonTest(classifier)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1      0.02148659      1.956557    0.55
##  Alternative hypothesis: rho != 0
```

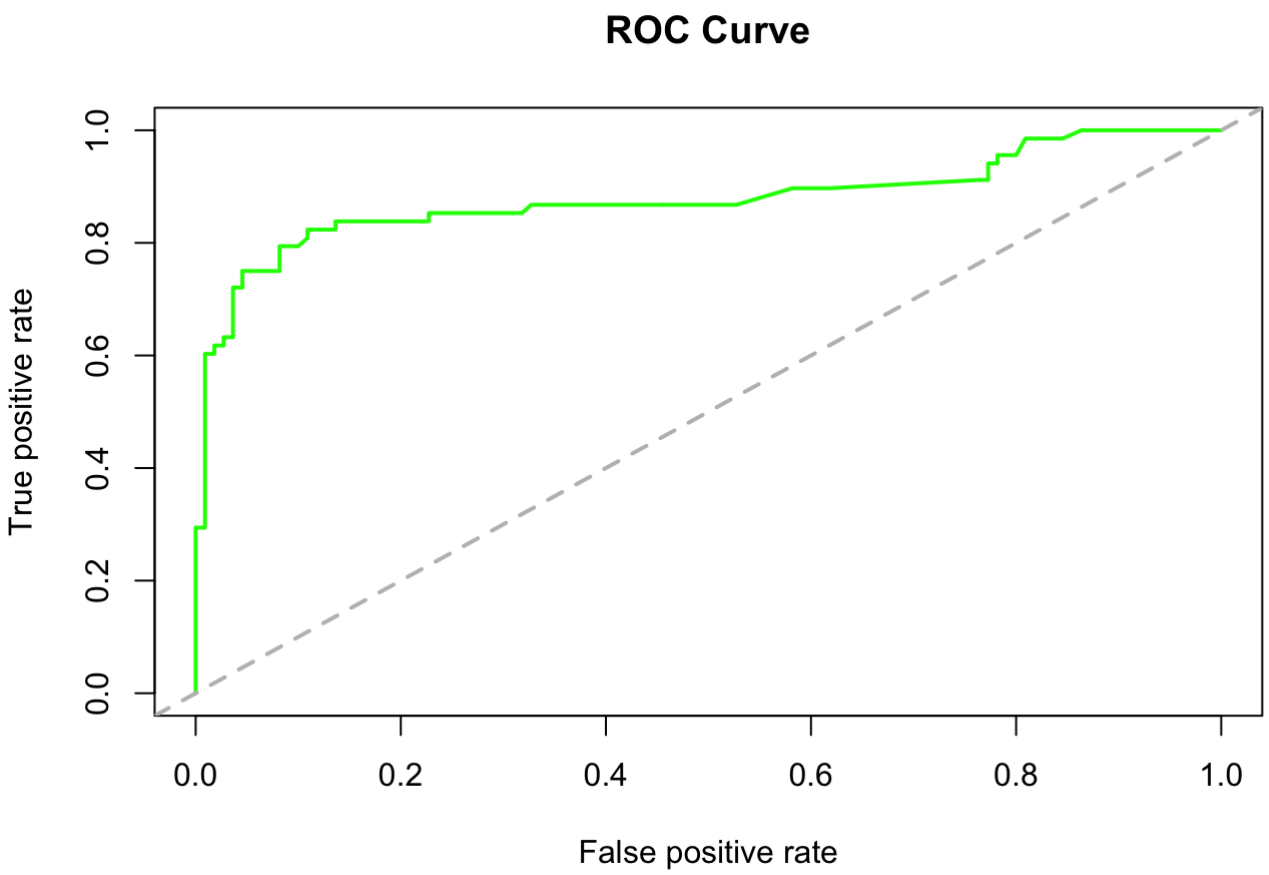
```
prob_pred = predict(classifier, type = 'response', newdata = test)
y_pred = ifelse(prob_pred > 0.5, 1, 0)
table(test$Survived, y_pred > 0.5)
```

```
##
##     FALSE TRUE
##   0   101    9
##   1    17   51
```

```
error <- mean(test$Survived != y_pred)
paste('Accuracy',round(1-error,4))
```

```
## [1] "Accuracy 0.8539"
```

```
fitpred = prediction(prob_pred, test$Survived)
fitperf = performance(fitpred,"tpr","fpr")
plot(fitperf,col="green",lwd=2,main="ROC Curve")
abline(a=0,b=1,lwd=2,lty=2,col="gray")
```



Support Vector Machines

```
paste('Age variance: ',var(train$Age),', SibSp variance: ',var(train$SibSp),', Parch var
iance: ',var(train$Parch),', Fare variance: ',var(train$Fare))
```

```
## [1] "Age variance:  173.992892791181 , SibSp variance:  1.26512441495816 , Parch vari
ance:  0.604594449784894 , Fare variance:  2291.51426667939"
```

```
standardized.train = cbind(select(train, Survived, Pclass, Sex, SibSp, Parch, Embarked,
Title, FamilySize), Age = scale(train$Age), Fare = scale(train$Fare))
paste('Age variance: ',var(standardized.train$Age),', Fare variance: ',var(standardized.
train$Fare))
```

```
## [1] "Age variance:  1 , Fare variance:  1"
```

```
standardized.test = cbind(select(test, Survived, Pclass, Sex, SibSp, Parch, Embarked, Ti
tle, FamilySize), Age = scale(test$Age), Fare = scale(test$Fare))
paste('Age variance: ',var(standardized.test$Age),', Fare variance: ',var(standardized.t
est$Fare))
```

```
## [1] "Age variance:  1 , Fare variance:  1"
```

```
classifier = svm(Survived ~ .,
                 data = standardized.train,
                 type = 'C-classification',
                 kernel = 'linear')

y_pred = predict(classifier, newdata = standardized.test[,-which(names(standardized.tes
t)=="Survived")])

table(test$Survived, y_pred)
```

```
##    y_pred
##       0   1
##   0 106   4
##   1  18  50
```

```
error <- mean(test$Survived != y_pred)
paste('Accuracy',round(1-error,4))
```

```
## [1] "Accuracy 0.8764"
```

```
classifier = svm(Survived ~ .,
                 data = standardized.train,
                 type = 'C-classification',
                 kernel = 'radial')

y_pred = predict(classifier, newdata = standardized.test[,-which(names(standardized.tes
t)=="Survived")])

table(test$Survived, y_pred)
```

```
##    y_pred
##       0   1
##   0 105   5
##   1  16  52
```

```
error <- mean(test$Survived != y_pred)
paste('Accuracy',round(1-error,4))
```

```
## [1] "Accuracy 0.882"
```

```
tune.results <- tune(svm,
                     Survived ~ .,
                     data = standardized.train,
                     kernel='radial',
                     ranges=list(cost=2^(-2:2), gamma=2^(-6:-2)))
summary(tune.results)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost gamma
##     4 0.125
##
## - best performance: 0.1794992
##
## - Detailed performance results:
##    cost    gamma     error dispersion
## 1  0.25 0.015625 0.2033646 0.02818232
## 2  0.50 0.015625 0.1936033 0.03253985
## 3  1.00 0.015625 0.1851721 0.03958896
## 4  2.00 0.015625 0.1851526 0.03885835
## 5  4.00 0.015625 0.1893388 0.03426396
## 6  0.25 0.031250 0.1865806 0.03383293
## 7  0.50 0.031250 0.1865415 0.03425425
## 8  1.00 0.031250 0.1865610 0.03670215
## 9  2.00 0.031250 0.1879695 0.03914563
## 10 4.00 0.031250 0.1865610 0.03729787
## 11 0.25 0.062500 0.1837441 0.03694279
## 12 0.50 0.062500 0.1851526 0.03718315
## 13 1.00 0.062500 0.1865806 0.03627857
## 14 2.00 0.062500 0.1838224 0.03791697
## 15 4.00 0.062500 0.1894171 0.04495805
## 16 0.25 0.125000 0.1865415 0.03492662
## 17 0.50 0.125000 0.1837637 0.03710588
## 18 1.00 0.125000 0.1865610 0.04113898
## 19 2.00 0.125000 0.1795579 0.04752411
## 20 4.00 0.125000 0.1794992 0.04997202
## 21 0.25 0.250000 0.1949531 0.03390228
## 22 0.50 0.250000 0.1823161 0.04411747
## 23 1.00 0.250000 0.1795579 0.05355751
## 24 2.00 0.250000 0.1837050 0.06249995
## 25 4.00 0.250000 0.1976330 0.06657420
```

```
classifier = svm(Survived ~ .,
                 data = standardized.train,
                 type = 'C-classification',
                 kernel = 'radial',
                 cost = 4,
                 gamma = 0.125)

y_pred = predict(classifier, newdata = standardized.test[,-which(names(standardized.tes
t)=="Survived")])

table(test$Survived, y_pred)
```

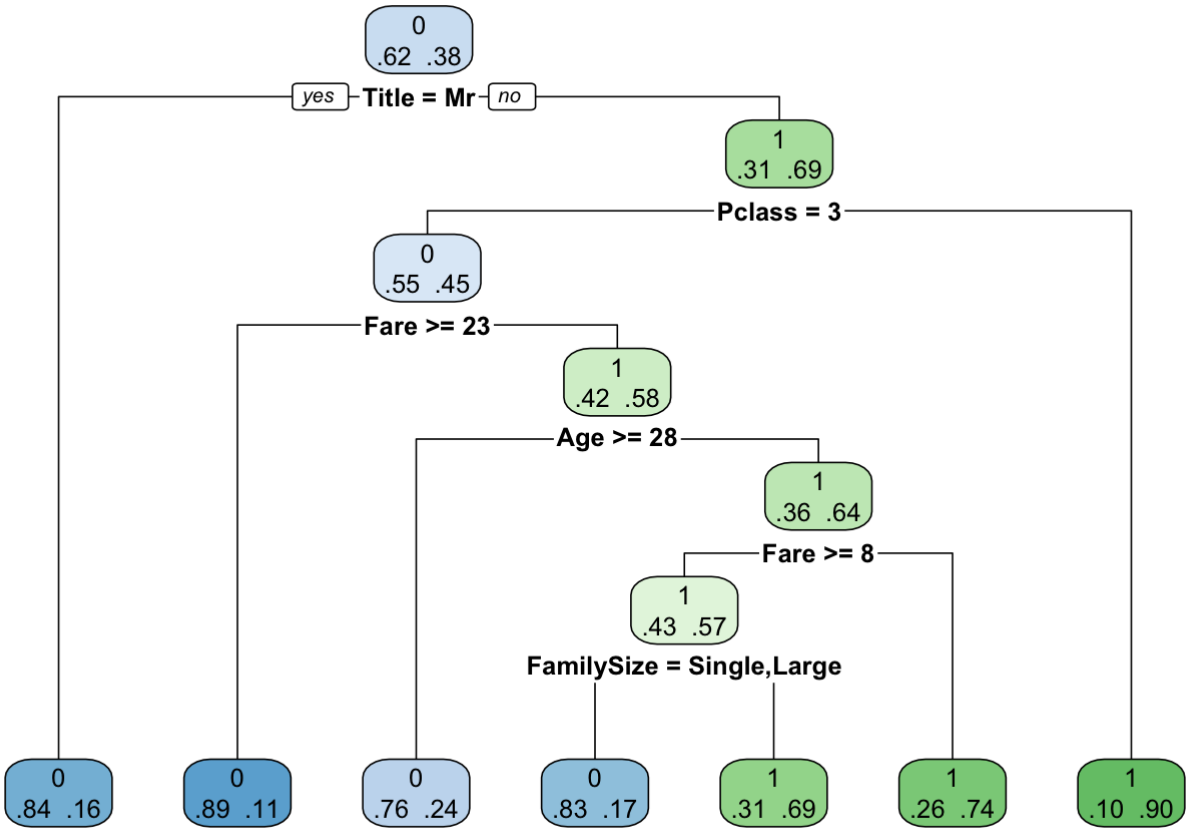
```
##     y_pred
##        0   1
##    0 104   6
##    1  24  44
```

```
error <- mean(test$Survived != y_pred)
paste('Accuracy',round(1-error,4))
```

```
## [1] "Accuracy 0.8315"
```

Decision Tree

```
classifier = rpart(Survived ~ ., data = train, method = 'class')
rpart.plot(classifier, extra=4)
```



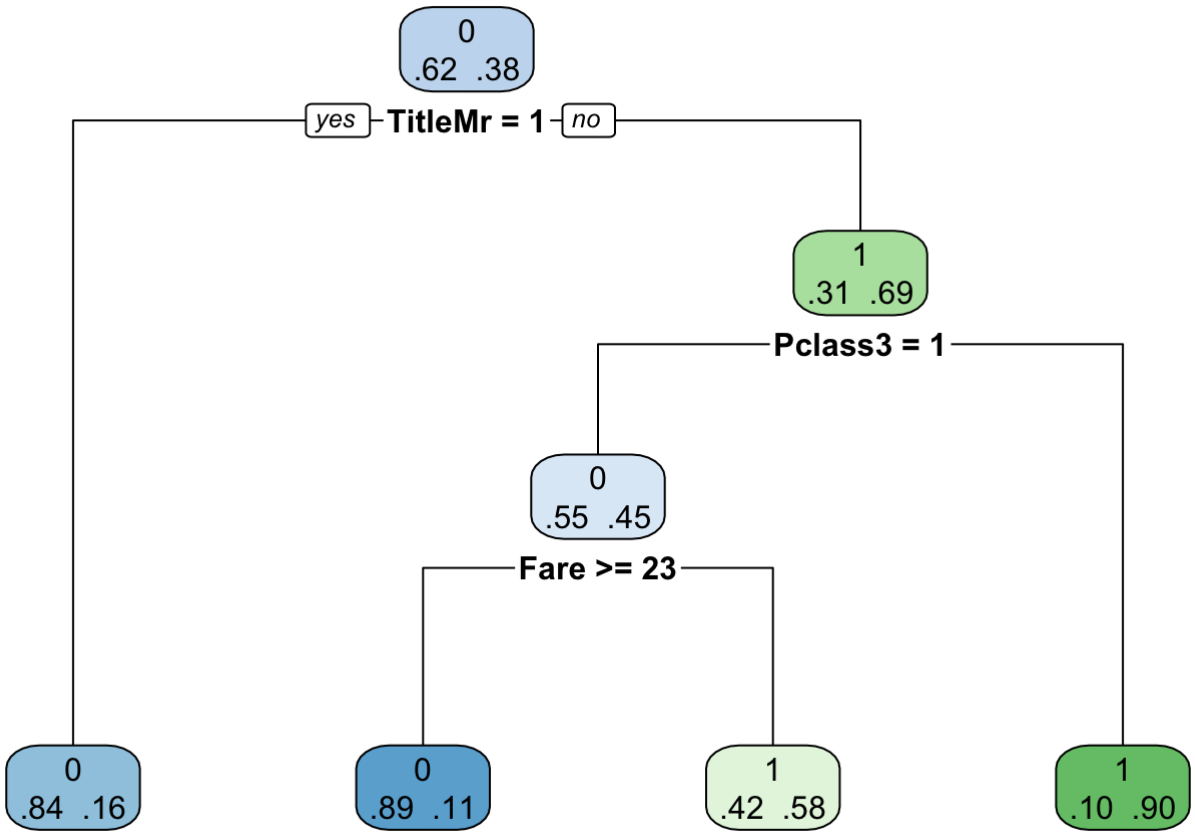

```
y_pred = predict(classifier, newdata = test[,-which(names(test)=="Survived")], type='cla
ss')
table(test$Survived, y_pred)
```

```
##     y_pred
##        0   1
##    0 102   8
##    1  21  47
```

```
error <- mean(test$Survived != y_pred)
paste('Accuracy',round(1-error,4))
```

```
## [1] "Accuracy 0.8371"
```

```
set.seed(789)
folds = createMultiFolds(train$Survived, k = 10, times = 5)
control <- trainControl(method = "repeatedcv", index = folds)
classifier_cv <- train(Survived ~ ., data = train, method = "rpart", trControl = contro
l)
rpart.plot(classifier_cv$finalModel, extra=4)
```




```
y_pred = predict(classifier_cv, newdata = test[,-which(names(test)=="Survived")])
table(test$Survived, y_pred)
```

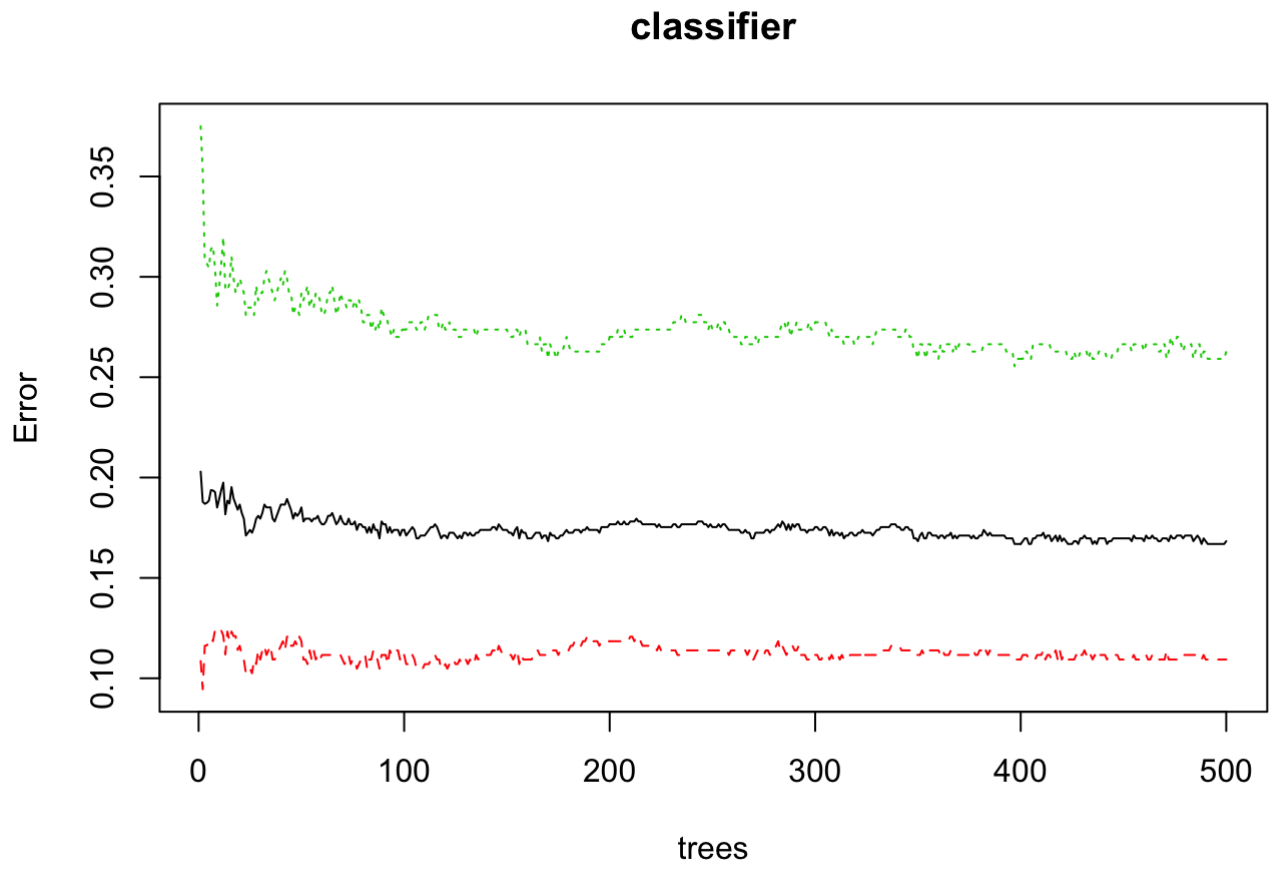
```
##     y_pred
##       0  1
##    0 99 11
##    1 17 51
```

```
error <- mean(test$Survived != y_pred)
paste('Accuracy',round(1-error,4))
```

```
## [1] "Accuracy 0.8427"
```

Random Forests

```
set.seed(432)
classifier = randomForest(Survived ~ ., data = train)
plot(classifier)
```




```
y_pred = predict(classifier, newdata = test[,-which(names(test)=="Survived")])
table(test$Survived, y_pred)
```

```
##    y_pred
##       0  1
##   0 99 11
##   1 18 50
```

```
error <- mean(test$Survived != y_pred)
paste('Accuracy',round(1-error,4))
```

```
## [1] "Accuracy 0.8371"
```

```
set.seed(651)
folds = createMultiFolds(train$Survived, k = 10)
control <- trainControl(method = "repeatedcv", index = folds)
classifier_cv <- train(Survived ~ ., data = train, method = "rf", trControl = control)
y_pred = predict(classifier_cv, newdata = test[,-which(names(test)=="Survived")])
table(test$Survived, y_pred)
```

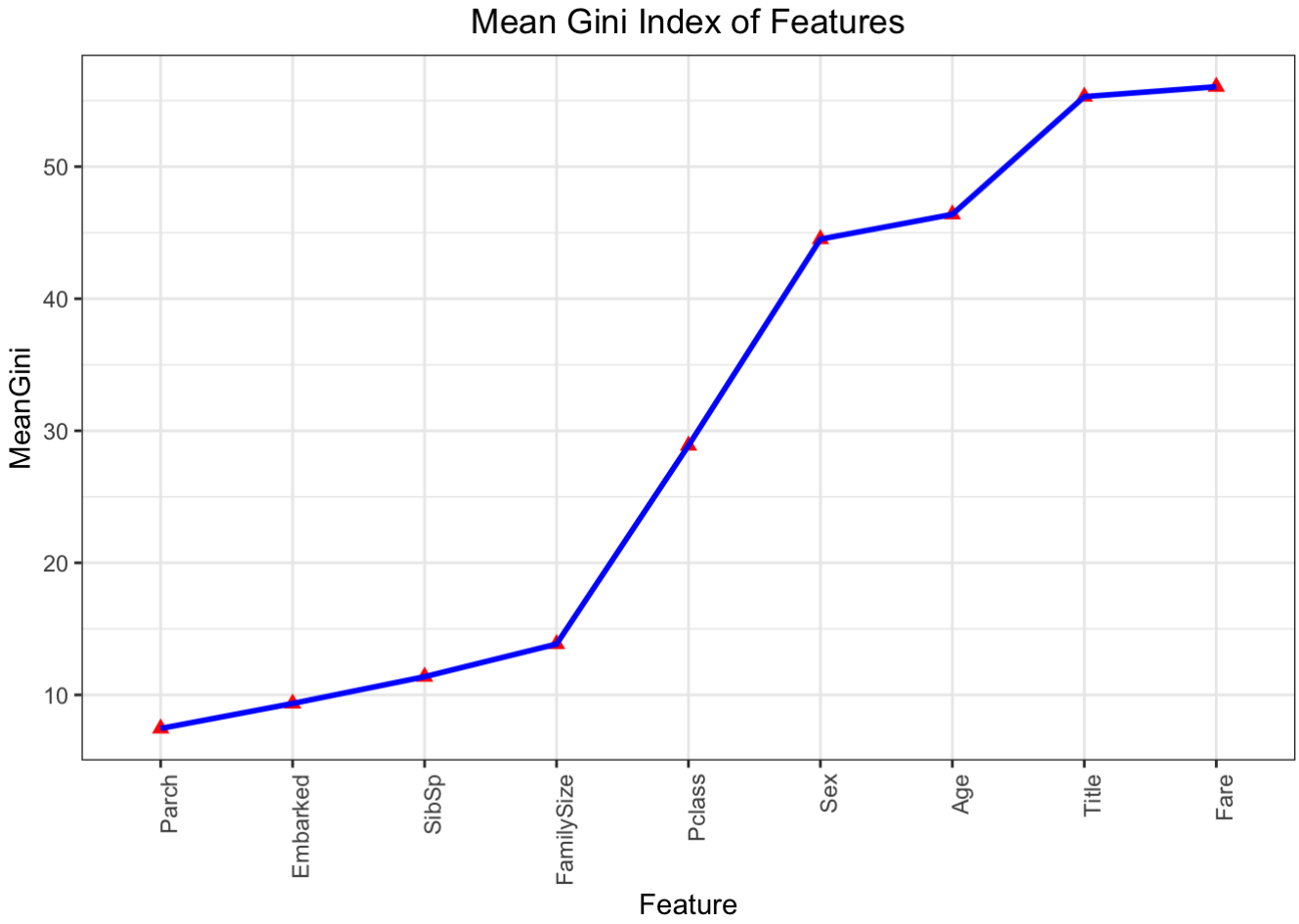
```
##    y_pred
##       0  1
##   0 94 16
##   1 19 49
```

```
error <- mean(test$Survived != y_pred)
paste('Accuracy',round(1-error,4))
```

```
## [1] "Accuracy 0.8034"
```

```
gini = as.data.frame(importance(classifier))
gini = data.frame(Feature = row.names(gini),
                  MeanGini = round(gini[ ,'MeanDecreaseGini'], 2))
gini = gini[order(-gini[,"MeanGini"]),]

ggplot(gini,aes(reorder(Feature,MeanGini), MeanGini, group=1)) +
  geom_point(color='red',shape=17,size=2) +
  geom_line(color='blue',size=1) +
  scale_y_continuous(breaks=seq(0,60,10)) +
  xlab("Feature") +
  ggtitle("Mean Gini Index of Features") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



Naive Bayes

```
classifier = naiveBayes(Survived ~ ., data = train)
y_pred = predict(classifier, newdata = test[,-which(names(test)=="Survived")])
table(test$Survived, y_pred)
```

```
##    y_pred
##       0  1
##   0 99 11
##   1 17 51
```

```
error <- mean(test$Survived != y_pred)
paste('Accuracy',round(1-error,4))
```

```
## [1] "Accuracy 0.8427"
```

Results

```
y_pred = predict(classifier, newdata = test_original)
results <- data.frame(PassengerID = titanic[892:1309,"PassengerId"], Survived = y_pred)
write.csv(results, file = 'PredictingTitanicSurvival.csv', row.names = FALSE, quote=FALS
E)
```