titanic.R

anweshaghosh

Fri Oct 21 18:28:12 2016

```r
library(data.table)
readData <- function(file.name, column.types, missing.types)
{
  read.csv(file.name,
        colClasses = column.types,
        na.strings=missing.types,
        stringsAsFactors = FALSE)
}
train.data.file <- "train.csv"
test.data.file <- "test.csv"
missing.types <- c("NA","")
train.column.types <- c('integer',   # PassengerId
                        'factor',    # Survived
                        'factor',    # Pclass
                        'character', # Name
                        'factor',    # Sex
                        'numeric',   # Age
                        'integer',   # SibSp
                        'integer',   # Parch
                        'character', # Ticket
                        'numeric',   # Fare
                        'character', # Cabin
                        'factor'     # Embarked
                        )
test.column.types <- train.column.types[-2]
## No Survived Column in test.csv

train <- readData(train.data.file, train.column.types, missing.types)
test <- readData(test.data.file, test.column.types, missing.types)


## Exploring Data
summary(train)
```

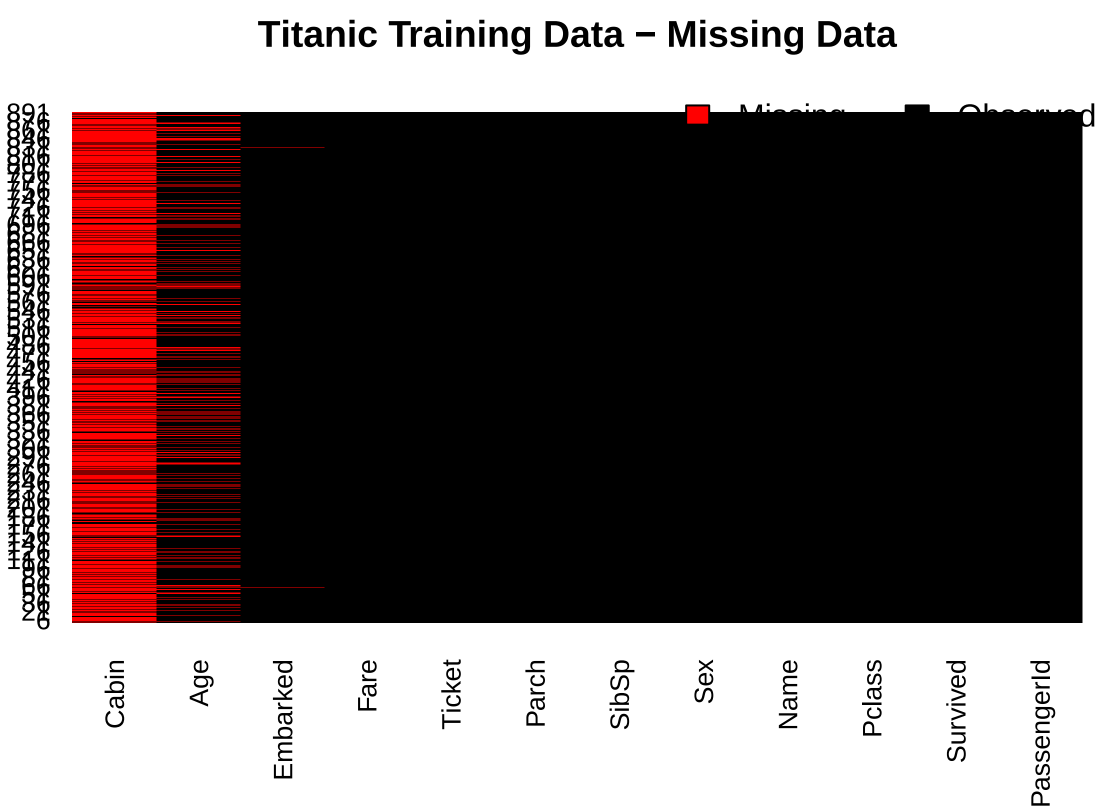
```
##   PassengerId    Survived Pclass      Name               Sex
##  Min.   :  1.0   0:549    1:216   Length:891         female:314
##  1st Qu.:223.5   1:342    2:184   Class :character   male  :577
##  Median :446.0            3:491   Mode  :character
##  Mean   :446.0
##  3rd Qu.:668.5
##  Max.   :891.0
##
##       Age            SibSp           Parch            Ticket
##  Min.   : 0.42   Min.   :0.000   Min.   :0.0000   Length:891
##  1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000   Class :character
##  Median :28.00   Median :0.000   Median :0.0000   Mode  :character
##  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
##       Fare           Cabin            Embarked
##  Min.   :  0.00   Length:891         C   :168
##  1st Qu.:  7.91   Class :character   Q   : 77
##  Median : 14.45   Mode  :character   S   :644
##  Mean   : 32.20                      NA's:  2
##  3rd Qu.: 31.00
##  Max.   :512.33
##
```

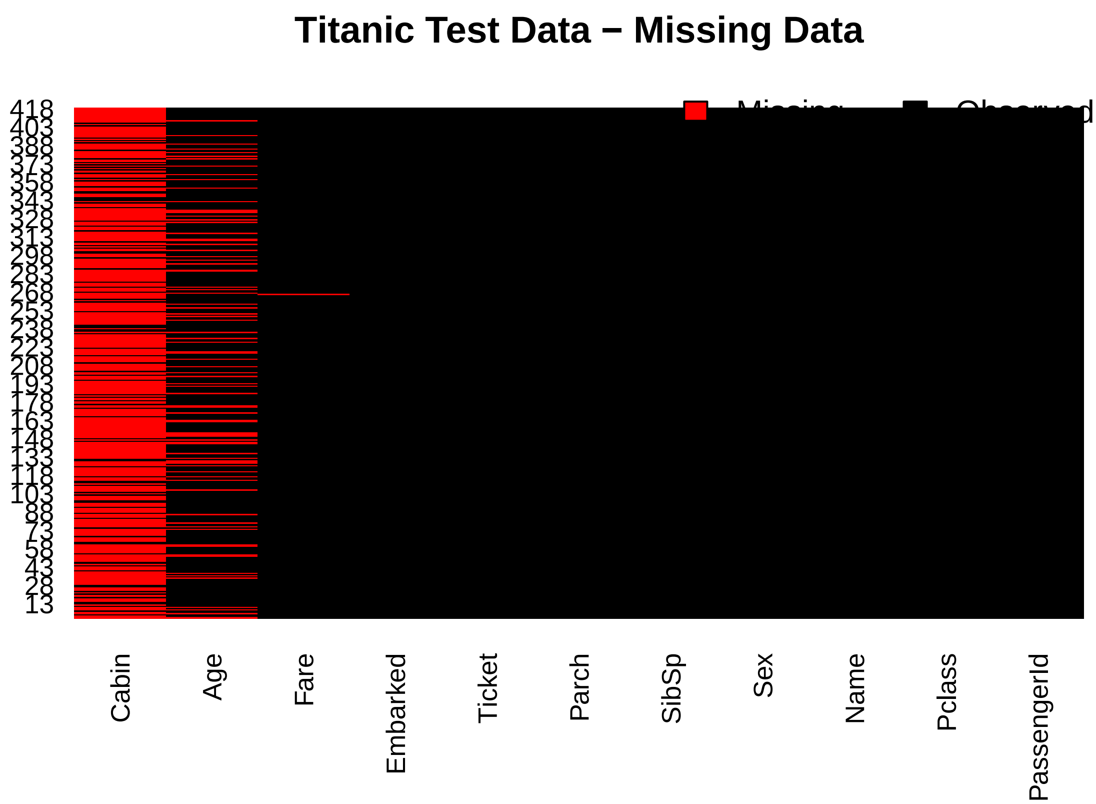
```r
## Missing Data
require(Amelia)
```

```
## Loading required package: Amelia

## Loading required package: Rcpp

## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.4, built: 2015-12-05)
## ## Copyright (C) 2005-2016 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

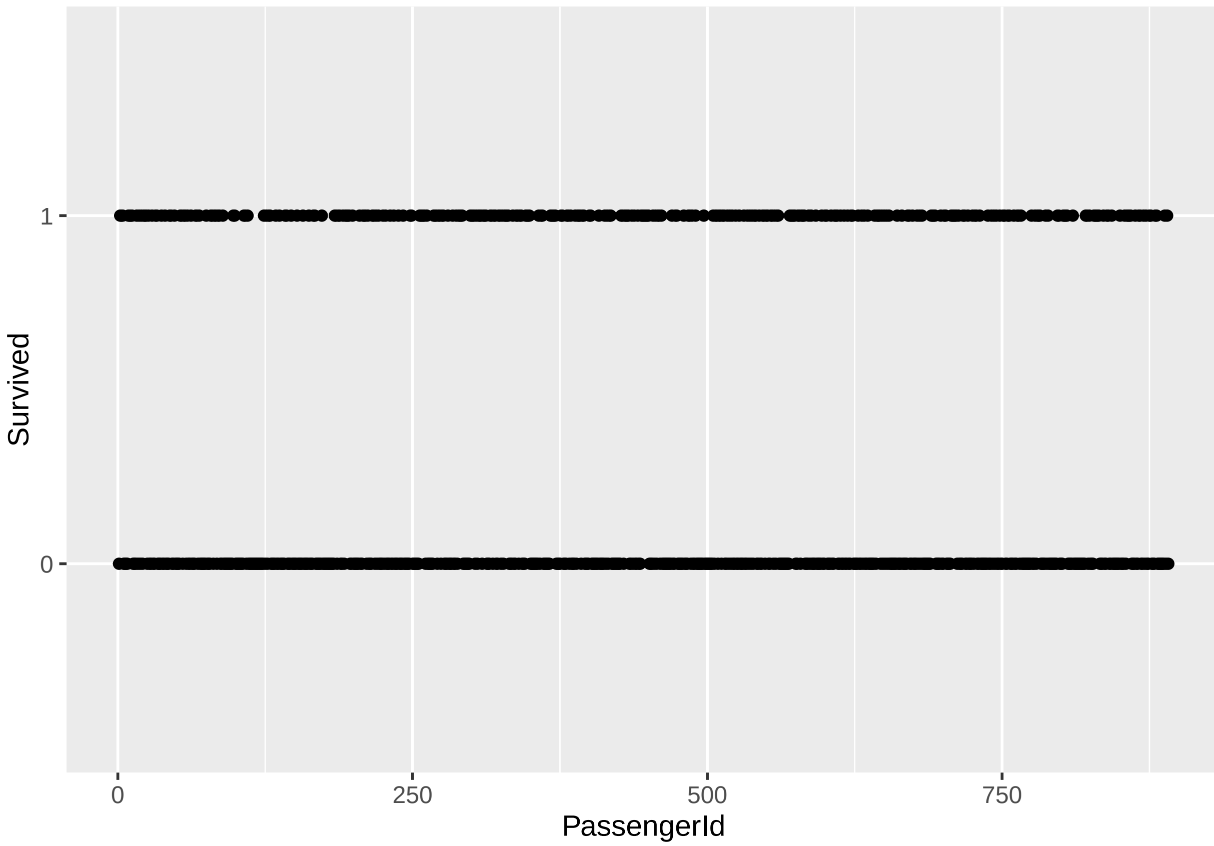
```r
missmap(train, main = "Titanic Training Data - Missing Data", col = c("red","black"))
```



```r
missmap(test, main = "Titanic Test Data - Missing Data", col = c("red","black"))

## Data Visualization
library(ggplot2)
```

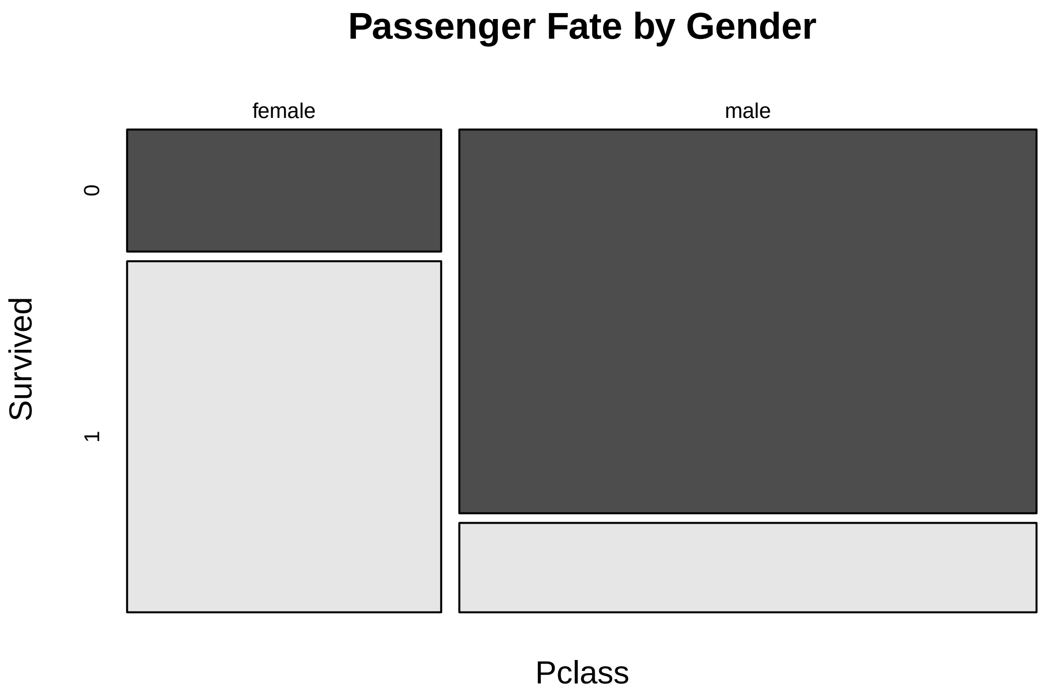



```r
library(ggthemes)

# PassengerId
pi1 <- ggplot(train, aes(x = PassengerId, y = Survived))
pi1 +geom_point()
```



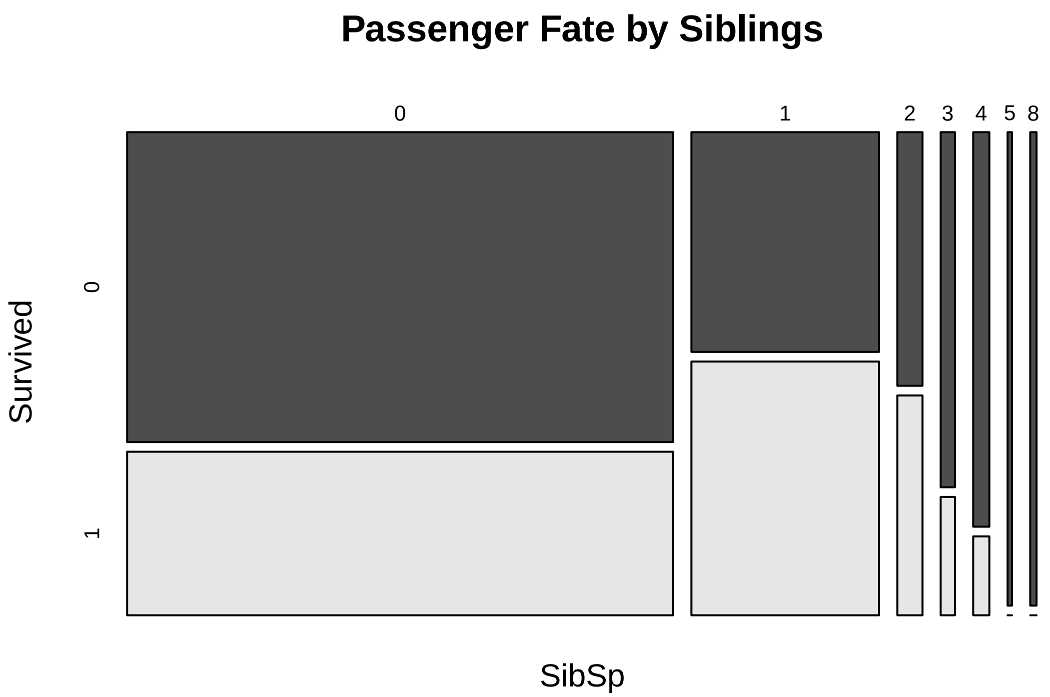

```r
# No obvious relationship found

# Pclass
mosaicplot(train$Pclass ~ train$Survived, main = "Passenger Fate by Traveling Class", shade = FALSE, col
```

```
# Sex - Yes
mosaicplot(train$Sex ~ train$Survived, main = "Passenger Fate by Gender", shade = FALSE, color = TRUE, 
```

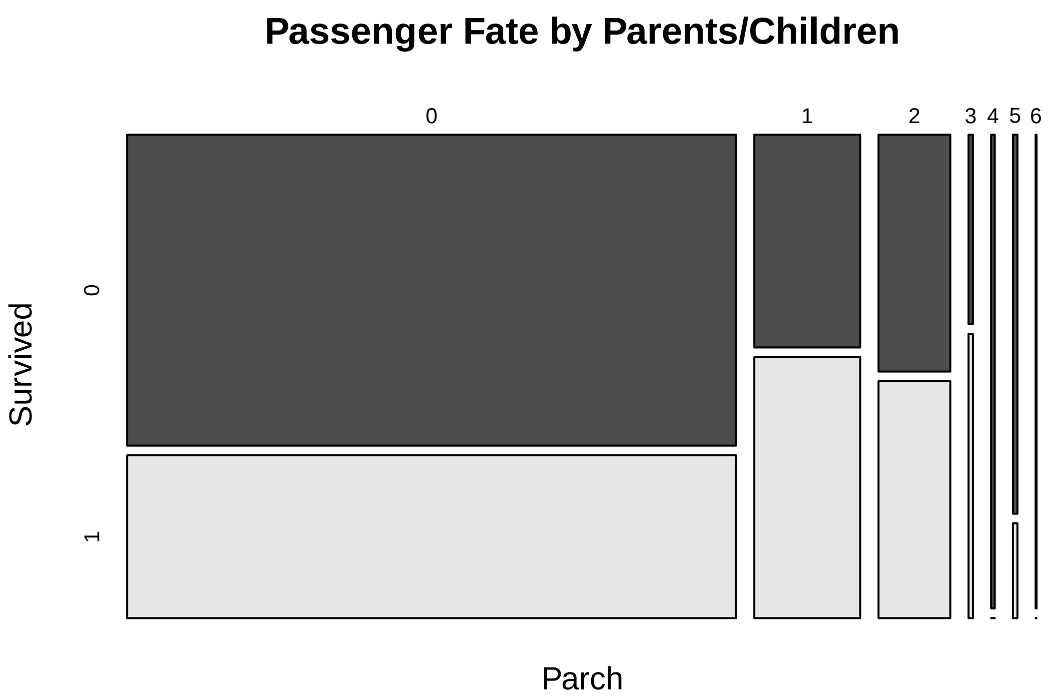



```
# Age - Yes
boxplot(train$Age ~ train$Survived, main= "Passenger Fate by Age", xlab = "Survived", ylab = "Age")
```

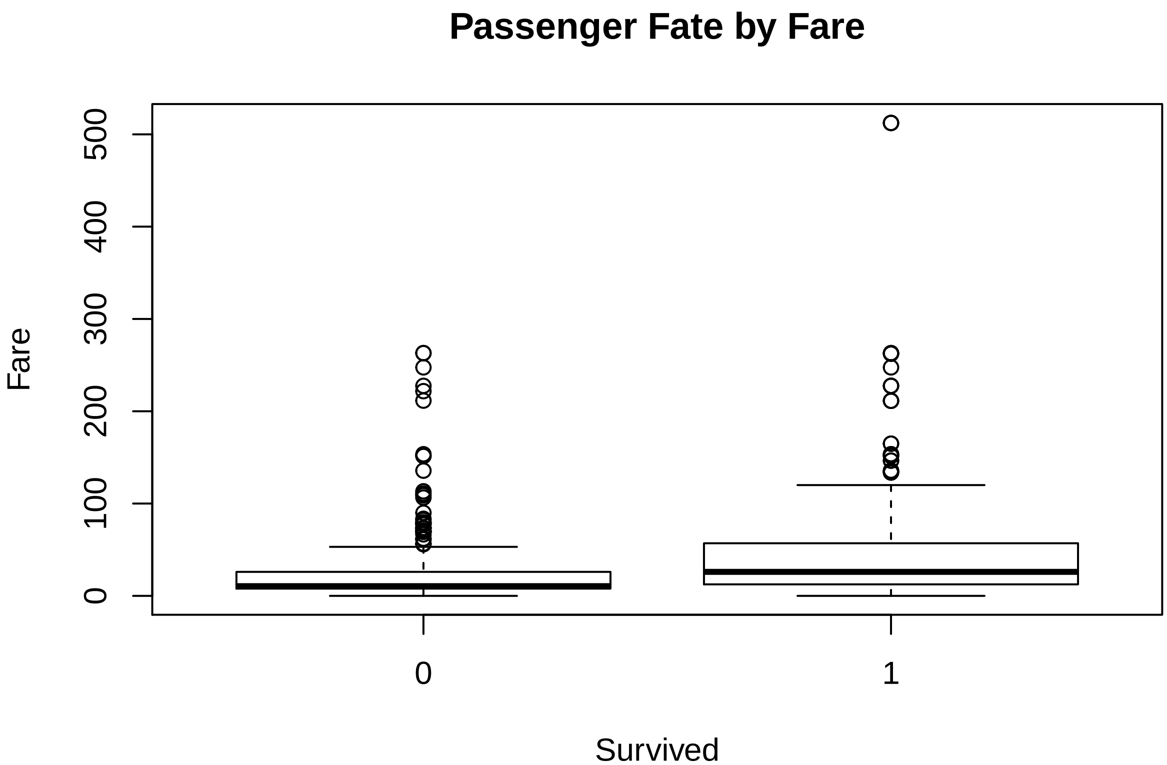
```r
# SibSp - Number of Siblings/Spouses Aboard - Mixed
mosaicplot(train$SibSp ~ train$Survived, main = "Passenger Fate by Siblings", shade = FALSE, color = TRU
```



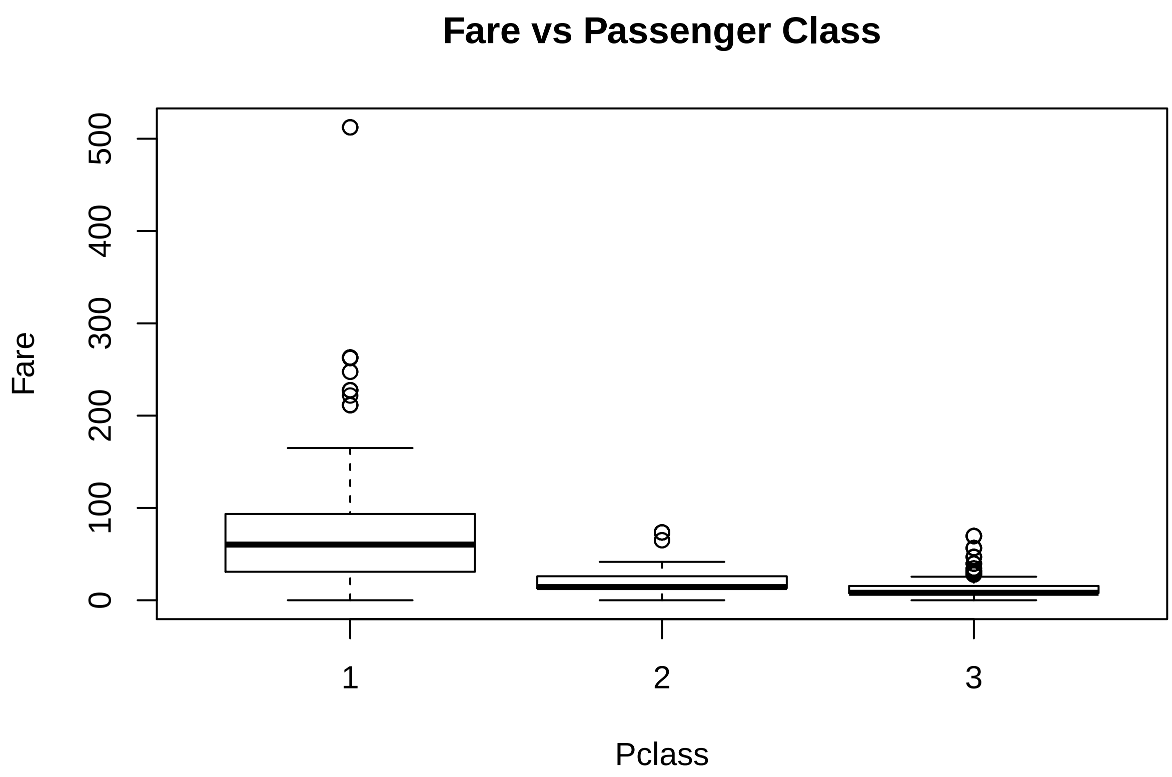

```r
# Parch - Number of Parents/Children Aboard - Yes for alone vs with parents
mosaicplot(train$Parch ~ train$Survived, main = "Passenger Fate by Parents/Children", shade = FALSE, col
```



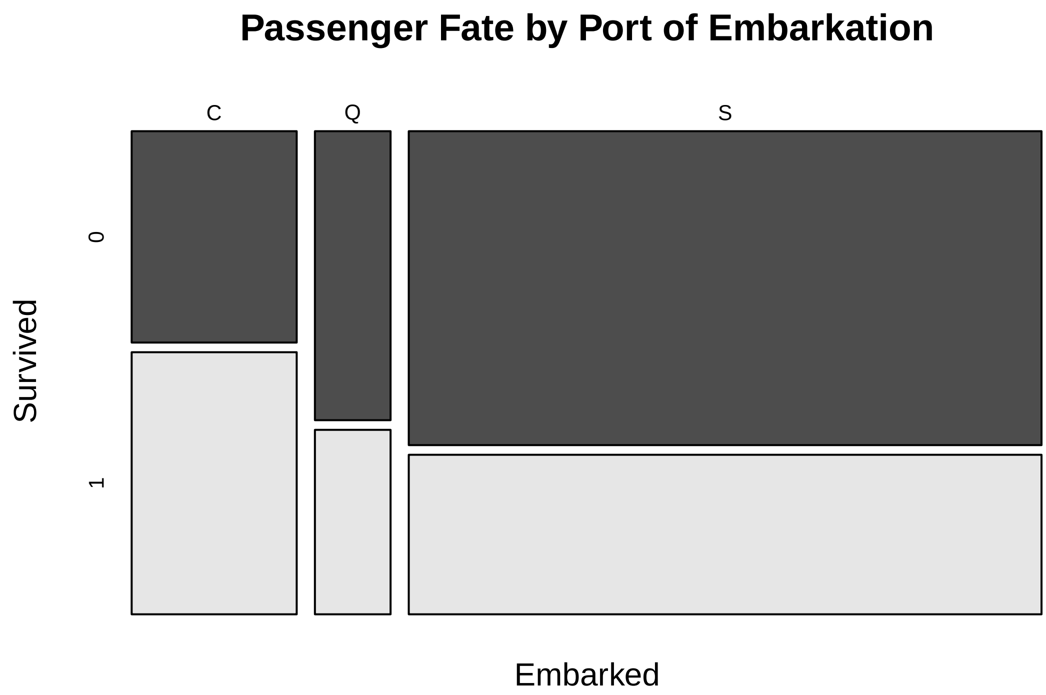
```r
# Fare - yes
boxplot(train$Fare ~ train$Survived, main= "Passenger Fate by Fare", xlab = "Survived", ylab = "Fare")
```



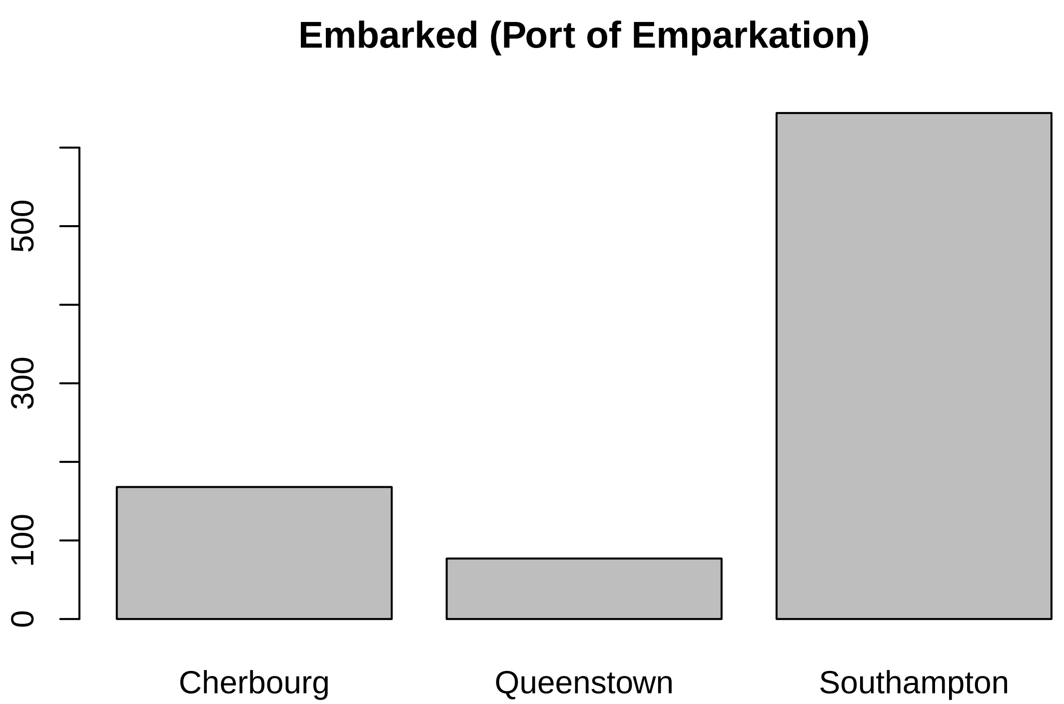

```r
# Are Fare and Pclass related? - Yes, so Passenger Class can be used as substitute for fare
boxplot(train$Fare ~ train$Pclass, main= "Fare vs Passenger Class", xlab = "Pclass", ylab = "Fare")
```




```r
# Embarked - Port of Embarkation
mosaicplot(train$Embarked ~ train$Survived, main = "Passenger Fate by Port of Embarkation", shade = FALS
```




```r
barplot(table(train$Embarked), names.arg = c("Cherbourg", "Queenstown", "Southampton"), main = "Embarke
```



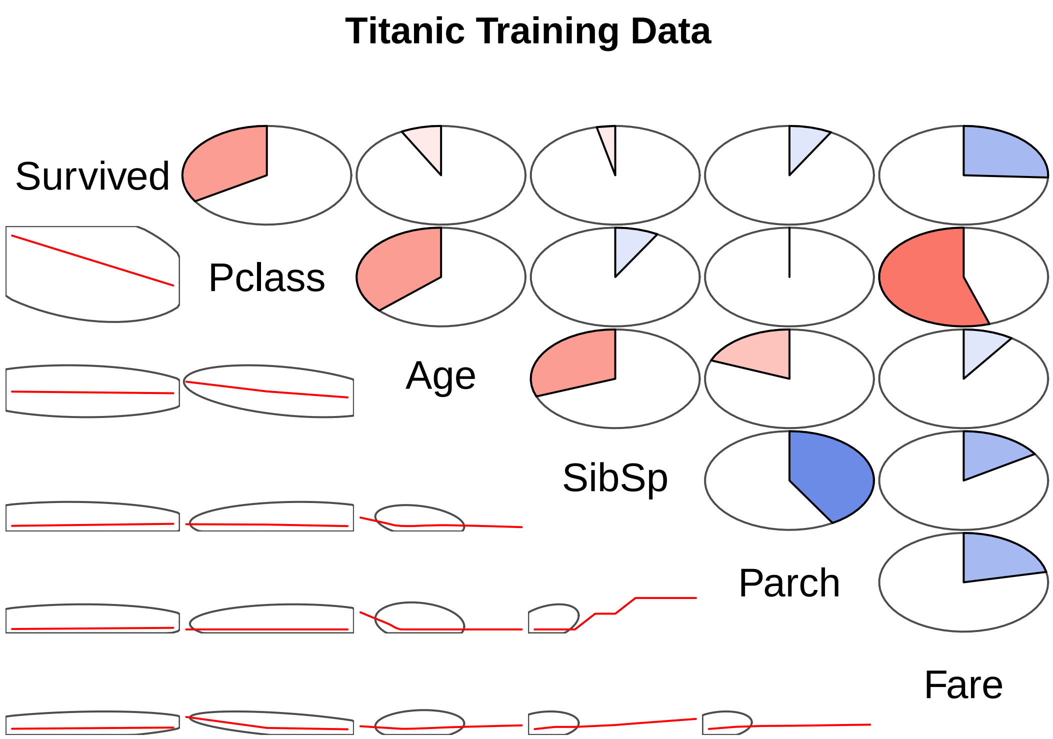

```
# Correlogram
require(corrgram)
```

```
## Loading required package: corrgram
```

```
require(plyr)
```

```
## Loading required package: plyr

##
## Attaching package: 'plyr'

## The following object is masked from 'package:corrgram':
##
##     baseball
```

```
corrgram.data <- train
## change features of factor type to numeric type for inclusion on correlogram
corrgram.data$Survived <- as.numeric(corrgram.data$Survived)
corrgram.data$Pclass <- as.numeric(corrgram.data$Pclass)
corrgram.data$Embarked <- revalue(corrgram.data$Embarked,
                                  c("C" = 1, "Q" = 2, "S" = 3))
## generate correlogram
corrgram.vars <- c("Survived", "Pclass", "Sex", "Age",
                   "SibSp", "Parch", "Fare", "Embarked")
corrgram(corrgram.data[0:891,corrgram.vars], order=FALSE,
         lower.panel=panel.ellipse, upper.panel=panel.pie,
         text.panel=panel.txt, main="Titanic Training Data")
```




```
## Replacing Fate ILO Survived and revaluing Fate factor
train$Fate <- train$Survived
train$Fate <- revalue(train$Fate, c("1" = "Survived", "0" = "Perished"))



# Individual's Name
## Obtaining titles
train$Title <- gsub('(.*, )|(\\..*)', '',train$Name)
table(train$Sex, train$Title)
```

```
##
##          Capt Col Don  Dr Jonkheer Lady Major Master Miss Mlle Mme  Mr Mrs
##   female    0   0   0   1        0    1     0      0  182    2   1   0 125
##   male      1   2   1   6        1    0     2     40    0    0   0 517   0
##
##           Ms Rev Sir the Countess
##   female   1   0   0            1
##   male     0   6   1            0
```

```
test$Title <- gsub('(.*, )|(\\..*)', '',test$Name)
table(test$Sex, test$Title)
```

```
##
##          Col Dona  Dr Master Miss  Mr Mrs  Ms Rev
##   female   0    1   0      0   78   0  72   1   0
##   male     2    0   1     21    0 240   0   0   2
```

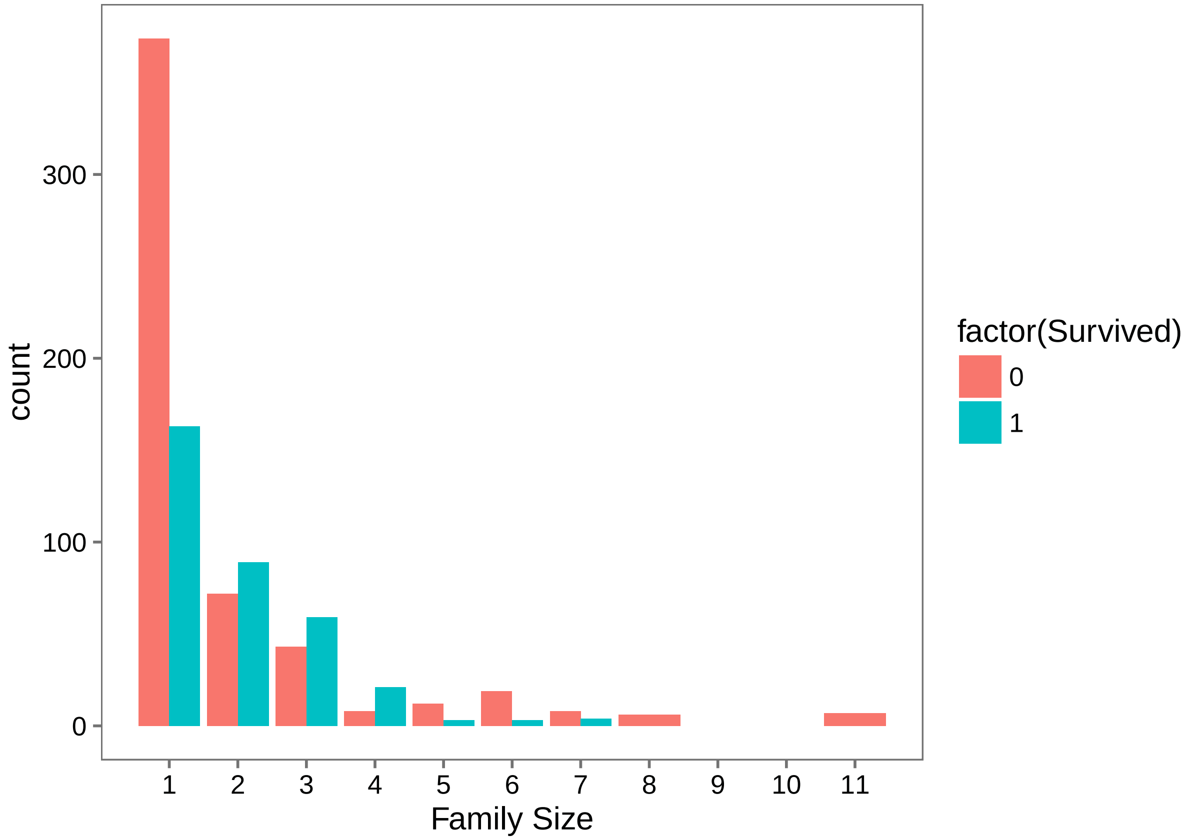
```r
## Combine all rare titles
rare_title <- c('Capt', 'Col','Don','Dona','Dr','Jonkheer', 'Lady', 'Major','Rev', 'Sir','the Countess')
## Reassignment of Mlle, Ms and Mme
train$Title[train$Title == 'Mlle'] <- 'Miss'
train$Title[train$Title == 'Ms'] <- 'Miss'
train$Title[train$Title == 'Mme'] <- 'Mrs'
train$Title[train$Title %in% rare_title] <- 'Rare'
test$Title[test$Title == 'Mlle'] <- 'Miss'
test$Title[test$Title == 'Ms'] <- 'Miss'
test$Title[test$Title == 'Mme'] <- 'Mrs'
test$Title[test$Title %in% rare_title] <- 'Rare'
table(train$Sex, train$Title )
```

```
##
##          Master Miss  Mr Mrs Rare
##   female      0  185   0 126    3
##   male       40    0 517   0   20
```

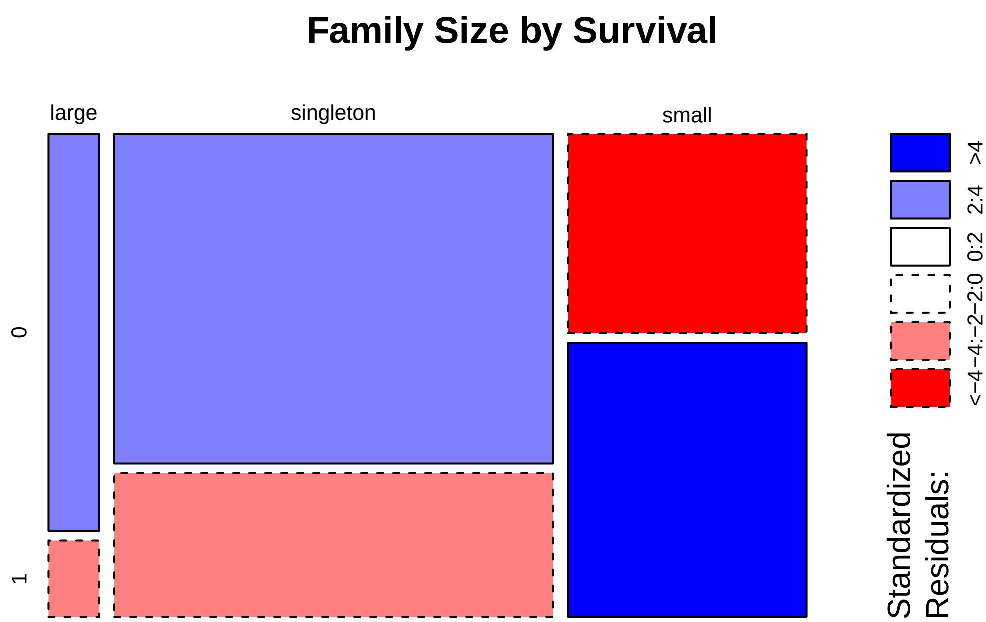
```r
table(test$Sex, test$Title )
```

```
##
##          Master Miss  Mr Mrs Rare
##   female      0   79   0  72    1
##   male       21    0 240   0    5
```

```r
## Obtaining Surnames
train$Surname <- sapply(train$Name, function(x) strsplit(x, split = '[,.]')[[1]][1])
test$Surname <- sapply(test$Name, function(x) strsplit(x, split = '[,.]')[[1]][1])

# Survival dependence on Family
train$Fsize = train$SibSp + train$Parch + 1
test$Fsize = test$SibSp + test$Parch + 1
train$Family <- paste(train$Surname, train$Fsize, sep='_')
test$Family <- paste(test$Surname, test$Fsize, sep='_')
ggplot(train, aes(x = Fsize, fill = factor(Survived))) +
  geom_bar(stat='count', position='dodge') +
  scale_x_continuous(breaks=c(1:11)) +
  labs(x = 'Family Size') +
  theme_few()
```




```r
# Discretize family size
train$FsizeD[train$Fsize == 1] <- 'singleton'
train$FsizeD[train$Fsize < 5 & train$Fsize > 1] <- 'small'
train$FsizeD[train$Fsize > 4] <- 'large'
test$FsizeD[test$Fsize == 1] <- 'singleton'
test$FsizeD[test$Fsize < 5 & test$Fsize > 1] <- 'small'
test$FsizeD[test$Fsize > 4] <- 'large'
# Show family size by survival using a mosaic plot
mosaicplot(table(train$FsizeD, train$Survived), main='Family Size by Survival', shade=TRUE)
```




```
# Survival dependence on Age: Discretized
## Predictive Imputation of Age using MICE (Multiple Imputation Using Chained Equations)
sum(is.na(train$Age))
```

```
## [1] 177
```

```
## Making variables factors
factor_vars <- c('PassengerId','Pclass','Sex','Embarked',
                 'Title','Surname','Family','FsizeD')
train[factor_vars] <- lapply(train[factor_vars], function(x) as.factor(x))
test[factor_vars] <- lapply(test[factor_vars], function(x) as.factor(x))
## Setting a random seed
set.seed(129)
## Performing MICE imputation, excluding certain less than useful variables
library(mice)
```

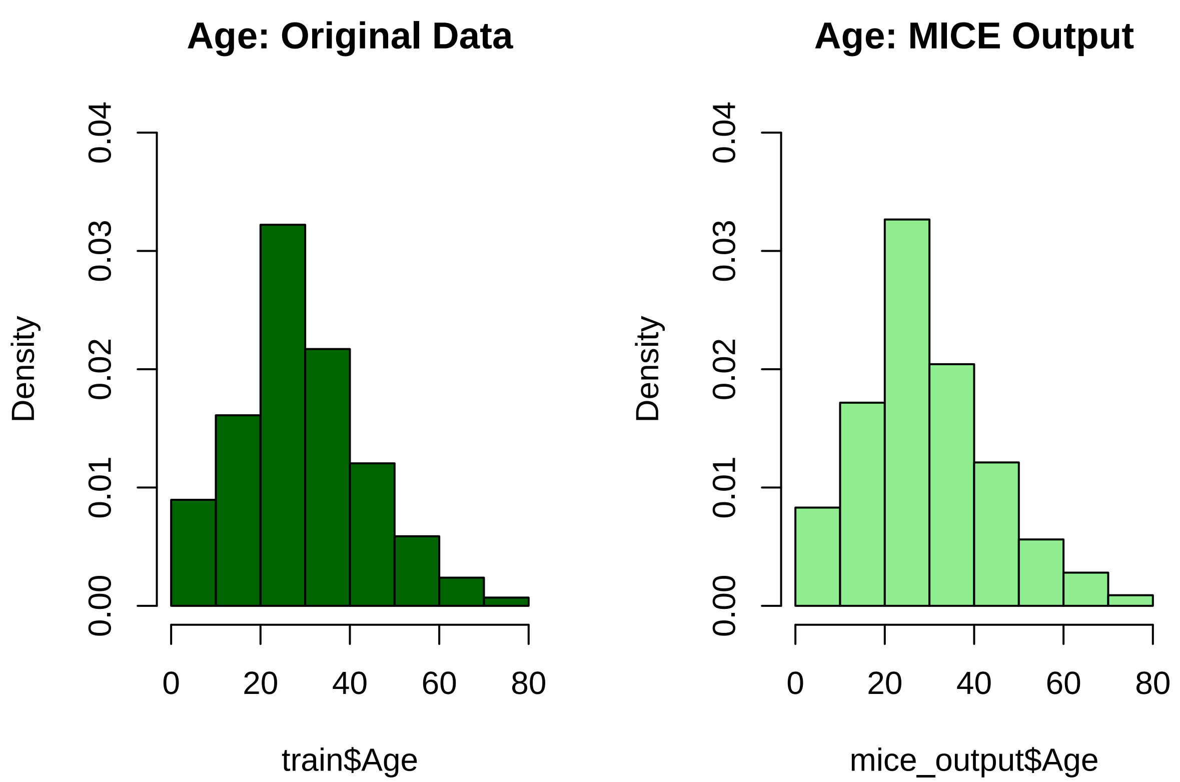
```
## mice 2.25 2015-11-09
```

```
mice_mod <- mice(train[, !names(train) %in% c('PassengerId','Name','Ticket','Cabin','Family','Surname',
```

```
##
##  iter imp variable
##   1   1  Age
##   1   2  Age
##   1   3  Age
##   1   4  Age
##   1   5  Age
##   2   1  Age
##   2   2  Age
##   2   3  Age
##   2   4  Age
##   2   5  Age
```

```
##     3    1  Age
##     3    2  Age
##     3    3  Age
##     3    4  Age
##     3    5  Age
##     4    1  Age
##     4    2  Age
##     4    3  Age
##     4    4  Age
##     4    5  Age
##     5    1  Age
##     5    2  Age
##     5    3  Age
##     5    4  Age
##     5    5  Age
```

```
mice_output <- complete(mice_mod)
## Plotting age distributions
par(mfrow=c(1,2))
hist(train$Age, freq=F, main='Age: Original Data',
     col='darkgreen', ylim=c(0,0.04))
hist(mice_output$Age, freq=F, main='Age: MICE Output',
     col='lightgreen', ylim=c(0,0.04))
```



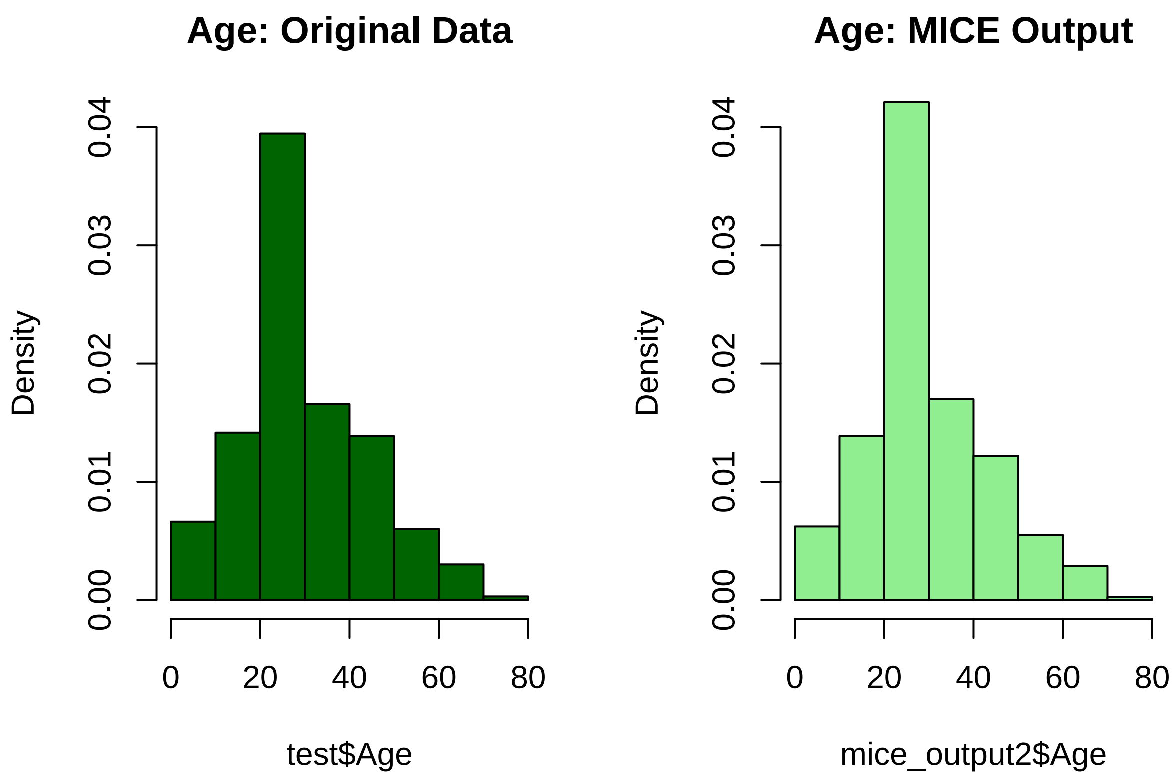

```
## Replacing Age variable with MICE model
train$Age <- mice_output$Age
sum(is.na(train$Age))
```

```
## [1] 0
```

```
### Doing the same to test
mice_mod2 <- mice(test[, !names(test) %in% c('PassengerId','Name','Ticket','Cabin','Family','Surname','S
```

```
##
##  iter imp variable
##   1   1  Age  Fare
##   1   2  Age  Fare
##   1   3  Age  Fare
##   1   4  Age  Fare
##   1   5  Age  Fare
##   2   1  Age  Fare
##   2   2  Age  Fare
##   2   3  Age  Fare
##   2   4  Age  Fare
##   2   5  Age  Fare
##   3   1  Age  Fare
##   3   2  Age  Fare
##   3   3  Age  Fare
##   3   4  Age  Fare
##   3   5  Age  Fare
##   4   1  Age  Fare
##   4   2  Age  Fare
##   4   3  Age  Fare
##   4   4  Age  Fare
##   4   5  Age  Fare
##   5   1  Age  Fare
##   5   2  Age  Fare
##   5   3  Age  Fare
##   5   4  Age  Fare
##   5   5  Age  Fare
```

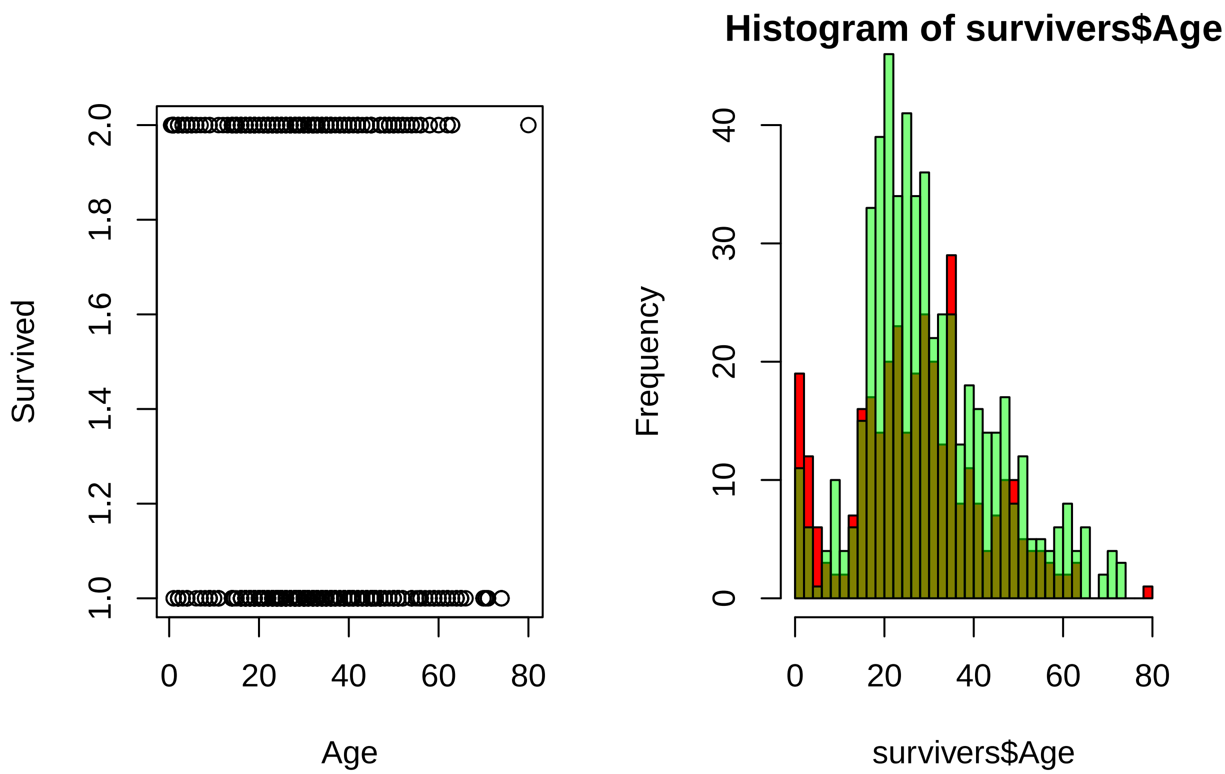
```
mice_output2 <- complete(mice_mod2)
## Plotting age distributions
par(mfrow=c(1,2))
hist(test$Age, freq=F, main='Age: Original Data',
     col='darkgreen', ylim=c(0,0.04))
hist(mice_output2$Age, freq=F, main='Age: MICE Output',
     col='lightgreen', ylim=c(0,0.04))
```




```
## Replacing Age variable with MICE model
test$Age <- mice_output2$Age
sum(is.na(test$Age))
```

```
## [1] 0
```

```
## Relationship with Age
plot(train$Age, train$Survived, xlab = "Age", ylab = "Survived")
survivers <- data.frame(train$Age[train$Survived == 1])
nonsurvivers <- data.frame(train$Age[train$Survived == 0])
survivers$title <- 'Survivers'
nonsurvivers$title <- 'Non-Survivers'
colnames(survivers)[1] <- "Age"
colnames(nonsurvivers)[1] <- "Age"
hist(survivers$Age, breaks = 32 ,xlim=c(0,80), ylim=c(0,40), col="red")
hist(nonsurvivers$Age,breaks = 32,  add=T, col=rgb(0,1,0,0.5))
```




```
##Discretize age
train$Agegroup[train$Age<14] <- 'child'
train$Agegroup[train$Age>=14] <- 'adult'
test$Agegroup[test$Age<14] <- 'child'
test$Agegroup[test$Age>=14] <- 'adult'
table(train$Agegroup, train$Survived)
```

```
##
##          0   1
##   adult 513 296
##   child  36  46
```

```
mosaicplot(table(train$Agegroup, train$Survived), main = "Age Group by Survival", shade = TRUE)
##Slight benefit of being a child

# Combined Effect of Age and Sex
ggplot(train, aes(Age, fill = factor(Survived))) +
  geom_histogram() +
  # Including Sex since we know (a priori) it's a significant predictor
  facet_grid(.~Sex) +
  theme_few()
```

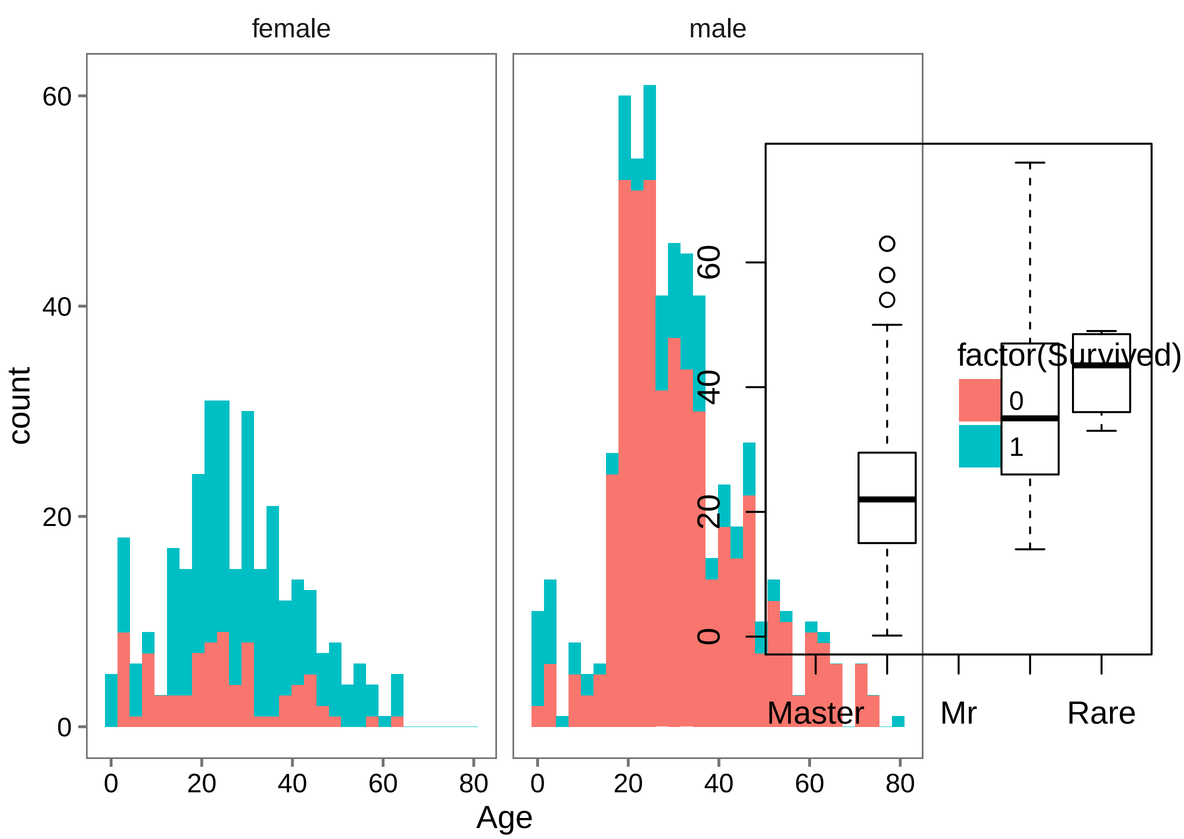
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
#Mothers may have survived: So maternity?
library(dplyr)
```

```
## ------------------------------------------------------------------------
```

```
## data.table + dplyr code now lives in dtplyr.
## Please library(dtplyr)!

## -------------------------------------------------------------------------

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:data.table':
##
##     between, last

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

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

```r
full1 <- bind_rows(select(train, Sex, Title, Age), select(test, Sex, Title, Age))
female_age <- full1 %>% filter(Sex == 'female')
plot(female_age$Title, female_age$Age)
```



```r
b <- female_age[female_age$Title == 'Mrs', ]
min(b$Age)
```

```
## [1] 14
```

```r
train$Mother <- 'Not Mother'
train$Mother[train$Sex == 'female' & train$Parch > 0 & train$Age > min(b$Age) & train$Title != 'Miss'] <
test$Mother <- 'Not Mother'
test$Mother[test$Sex == 'female' & test$Parch > 0 & test$Age > min(b$Age) & test$Title != 'Miss'] <- 'Mo
table(train$Mother, train$Survived)
```

```
##
##               0   1
##   Mother     16  40
##   Not Mother 533 302
```

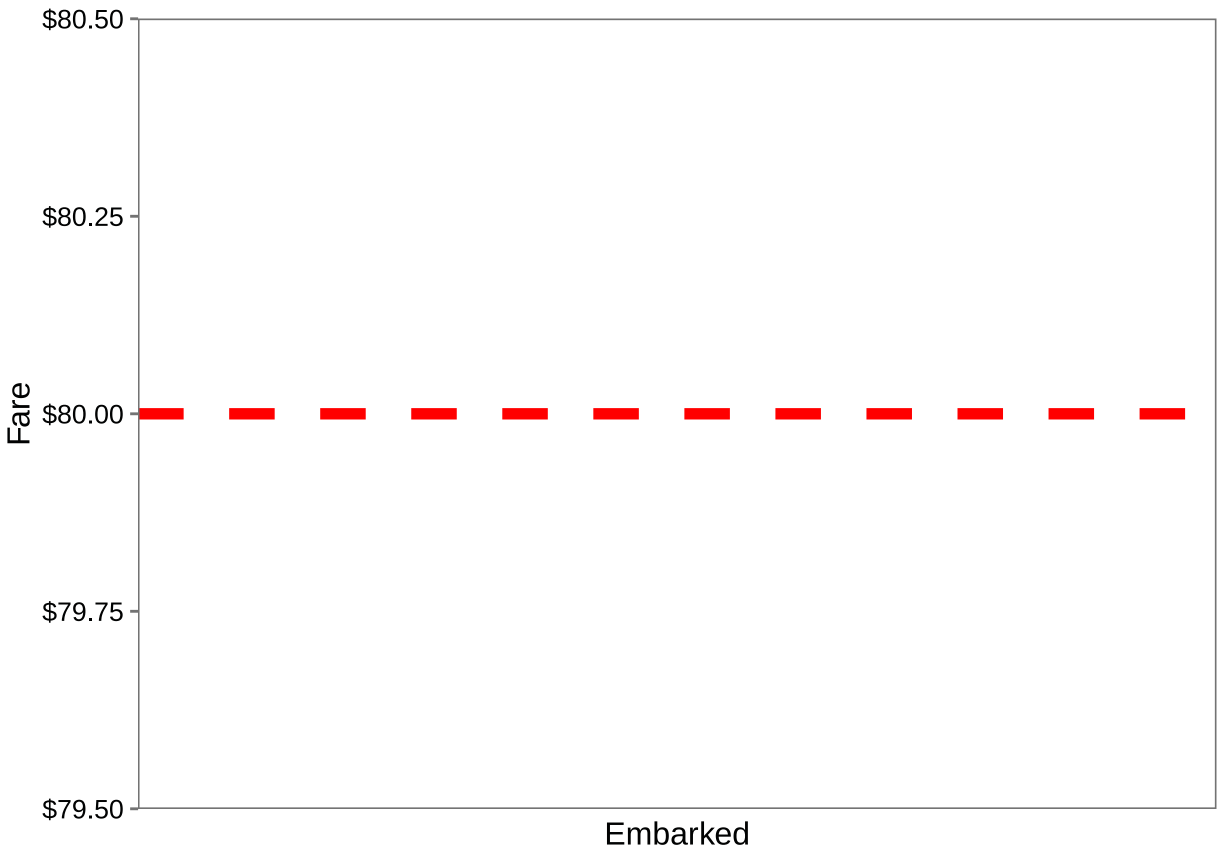
```r
## Factorizing our two new factor variables
train$Agegroup  <- factor(train$Agegroup)
train$Mother <- factor(train$Mother)
test$Agegroup  <- factor(test$Agegroup)
test$Mother <- factor(test$Mother)

# Embarkment completion
table(is.na(train$Embarked))
```

```
##
## FALSE  TRUE
##   889     2
```

```r
table(is.na(test$Embarked))
```

```
##
## FALSE
##   418
```

```r
## Can the data be extrapolated from Passenger Class and Fare?
## Removing the entries without Embarked Info and adding info from test data
library(dplyr)
full <- bind_rows(select(train, Embarked, Pclass, Fare), select(test, Embarked, Pclass, Fare))
embark_fare <- full %>% filter(Embarked == "NA")
library(scales)
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(labels=dollar_format()) +
  theme_few()
```




```
## Median = $80
train[is.na(train$Embarked),]
```

```
##     PassengerId Survived Pclass                                      Name
## 62           62        1      1                       Icard, Miss. Amelie
## 830         830        1      1 Stone, Mrs. George Nelson (Martha Evelyn)
##        Sex Age SibSp Parch Ticket Fare Cabin Embarked     Fate Title
## 62  female  38     0     0 113572   80   B28     <NA> Survived  Miss
## 830 female  62     0     0 113572   80   B28     <NA> Survived   Mrs
##     Surname Fsize  Family    FsizeD Agegroup     Mother
## 62    Icard     1 Icard_1 singleton    adult Not Mother
## 830   Stone     1 Stone_1 singleton    adult Not Mother
```

```
## Entries are 62 and 830
train$Embarked[c(62, 830)] <- 'C'
table(is.na(train$Embarked))
```

```
##
## FALSE
##   891
```

```
# Fixing Fare
table(is.na(train$Fare))
```

```
##
```

```
## FALSE
##   891
```

```r
table(is.na(test$Fare))
```

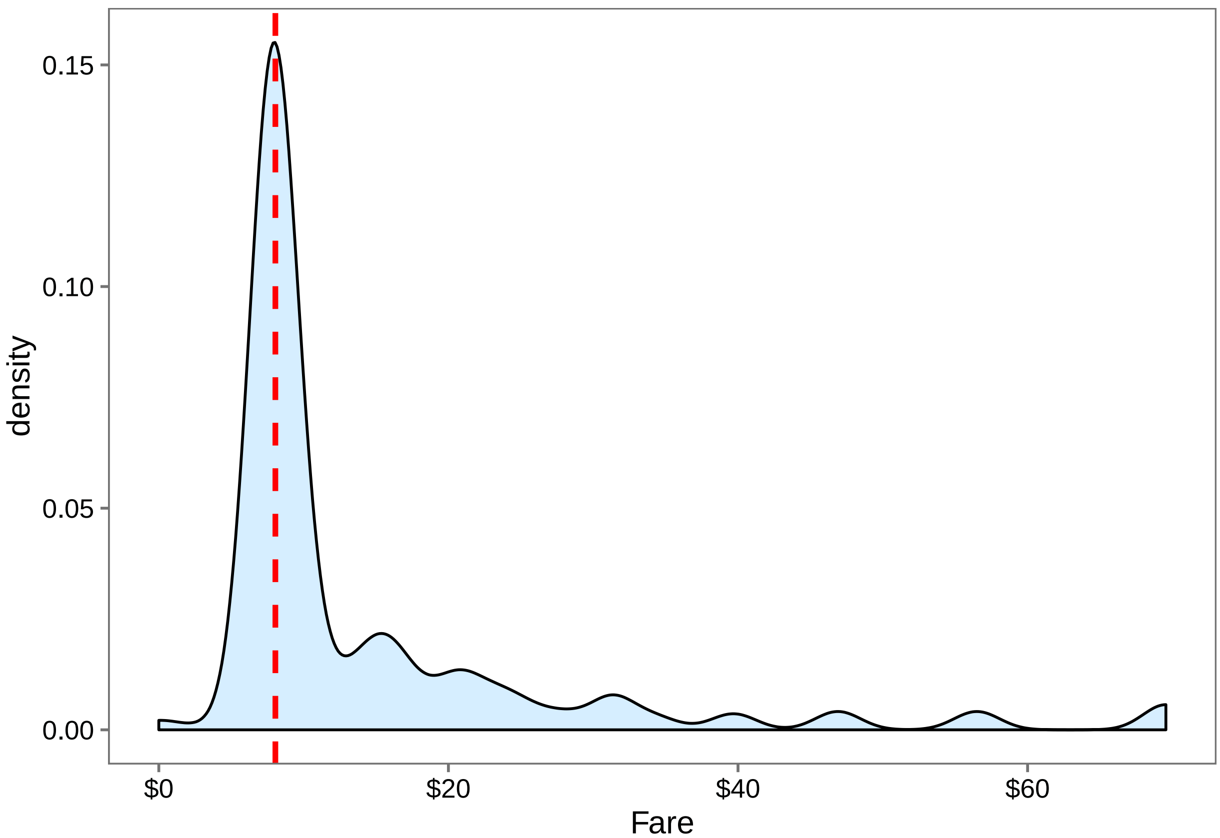
```
##
## FALSE  TRUE
##   417     1
```

```r
## One entry in test does not have fare
test[is.na(test$Fare),]
```

```
##     PassengerId Pclass               Name  Sex  Age SibSp Parch Ticket
## 153        1044      3 Storey, Mr. Thomas male 60.5     0     0   3701
##     Fare Cabin Embarked Title Surname Fsize   Family    FsizeD Agegroup
## 153   NA  <NA>        S    Mr  Storey     1 Storey_1 singleton    adult
##         Mother
## 153 Not Mother
```

```r
## It is entry no. 1044/ test no. 153 and his Pclass is 3; Embarked is S
ggplot(full[full$Pclass == '3' & full$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) +
  scale_x_continuous(labels=dollar_format()) +
  theme_few()
```

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




```
a <- full[full$Pclass == '3' & full$Embarked == 'S', ]
a <- a[is.na(a$Fare)==FALSE,]
median(a$Fare)
```

```
## [1] 8.05
```

```
## Median is $8.05
test$Fare[153] <- median(a$Fare)

# Building Model
##set.seed(754)
##rf_model <- randomForest(factor(Survived) ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked + Ti
## Show model error
##plot(rf_model, ylim = c(0,0.36))
##legend('topright', colnames(rf_model$err.rate),col=1:3, fill=1:3)

# Get importance
##importance    <- importance(rf_model)
##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
```

```r
##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() +
##  theme_few()
# Predict using the test set
## prediction <- predict(rf_model, test)

# Save the solution to a dataframe with two columns: PassengerId and Survived (prediction)
## solution <- data.frame(PassengerID = test$PassengerId, Survived = prediction)

# Write the solution to file
## write.csv(solution, file = 'rf_mod_Solution.csv', row.names = F)

# Comparing algorithms
library(caret)
```

```
## Loading required package: lattice
```

```r
## Testing harness with 10-fold cross validation
# Run algorithms using 10-fold cross validation
control <- trainControl(method="cv", number=10)
metric <- "Accuracy"
## a) linear algorithms
set.seed(7)
fit.lda <- train(factor(Survived) ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked + Title + Fsize
```

```
## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```r
## b) nonlinear algorithms
## CART
set.seed(7)
fit.cart <- train(factor(Survived) ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked + Title + Fsi
```

```
## Loading required package: rpart
```

```r
## kNN
set.seed(7)
fit.knn <- train(factor(Survived) ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked + Title + Fsize
## c) advanced algorithms
## SVM
set.seed(7)
fit.svm <- train(factor(Survived) ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked + Title + Fsize
```

```
## Loading required package: kernlab

##
## Attaching package: 'kernlab'

## The following object is masked from 'package:scales':
##
##     alpha

## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```r
## Random Forest
set.seed(7)
fit.rf <- train(factor(Survived) ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked + Title + FsizeD
```

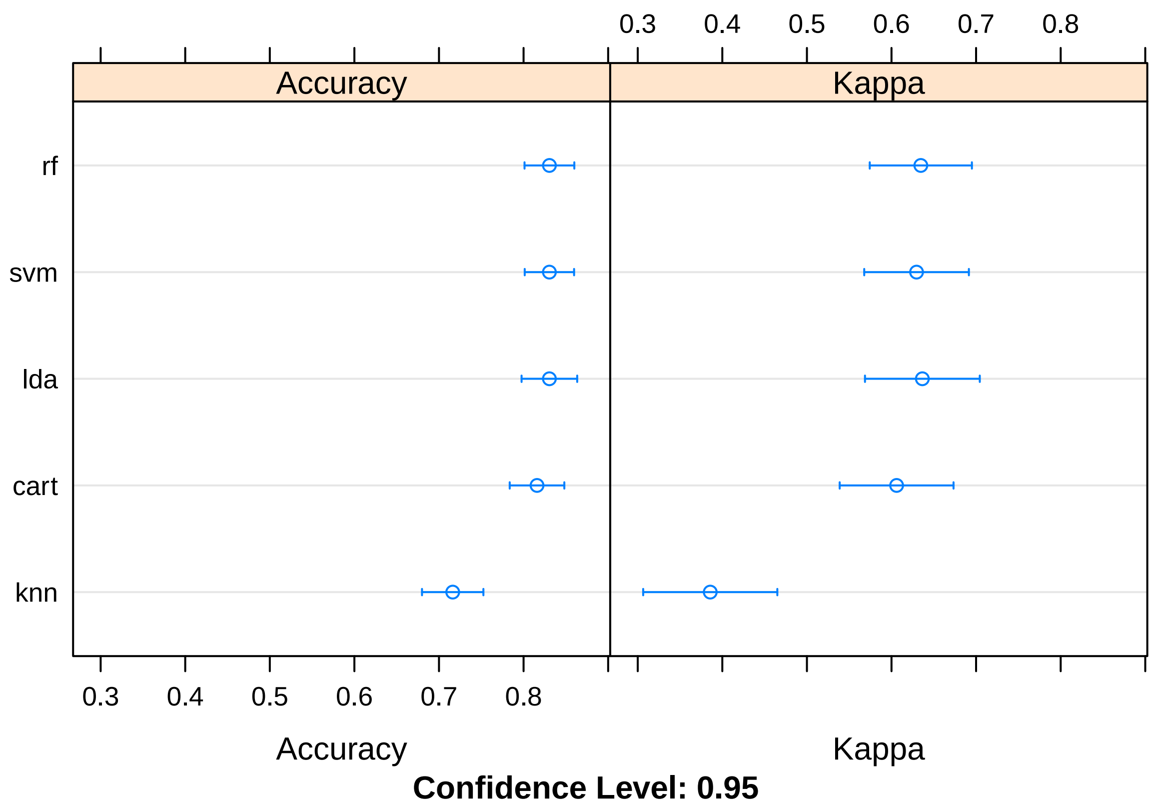
```
## Loading required package: randomForest

## randomForest 4.6-12

## 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
```

```r
## Comparison of algorithms
# summarize accuracy of models
results <- resamples(list(lda=fit.lda, cart=fit.cart, knn=fit.knn, svm=fit.svm, rf=fit.rf))
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: lda, cart, knn, svm, rf
## Number of resamples: 10
##
## Accuracy
##        Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
## lda  0.7416  0.8202 0.8325 0.8305  0.8539 0.8889    0
## cart 0.7416  0.7893 0.8146 0.8159  0.8440 0.8778    0
## knn  0.6404  0.6889 0.7022 0.7162  0.7360 0.8182    0
```

```
## svm  0.7528  0.8062 0.8427 0.8305  0.8444 0.8876    0
## rf   0.7416  0.8202 0.8371 0.8306  0.8516 0.8977    0
##
## Kappa
##         Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
## lda  0.4557  0.6071 0.6386 0.6365  0.6871 0.7662    0
## cart 0.4557  0.5480 0.5996 0.6060  0.6651 0.7415    0
## knn  0.2552  0.3115 0.3527 0.3857  0.4314 0.6166    0
## svm  0.4765  0.5859 0.6491 0.6296  0.6613 0.7566    0
## rf   0.4557  0.6082 0.6418 0.6347  0.6796 0.7758    0
```

```r
dotplot(results)
```




```r
## Random Forest is best model
print(fit.rf)
```

```
## Random Forest
##
## 891 samples
##  11 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 802, 801, 802, 802, 803, 802, ...
## Resampling results across tuning parameters:
##
```

```
##   mtry  Accuracy   Kappa    
##    2    0.8305592  0.6346562
##    9    0.8182874  0.6080777
##   17    0.8026189  0.5786917
## 
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 2.
```

```r
# Get importance
varImportance <- data.frame(varImp(fit.rf)$importance)
varImportance$Vars <- row.names(varImportance)
varImportance[order(-varImportance$Overall),]
```

```
##                       Overall              Vars
## Sexmale            100.000000           Sexmale
## TitleMr             99.088288           TitleMr
## Fare                57.286404              Fare
## Pclass3             42.548643           Pclass3
## Age                 39.367523               Age
## TitleMiss           37.476197         TitleMiss
## TitleMrs            28.635316          TitleMrs
## FsizeDsmall         19.312554       FsizeDsmall
## SibSp               14.338184             SibSp
## Parch                9.143224             Parch
## Agegroupchild        7.725809     Agegroupchild
## EmbarkedS            5.974658         EmbarkedS
## Pclass2              4.994217           Pclass2
## FsizeDsingleton      4.949275   FsizeDsingleton
## MotherNot Mother     2.574848 MotherNot Mother
## EmbarkedQ            1.723891         EmbarkedQ
## TitleRare            0.000000         TitleRare
```

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

# Predict using the test set
prediction <- predict(fit.rf, test)

# Save the solution to a dataframe with two columns: PassengerId and Survived (prediction)
solution <- data.frame(PassengerID = test$PassengerId, Survived = prediction)

# Write the solution to file
write.csv(solution, file = 'rf_mod_Solution.csv', row.names = F)
```