TitanicSurvival

Lance J. Fernando

First lets import the training and testing sets

```
tit_test <- read.csv("/Users/lancefernando/Desktop/DataMining/PythonProjects/Titanic/
titanic_test.csv", header = TRUE)
tit_train <- read.csv("/Users/lancefernando/Desktop/DataMining/PythonProjects/Titanic
/titanic_train.csv", header = TRUE)
head(tit_train)
```

```
##   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
```

Notice that all passengers have a title after their surname. In order to extract this, lets split up each name observation using [,.] as our delimeter. With this we can extract titles.

```
names.split <- strsplit(as.character(tit_train$Name), "[,.]")
test.names.split <- strsplit(as.character(tit_test$Name), "[,.]")

title <- rep(NA, length(names.split))
test.title <- rep(NA, length(test.names.split))

for(i in 1:length(names.split)){
  title[i] <- trimws(names.split[[i]][2])
}

for(i in 1:length(test.names.split)){
  test.title[i] <- trimws(test.names.split[[i]][2])
}
table(title)
```

```
## title
##         Capt          Col          Don           Dr     Jonkheer
##            1            2            1            7            1
##         Lady        Major       Master         Miss         Mlle
##            1            2           40          182            2
##          Mme           Mr          Mrs           Ms          Rev
##            1          517          125            1            6
##          Sir the Countess
##            1            1
```

```
table(test.title)
```

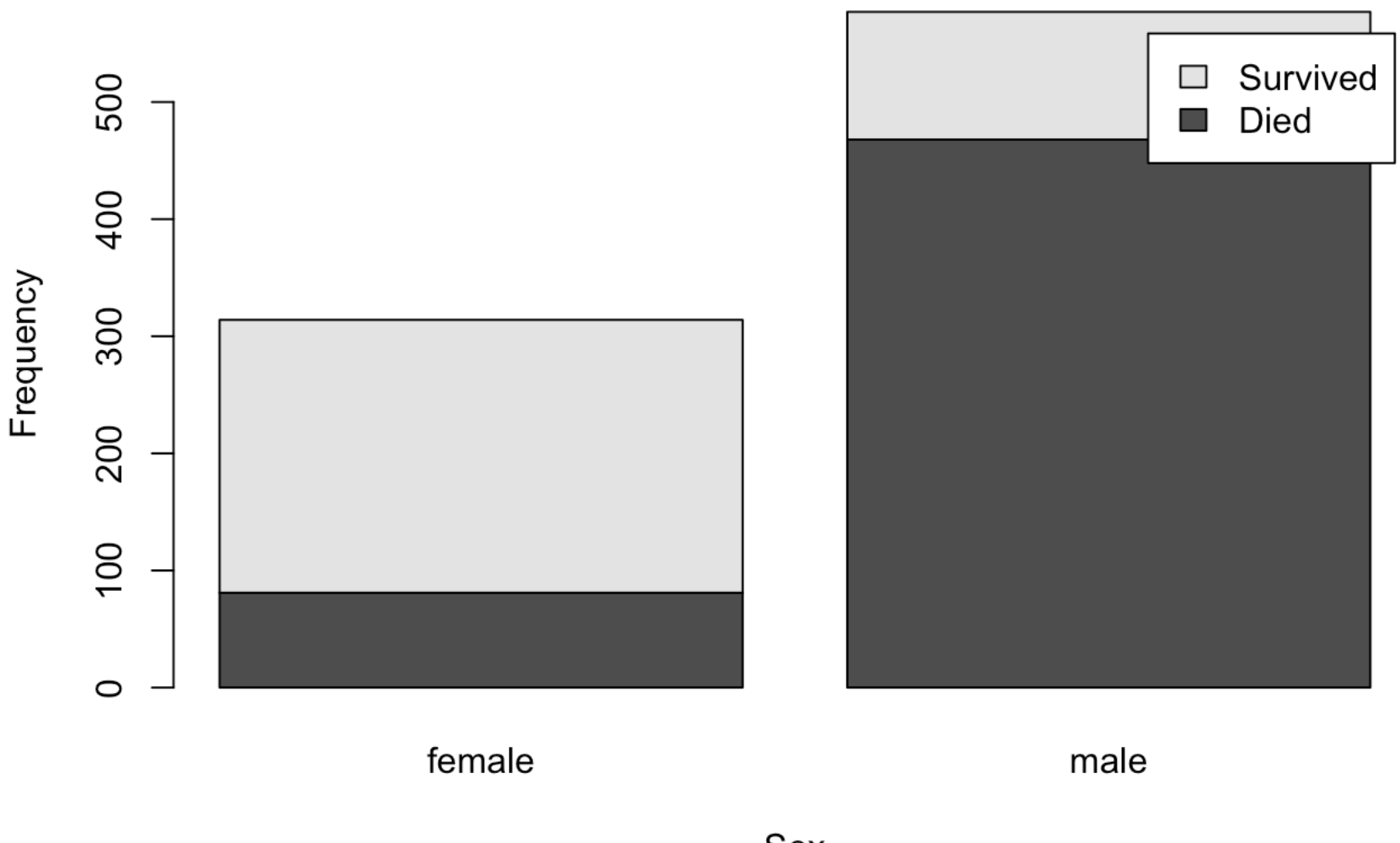
```
## test.title
##    Col   Dona     Dr Master   Miss     Mr    Mrs     Ms    Rev
##      2      1      1     21     78    240     72      1      2
```

Lets anaylze the survival rate based on Sex.

```
table(Survived = tit_train$Survived, Sex = tit_train$Sex)
```

```
##          Sex
## Survived female male
##        0     81  468
##        1    233  109
```

```
barplot(table(tit_train$Survived, tit_train$Sex), legend = c('Died', 'Survived'),
        xlab = 'Sex', ylab = 'Frequency', main = 'Female vs Male Survival Rate')
```



Age is an important feature to include however many observations are missing from that column. In order to extract the essence of age out of each passenger, we can use their title along with other features. Since mostly women survived, lets anaylze which among those women were more likely to make it out.

```
table(Survived = tit_train$Survived[tit_train$Sex == 'female'], title[tit_train$Sex =
= 'female'])
```

```
##
## Survived  Dr Lady Miss Mlle Mme Mrs  Ms the Countess
##        0   0    0   55    0   0  26   0            0
##        1   1    1  127    2   1  99   1            1
```

Lets change the Sex values to 0 and 1 for female and male respectively.

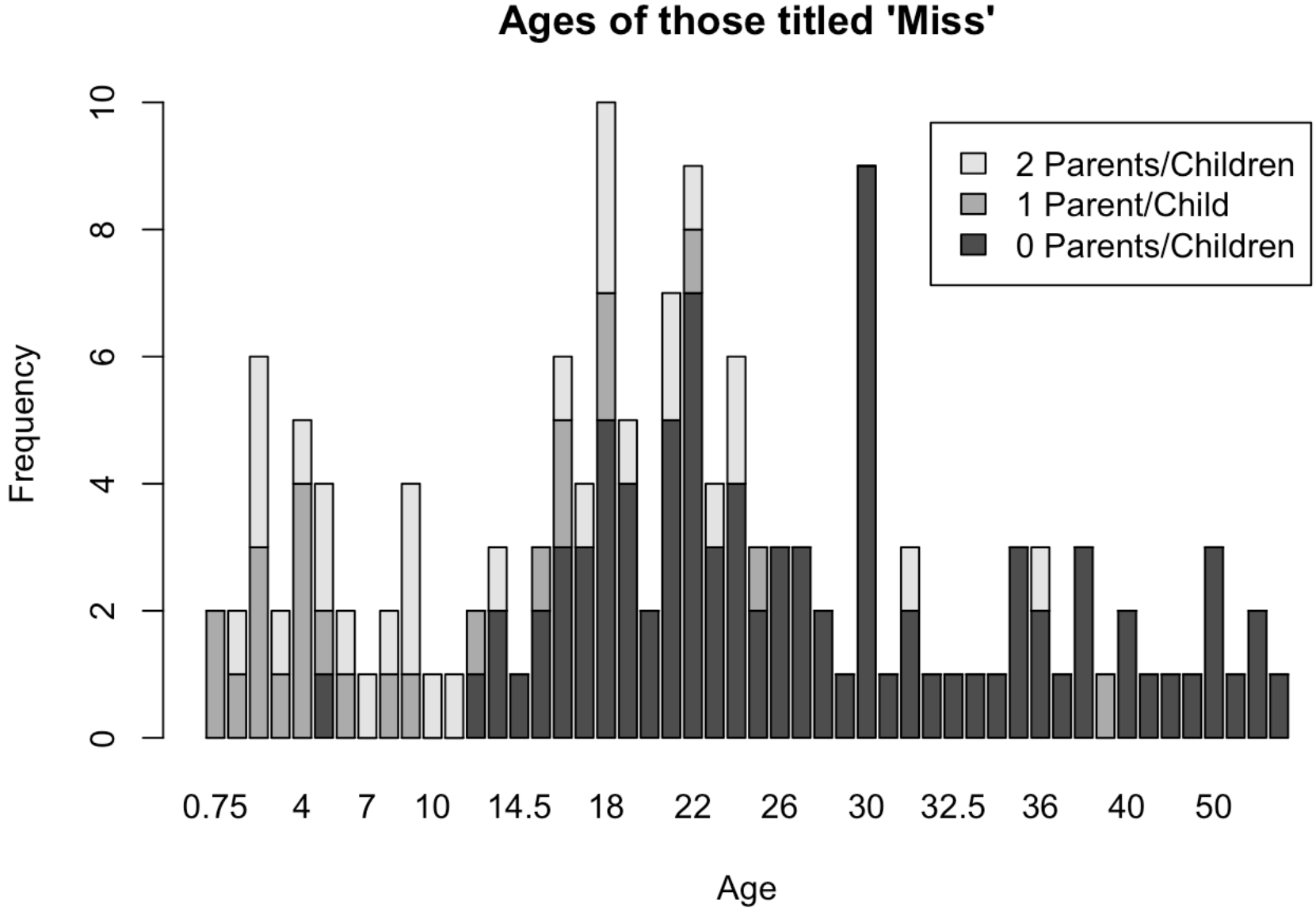
```
tit_train$Sex <- ifelse(tit_train$Sex == "male", 1, 0)
tit_test$Sex <- ifelse(tit_test$Sex == "male", 1, 0)
```

Women with titles ‘Miss’ and ‘Mrs’ occur more often so lets dive further into that data. We can easily categorize women with the title of ‘Mrs’ to be married so lets make that into a feature. In addition, the title of ‘Master’ for men means that they are under the age of 18. Lets also create that feature.

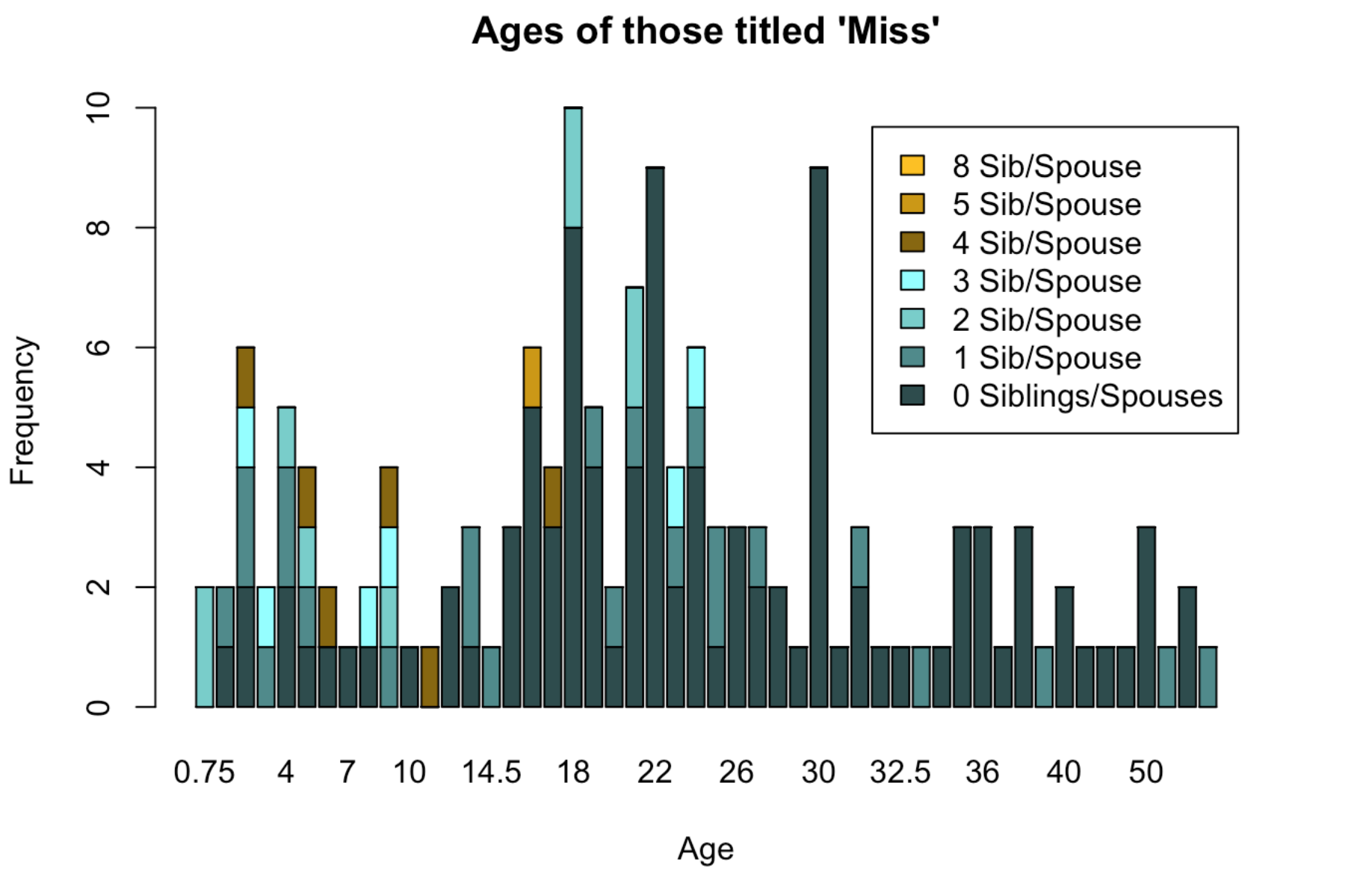
```
marrWom <- rep(0, nrow(tit_train))
marrWom.test <- rep(0,nrow(tit_test))
marrWom <- ifelse(title == "Mrs", 1, 0)
marrWom.test <- ifelse(test.title == "Mrs", 1, 0)
tit_train$marrWom <- marrWom
tit_test$marrWom <- marrWom.test

isBoy <- rep(0, nrow(tit_train))
isBoy.test <- rep(0,nrow(tit_test))
isBoy <- ifelse(title == "Master", 1, 0)
isBoy.test <- ifelse(test.title == "Master", 1, 0)

tit_train$isBoy <- isBoy
tit_test$isBoy <- isBoy.test
```

Lets analyze the age range for those with the title of ‘Miss’ or ‘Ms’. We can also include the column regarding the number of Siblings/Spouses and Parents/Children.

```
barplot(table(tit_train$Parch[title=="Miss"],tit_train$Age[title == "Miss"]),
        legend = c("0 Parents/Children", "1 Parent/Child", "2 Parents/Children"),
        xlab = "Age", ylab = "Frequency", main = "Ages of those titled 'Miss'")
```



```
barplot(table(tit_train$SibSp[title=="Miss"],tit_train$Age[title == "Miss"]),
        legend = c("0 Siblings/Spouses", "1 Sib/Spouse", "2 Sib/Spouse",
                   "3 Sib/Spouse", "4 Sib/Spouse", "5 Sib/Spouse", "8 Sib/Spouse"),
        col = c("darkslategrey", "darkslategray4", "darkslategray3", "darkslategray1"
,
                "goldenrod4", "goldenrod3", "goldenrod1"),
        xlab = "Age", ylab = "Frequency", main = "Ages of those titled 'Miss'")
```




```
mean(tit_train$Age[title == 'Miss' | title == 'Ms'], na.rm = TRUE)
```

```
## [1] 21.81633
```

The average age for ‘Miss’/‘Ms’ is 21.8. In addition, those that are older tend to have smaller family sizes; and more importantly travel alone. So using this information we can create another feature that will categorize ‘Miss’/‘Ms’ whether they are under or over this mean age.

```
youngFem <- rep(0, nrow(tit_train))
youngFem.test <- rep(0,nrow(tit_test))
miss <- c(which(title == 'Miss'), which(title == 'Ms'))
miss.test <- c(which(test.title == 'Miss'), which(test.title == 'Ms'))
for(i in miss){
  if(is.na(tit_train$Age[i])){
    if(tit_train$SibSp[i] >= 1 || tit_train$Parch[i] >= 1)
      youngFem[i] <- 1
  }
  else if(tit_train$Age[i] <= 22)
    youngFem[i] <- 1
}
for(i in miss.test){
  if(is.na(tit_test$Age[i])){
    if(tit_test$SibSp[i] >= 1 || tit_test$Parch[i] >= 1)
      youngFem.test[i] <- 1
  }
  else if(tit_test$Age[i] <= 22)
    youngFem.test[i] <- 1
}

tit_train$youngFem <- youngFem
tit_test$youngFem <- youngFem.test
```

Lets also add a variable for family size. We add 1 to account for the passengers themselves as well.

```
tit_train$famSize <- tit_train$SibSp + tit_train$Parch + 1
tit_test$famSize <- tit_test$SibSp + tit_test$Parch + 1
```

In the test set there seems to be an NA value for Fare. The passenger is of Pclass 3. Lets analyze how Pclass and Fare are correlated and give that passenger the mean Fare price for those of Pclass 3.

```
missing.Fare <- which(is.na(tit_test$Fare))
tit_test[missing.Fare,]
```

```
##     PassengerId Pclass                Name Sex  Age SibSp Parch Ticket Fare
## 153        1044      3 Storey, Mr. Thomas   1 60.5     0     0   3701   NA
##     Cabin Embarked marrWom isBoy youngFem famSize
## 153              S       0     0        0       1
```

```
cor(tit_train$Fare, tit_train$Pclass)
```

```
## [1] -0.5494996
```

```
tit_test$Fare[missing.Fare] <- mean(tit_train$Fare[tit_train$Pclass == 3])
```

Where people embarked from may turn out to be a significant feature. Analyzing the levels there are two passengers that have missing embark values. With a simple google search, these passengers boarded the Titanic at Southampton or 'S'. Lets just input that into our dataset

```
levels(tit_train$Embarked)
```

```
## [1] ""  "C" "Q" "S"
```

```
levels(tit_test$Embarked)
```

```
## [1] "C" "Q" "S"
```

```
missing.Embarked <- which(tit_train$Embarked == "")
tit_train$Embarked[missing.Embarked] <- "S"
tit_train[missing.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 marrWom isBoy youngFem
## 62    0  38     0     0 113572   80   B28        S       0     0        0
## 830   0  62     0     0 113572   80   B28        S       1     0        0
##     famSize
## 62        1
## 830       1
```

Now lets analyze the survival rate of each boarding station and create a new feature for each one.

```
table(tit_train$Survived, tit_train$Embarked)
```

```
##
##            C   Q   S
##   0    0  75  47 427
##   1    0  93  30 219
```

```
tit_train$Embarked.S <- ifelse(tit_train$Embarked == 'S', 1, 0)
tit_train$Embarked.Q <- ifelse(tit_train$Embarked == 'Q', 1, 0)
tit_train$Embarked.C <- ifelse(tit_train$Embarked == 'C', 1, 0)

tit_test$Embarked.S <- ifelse(tit_test$Embarked == 'S', 1, 0)
tit_test$Embarked.Q <- ifelse(tit_test$Embarked == 'Q', 1, 0)
tit_test$Embarked.C <- ifelse(tit_test$Embarked == 'C', 1, 0)
```

Lets also take a look at the cabin feature. This may or may not be important in our final model but it will not hurt to extract info out of it. We will create a feature for each cabin. Those that do not have their cabin labeled explicitly will get a ‘U’ for unknown.

```
tit_train$cab <- ifelse(tit_train$Cabin == "" | tit_train$Cabin == "T", "U", substr(t
it_train$Cabin,1,1))
tit_test$cab <- ifelse(tit_test$Cabin == "" | tit_test$Cabin == "T", "U", substr(tit_
test$Cabin,1,1))

table(tit_train$Survived, tit_train$cab)
```

```
##
##        A   B   C   D   E   F   G   U
##   0    8  12  24   8   8   5   2 482
##   1    7  35  35  25  24   8   2 206
```

```
tit_train$cab.A <- ifelse(tit_train$cab == 'A', 1, 0)
tit_train$cab.B <- ifelse(tit_train$cab == 'B', 1, 0)
tit_train$cab.C <- ifelse(tit_train$cab == 'C', 1, 0)
tit_train$cab.D <- ifelse(tit_train$cab == 'D', 1, 0)
tit_train$cab.E <- ifelse(tit_train$cab == 'E', 1, 0)
tit_train$cab.F <- ifelse(tit_train$cab == 'F', 1, 0)
tit_train$cab.G <- ifelse(tit_train$cab == 'G', 1, 0)
tit_train$cab.U <- ifelse(tit_train$cab == 'U', 1, 0)

tit_test$cab.A <- ifelse(tit_test$cab == 'A', 1, 0)
tit_test$cab.B <- ifelse(tit_test$cab == 'B', 1, 0)
tit_test$cab.C <- ifelse(tit_test$cab == 'C', 1, 0)
tit_test$cab.D <- ifelse(tit_test$cab == 'D', 1, 0)
tit_test$cab.E <- ifelse(tit_test$cab == 'E', 1, 0)
tit_test$cab.F <- ifelse(tit_test$cab == 'F', 1, 0)
tit_test$cab.G <- ifelse(tit_test$cab == 'G', 1, 0)
tit_test$cab.U <- ifelse(tit_test$cab == 'U', 1, 0)
```

Now that we have an abundance of features lets see what we have analyze their correlation to survival.

```
head(tit_train[,c(2,3,5,7,8,10,c(13:19), c(21:27))])
```

```
##   Survived Pclass Sex SibSp Parch    Fare marrWom isBoy youngFem famSize
## 1        0      3   1     1     0  7.2500       0     0         0        2
## 2        1      1   0     1     0 71.2833       1     0         0        2
## 3        1      3   0     0     0  7.9250       0     0         0        1
## 4        1      1   0     1     0 53.1000       1     0         0        2
## 5        0      3   1     0     0  8.0500       0     0         0        1
## 6        0      3   1     0     0  8.4583       0     0         0        1
##   Embarked.S Embarked.Q Embarked.C cab.A cab.B cab.C cab.D cab.E cab.F
## 1          1          0          0     0     0     0     0     0     0
## 2          0          0          1     0     0     1     0     0     0
## 3          1          0          0     0     0     0     0     0     0
## 4          1          0          0     0     0     1     0     0     0
## 5          1          0          0     0     0     0     0     0     0
## 6          0          1          0     0     0     0     0     0     0
##   cab.G
## 1     0
## 2     0
## 3     0
## 4     0
## 5     0
## 6     0
```

```
cor(tit_train[,c(2,3,5,7,8,10,c(13:19), c(21:27))])
```

```
##                Survived      Pclass         Sex       SibSp       Parch
## Survived    1.000000000 -0.33848104 -0.543351381 -0.035322499  0.08162941
## Pclass     -0.338481036  1.00000000  0.131900491  0.083081363  0.01844267
## Sex        -0.543351381  0.13190049  1.000000000 -0.114630810 -0.24548896
## SibSp      -0.035322499  0.08308136 -0.114630810  1.000000000  0.41483770
## Parch       0.081629407  0.01844267 -0.245488960  0.414837699  1.00000000
## Fare        0.257306522 -0.54949962 -0.182332834  0.159651043  0.21622494
## marrWom     0.339040251 -0.14920940 -0.547600334  0.063406878  0.22585153
## isBoy       0.085220561  0.08208138  0.159934491  0.349558681  0.26734379
## youngFem    0.172482345  0.09767676 -0.476540182  0.204221026  0.22532616
## famSize     0.016638989  0.06599691 -0.200988444  0.890711672  0.78311078
## Embarked.S -0.149682723  0.07405279  0.119223750  0.068733586  0.06081361
## Embarked.Q  0.003650383  0.22100892 -0.074115123 -0.026353729 -0.08122810
## Embarked.C  0.168240431 -0.24329208 -0.082853469 -0.059528215 -0.01106877
## cab.A       0.022286954 -0.20493446  0.078270705 -0.046266320 -0.04032543
## cab.B       0.175095034 -0.36957205 -0.109689073 -0.034537975  0.05649763
## cab.C       0.114652115 -0.41704772 -0.058649358  0.029250559  0.03073575
## cab.D       0.150715644 -0.27869030 -0.079248134 -0.017574690 -0.01912545
## cab.E       0.145321443 -0.23009131 -0.047002523 -0.036865157 -0.01655369
## cab.F       0.057934947  0.01106335 -0.008202329  0.001706184  0.02369388
## cab.G       0.016040183  0.05556122 -0.091031410 -0.001401889  0.07238842
##                    Fare      marrWom       isBoy    youngFem      famSize
## Survived    0.25730652  0.339040251  0.08522056  0.17248234  0.016638989
## Pclass     -0.54949962 -0.149209397  0.08208138  0.09767676  0.065996908
```

```
## Sex         -0.18233283 -0.547600334  0.15993449 -0.47654018 -0.200988444
## SibSp        0.15965104  0.063406878  0.34955868  0.20422103  0.890711672
## Parch        0.21622494  0.225851531  0.26734379  0.22532616  0.783110775
## Fare         1.00000000  0.105203278  0.01090842  0.01042077  0.217138407
## marrWom      0.10520328  1.000000000 -0.08758018 -0.14200939  0.156168113
## isBoy        0.01090842 -0.087580181  1.00000000 -0.07621521  0.372471876
## youngFem     0.01042077 -0.142009391 -0.07621521  1.00000000  0.252147273
## famSize      0.21713841  0.156168113  0.37247188  0.25214727  1.000000000
## Embarked.S  -0.16218419  0.002688805  0.02426442 -0.04059351  0.077358516
## Embarked.Q  -0.11721599 -0.089739327  0.01047835  0.04507664 -0.058592086
## Embarked.C   0.26933473  0.061394633 -0.03522489  0.01395681 -0.046215264
## cab.A        0.01954896 -0.052860857  0.01375943 -0.04600129 -0.051767355
## cab.B        0.38629710  0.049245795 -0.02691419  0.04542295  0.004619762
## cab.C        0.36431778  0.074375167 -0.03593680 -0.07918818  0.035346823
## cab.D        0.09887783  0.074786971 -0.04251856 -0.01195949 -0.021566454
## cab.E        0.05371671  0.043603869 -0.01271696 -0.02929527 -0.033466019
## cab.F       -0.03309341  0.004749100  0.10922708  0.01705671  0.013003191
## cab.G       -0.02518035  0.069554755 -0.01455906  0.08370923  0.035205917
##               Embarked.S   Embarked.Q  Embarked.C        cab.A        cab.B
## Survived   -0.149682723  0.003650383  0.16824043  0.022286954  0.175095034
## Pclass      0.074052785  0.221008920 -0.24329208 -0.204934458 -0.369572047
## Sex         0.119223750 -0.074115123 -0.08285347  0.078270705 -0.109689073
## SibSp       0.068733586 -0.026353729 -0.05952822 -0.046266320 -0.034537975
## Parch       0.060813608 -0.081228104 -0.01106877 -0.040325427  0.056497626
## Fare       -0.162184188 -0.117215990  0.26933473  0.019548956  0.386297101
## marrWom     0.002688805 -0.089739327  0.06139463 -0.052860857  0.049245795
## isBoy       0.024264416  0.010478353 -0.03522489  0.013759433 -0.026914193
## youngFem   -0.040593506  0.045076638  0.01395681 -0.046001291  0.045422946
## famSize     0.077358516 -0.058592086 -0.04621526 -0.051767355  0.004619762
## Embarked.S  1.000000000 -0.499420514 -0.78274213 -0.056180053 -0.102062920
## Embarked.Q -0.499420514  1.000000000 -0.14825818 -0.040246372 -0.072578981
## Embarked.C -0.782742129 -0.148258176  1.00000000  0.093040297  0.168641547
## cab.A      -0.056180053 -0.040246372  0.09304030  1.000000000 -0.030879573
## cab.B      -0.102062920 -0.072578981  0.16864155 -0.030879573  1.000000000
## cab.C      -0.068502455 -0.049776135  0.11395198 -0.034846400 -0.062840850
## cab.D      -0.052254042 -0.060317938  0.10297709 -0.025662969 -0.046279753
## cab.E       0.037811645 -0.037896542 -0.01593912 -0.025256431 -0.045546616
## cab.F       0.033009537 -0.004112597 -0.03472608 -0.015922747 -0.028714557
## cab.G       0.041355661 -0.020653866 -0.03237082 -0.008787428 -0.015846957
##                  cab.C       cab.D       cab.E        cab.F        cab.G
## Survived    0.11465212  0.15071564  0.14532144  0.057934947  0.016040183
## Pclass     -0.41704772 -0.27869030 -0.23009131  0.011063349  0.055561218
## Sex        -0.05864936 -0.07924813 -0.04700252 -0.008202329 -0.091031410
## SibSp       0.02925056 -0.01757469 -0.03686516  0.001706184 -0.001401889
## Parch       0.03073575 -0.01912545 -0.01655369  0.023693884  0.072388424
## Fare        0.36431778  0.09887783  0.05371671 -0.033093414 -0.025180348
## marrWom     0.07437517  0.07478697  0.04360387  0.004749100  0.069554755
## isBoy      -0.03593680 -0.04251856 -0.01271696  0.109227079 -0.014559062
## youngFem   -0.07918818 -0.01195949 -0.02929527  0.017056705  0.083709233
## famSize     0.03534682 -0.02156645 -0.03346602  0.013003191  0.035205917
```

```
## Embarked.S -0.06850246 -0.05225404  0.03781165  0.033009537  0.041355661
## Embarked.Q -0.04977614 -0.06031794 -0.03789654 -0.004112597 -0.020653866
## Embarked.C  0.11395198  0.10297709 -0.01593912 -0.034726083 -0.032370818
## cab.A      -0.03484640 -0.02566297 -0.02525643 -0.015922747 -0.008787428
## cab.B      -0.06284085 -0.04627975 -0.04554662 -0.028714557 -0.015846957
## cab.C       1.00000000 -0.05222491 -0.05139759 -0.032403264 -0.017882677
## cab.D      -0.05222491  1.00000000 -0.03785225 -0.023863698 -0.013169871
## cab.E      -0.05139759 -0.03785225  1.00000000 -0.023485663 -0.012961241
## cab.F      -0.03240326 -0.02386370 -0.02348566  1.000000000 -0.008171327
## cab.G      -0.01788268 -0.01316987 -0.01296124 -0.008171327  1.000000000
```

Lets start modeling. We will begin with logistic regression and validate our misclassification error rate using cross validation. In order to show how to script cross validation lets try it with a simple model using Pclass and Sex. We get a success rate of 78.5% which is not too bad. Remember we do not want to overfit our model on the training set because that would lead to a reduction of accuracy on the test set.

```
attach(tit_train)
```

```
## The following objects are masked _by_ .GlobalEnv:
##
##     isBoy, marrWom, youngFem
```

```
K = 5
folds <- sample(1:K, nrow(tit_train), replace = TRUE)
error <- rep(0, 5)

for(i in 1:K){
  log.fit <- glm(Survived~Pclass + Sex,
                 data = tit_train,
                 subset = which(folds != i),
                 family = binomial)
  log.train.probs <- predict(log.fit, newdata = tit_train[folds == i,], type = "respo
nse")
  log.train.preds <- ifelse(log.train.probs >= 0.5, 1, 0)
  error[i] <- mean(log.train.preds != Survived[folds == i])
}
print(paste('Error rate: ', mean(error)))
```

```
## [1] "Error rate:  0.21322582972583"
```

```
print(paste('Success rate: ', 1-mean(error)))
```

```
## [1] "Success rate:  0.78677417027417"
```

Now lets create a model using random forests. This is the model used to score 0.80383 and landed me at number 1076 on the leaderboard as of Tue, 03 Jan 2017 22:06:22. Notice that it does not utilize all the features we had created initially.

```
library(randomForest)
```

```
## randomForest 4.6-12
```

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

```
set.seed(1)
rf.fit <- randomForest(factor(Survived) ~Pclass + Sex + famSize+
                         isBoy + youngFem + marrWom
                       + Embarked.S + Embarked.C + Embarked.Q,
                       data = tit_train, mtry = 2, ntree = 15000, nodesize = 1,
                       importance = FALSE)
rf.pred.train <- predict(rf.fit, newdata = tit_train, type = "class")
print(paste('Success rate on training data: ', mean(rf.pred.train == tit_train$Surviv
ed)))
```

```
## [1] "Success rate on training data:  0.836139169472503"
```

```
rf.pred.out <- predict(rf.fit, newdata = tit_test, type = "class")
```

Using our predictions from the model we can write it out to a csv file of choice. Simply run the method using the filename of choice as the parameter

```
results <- data.frame(PassengerId = 892:1309, Survived = rf.pred.out)

write_results_csv <- function(filename){
  write.csv(results, filename, row.names = FALSE)
}
```