Kaggle Titanic

John Cook

Introduction

In the below RMarkdown File I develop a Random Forest for the Kaggle Titanic Competition.

The competition is to predict who lived and who died on the Titanic. We are given a training set of 891 observations and a test set of 418 observations. The two sets contain 11 columns containing information such as name, age, ticket no., social class, gender,etc. Additionally the training set contains a binary "Survived" column where a 0 indicates death. More information can be found at .

Setup

```
# setwd to git repo
setwd('/Users/johncook/repos/Kaggle_Titanic/')
```

```
#read in data
train=data.frame(read.csv('Data/train.csv',header = 1))
test=data.frame(read.csv('Data/test.csv',header = 1))
```

```
#Check to see if necessary packages installed, if not install
list.of.packages <- c("randomForest", "data.table","tree","ggplot2","dplyr","gsubfn")
new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]
if(length(new.packages)) install.packages(new.packages)
```

```
#Load Libraries:
library(randomForest)
```

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

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

```
library(data.table)
library(tree)
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:randomForest':
##
##     margin
```

```
library(dplyr)
```

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

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

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

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

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

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

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

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

```
library(gsubfn)
```

```
## Loading required package: proto
```

Initial Analysis

```
head(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
```

Quick Look at contingency tables and histograms to see which variables seperate the most between died and survived passengers. Clearly Sex is the largest single predictor.

```
prop.table(table(train$Survived,train$Sex),2)
```

```
##
##        female      male
##   0 0.2579618 0.8110919
##   1 0.7420382 0.1889081
```

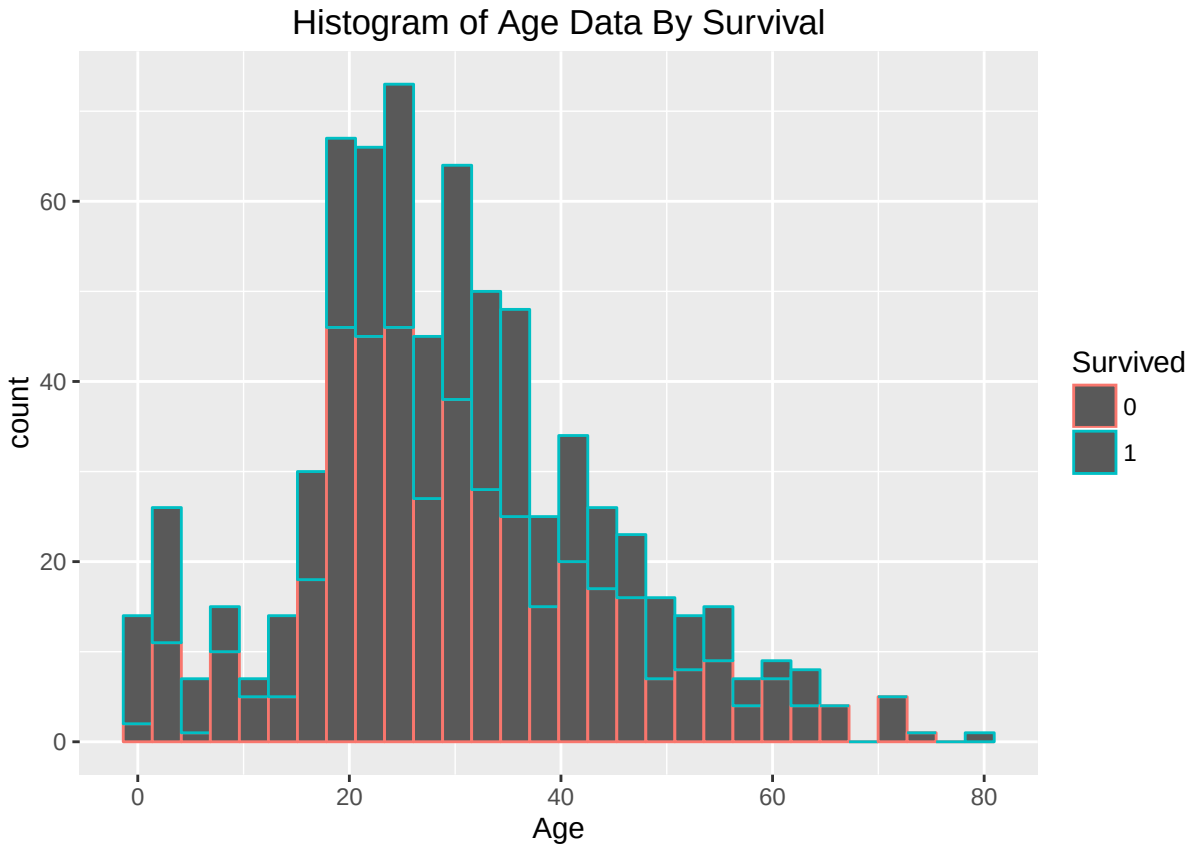
```
prop.table(table(train$Survived,train$Pclass),2)
```

```
##
##             1         2         3
##   0 0.3703704 0.5271739 0.7576375
##   1 0.6296296 0.4728261 0.2423625
```

```
ggplot(train,aes(Age,colour=as.factor(Survived)))+geom_histogram()+ggtitle("Histogram of Age Data By Su
```

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

```
## Warning: Removed 177 rows containing non-finite values (stat_bin).
```




```
prop.table(table(train$SibSp,train$Survived),1)
```

```
##
##             0         1
##   0 0.6546053 0.3453947
##   1 0.4641148 0.5358852
##   2 0.5357143 0.4642857
##   3 0.7500000 0.2500000
##   4 0.8333333 0.1666667
##   5 1.0000000 0.0000000
##   8 1.0000000 0.0000000
```

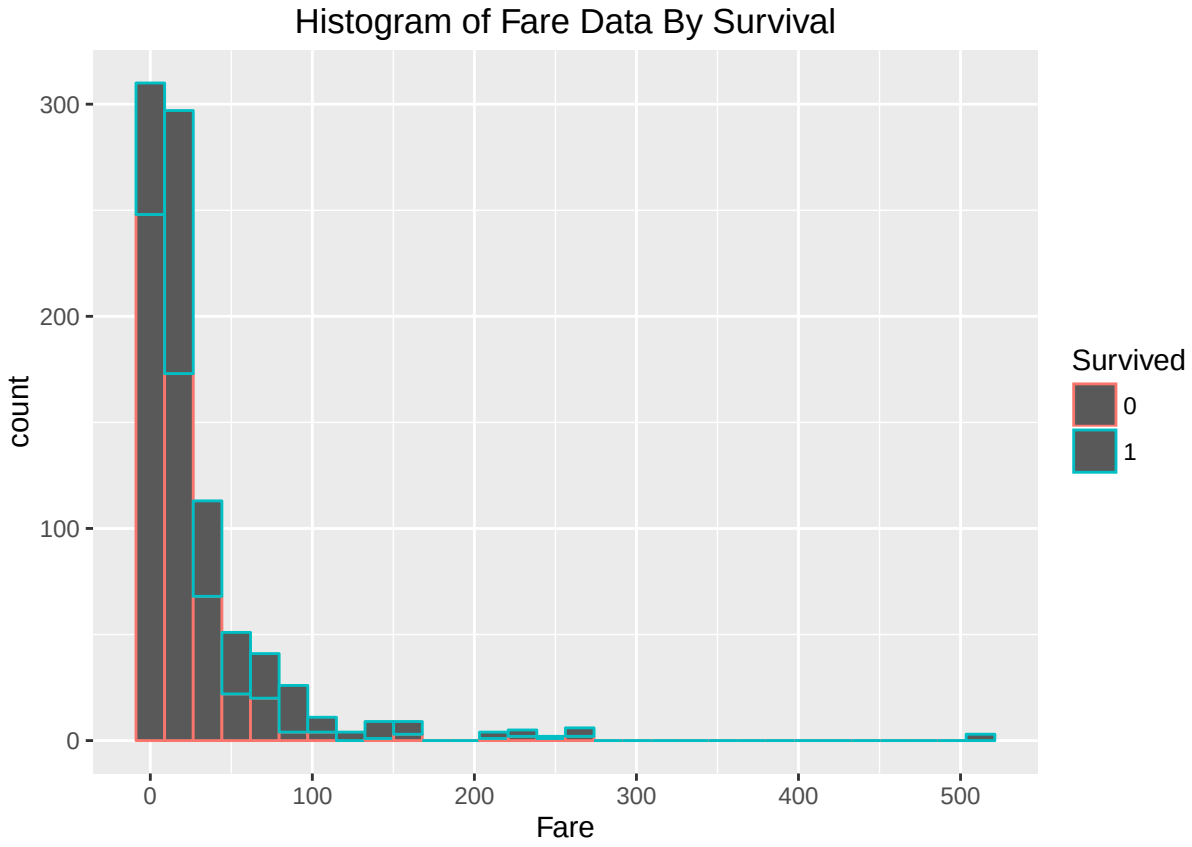
```
prop.table(table(train$Parch,train$Survived),1)
```

```
##
##             0         1
```

```
##   0 0.6563422 0.3436578
##   1 0.4491525 0.5508475
##   2 0.5000000 0.5000000
##   3 0.4000000 0.6000000
##   4 1.0000000 0.0000000
##   5 0.8000000 0.2000000
##   6 1.0000000 0.0000000
```

```
ggplot(train,aes(Fare,colour=as.factor(Survived)))+geom_histogram()+ggtitle("Histogram of Fare Data By S
```

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




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

```
##
##       0   1
##       0   2
##   C  75  93
##   Q  47  30
##   S 427 217
```

Creating New Variables

In this section I create new variables from given columns.

Making Column of Titles

In this section I take out the title from each person's name and create a new column. I then group together like title's that are sparse. Combining like titles will prevent overfitting by reducing the variables cardinality(number possible values). Note the underlying problem are the titles that have only a few entries as their Survived proportion's can vary significantly between the training and testing sets.

```
#Strip out the titles using regex
train$Title <- strapplyc(as.character(train$Name), ", (.*?)\\.",simplify=T)
table(train$Title,train$Survived)
```

```
##
##                  0   1
##   Capt           1   0
##   Col            1   1
##   Don            1   0
##   Dr             4   3
##   Jonkheer       1   0
##   Lady           0   1
##   Major          1   1
##   Master        17  23
##   Miss          55 127
##   Mlle           0   2
##   Mme            0   1
##   Mr           436  81
##   Mrs           26  99
##   Ms             0   1
##   Rev            6   0
##   Sir            0   1
##   the Countess   0   1
```

```
test$Title <- strapplyc(as.character(test$Name), ", (.*?)\\.",simplify=T)
table(test$Title)
```

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

I reduce the number of titles into: Mr Mrs Miss Master

and my new classes of: Job Title Formal Title

```
train$Title <- as.character(train$Title)
test$Title <- as.character(test$Title)
replTitle<- function(repl,rwith){
  train[train$Title %in% repl,]['Title']=rwith
  test[test$Title %in% repl,]['Title']=rwith
  return(train,test)
}
Jobs <- c('Capt','Col','Major','Dr',"Rev")
train[train$Title %in% Jobs,]['Title'] ='Job'
test[test$Title %in% Jobs,]['Title'] ='Job'

FTitles <- c('Jonkheer','Don','Sir','the Countess','Dona','Lady')
train[train$Title %in% FTitles,]['Title'] ='Ftitle'
test[test$Title %in% FTitles,]['Title'] ='Ftitle'
```

```
MrsTitles <- c('Mme','Ms')
train[train$Title %in% MrsTitles,]['Title'] ='Mrs'
test[test$Title %in% MrsTitles,]['Title'] ='Mrs'

MissTitles <- c('Mlle')
train[train$Title %in% MissTitles,]['Title'] ='Miss'
#test[test$Title %in% MissTitles,]['Title'] ='Miss'
#Above edited out because Mlle isn't present

train$Title <- as.factor(train$Title)
test$Title <- factor(test$Title,levels=levels(train$Title))

#At least now the training set has at minimum 5 entries

table(train$Title)
```

```
##
## Ftitle    Job Master   Miss     Mr    Mrs
##      5     18     40    184    517    127
```

Making Columns of Cabin Sections, Number of Cabin rooms, And Room Number.

In this section I split up the Cabin column into the section (the beginning letter of the cabin), the number of cabin rooms booked, and the room number. For example if an entry is "C85 C86". The associated columns would be section C, 2 rooms, and room number 85.

I take the first room number as no cabin rooms purchased by the same person is more than a few rooms apart (Mainly multiple purchasers are heads of a household and buy rooms for there family close to each other).

Unfortunately relatively few passengers have Cabin information so these will be fairly sparse columns.

```
Cabs=strsplit(as.character(train$Cabin), " ")
Cabstest=strsplit(as.character(test$Cabin), " ")
train$Section <- substr(as.character(train$Cabin),1,1)
test$Section <- substr(as.character(test$Cabin),1,1)
train$NumRms <- sapply(Cabs,length)
test$NumRms <- sapply(Cabstest,length)
#Create a function to deal with character(0) problem for string manipulation in r
substrMY <- function(x){
  if (identical(x,character(0))){
    return(NA)
  }
  else{
  return(substr(x[[1]][1], 2, nchar(x)))
  }
}
train$RNum <- unlist(sapply(Cabs, substrMY))
test$RNum <- unlist(sapply(Cabstest, substrMY))
```

Calculating Family Size

Here we calculate family size by adding the Sibling/Spouse column to the Parents/Children column.

```
train$FSize <- train$SibSp + train$Parch
test$FSize <- test$SibSp + test$Parch
```

Identifying Missing and Strange Values

The columns that we will look at in this section are Age and Fare. We have already mentioned the missing values in Cabin.

First we will look at the Fare variable and then at Age.

Correcting Fare Column

There's 1 NA for Fare in the Test dataset. Additionally we will see below that there are strange values for Fare as well which we will attempt to correct.

We fill in the 1 NA for Fare in the test dataset by using the mean of our data which fits criteria most similar to our missing value passenger.

```
test <- data.frame(test)
train <- data.frame(train)
test %>% summarise_each(funs(sum(is.na(.))))
```

```
##   PassengerId Pclass Name Sex Age SibSp Parch Ticket Fare Cabin Embarked
## 1           0      0    0   0  86     0     0      0    1     0        0
##   Title Section NumRms RNum FSize
## 1     0       0      0  327     0
```

```
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 Section NumRms RNum FSize
## 153   NA              S    Mr              0 <NA>     0
```

```
#No alike entries in test df but alike entries in train
test[which(test$Age >50 & test$Pclass=='3' & test$Sex=='male'),]
```

```
##     PassengerId Pclass               Name  Sex  Age SibSp Parch Ticket
## 153        1044      3 Storey, Mr. Thomas male 60.5     0     0   3701
##     Fare Cabin Embarked Title Section NumRms RNum FSize
## 153   NA              S    Mr              0 <NA>     0
```

```
train[which(train$Age >50 & train$Pclass=='3' & train$Sex=='male'),]
```

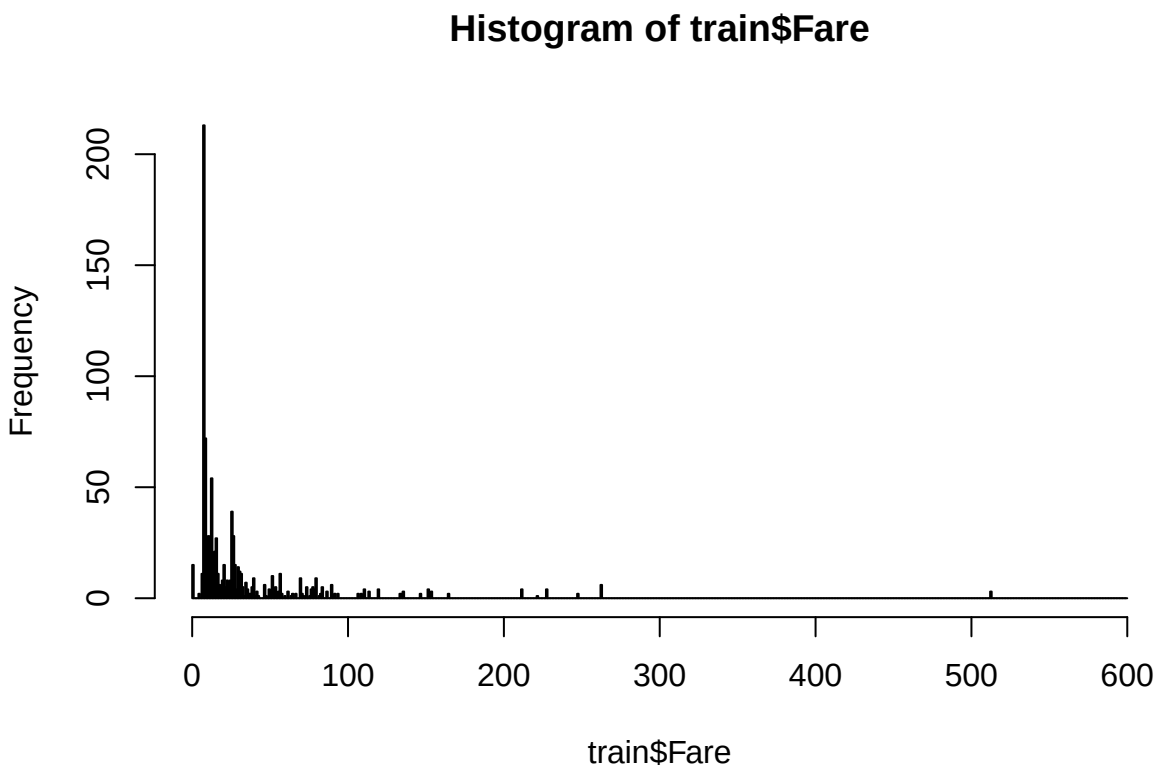
```
##     PassengerId Survived Pclass                            Name  Sex  Age
## 95           95        0      3                Coxon, Mr. Daniel male 59.0
## 117         117        0      3             Connors, Mr. Patrick male 70.5
## 153         153        0      3                 Meo, Mr. Alfonzo male 55.5
## 223         223        0      3          Green, Mr. George Henry male 51.0
## 281         281        0      3                 Duane, Mr. Frank male 65.0
## 327         327        0      3        Nysveen, Mr. Johan Hansen male 61.0
## 407         407        0      3 Widegren, Mr. Carl/Charles Peter male 51.0
## 632         632        0      3      Lundahl, Mr. Johan Svensson male 51.0
## 852         852        0      3               Svensson, Mr. Johan male 74.0
##     SibSp Parch     Ticket   Fare Cabin Embarked Title Section NumRms RNum
```

```
## 95      0     0     364500 7.2500             S    Mr              0 <NA>
## 117     0     0     370369 7.7500             Q    Mr              0 <NA>
## 153     0     0 A.5. 11206 8.0500             S    Mr              0 <NA>
## 223     0     0      21440 8.0500             S    Mr              0 <NA>
## 281     0     0     336439 7.7500             Q    Mr              0 <NA>
## 327     0     0     345364 6.2375             S    Mr              0 <NA>
## 407     0     0     347064 7.7500             S    Mr              0 <NA>
## 632     0     0     347743 7.0542             S    Mr              0 <NA>
## 852     0     0     347060 7.7750             S    Mr              0 <NA>
##     FSize
## 95      0
## 117     0
## 153     0
## 223     0
## 281     0
## 327     0
## 407     0
## 632     0
## 852     0
```

```
#Set it equal to the mean Fare of these rows
test[is.na(test$Fare),][c('Fare')] <- sum(train[which(train$Age >50 & train$Pclass=='3' & train$Sex=='ma
```

Below we look at the distirbution of Fares and investigate interesting outliers with Fares =0, and >200. If the fares >200 look normal in the other columns, i.e. Should be high class and have cabins.

```
hist(train$Fare,breaks = c(0:600))
```



Investigate fares >200. We see that these are all passengers of the top class and most have cabins in B or C. Because of this I will leave these fares as they are as I have little evidence they are "bad data".

```
train[train$Fare > 200,]
```

```
##     PassengerId Survived Pclass
## 28           28        0      1
## 89           89        1      1
## 119         119        0      1
## 259         259        1      1
## 300         300        1      1
## 312         312        1      1
## 342         342        1      1
## 378         378        0      1
## 381         381        1      1
## 439         439        0      1
## 528         528        0      1
## 558         558        0      1
## 680         680        1      1
## 690         690        1      1
## 701         701        1      1
## 717         717        1      1
## 731         731        1      1
## 738         738        1      1
## 743         743        1      1
## 780         780        1      1
##                                                    Name    Sex Age SibSp
## 28                            Fortune, Mr. Charles Alexander   male  19     3
## 89                                Fortune, Miss. Mabel Helen female  23     3
## 119                                 Baxter, Mr. Quigg Edmond   male  24     0
## 259                                         Ward, Miss. Anna female  35     0
## 300       Baxter, Mrs. James (Helene DeLaudeniere Chaput) female  50     0
## 312                               Ryerson, Miss. Emily Borie female  18     2
## 342                            Fortune, Miss. Alice Elizabeth female  24     3
## 378                                Widener, Mr. Harry Elkins   male  27     0
## 381                                    Bidois, Miss. Rosalie female  42     0
## 439                                        Fortune, Mr. Mark   male  64     1
## 528                                     Farthing, Mr. John   male  NA     0
## 558                                     Robbins, Mr. Victor   male  NA     0
## 680                         Cardeza, Mr. Thomas Drake Martinez   male  36     0
## 690                          Madill, Miss. Georgette Alexandra female  15     0
## 701     Astor, Mrs. John Jacob (Madeleine Talmadge Force) female  18     1
## 717                             Endres, Miss. Caroline Louise female  38     0
## 731                             Allen, Miss. Elisabeth Walton female  29     0
## 738                                   Lesurer, Mr. Gustave J   male  35     0
## 743                    Ryerson, Miss. Susan Parker "Suzette" female  21     2
## 780 Robert, Mrs. Edward Scott (Elisabeth Walton McMillan) female  43     0
##     Parch   Ticket     Fare           Cabin Embarked Title Section NumRms
## 28      2    19950 263.0000     C23 C25 C27        S    Mr       C      3
## 89      2    19950 263.0000     C23 C25 C27        S  Miss       C      3
## 119     1 PC 17558 247.5208         B58 B60        C    Mr       B      2
## 259     0 PC 17755 512.3292                        C  Miss                0
## 300     1 PC 17558 247.5208         B58 B60        C   Mrs       B      2
## 312     2 PC 17608 262.3750 B57 B59 B63 B66        C  Miss       B      4
## 342     2    19950 263.0000     C23 C25 C27        S  Miss       C      3
## 378     2   113503 211.5000             C82        C    Mr       C      1
## 381     0 PC 17757 227.5250                        C  Miss                0
```

```
## 439     4    19950 263.0000     C23 C25 C27        S    Mr       C      3
## 528     0 PC 17483 221.7792             C95        S    Mr       C      1
## 558     0 PC 17757 227.5250                        C    Mr                0
## 680     1 PC 17755 512.3292     B51 B53 B55        C    Mr       B      3
## 690     1    24160 211.3375              B5        S  Miss       B      1
## 701     0 PC 17757 227.5250         C62 C64        C   Mrs       C      2
## 717     0 PC 17757 227.5250             C45        C  Miss       C      1
## 731     0    24160 211.3375              B5        S  Miss       B      1
## 738     0 PC 17755 512.3292            B101        C    Mr       B      1
## 743     2 PC 17608 262.3750 B57 B59 B63 B66        C  Miss       B      4
## 780     1    24160 211.3375              B3        S   Mrs       B      1
##     RNum FSize
## 28    23     5
## 89    23     5
## 119   58     1
## 259 <NA>     0
## 300   58     1
## 312   57     4
## 342   23     5
## 378   82     2
## 381 <NA>     0
## 439   23     5
## 528   95     0
## 558 <NA>     0
## 680   51     1
## 690    5     1
## 701   62     1
## 717   45     0
## 731    5     0
## 738  101     0
## 743   57     4
## 780    3     1
```

```
test[test$Fare > 200,]
```

```
##     PassengerId Pclass
## 25          916      1
## 54          945      1
## 60          951      1
## 65          956      1
## 70          961      1
## 75          966      1
## 76          967      1
## 82          973      1
## 115        1006      1
## 143        1034      1
## 157        1048      1
## 185        1076      1
## 203        1094      1
## 219        1110      1
## 325        1216      1
## 344        1235      1
## 376        1267      1
## 408        1299      1
##                                                      Name    Sex
```

```
## 25                  Ryerson, Mrs. Arthur Larned (Emily Maria Borie) female
## 54                                       Fortune, Miss. Ethel Flora female
## 60                                     Chaudanson, Miss. Victorine female
## 65                                     Ryerson, Master. John Borie   male
## 70                                Fortune, Mrs. Mark (Mary McDougald) female
## 75                                             Geiger, Miss. Amalie female
## 76                                               Keeping, Mr. Edwin   male
## 82                                               Straus, Mr. Isidor   male
## 115                           Straus, Mrs. Isidor (Rosalie Ida Blun) female
## 143                                    Ryerson, Mr. Arthur Larned   male
## 157                                              Bird, Miss. Ellen female
## 185            Douglas, Mrs. Frederick Charles (Mary Helene Baxter) female
## 203                                          Astor, Col. John Jacob   male
## 219                     Widener, Mrs. George Dunton (Eleanor Elkins) female
## 325                                          Kreuchen, Miss. Emilie female
## 344 Cardeza, Mrs. James Warburton Martinez (Charlotte Wardle Drake) female
## 376                                        Bowen, Miss. Grace Scott female
## 408                                      Widener, Mr. George Dunton   male
##      Age SibSp Parch   Ticket     Fare           Cabin Embarked  Title
## 25  48.0     1     3 PC 17608 262.3750 B57 B59 B63 B66        C    Mrs
## 54  28.0     3     2    19950 263.0000     C23 C25 C27        S   Miss
## 60  36.0     0     0 PC 17608 262.3750             B61        C   Miss
## 65  13.0     2     2 PC 17608 262.3750 B57 B59 B63 B66        C Master
## 70  60.0     1     4    19950 263.0000     C23 C25 C27        S    Mrs
## 75  35.0     0     0   113503 211.5000            C130        C   Miss
## 76  32.5     0     0   113503 211.5000            C132        C     Mr
## 82  67.0     1     0 PC 17483 221.7792         C55 C57        S     Mr
## 115 63.0     1     0 PC 17483 221.7792         C55 C57        S    Mrs
## 143 61.0     1     3 PC 17608 262.3750 B57 B59 B63 B66        C     Mr
## 157 29.0     0     0 PC 17483 221.7792             C97        S   Miss
## 185 27.0     1     1 PC 17558 247.5208         B58 B60        C    Mrs
## 203 47.0     1     0 PC 17757 227.5250         C62 C64        C    Job
## 219 50.0     1     1   113503 211.5000             C80        C    Mrs
## 325 39.0     0     0    24160 211.3375                        S   Miss
## 344 58.0     0     1 PC 17755 512.3292     B51 B53 B55        C    Mrs
## 376 45.0     0     0 PC 17608 262.3750                        C   Miss
## 408 50.0     1     1   113503 211.5000             C80        C     Mr
##     Section NumRms RNum FSize
## 25        B      4   57     4
## 54        C      3   23     5
## 60        B      1   61     0
## 65        B      4   57     4
## 70        C      3   23     5
## 75        C      1  130     0
## 76        C      1  132     0
## 82        C      2   55     1
## 115       C      2   55     1
## 143       B      4   57     4
## 157       C      1   97     0
## 185       B      2   58     2
## 203       C      2   62     1
## 219       C      1   80     2
## 325              0 <NA>     0
## 344       B      3   51     1
```

```
## 376              0 <NA>     0
## 408       C      1   80     2
```

```
# Doesn't look like anything out of the ordinary will look at the two tickes >500
train[train$Fare>500,]
```

```
##     PassengerId Survived Pclass                                Name    Sex
## 259         259        1      1                    Ward, Miss. Anna female
## 680         680        1      1 Cardeza, Mr. Thomas Drake Martinez   male
## 738         738        1      1             Lesurer, Mr. Gustave J   male
##     Age SibSp Parch   Ticket     Fare       Cabin Embarked Title Section
## 259  35     0     0 PC 17755 512.3292                   C  Miss
## 680  36     0     1 PC 17755 512.3292 B51 B53 B55        C    Mr       B
## 738  35     0     0 PC 17755 512.3292        B101        C    Mr       B
##     NumRms RNum FSize
## 259      0 <NA>     0
## 680      3   51     1
## 738      1  101     0
```

```
test[test$Fare > 500,]
```

```
##     PassengerId Pclass
## 344        1235      1
##                                                      Name    Sex
## 344 Cardeza, Mrs. James Warburton Martinez (Charlotte Wardle Drake) female
##     Age SibSp Parch   Ticket     Fare       Cabin Embarked Title Section
## 344  58     0     1 PC 17755 512.3292 B51 B53 B55        C   Mrs       B
##     NumRms RNum FSize
## 344      3   51     1
```

```
#I can believe that these Fares are correct that a really expensive ticket was bought instead of this b
```

Investigating fares =0. It is interesting that these people are all male and all but one died. Additionally they all Embarked from the same place. I would suggest that these may be crewman as they are all males of working age but the variation in the other columns refutes this hypothesis. Additionaly this is supposed to be a passenger manifest.

```
train[train$Fare ==0,]
```

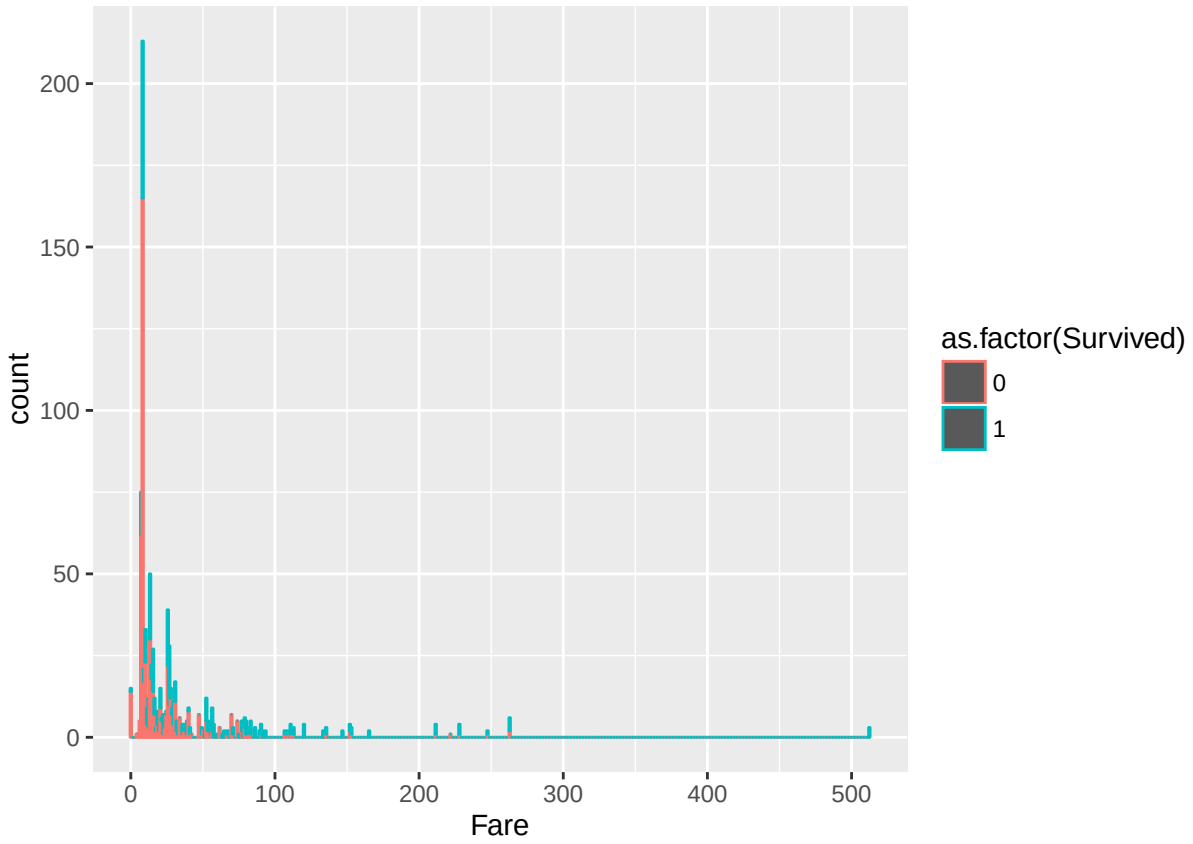
```
##     PassengerId Survived Pclass                                Name  Sex Age
## 180         180        0      3                Leonard, Mr. Lionel male  36
## 264         264        0      1              Harrison, Mr. William male  40
## 272         272        1      3       Tornquist, Mr. William Henry male  25
## 278         278        0      2        Parkes, Mr. Francis "Frank" male  NA
## 303         303        0      3  Johnson, Mr. William Cahoone Jr male  19
## 414         414        0      2   Cunningham, Mr. Alfred Fleming male  NA
## 467         467        0      2              Campbell, Mr. William male  NA
## 482         482        0      2 Frost, Mr. Anthony Wood "Archie" male  NA
## 598         598        0      3                Johnson, Mr. Alfred male  49
## 634         634        0      1    Parr, Mr. William Henry Marsh male  NA
## 675         675        0      2       Watson, Mr. Ennis Hastings male  NA
## 733         733        0      2              Knight, Mr. Robert J male  NA
## 807         807        0      1            Andrews, Mr. Thomas Jr male  39
## 816         816        0      1                  Fry, Mr. Richard male  NA
## 823         823        0      1  Reuchlin, Jonkheer. John George male  38
##     SibSp Parch Ticket Fare Cabin Embarked  Title Section NumRms RNum
## 180     0     0   LINE    0              S     Mr              0 <NA>
```

```
## 264     0      0 112059    0   B94        S      Mr        B       1   94
## 272     0      0   LINE    0              S      Mr                0 <NA>
## 278     0      0 239853    0              S      Mr                0 <NA>
## 303     0      0   LINE    0              S      Mr                0 <NA>
## 414     0      0 239853    0              S      Mr                0 <NA>
## 467     0      0 239853    0              S      Mr                0 <NA>
## 482     0      0 239854    0              S      Mr                0 <NA>
## 598     0      0   LINE    0              S      Mr                0 <NA>
## 634     0      0 112052    0              S      Mr                0 <NA>
## 675     0      0 239856    0              S      Mr                0 <NA>
## 733     0      0 239855    0              S      Mr                0 <NA>
## 807     0      0 112050    0   A36        S      Mr        A       1   36
## 816     0      0 112058    0  B102        S      Mr        B       1  102
## 823     0      0  19972    0              S Ftitle                0 <NA>
##     FSize
## 180     0
## 264     0
## 272     0
## 278     0
## 303     0
## 414     0
## 467     0
## 482     0
## 598     0
## 634     0
## 675     0
## 733     0
## 807     0
## 816     0
## 823     0
```

```
test[test$Fare ==0,]
```

```
##     PassengerId Pclass                                    Name  Sex Age
## 267        1158      1 Chisholm, Mr. Roderick Robert Crispin male  NA
## 373        1264      1               Ismay, Mr. Joseph Bruce male  49
##     SibSp Parch Ticket Fare       Cabin Embarked Title Section NumRms RNum
## 267     0     0 112051    0                   S    Mr              0 <NA>
## 373     0     0 112058    0 B52 B54 B56        S    Mr       B      3   52
##     FSize
## 267     0
## 373     0
```

```
ggplot(train,aes(Fare,colour=as.factor(Survived)))+geom_histogram(bins = 500)
```



Below I make sure that passengers with the same Ticket No. paid about the same amount. The below code only outputs values if two different Fares are present for the same Ticket. Unfortunately this is not the case for the Fares of 0 so Ticket # will not be of help.

```
train$TicketCl <- as.character(train$Ticket)
uniqs <- unique(train$TicketCl)
for (i in 1:length(uniqs)){
  vals<-c()
  for (j in 1:length(train$TicketCl)){
  if ((uniqs[i]==train$TicketCl[j])){
    vals <- append(vals,train$Fare[j])
    }
  }
  if (length(unique(vals))>1){
  print(vals)
  print(uniqs[i])
  print(sqrt(var(vals)))
  }
}
```

```
## [1] 9.2167 9.8458
## [1] "7534"
## [1] 0.4448409
```

I'll set these Fares equal to the median Fare of the corresponding Pclass.

```
train$FareCl <- train$Fare
test$FareCl <- test$Fare
aggregate(Fare~Pclass,train,median)
```

```
##   Pclass    Fare
## 1      1 60.2875
## 2      2 14.2500
## 3      3  8.0500
```

```
train[(train$FareCl == 0)&(train$Pclass==1),]['FareCl'] <- 60.2875
train[(train$FareCl == 0)&(train$Pclass==2),]['FareCl'] <- 14.2500
train[(train$FareCl == 0)&(train$Pclass==3),]['FareCl'] <- 8.0500
test[(test$FareCl == 0)&(test$Pclass==1),]['FareCl'] <- 60.2875
```

Missing Values in Age

```
#All in Age for the training set
train %>% group_by(Survived) %>%  summarise_each(funs(sum(is.na(.))))
```

```
## # A tibble: 2 × 19
##   Survived PassengerId Pclass  Name   Sex   Age SibSp Parch Ticket  Fare
##      <int>       <int>  <int> <int> <int> <int> <int> <int>  <int> <int>
## 1        0           0      0     0     0   125     0     0      0     0
## 2        1           0      0     0     0    52     0     0      0     0
## # ... with 9 more variables: Cabin <int>, Embarked <int>, Title <int>,
## #   Section <int>, NumRms <int>, RNum <int>, FSize <int>, TicketCl <int>,
## #   FareCl <int>
```

```
#Is the percentage of survived significantly different of the NAs then the total population?
phat=52/(125+52)
p0=sum(train$Survived)/nrow(train)
zscore=(phat-p0)/sqrt((p0*(1-p0))/177)
pvalue2sided=2*pnorm(-abs(zscore))
pvalue2sided
```

```
## [1] 0.01375638
```

Above the two sided proportion test suggests that the survived proportion is significantly different among the people where age is NA than where age is available. This suggests that a easy technique such as taking the mean or median of Age is not a great approach. This is because some of the variables that are affecting Survival are also affecting who has their Age missing.

Building Model to Predict Age

Age predictor By Random Forest

```
str(train)
```

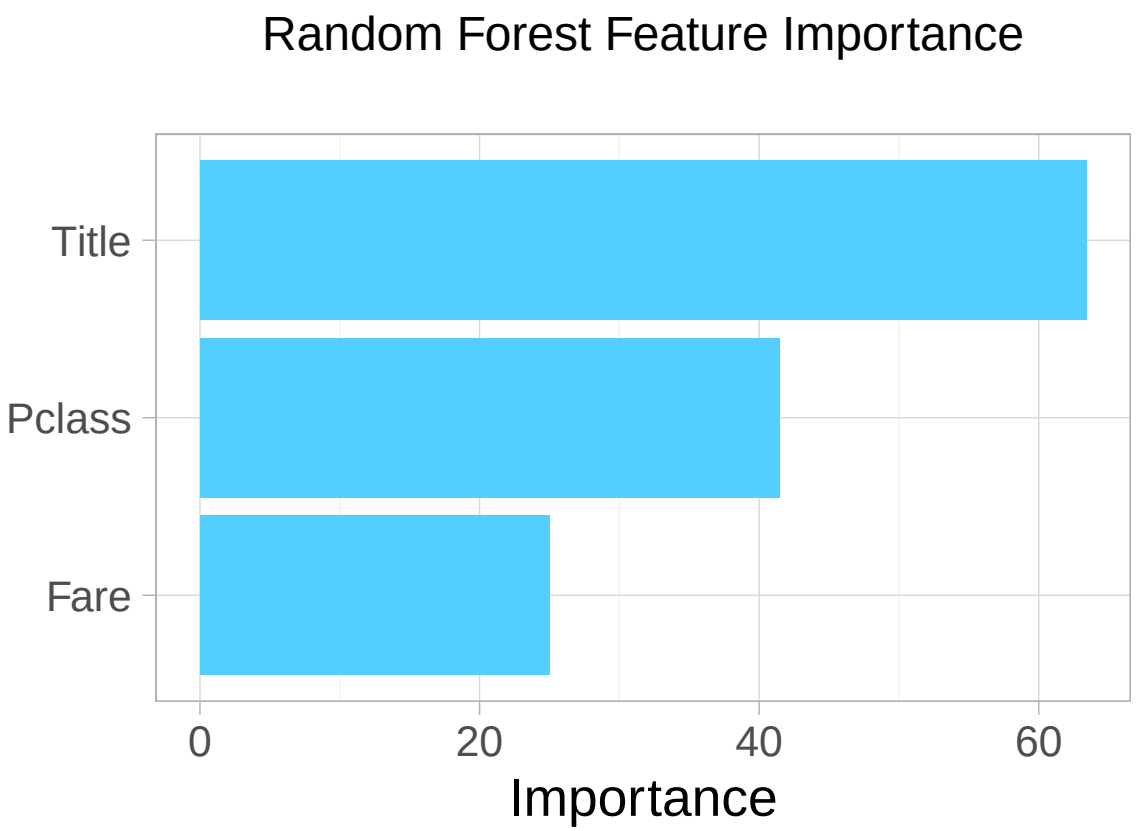
```
## 'data.frame':    891 obs. of  19 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : Factor w/ 891 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520 629 417 581
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ 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     : Factor w/ 681 levels "110152","110413",..: 524 597 670 50 473 276 86 396 345 133 ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : Factor w/ 148 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 1 1 ...
##  $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
##  $ Title      : Factor w/ 6 levels "Ftitle","Job",..: 5 6 4 6 5 5 5 3 6 6 ...
##  $ Section    : chr  "" "C" "" "C" ...
##  $ NumRms     : int  0 1 0 1 0 0 1 0 0 0 ...
##  $ RNum       : chr  NA "85" NA "123" ...
##  $ FSize      : int  1 1 0 1 0 0 0 4 2 1 ...
##  $ TicketCl   : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ FareCl     : num  7.25 71.28 7.92 53.1 8.05 ...
```

```
train$Section <- as.factor(train$Section)
test$Section <- factor(test$Section,levels=levels(train$Section))

train$AgeNas<- is.na(train$Age)
test$AgeNas <- is.na(test$Age)
cols=c('Age','Title','Pclass','Fare')
rf <- randomForest(Age~Title+Pclass+Fare,data=rbind(train[!is.na(train$Age),][cols],test[!is.na(test$Ag
rf
```

```
##
## Call:
##  randomForest(formula = Age ~ Title + Pclass + Fare, data = rbind(train[!is.na(train$Age),      ][co
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 120.7087
##                     % Var explained: 41.84
```

```
imp <- importance(rf, type=1)
featureImportance <- data.frame(Feature=row.names(imp), Importance=imp[,1])

p <- ggplot(featureImportance, aes(x=reorder(Feature, Importance), y=Importance)) +
     geom_bar(stat="identity", fill="#53cfff") +
     coord_flip() +
     theme_light(base_size=20) +
     xlab("") +
     ylab("Importance") +
     ggtitle("Random Forest Feature Importance\n") +
     theme(plot.title=element_text(size=18))

p
```




```
preds <- predict(rf, rbind(train[is.na(train$Age),][cols],test[is.na(test$Age),][cols]))
sep <- nrow(train[is.na(train$Age),])
test[is.na(test$Age),]['Age'] <- preds[(sep+1):length(preds)]
train[is.na(train$Age),]['Age'] <- preds[1:sep]
```

Final Variable: Find Relatives who Died/Survived.

The ability to incorporate this variable into the model depends on the question we are trying to answer. If we are trying to predict whether, given peoples information from the travel manifest, they survived or not then use of the given training set in this manner would be unhelpful. However, if answering who among the passengers in their test set survived it seems clear that this variable could be helpful. The underlying assumption to this are that the passengers survival is not independent people's outcome should be related to the outcome of those they were around when the boat crashed, i.e. either people they're related to or who they came with on the boat.

Relatives who died are for people in both train and test who have relatives in train.

```
train$LName<- strapplyc(as.character(train$Name), "(.*?),",simplify=T)
test$LName<- strapplyc(as.character(test$Name), "(.*?),",simplify=T)
```

At first I tried to do this by last name. I knew that there would be mistakes for common last names. Thus I decided to use ticket #. For example, one can see in the last name Andersson that the ticket numbers match the expected familial relations among passengers of the last name Andersson.

```
train[train$LName =='Andersson',]
```

```
##     PassengerId Survived Pclass
## 14           14        0      3
```

```
## 69           69        1      3
## 120         120        0      3
## 147         147        1      3
## 542         542        0      3
## 543         543        0      3
## 611         611        0      3
## 814         814        0      3
## 851         851        0      3
##                                                    Name    Sex Age
## 14                              Andersson, Mr. Anders Johan   male  39
## 69                            Andersson, Miss. Erna Alexandra female  17
## 120                          Andersson, Miss. Ellis Anna Maria female   2
## 147              Andersson, Mr. August Edvard ("Wennerstrom")   male  27
## 542                       Andersson, Miss. Ingeborg Constanzia female   9
## 543                          Andersson, Miss. Sigrid Elisabeth female  11
## 611 Andersson, Mrs. Anders Johan (Alfrida Konstantia Brogren) female  39
## 814                          Andersson, Miss. Ebba Iris Alfrida female   6
## 851                    Andersson, Master. Sigvard Harald Elias   male   4
##     SibSp Parch  Ticket    Fare Cabin Embarked  Title Section NumRms RNum
## 14      1     5  347082 31.2750              S     Mr              0 <NA>
## 69      4     2 3101281  7.9250              S   Miss              0 <NA>
## 120     4     2  347082 31.2750              S   Miss              0 <NA>
## 147     0     0  350043  7.7958              S     Mr              0 <NA>
## 542     4     2  347082 31.2750              S   Miss              0 <NA>
## 543     4     2  347082 31.2750              S   Miss              0 <NA>
## 611     1     5  347082 31.2750              S    Mrs              0 <NA>
## 814     4     2  347082 31.2750              S   Miss              0 <NA>
## 851     4     2  347082 31.2750              S Master              0 <NA>
##     FSize TicketCl  FareCl AgeNas     LName
## 14      6   347082 31.2750  FALSE Andersson
## 69      6  3101281  7.9250  FALSE Andersson
## 120     6   347082 31.2750  FALSE Andersson
## 147     0   350043  7.7958  FALSE Andersson
## 542     6   347082 31.2750  FALSE Andersson
## 543     6   347082 31.2750  FALSE Andersson
## 611     6   347082 31.2750  FALSE Andersson
## 814     6   347082 31.2750  FALSE Andersson
## 851     6   347082 31.2750  FALSE Andersson
```

In fact I later decided to drop the use of last name altogether. This is because I found some instances where people with the same ticket number do not have the same last name. These people I assume are either related in some way or good enough friends that as the ship was sinking they would group together. This idea of people grouping together is what I am really trying to replicate, i.e. how many people who they would've grouped together with survived/died. Thus last name seems like an unecnessarily strict criteria. However I do both in order to test my hypothesis. It turned out that using ticket no. resulted in a better overall predictor.

```
#Using Last Name and Ticket Number

train$FamDiedCat <- "Unknown"
train$FamDiedCont <- 0
train$FamSurvivedCont <- 0
train$Ticket <- as.character(train$Ticket)
for (i in 1:length(train$Ticket)){
  for (j in 1:length(train$Ticket)){
  if ((train$Ticket[i]==train$Ticket[j])&(i!=j)&(train$LName[i]==train$LName[j])){
```

```
    if (train$Survived[j]==0){
      train$FamDiedCont[i]=train$FamDiedCont[i]+1
    }
    else{
      train$FamSurvivedCont[i]=train$FamSurvivedCont[i]+1
    }
  }
}
}

#Test set

test$FamDiedCat <- "Unknown"
test$FamDiedCont <- 0
test$FamSurvivedCont <- 0
test$Ticket <- as.character(test$Ticket)
for (i in 1:length(test$Ticket)){
  for (j in 1:length(train$Ticket)){
  if ((test$Ticket[i]==train$Ticket[j])&(test$LName[i]==train$LName[j])){
    if (train$Survived[j]==0){
      test$FamDiedCont[i]=test$FamDiedCont[i]+1
    }
    else{
      test$FamSurvivedCont[i]=test$FamSurvivedCont[i]+1
    }
  }
}
}

#Using Ticket Number Except for If Ticket == "LINE" using LName

train$FamDiedCat <- "Unknown"
train$TickDiedCont <- 0
train$TickSurvivedCont <- 0
train$Ticket<- as.character(train$Ticket)
train$TicketCl <- train$Ticket
LINErows <- train$Ticket=="LINE"
train[LINErows,]["TicketCl"]<-paste(train$Ticket[LINErows],train$LName[LINErows])
for (i in 1:length(train$TicketCl)){
  for (j in 1:length(train$TicketCl)){
  if ((train$TicketCl[i]==train$TicketCl[j])&(i!=j)){
    if (train$Survived[j]==0){
      train$TickDiedCont[i]=train$TickDiedCont[i]+1
    }
    else{
      train$TickSurvivedCont[i]=train$TickSurvivedCont[i]+1
    }
  }
}
}

#Test set
```

```
test$TickDiedCat <- "Unknown"
test$TickDiedCont <- 0
test$TickSurvivedCont <- 0
test$TicketCl <- as.character(test$Ticket)
LINErows <- test$Ticket=="LINE"
test[LINErows,]["TicketCl"]<-paste(test$Ticket[LINErows],test$LName[LINErows])
for (i in 1:length(test$TicketCl)){
  for (j in 1:length(train$TicketCl)){
  if ((test$TicketCl[i]==train$TicketCl[j])){
    if (train$Survived[j]==0){
      test$TickDiedCont[i]=test$TickDiedCont[i]+1
    }
    else{
      test$TickSurvivedCont[i]=test$TickSurvivedCont[i]+1
    }
  }
}
}
```

Models

Baseline Model

From the preliminary analysis it is clear that the variable sex clearly has an effect on survival. Thus the easiest baseline model is one in which females survive and males die.

```
test$SimplestPred<-0
test[test$Sex=='female',]['SimplestPred']<-1
ToTest<-test[c('PassengerId','SimplestPred')]
colnames(ToTest) <- c('PassengerId','Survived')
write.csv(ToTest,'./SimplestModel.csv',row.names = F)
```

Thsi model gets a score of 76.5% on the test data. This is a very high score for such a simple model, however we are able to improve accuracy using other variables in a random forest model.

Random Forest Model Without Relatives Survival as variables

The below model uses Passenger class, Sex, Age, Fare, Family size, Section, Embarked, and Title. I chose the parameters to include based on the results from both the 1/10th of the data I set aside for testing as well as the accuracy on the testing set. The model tested at 78.5 % according to Kaggle. Below are in-depth reports as to the performance of the random forest and a bar chart of variable importance.

```
str(train)
```

```
## 'data.frame':    891 obs. of  26 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            : Factor w/ 891 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520 629 4
##  $ Sex             : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age             : num  22 38 26 35 35 ...
##  $ 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           : Factor w/ 148 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 1 1 ...
##  $ Embarked        : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
##  $ Title           : Factor w/ 6 levels "Ftitle","Job",..: 5 6 4 6 5 5 5 3 6 6 ...
##  $ Section         : Factor w/ 9 levels "","A","B","C",..: 1 4 1 4 1 1 6 1 1 1 ...
##  $ NumRms          : int  0 1 0 1 0 0 1 0 0 0 ...
##  $ RNum            : chr  NA "85" NA "123" ...
##  $ FSize           : int  1 1 0 1 0 0 0 4 2 1 ...
##  $ TicketCl        : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ FareCl          : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ AgeNas          : logi  FALSE FALSE FALSE FALSE FALSE TRUE ...
##  $ LName           : chr  "Braund" "Cumings" "Heikkinen" "Futrelle" ...
##  $ FamDiedCat      : chr  "Unknown" "Unknown" "Unknown" "Unknown" ...
##  $ FamDiedCont     : num  0 0 0 1 0 0 0 3 0 1 ...
##  $ FamSurvivedCont : num  0 0 0 0 0 0 0 0 2 0 ...
##  $ TickDiedCont    : num  0 0 0 1 0 0 0 3 0 1 ...
##  $ TickSurvivedCont: num  0 0 0 0 0 0 0 0 2 0 ...
```

```
test$Embarked <- as.character(test$Embarked)
test$Embarked <- factor(test$Embarked,levels=levels(train$Embarked))
train$Survived <- as.factor(train$Survived)
train$Title <- as.factor(train$Title)
test$Title <- factor(test$Title,levels=levels(train$Title))
train$Section <- as.factor(train$Section)
test$Section <- factor(test$Section,levels=levels(train$Section))
library(randomForest)
set.seed(1234)
train1ind=sample(nrow(train),floor(nrow(train)/10))
trainTest <- train[train1ind,]
train1 <- train[-train1ind,]
params <- c("Pclass","Sex","Age","Fare","FSize","Section","Embarked","Title")
fit.rf= randomForest(train1[params], as.factor(train1$Survived),xtest = trainTest[params],trainTest$Surv
fit.rf
```

```
##
## Call:
##  randomForest(x = train1[params], y = as.factor(train1$Survived),      xtest = trainTest[params], yt
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 17.08%
## Confusion matrix:
##     0   1 class.error
## 0 439  48  0.09856263
## 1  89 226  0.28253968
##                 Test set error rate: 16.85%
## Confusion matrix:
##    0  1 class.error
## 0 56  6  0.09677419
## 1  9 18  0.33333333
```

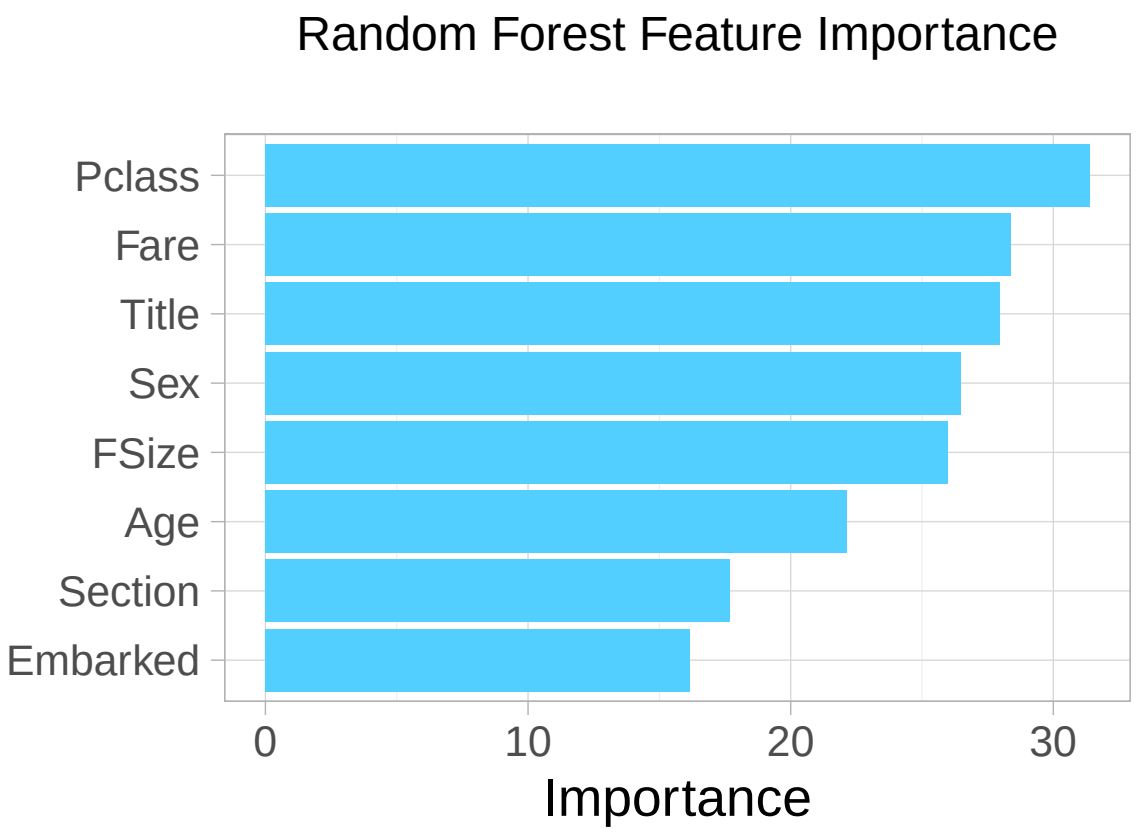
```r
sum(abs(as.numeric(fit.rf$test$predicted) - as.numeric(trainTest$Survived)))/nrow(trainTest)
```

```
## [1] 0.1685393
```

```r
round(importance(fit.rf), 2)
```

```
##              0     1 MeanDecreaseAccuracy MeanDecreaseGini
## Pclass   21.76 20.99                31.38            24.38
## Sex      22.57 23.41                26.48            51.37
## Age      13.91 15.70                22.16            41.01
## Fare     16.83 19.41                28.37            42.70
## FSize    23.28  9.70                25.98            23.47
## Section  16.32  4.07                17.68            20.83
## Embarked  6.92 13.61                16.18             8.43
## Title    24.23 24.01                27.97            65.26
```

```r
imp <- importance(fit.rf, type=1)
featureImportance <- data.frame(Feature=row.names(imp), Importance=imp[,1])

p <- ggplot(featureImportance, aes(x=reorder(Feature, Importance), y=Importance)) +
     geom_bar(stat="identity", fill="#53cfff") +
     coord_flip() +
     theme_light(base_size=20) +
     xlab("") +
     ylab("Importance") +
     ggtitle("Random Forest Feature Importance\n") +
     theme(plot.title=element_text(size=18))

p
```




```
levels(test$Section)<- levels(train$Section)
test$RF1pred=predict(fit.rf, test[params])
table(test$RF1pred)
```

```
##
##   0   1
## 271 147
```

```
ToTest<-test[c('PassengerId','RF1pred')]
colnames(ToTest) <- c('PassengerId','Survived')
write.csv(ToTest,'./RF1.csv',row.names = F)
```

Random Forest Model With Relatives Survival as variables

The below model uses Passenger class, Sex, Age, Fare, Family size, Section, Embarked, Title, and the count of know deaths and survival among same Ticket members. I chose the parameters to include based on the results from both the 1/10th of the data I set aside for testing as well as the accuracy on the testing set. The model tested at 82.3 % according to Kaggle. Below are in-depth reports as to the performance of the random forest and a bar chart of variable importance.

```
str(train)
```

```
## 'data.frame':    891 obs. of  26 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          : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name            : Factor w/ 891 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520 629 4
##  $ Sex             : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
```

```
##  $ Age             : num  22 38 26 35 35 ...
##  $ 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           : Factor w/ 148 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 1 1 ...
##  $ Embarked        : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
##  $ Title           : Factor w/ 6 levels "Ftitle","Job",..: 5 6 4 6 5 5 5 3 6 6 ...
##  $ Section         : Factor w/ 9 levels "","A","B","C",..: 1 4 1 4 1 1 6 1 1 1 ...
##  $ NumRms          : int  0 1 0 1 0 0 1 0 0 0 ...
##  $ RNum            : chr  NA "85" NA "123" ...
##  $ FSize           : int  1 1 0 1 0 0 0 4 2 1 ...
##  $ TicketCl        : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ FareCl          : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ AgeNas          : logi  FALSE FALSE FALSE FALSE FALSE TRUE ...
##  $ LName           : chr  "Braund" "Cumings" "Heikkinen" "Futrelle" ...
##  $ FamDiedCat      : chr  "Unknown" "Unknown" "Unknown" "Unknown" ...
##  $ FamDiedCont     : num  0 0 0 1 0 0 0 3 0 1 ...
##  $ FamSurvivedCont : num  0 0 0 0 0 0 0 0 2 0 ...
##  $ TickDiedCont    : num  0 0 0 1 0 0 0 3 0 1 ...
##  $ TickSurvivedCont: num  0 0 0 0 0 0 0 0 2 0 ...
```

```
test$Embarked <- as.character(test$Embarked)
test$Embarked <- factor(test$Embarked,levels=levels(train$Embarked))
train$Survived <- as.factor(train$Survived)
train$Title <- as.factor(train$Title)
test$Title <- factor(test$Title,levels=levels(train$Title))
train$Section <- as.factor(train$Section)
test$Section <- factor(test$Section,levels=levels(train$Section))
library(randomForest)
set.seed(1234)
train1ind=sample(nrow(train),floor(nrow(train)/10))
trainTest <- train[train1ind,]
train1 <- train[-train1ind,]
params <- c("Pclass","Sex","Age","Fare","FSize","Section","Embarked","Title","TickDiedCont","TickSurvive
fit.rf= randomForest(train1[params], as.factor(train1$Survived),xtest = trainTest[params],trainTest$Surv
fit.rf
```

```
##
## Call:
##  randomForest(x = train1[params], y = as.factor(train1$Survived),      xtest = trainTest[params], yt
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 16.21%
## Confusion matrix:
##     0   1 class.error
## 0 442  45  0.09240246
## 1  85 230  0.26984127
##                 Test set error rate: 10.11%
## Confusion matrix:
##    0  1 class.error
## 0 61  1  0.01612903
## 1  8 19  0.29629630
```

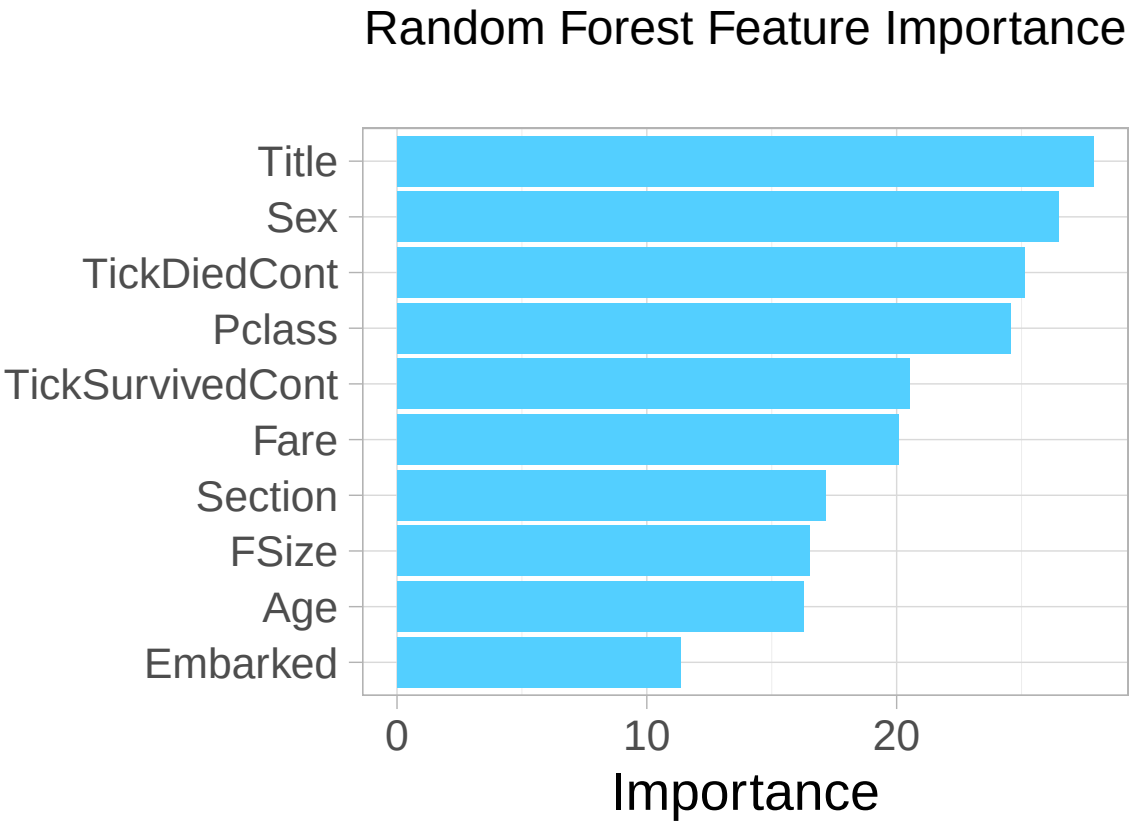
```
sum(abs(as.numeric(fit.rf$test$predicted) - as.numeric(trainTest$Survived)))/nrow(trainTest)
```

```
## [1] 0.1011236
```

```
round(importance(fit.rf), 2)
```

```
##                      0     1 MeanDecreaseAccuracy MeanDecreaseGini
## Pclass           16.27 18.42                24.54            19.98
## Sex              22.78 23.31                26.47            49.93
## Age               9.90 11.30                16.27            30.35
## Fare             10.56 16.64                20.09            33.58
## FSize            15.08  6.14                16.49            14.62
## Section          14.64  6.28                17.16            17.84
## Embarked          6.60  9.22                11.36             6.76
## Title            23.71 23.39                27.89            64.27
## TickDiedCont     21.36 18.36                25.12            16.23
## TickSurvivedCont 14.48 16.66                20.52            23.36
```

```
imp <- importance(fit.rf, type=1)
featureImportance <- data.frame(Feature=row.names(imp), Importance=imp[,1])

p <- ggplot(featureImportance, aes(x=reorder(Feature, Importance), y=Importance)) +
     geom_bar(stat="identity", fill="#53cfff") +
     coord_flip() +
     theme_light(base_size=20) +
     xlab("") +
     ylab("Importance") +
     ggtitle("Random Forest Feature Importance\n") +
     theme(plot.title=element_text(size=18))

p
```




```
trainpred=predict(fit.rf, train[params])
levels(test$Section)<- levels(train$Section)
test$RF1pred=predict(fit.rf, test[params])
table(test$RF1pred)
```

```
##
##   0   1
## 272 146
```

```
ToTest<-test[c('PassengerId','RF1pred')]
colnames(ToTest) <- c('PassengerId','Survived')
write.csv(ToTest,'./RF1.csv',row.names = F)
```