Building a Random Forest Model to Predict Titanic Survivorship

Joelle Strom

5/3/2021

Introduction

The Titanic challenge is one of the more popular Kaggle competitions. In this classic machine learning project, participants attempt to develop a model that most accurately predicts which passengers survived the sinking of the ship Titanic. In this tragic event, 1502 of 2224 passengers perished due to lack of sufficient safety features, such as an adequate number of lifeboats to support all passengers and crew. The patterns of survival were not random; some passengers were more likely to escape than others, especially women, children, and the wealthy.

This report examines the Titanic data set, teases out some pertinent information from descriptive variables, and combines these new features with other predictors to create a predictive model. This model is used to predict survival status of passengers within the supplied test data set, and these predictions were submitted to the Kaggle competition.

The Data

The training data set contains 891 observations and the test data set contains 418 observations. Variables include a passenger ID and 10 possible predictors: three categorical variables, one ordinal; four numerical variables, two continuous, two discrete; and three character variables that are varied enough to resemble descriptions more than categorical information. The response variable of interest is the survival status, a binary response where 0 = did not survive and 1 = survived. This information is not included in the test data set. A summary is shown below.

```r
#Read in data
train <- read.csv("D:/Documents/Applied Stats MS/Spring 2021/STAT 488/train.csv")
test <- read.csv("D:/Documents/Applied Stats MS/Spring 2021/STAT 488/test.csv")
test$Survived <- rep(NA, dim(test)[1])
total <- rbind(train,test)

#Create factors
total$Survived <- factor(total$Survived)
total$Sex <- factor(total$Sex)
total$Embarked <- factor(total$Embarked)

summary(total[!is.na(total$Survived),])
```

```
##   PassengerId    Survived     Pclass          Name               Sex     
##  Min.   :  1.0   0:549    Min.   :1.000   Length:891         female:314  
##  1st Qu.:223.5   1:342    1st Qu.:2.000   Class :character   male  :577  
##  Median :446.0            Median :3.000   Mode  :character               
##  Mean   :446.0            Mean   :2.309                                  
##  3rd Qu.:668.5            3rd Qu.:3.000                                  
##  Max.   :891.0            Max.   :3.000                                  
##                                                                          
##      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          :  2   
##  1st Qu.:  7.91   Class :character   C:168   
##  Median : 14.45   Mode  :character   Q: 77   
##  Mean   : 32.20                      S:644   
##  3rd Qu.: 31.00                              
##  Max.   :512.33                              
## 
```

Note the missing values in Age. In addition to the missing data here, there are also missing data values for variables Cabin and Fare, although the latter only has missing values in the test data set. The missing values for Age and Fare will be imputed as described in the following methods section. Missing Cabin data is too numerous to reliably impute, so these instances will be coded as a new “NA” category and carried over to any engineered features.

To begin exploring the data, some visualizations of survival status as it varies by several predictors are shown below. Survival status appears to vary somewhat according to the embarkment location, but the more drastic difference appears when passengers are grouped according to sex. Females had a much higher chance of survival compared to males (which follows the often-referenced “women and children first” strategy of evacuating the boat).

```
set.seed(2345)
library(mice)
```

```
## Warning: package 'mice' was built under R version 4.0.5
```

```
## 
## Attaching package: 'mice'
```

```
## The following object is masked from 'package:stats':
## 
##     filter
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
totalmice <- total[,c(-1,-2,-4,-9)]
miceout <- mice(totalmice, m=5, method="rf")
```

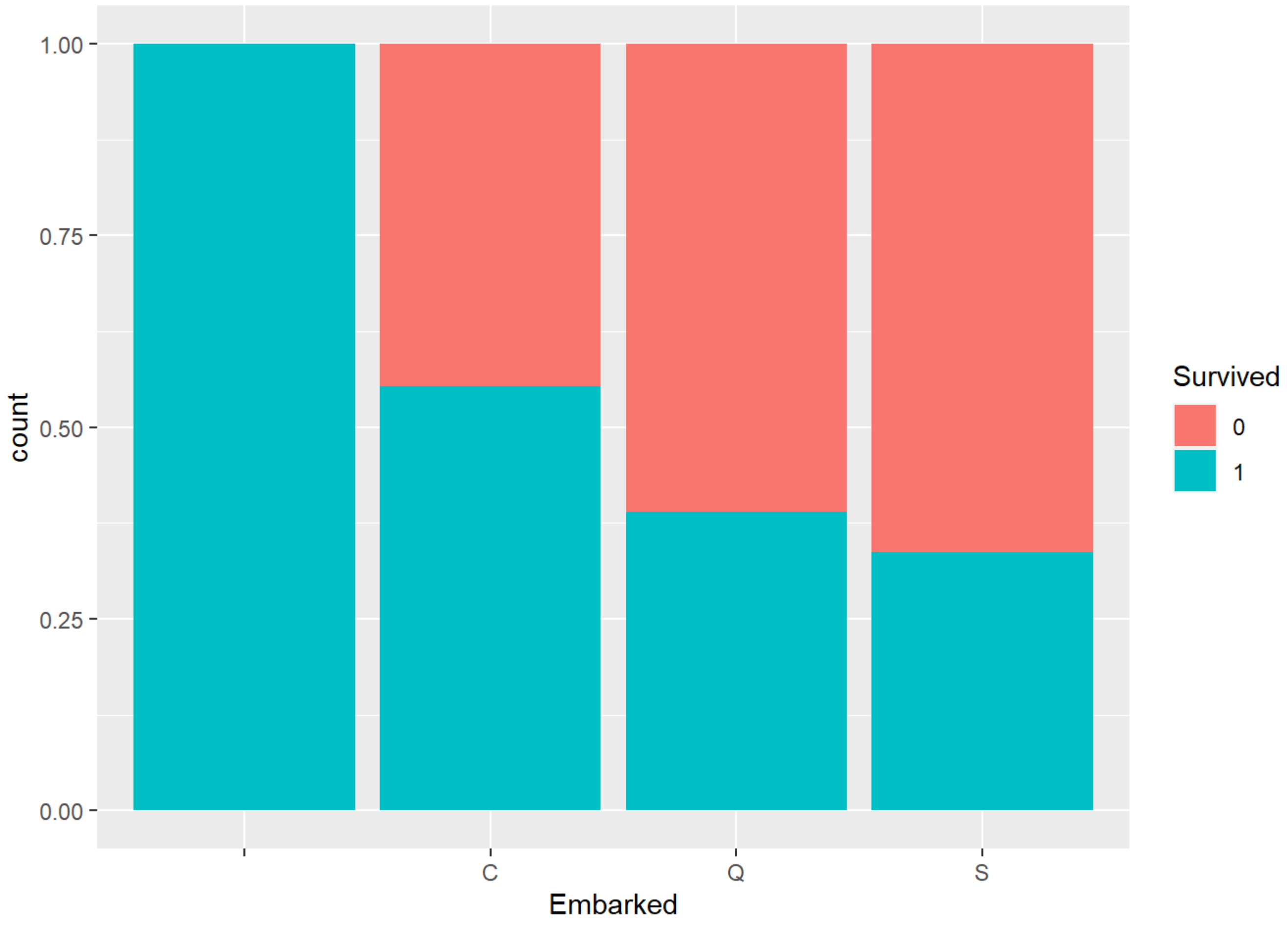
```
##
##  iter imp variable
##   1   1  Age  Fare
##   1   2  Age  Fare
##   1   3  Age  Fare
##   1   4  Age  Fare
##   1   5  Age  Fare
##   2   1  Age  Fare
##   2   2  Age  Fare
##   2   3  Age  Fare
##   2   4  Age  Fare
##   2   5  Age  Fare
##   3   1  Age  Fare
##   3   2  Age  Fare
##   3 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

```
## Warning: Number of logged events: 1
```

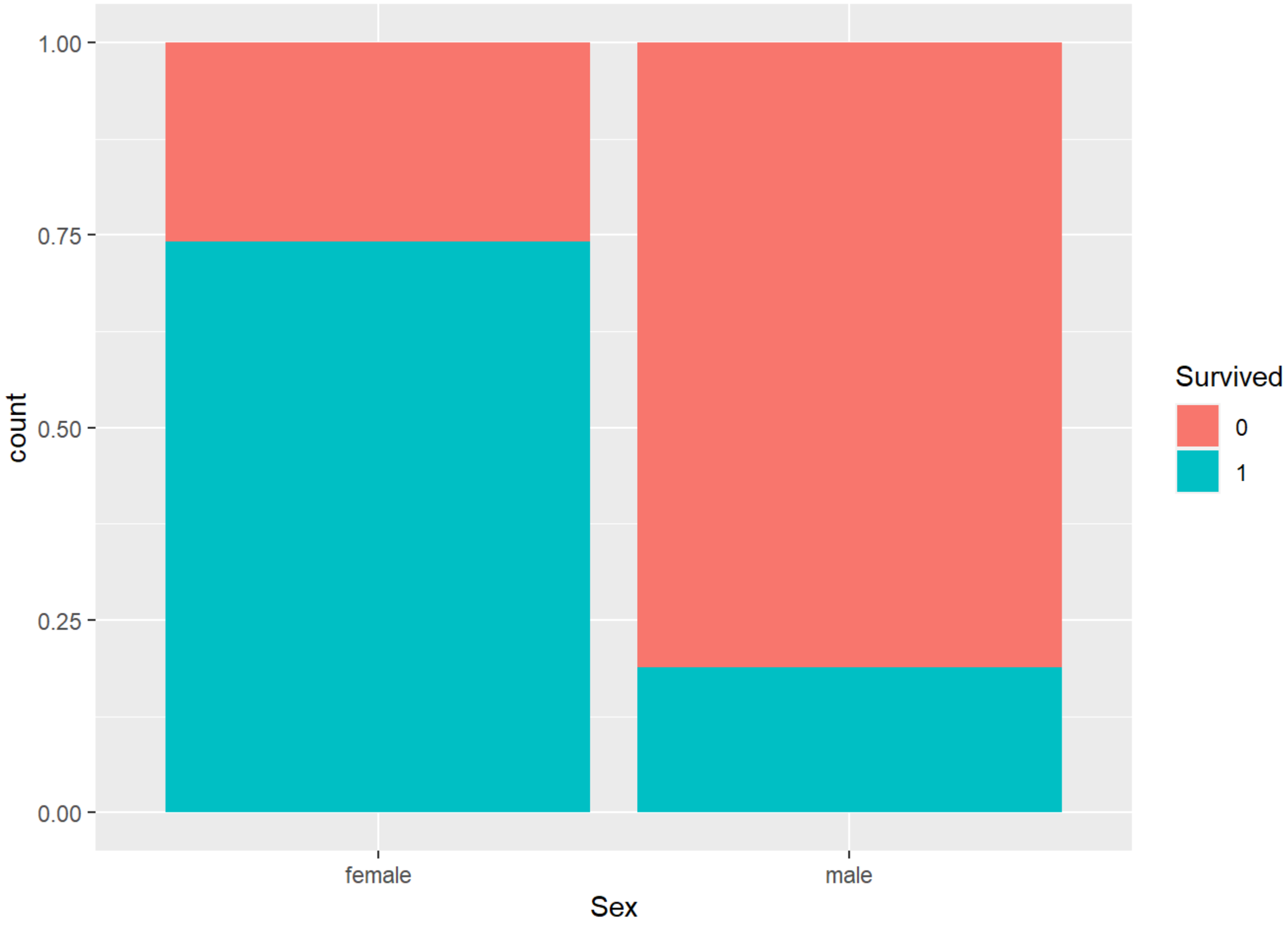
```
totalmice <- complete(miceout,1)
total$Age <- totalmice$Age
total$Fare <- totalmice$Fare

#Split into training and test data sets
train <- total[!is.na(total$Survived),]
test <- total[is.na(total$Survived),]

#Comprehensive bivariate visualization
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.4
```

```
ggplot(aes(Embarked),data=train) + geom_bar(aes(fill=Survived),position="fill")
```

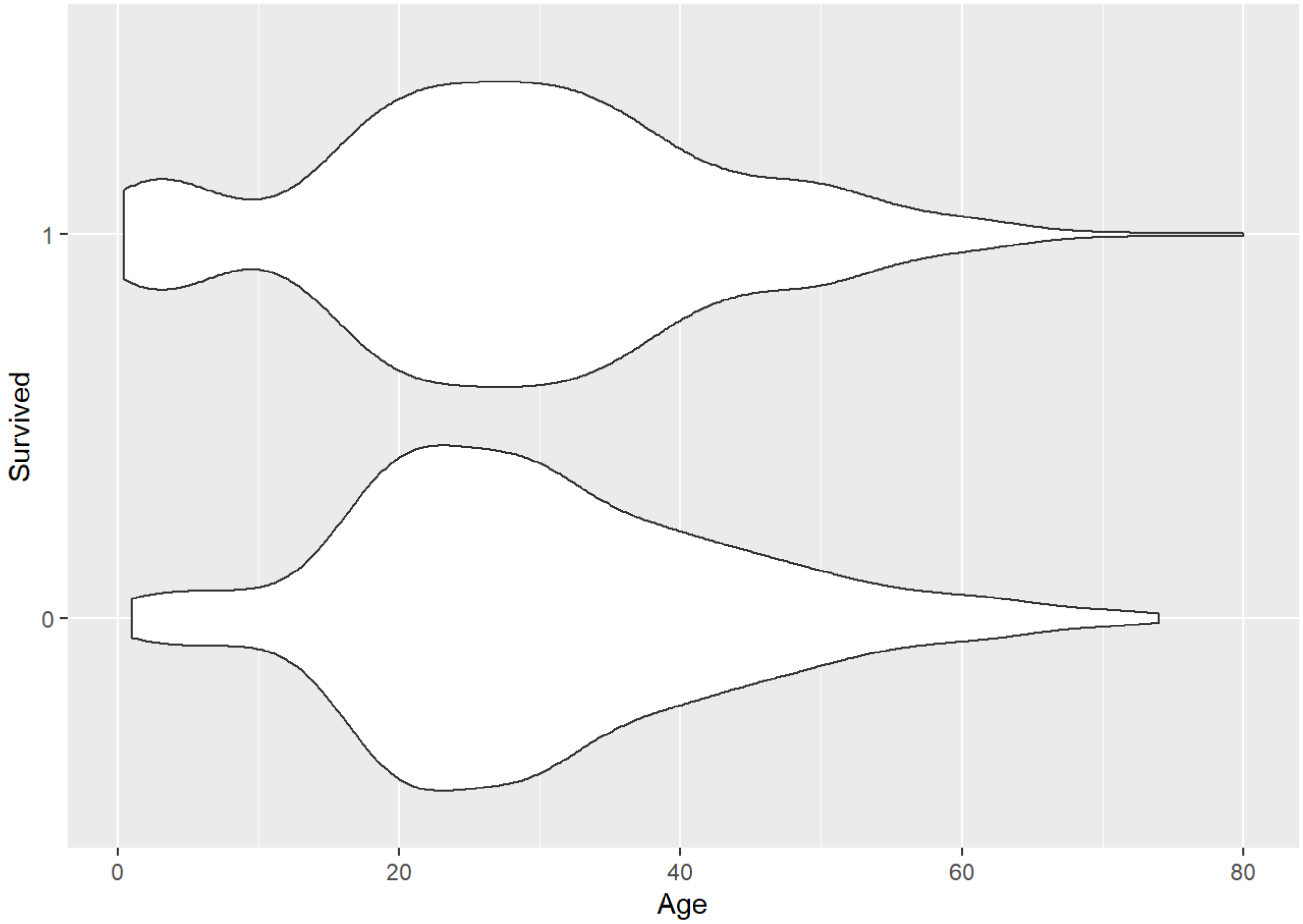


```
ggplot(aes(Sex),data=train) + geom_bar(aes(fill=Survived),position="fill")
```



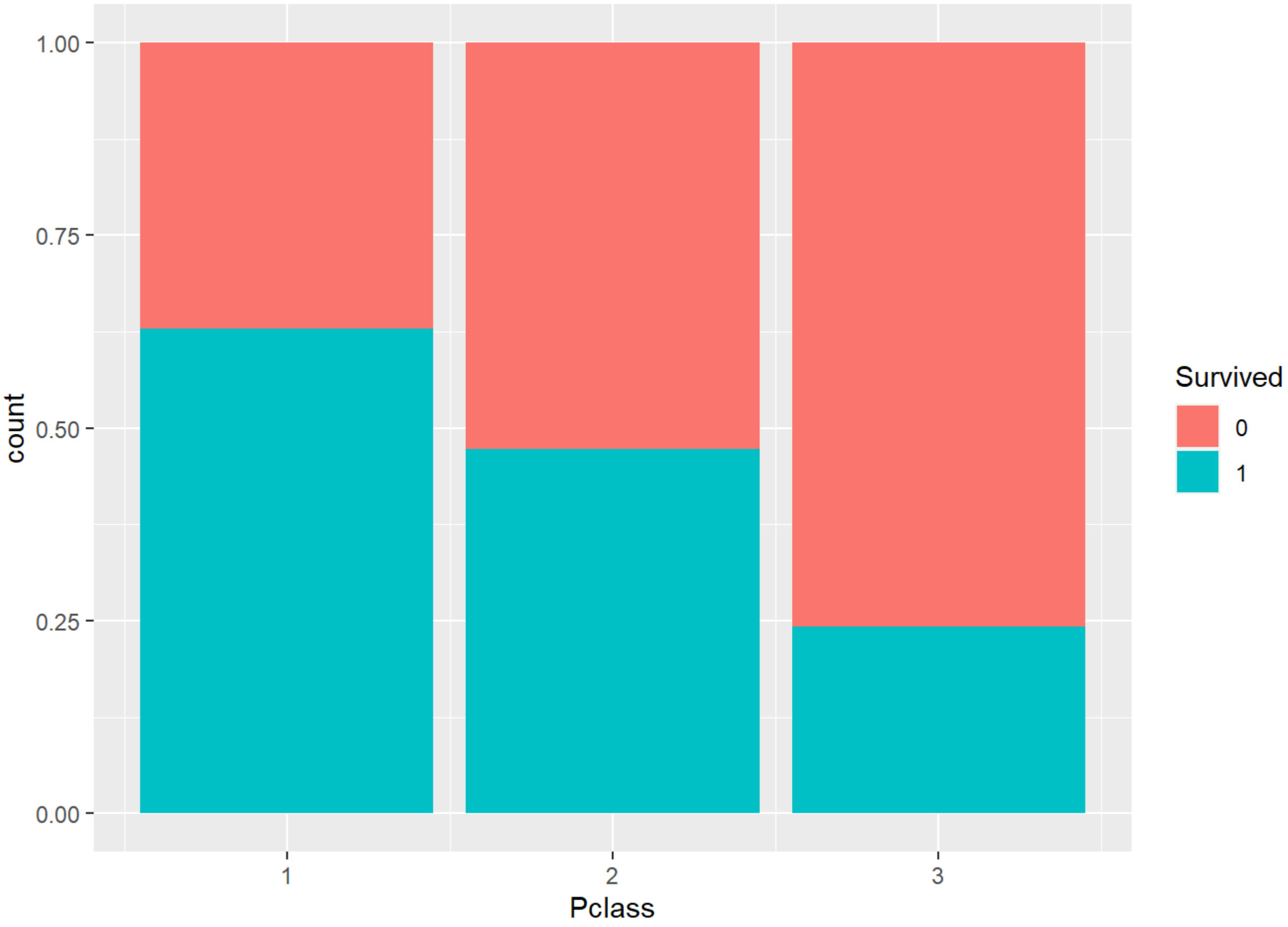
Evidence of this mentality is also apparent in the following violin plot comparing age to survival status. Passengers below 10 years of age seem to have been more likely to survive. This distinction is not as apparent as expected, however. Data was imputed (see above) before constructing this graph. The imputation methods will be discussed in more depth later.

```
ggplot(aes(Age,Survived),data=train) + geom_violin()
```

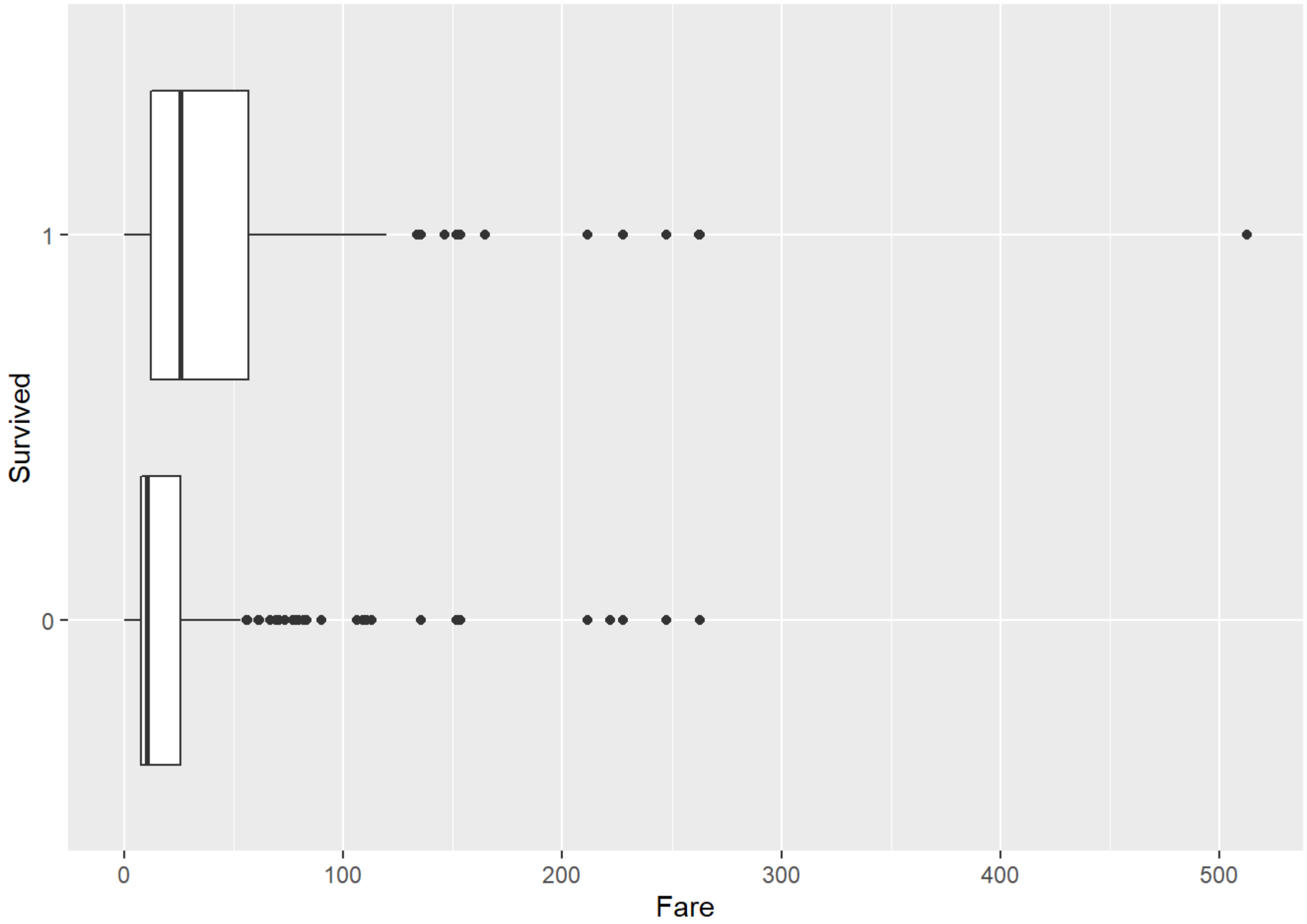


Finally, we look at the relationship between wealth or status and survival. The plots below display evidence that passengers who paid higher fare and were of higher class were more likely to survive.

```
ggplot(aes(Pclass),data=train) + geom_bar(aes(fill=Survived),position="fill")
```

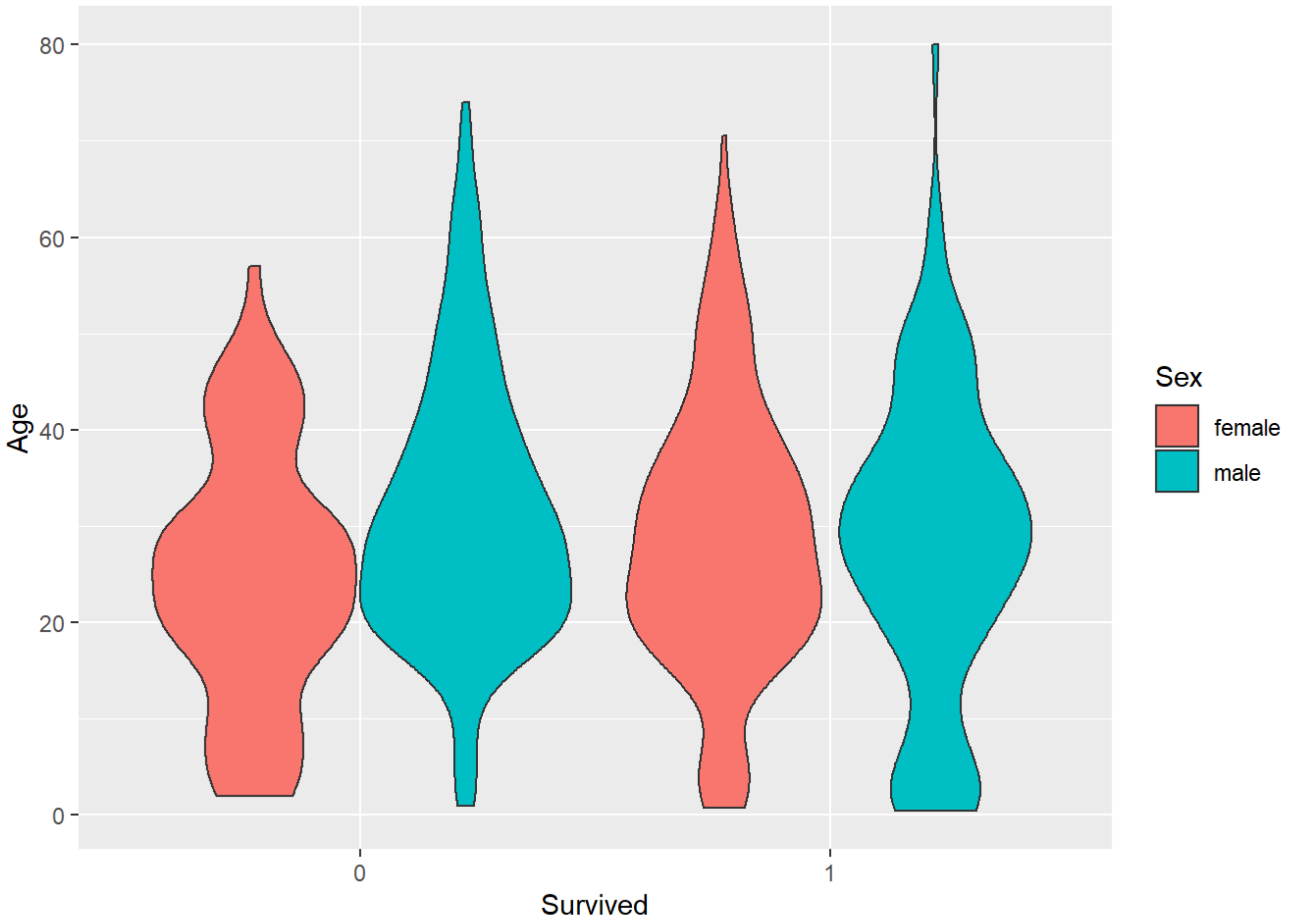


```
ggplot(aes(Fare,Survived),data=train) + geom_boxplot()
```



There is also likely some interaction between Age and Sex, as shown in the grouped violin plot below. For example, males under the age of 10 were more likely to survive, but not females in this age group. This interaction will be revisited when discussing derived variables.

```
ggplot(aes(fill=Sex,y=Age,x=Survived), data=train) + geom_violin()
```



Methods

Before building the model, a number of derived variables were created and missing data was imputed. Data imputation was performed by Multiple Imputation by Chained Equations (MICE), with 5 imputations and 5 iterations, with a random forest imputation method. The data set was completed with the first imputation for Age and Fare; this was performed with the test and training data sets combined, with the survival variable and descriptors masked.

Some variables were derived from numerical variables. For example, the SibSp and Parch variables (one indicating the number of siblings and/or spouses aboard, the other the number of parents and/or children) were added together to create the Family variable, an attempt to capture the full family size of each passenger. Categorical variables were created from Age and Fare, with breaks set at the 25%, 50%, and 75% quantiles.

```
#Assess individual contribution of Parch and SibSp and whether they can be combined
prop.table(table(total$Survived,total$Parch),2) #Add 1 or 2 for row/col perc
```

```
## 
##             0         1         2         3         4         5         6 9
##   0 0.6563422 0.4491525 0.5000000 0.4000000 1.0000000 0.8000000 1.0000000
##   1 0.3436578 0.5508475 0.5000000 0.6000000 0.0000000 0.2000000 0.0000000
```

```
prop.table(table(total$Survived,total$SibSp),2)
```

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

```
total$Family <- total$Parch + total$SibSp
prop.table(table(total$Survived,total$Family),2)
```

```
## 
##             0         1         2         3         4         5         6
##   0 0.6964618 0.4472050 0.4215686 0.2758621 0.8000000 0.8636364 0.6666667
##   1 0.3035382 0.5527950 0.5784314 0.7241379 0.2000000 0.1363636 0.3333333
## 
##             7        10
##   0 1.0000000 1.0000000
##   1 0.0000000 0.0000000
```

```
#Create categorical variables for continuous variables
quanta <- quantile(total$Age, c(0,0.25,0.5,0.75,1))
quanta[1] <- 0
quantf <- quantile(total$Fare, c(0,0.25,0.5,0.75,1))
quantf[1] <- -.0001
total$Agecat <- cut(total$Age, breaks=quanta)
total$Farecat <- cut(total$Fare, breaks=quantf)
```

Variables derived from the descriptors were more complex. The passenger’s title was extracted from the Name variable, which resulted in numerous different titles. The most common titles were “Mr.”, “Mrs.”, “Miss”, and “Master”. Less common titles, such as “Mlle” and “Mme” were assumed to be analogous to “Miss” and “Mrs.” respectively and were combined with those categories accordingly. The remaining rare titles were all indicators of higher class passengers, and thus were combined into a single “Other” category.

```
library(stringr)
```

```
## Warning: package 'stringr' was built under R version 4.0.5
```

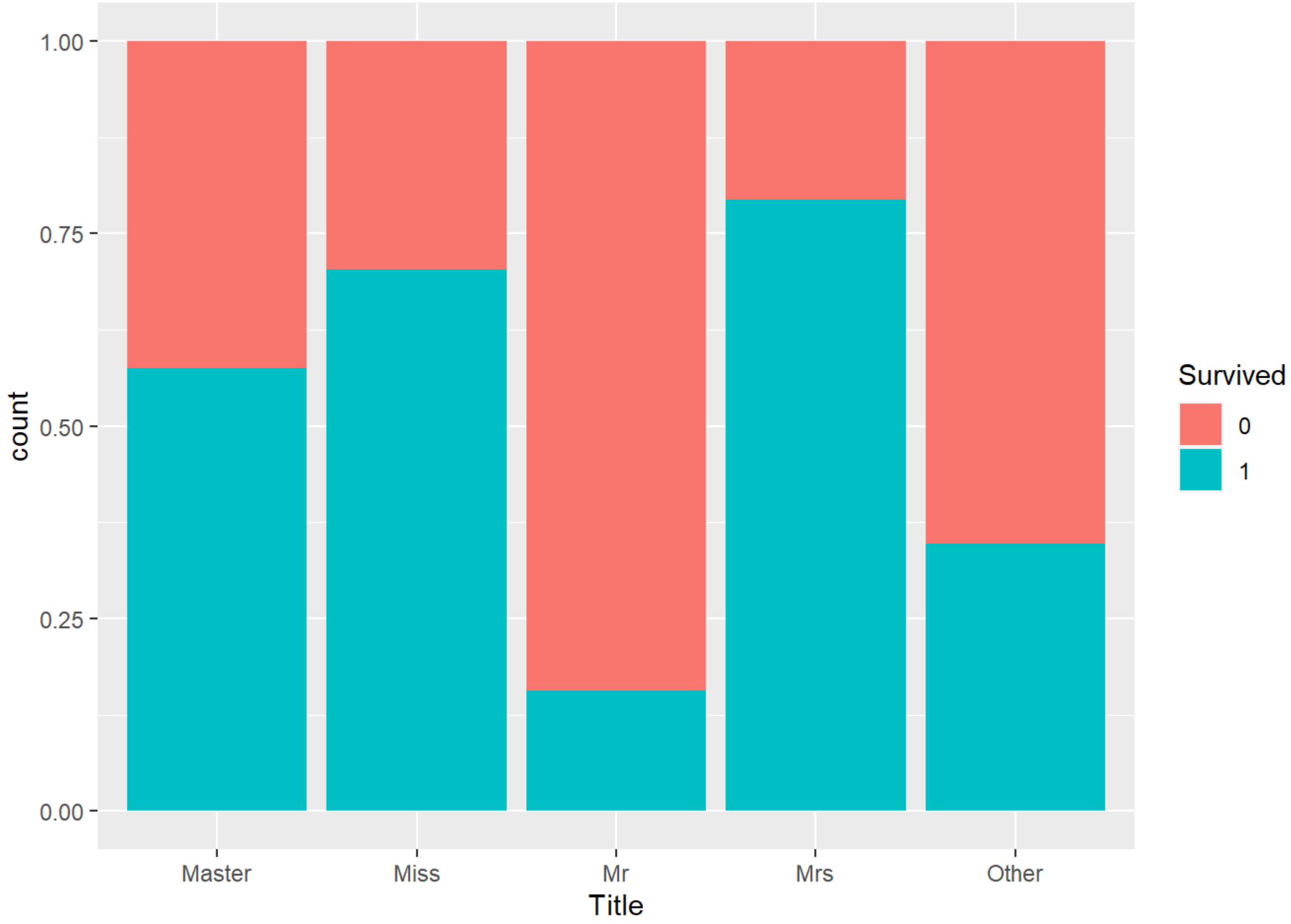
```
#Extract titles from Name variable
total$Title <- str_extract(total$Name, ",(.*?)\\.")
total$Title <- substring(total$Title, 3, nchar(total$Title)-1)
table(total$Title)
```

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

```
#Consolidate some categories; the unusual titles will be combined into "Other" because they all
 denote a higher class
total$Title[which(total$Title=="Mlle" | total$Title=="Ms")] <-  "Miss"
total$Title[which(total$Title=="Mme")] <- "Mrs"
total$Title[which(total$Title!="Master" &
                    total$Title!="Miss" &
                    total$Title!="Mr" &
                     total$Title!="Mrs")] <- "Other"
table(total$Title)
```

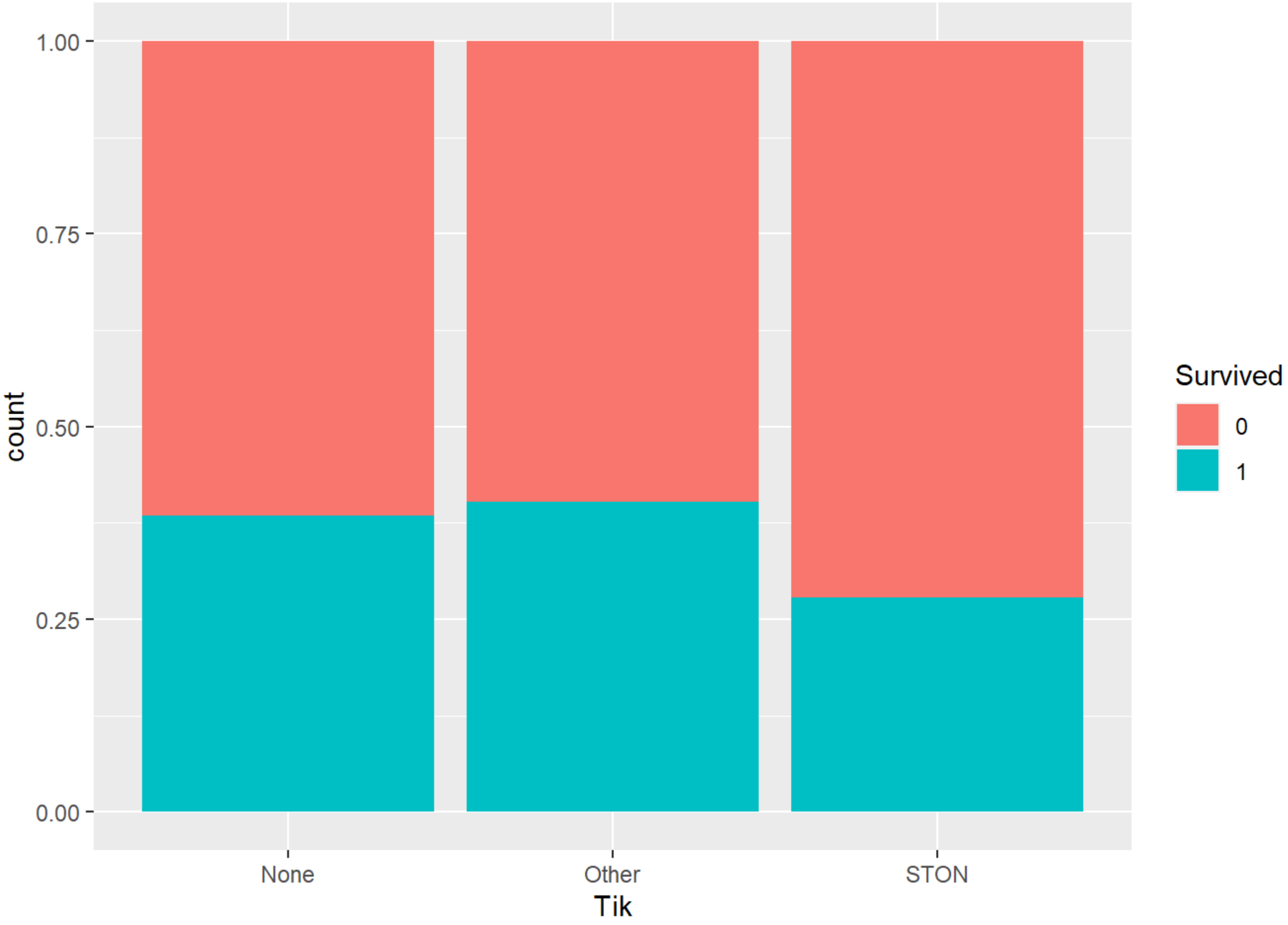
```
##
## Master   Miss     Mr    Mrs  Other
##     61    264    757    198     29
```

```
ggplot(aes(Title), data=total[!is.na(total$Survived),]) + geom_bar(aes(fill=Survived),position=
"fill")
```



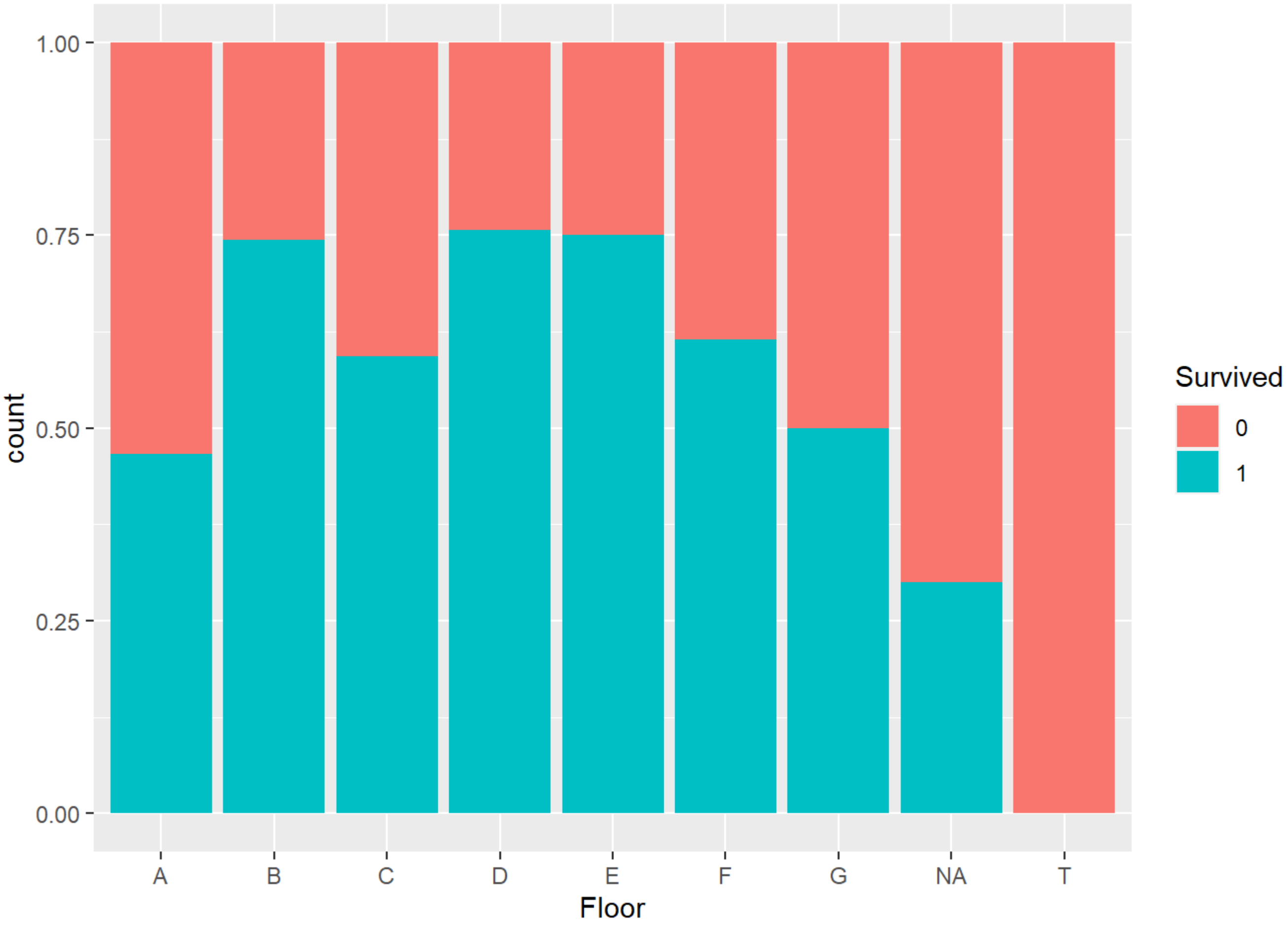
Passengers with tickets with distinctive patterns “STON” and “SOTON” were identified, and tickets with any other letters were identified separately. This information was combined to form a new variable with categories “STON”, “Other”, and “None”, indicating presence of “STON” or “SOTON”, other letter patterns, or no letters. This variable was denoted Tik.

```
#Identify tickets with common patterns "STON" and "SOTON"
total$STON <- str_detect(total$Ticket, "STON") | str_detect(total$Ticket, "SOTON")
#Identify all other tickets with letters
total$Letter <- str_detect(total$Ticket, "[A-Z]")
total$Letter <- (total$Letter & !total$STON)
#Create new variable with categories "STON/SOTON", "Other letters", and "None"
total$Tik <- rep("None", length(total$STON))
total$Tik[which(total$STON)] <- "STON"
total$Tik[which(total$Letter)] <- "Other"
ggplot(aes(Tik), data=total[!is.na(total$Survived),]) + geom_bar(aes(fill=Survived),position="fill")
```



The leading letter of the Cabin variable was extracted to create a new variable Floor that indicated the floor of the ship on which the passenger was staying. Passengers with no cabin listed were labeled “NA” in the Floor variable.

```
#Extract level from cabin variable, label missing data
total$Floor <- str_extract(total$Cabin,"[^ ]")
total$Floor[is.na(total$Floor)] <- "NA"
ggplot(aes(Floor), data=total[!is.na(total$Survived),]) + geom_bar(aes(fill=Survived),position="fill")
```



Summaries of both training and test data sets after the creation of derived variables and data imputation are shown below.

```
#Split into training and test data sets
train <- total[!is.na(total$Survived),]
test <- total[is.na(total$Survived),]

summary(train)
```

```
##   PassengerId    Survived     Pclass          Name               Sex
##  Min.   :  1.0   0:549    Min.   :1.000   Length:891         female:314
##  1st Qu.:223.5   1:342    1st Qu.:2.000   Class :character   male  :577
##  Median :446.0            Median :3.000   Mode  :character
##  Mean   :446.0            Mean   :2.309
##  3rd Qu.:668.5            3rd Qu.:3.000
##  Max.   :891.0            Max.   :3.000
##       Age            SibSp           Parch           Ticket
##  Min.   : 0.42   Min.   :0.000   Min.   :0.0000   Length:891
##  1st Qu.:21.00   1st Qu.:0.000   1st Qu.:0.0000   Class :character
##  Median :28.00   Median :0.000   Median :0.0000   Mode  :character
##  Mean   :29.68   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
##       Fare           Cabin           Embarked     Family           Agecat
##  Min.   :  0.00   Length:891          :  2    Min.   : 0.0000   (0,21] :257
##  1st Qu.:  7.91   Class :character   C:168    1st Qu.: 0.0000   (21,28]:190
##  Median : 14.45   Mode  :character   Q: 77    Median : 0.0000   (28,38]:228
##  Mean   : 32.20                      S:644    Mean   : 0.9046   (38,80]:216
##  3rd Qu.: 31.00                               3rd Qu.: 1.0000
##  Max.   :512.33                               Max.   :10.0000
##           Farecat       Title              STON           Letter
##  (-0.0001,7.9]:223   Length:891         Mode :logical   Mode :logical
##  (7.9,14.5]   :224   Class :character   FALSE:855       FALSE:697
##  (14.5,31.3]  :229   Mode  :character   TRUE :36        TRUE :194
##  (31.3,512]   :215
##
##
##      Tik               Floor
##  Length:891         Length:891
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

```
summary(test)
```

```
##   PassengerId     Survived       Pclass          Name               Sex
##  Min.   : 892.0   0   :  0   Min.   :1.000   Length:418         female:152
##  1st Qu.: 996.2   1   :  0   1st Qu.:1.000   Class :character   male  :266
##  Median :1100.5   NA's:418   Median :3.000   Mode  :character
##  Mean   :1100.5              Mean   :2.266
##  3rd Qu.:1204.8              3rd Qu.:3.000
##  Max.   :1309.0              Max.   :3.000
##       Age            SibSp            Parch           Ticket
##  Min.   : 0.17   Min.   :0.0000   Min.   :0.0000   Length:418
##  1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.0000   Class :character
##  Median :27.50   Median :0.0000   Median :0.0000   Mode  :character
##  Mean   :30.39   Mean   :0.4474   Mean   :0.3923
##  3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :76.00   Max.   :8.0000   Max.   :9.0000
##       Fare            Cabin            Embarked     Family            Agecat
##  Min.   :  0.000   Length:418          :  0    Min.   : 0.0000   (0,21] :107
##  1st Qu.:  7.896   Class :character   C:102    1st Qu.: 0.0000   (21,28]:110
##  Median : 14.454   Mode  :character   Q: 46    Median : 0.0000   (28,38]: 92
##  Mean   : 35.559                      S:270    Mean   : 0.8397   (38,80]:109
##  3rd Qu.: 31.472                               3rd Qu.: 1.0000
##  Max.   :512.329                               Max.   :10.0000
##           Farecat       Title             STON         Letter
##  (-0.0001,7.9]:115   Length:418         Mode :logical   Mode :logical
##  (7.9,14.5]   : 96   Class :character   FALSE:404       FALSE:310
##  (14.5,31.3]  : 99   Mode  :character   TRUE :14        TRUE :108
##  (31.3,512]   :108
##
##
##      Tik               Floor
##  Length:418         Length:418
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

A k-fold cross-validation random forest algorithm was used to investigate the optimal number of variables. The results are shown below, and suggest that 8 predictors provides the most accurate results.

```
library(randomForest)
```

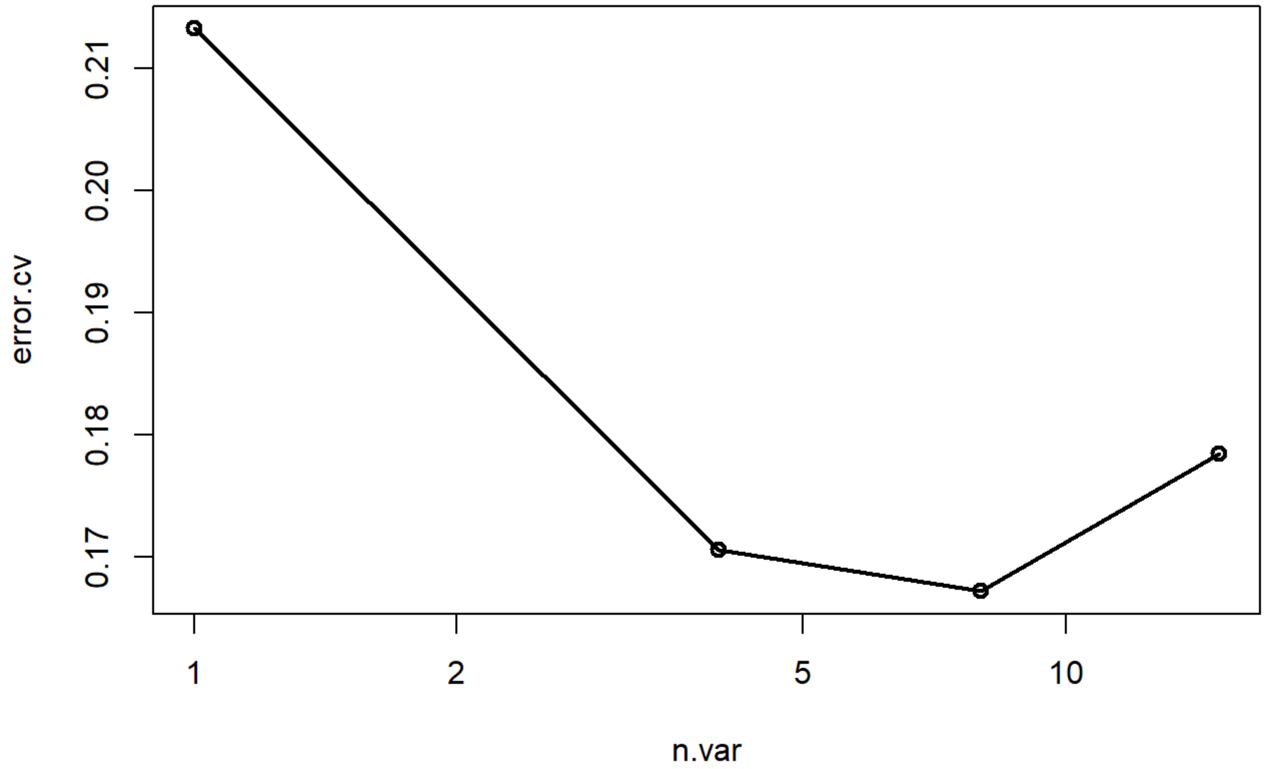
```
## Warning: package 'randomForest' was built under R version 4.0.4
```

```
## randomForest 4.6-14
```

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

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

```
cvout <- rfcv(trainx=train[,c(-1,-2,-4,-9,-11)],trainy=train[,2])
with(cvout, plot(n.var, error.cv, log="x", type="o", lwd=2))
```



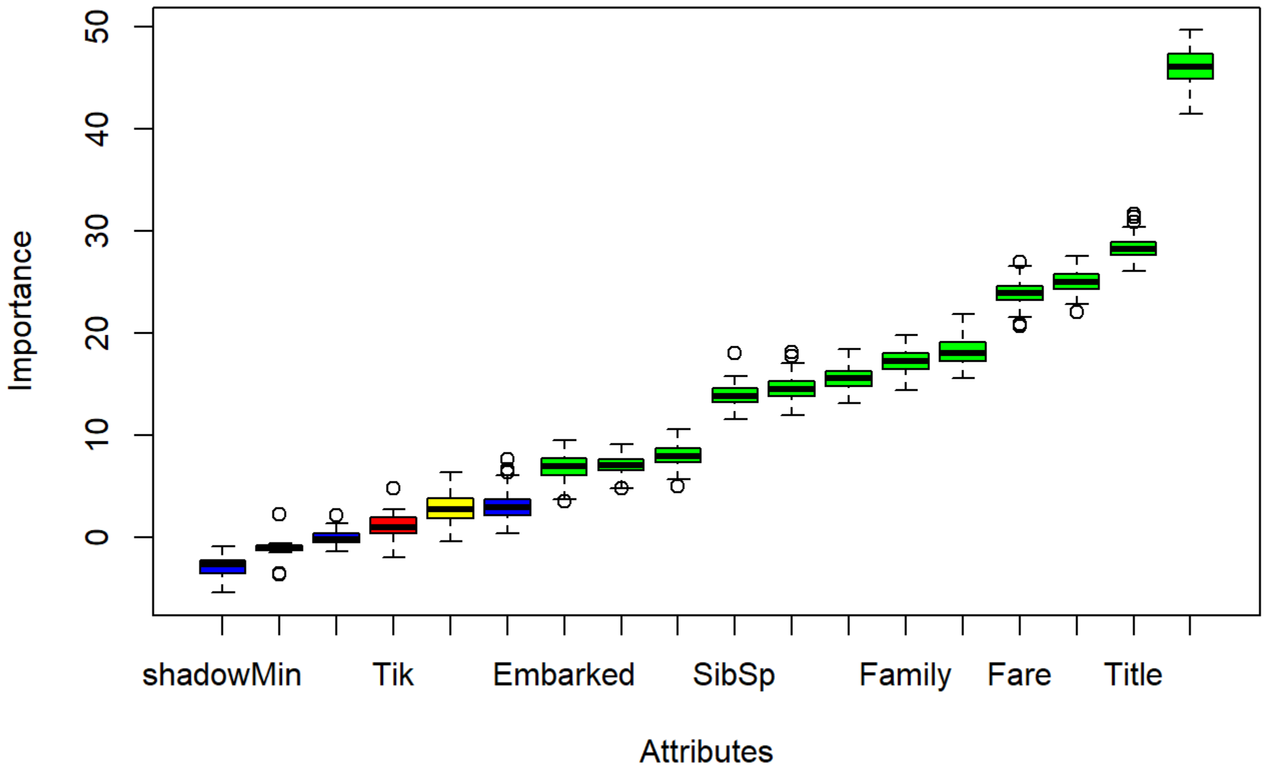
```
#About 8 variables seems to reduce error the most
```

The Boruta algorithm was also applied to visualize the importance of all variables. The only variables below the importance threshold of the random data were the derived variables from Ticket. The eight most important variables are Sex, Title, Pclass, Fare, Floor, Family, Farecat, and Age. The remaining variables above the importance threshold are SibSp, Agecat, Parch, and Embarked. Because it does not make sense to include two variables that are correlated (Fare and Farecat, Age and Agecat, Family and SibSp + Parch), the best variable of the correlated groups will be tested, and Embarked will be tested as an addition afterward.

```
library(Boruta)
```

```
## Warning: package 'Boruta' was built under R version 4.0.4
```

```
bor <- Boruta(Survived ~ Pclass + Sex + Age + Fare + SibSp + Parch + Family + Floor + Title + ST
ON + Letter + Agecat + Farecat + Embarked + Tik, data=train)
plot(bor)
```



```
#None of the derived variables from the Ticket variable are important, everything else is above
 the importance
#threshold of the random variable
```

With this information, a preliminary random forest model included the given variables Age, Fare, Pclass, Sex and derived variables Family, Title, and Floor. The training data set was randomly split in a 75-25 ratio into a new training and test data set for cross validation. A new model was then fit with SibSp and Parch to compare these separate variables with the combined Family. Age seems to be more important than its derived counterpart Agecat, but Fare vs. Farecat will be tested because both were in the top eight important variables found above. One model was fit with Fare, another with Farecat. Once the best of the above combinations was selected, Embarked was added. All of these models were fit with default parameters.

```
library(caTools)
```

```
## Warning: package 'caTools' was built under R version 4.0.4
```

```
set.seed(418)
split <- sample.split(train,SplitRatio=0.75)
train_cv <- train[split,]
test_cv <- train[!split,]

rf1 <- randomForest(Survived ~ Age + Sex + Title + Family + Fare + Pclass + Floor, data=train_cv, importance=TRUE)
pred1 <- predict(rf1, newdata=test_cv)
mean(test_cv[,2] == pred1)*100
```

```
## [1] 78.47534
```

```
rf2 <- randomForest(Survived ~ Age + Sex + Title + Parch + SibSp + Fare + Pclass + Floor, data=train_cv, importance=TRUE)
pred2 <- predict(rf2, newdata=test_cv)
mean(test_cv[,2] == pred2)*100
```

```
## [1] 79.82063
```

```
#Results of random forest indicate that SibSp and Parch separately performs better than the combined Family, retain SibSp + Parch

rf3 <- randomForest(Survived ~ Age + Fare + Pclass + Title + SibSp + Parch + Sex + Floor, data=train_cv, importance=TRUE)
pred3 <- predict(rf3, newdata=test_cv)
mean(test_cv[,2] == pred3)*100
```

```
## [1] 79.82063
```

```
rf4 <- randomForest(Survived ~ Age + Farecat + Pclass + Title + SibSp + Parch + Sex + Floor, data=train_cv, importance=TRUE)
pred4 <- predict(rf4, newdata=test_cv)
mean(test_cv[,2] == pred4)*100
```

```
## [1] 79.3722
```

```
rf5 <- randomForest(Survived ~ Age + Fare + Pclass + Title + SibSp + Parch + Sex + Floor + Embarked, data=train_cv, importance=TRUE)
pred5 <- predict(rf5, newdata=test_cv)
mean(test_cv[,2] == pred5)*100
```

```
## [1] 79.82063
```

```
#Continuous variable Fare performed better than categorical variable Farecat, retain Fare
#Addition of Embarked doesn't improve model accuracy, so it will be left out
#Formula now has 8 variables, the number that reduced error in the k-fold CV; this will be the f
inal formula
```

Tuning of hyperparameters was conducted two separate ways. A grid was created with values of mtry ranging from 1 to 5 and values of min.node.size ranging from 1 to 10 and these values were tested with the ranger function. The combination of parameters resulting in the lowest prediction error were chosen for a final ranger model.

```
#Move on to tuning parameter choice
library(ranger)
```

```
## Warning: package 'ranger' was built under R version 4.0.5
```

```
##
## Attaching package: 'ranger'
```

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

```
hyper_grid <- expand.grid(
  mtry       = 1:5,
  node_size  = 1:10,
  num.trees = 500,
  OOB_RMSE   = 0
)
for(i in 1:nrow(hyper_grid)) {
  # train model
  rf <- ranger(
    formula        = Survived ~ Age + Fare + Title + Pclass + SibSp + Parch + Sex + Floor,
    data           = train,
    num.trees      = hyper_grid$num.trees[i],
    mtry           = hyper_grid$mtry[i],
    min.node.size  = hyper_grid$node_size[i],
    importance = 'impurity')
  # add OOB error to grid
  hyper_grid$OOB_RMSE[i] <- sqrt(rf$prediction.error)
}
position <- which.min(hyper_grid$OOB_RMSE)
```

A second grid was created for the randomForest function, including mtry values 1 to 5, mincut values 2 to 8, and 10 mindev values from 0.001 to 0.5. The parameters resulting in the highest cross-validation accuracy were chosen for a final randomForest model.

```
#Try grid search with randomForest()
hyper_grid2 <- expand.grid(
  mtry    = 1:5,
  mincut  = 2:8,
  mindev  = seq(0.001,0.5,length.out=10),
  OOB_RMSE   = 0
)

for(i in 1:nrow(hyper_grid2)){
  rf <- randomForest(Survived ~ Age + Fare + Title + Pclass + SibSp + Parch + Sex + Floor, data=
train_cv, mtry=hyper_grid2$mtry[i],
                     control=tree.control(nobs=668,mincut=hyper_grid2$mincut[i],minsize=2*hyper_
grid2$mincut[i],mindev=hyper_grid2$mindev[i]))
  hyper_grid2$OOB_RMSE[i] <- mean(predict(rf,test_cv)==test_cv[,2])*100
}
position <- which.max(hyper_grid2$OOB_RMSE)
```

Results

The models were compared by calculating predictive accuracy on the cross-validation test data set. The initial model returned an accuracy of 78.48%. When the Family variable was swapped out for the separate SibSp and Parch variables, the accuracy increased to 79.82%. Importance plots show that SibSp is more important than Parch, but about as important as Family (relative to the other variables that stay constant).

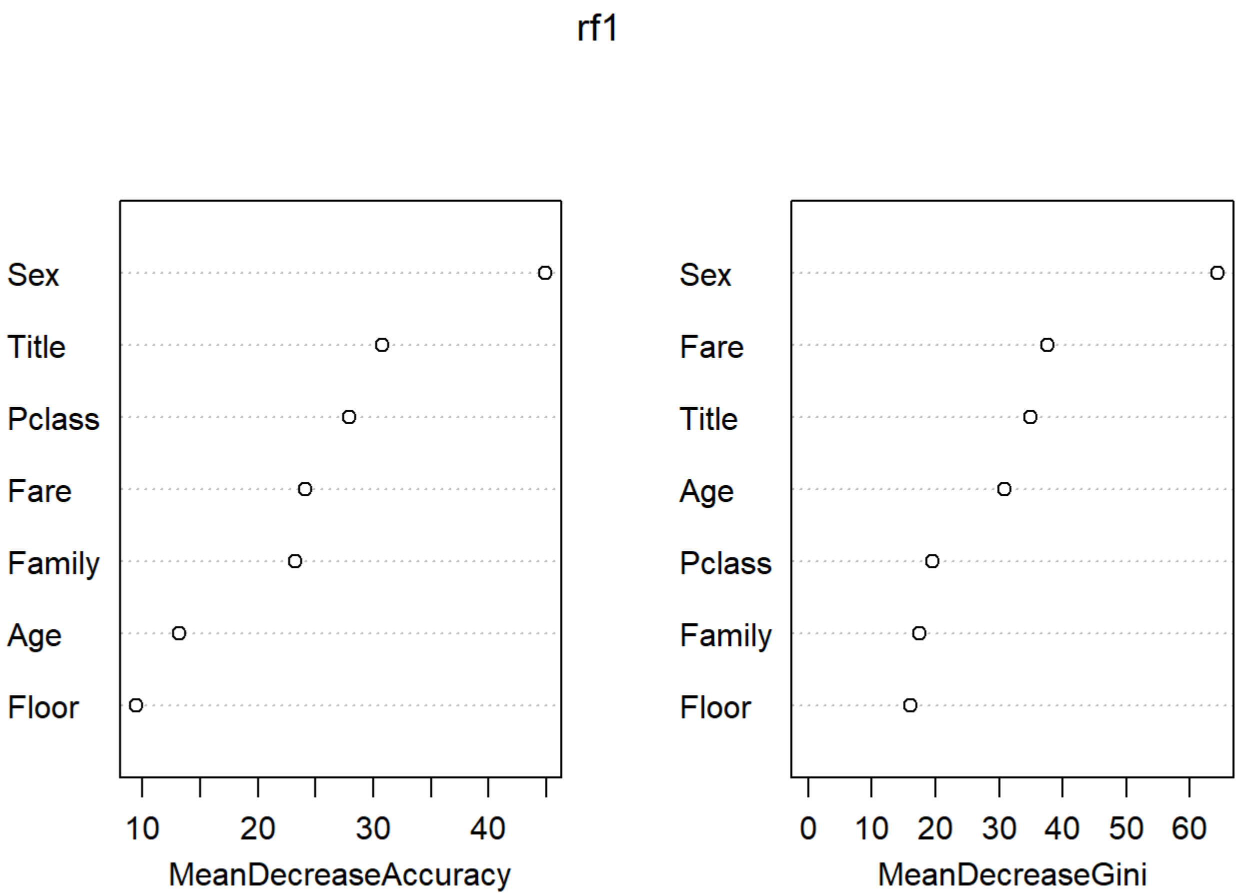
```
mean(test_cv[,2] == pred1)*100 #Accuracy of model with Family but not SibSp + Parch
```

```
## [1] 78.47534
```

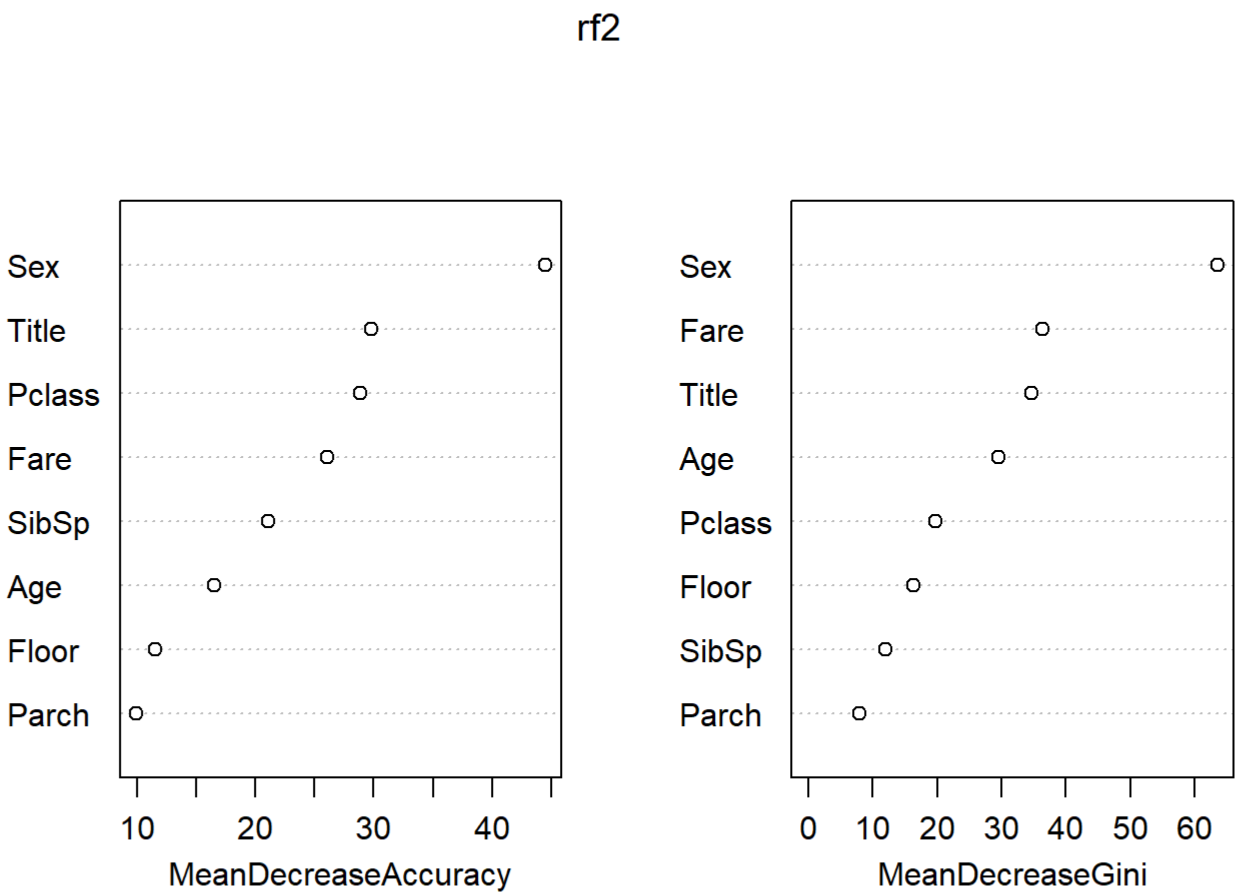
```
mean(test_cv[,2] == pred2)*100 #Accuracy of model with SibSp + Parch but not Family
```

```
## [1] 79.82063
```

```
varImpPlot(rf1) #Model with Family
```



```
varImpPlot(rf2) #Model with SibSp + Parch
```



The replacement of Fare with Farecat reduced accuracy from 79.82% to 79.37%. Although this is not a large reduction, Fare will remain in the model instead of Farecat. The addition of Embarked did not increase accuracy, and was shown to be less important than the other variables in the Boruta analysis, despite the results of the importance plot below. For these reasons, it will be left out of future models.

```
mean(test_cv[,2] == pred3)*100 #Accuracy of model with Fare
```

```
## [1] 79.82063
```

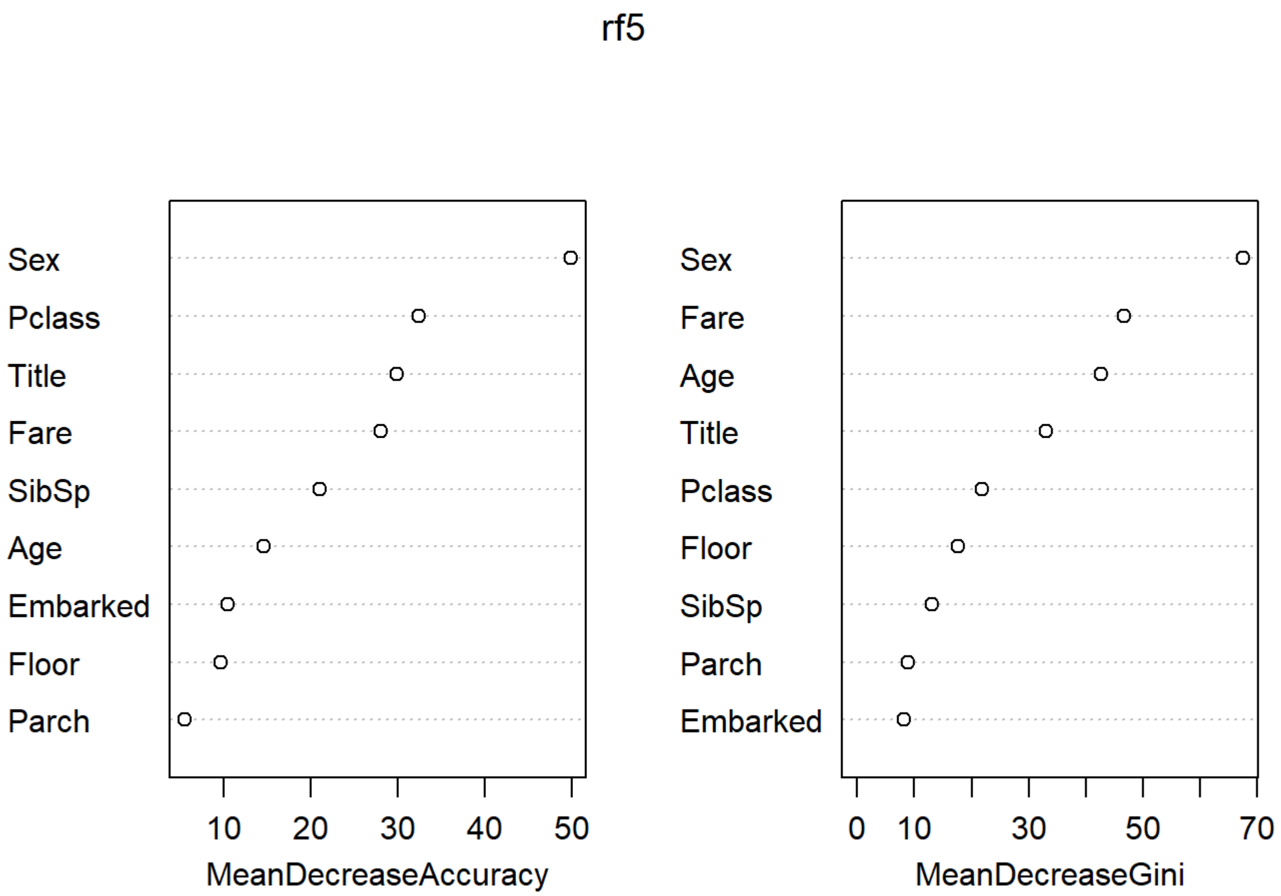
```
mean(test_cv[,2] == pred4)*100 #Accuracy of model with Farecat
```

```
## [1] 79.3722
```

```
mean(test_cv[,2] == pred5)*100 #Accuracy of model with the addition of Embarked
```

```
## [1] 79.82063
```

```
varImpPlot(rf5)
```



The grid search for hyperparameter tuning with the ranger function demonstrated that an mtry value of 5 and min.node.size of 10 reduced error, but the cross-validation accuracy of this model was 78.47%, lower than the default randomForest models that had been fit earlier.

```
head(hyper_grid[order(hyper_grid$OOB_RMSE),],5)
```

```
##    mtry node_size num.trees  OOB_RMSE
## 50    5        10       500 0.3935507
## 35    5         7       500 0.3963923
## 34    4         7       500 0.3978054
## 9     4         2       500 0.3992136
## 25    5         5       500 0.3992136
```

```
rfran <- ranger(formula=Survived ~ Age + Fare + Title + Pclass + SibSp + Parch + Sex + Floor, da
ta=train_cv, num.trees=1000,
             mtry = 5, min.node.size=10)
rfran
```

```
## Ranger result
##
## Call:
##  ranger(formula = Survived ~ Age + Fare + Title + Pclass + SibSp +      Parch + Sex + Floor,
data = train_cv, num.trees = 1000, mtry = 5,      min.node.size = 10)
##
## Type:                             Classification
## Number of trees:                  1000
## Sample size:                      668
## Number of independent variables:  8
## Mtry:                             5
## Target node size:                 10
## Variable importance mode:         none
## Splitrule:                        gini
## OOB prediction error:             16.02 %
```

```
pred <- predict(rfran,test_cv)
mean(test_cv[,2] == pred$predictions)*100 #This is lower than randomForest models with default p
arameters
```

```
## [1] 78.47534
```

The grid search for the randomForest model indicated that an mtry value of 4, a mincut value of 7, and a mindev value of 0.1 increased accuracy the most. When this model was fit again, the cross-validation error decreased to 78.92%, despite finding an error rate of over 80% for the model with these hyperparameters during the grid search. The maximum accuracy model during the grid search may have been an artifice of random variability in the random forest algorithm. Because this result is not replicable, the tuning parameters will be left to default values in the final model.

```
tail(hyper_grid2[order(hyper_grid2$OOB_RMSE),],5)
```

```
##     mtry mincut    mindev OOB_RMSE
## 286    1      3 0.4445556 80.26906
## 296    1      5 0.4445556 80.26906
## 337    2      6 0.5000000 80.26906
## 7      2      3 0.0010000 80.71749
## 99     4      7 0.1118889 80.71749
```

```
rf <- randomForest(Survived ~ Age + Fare + Title + Pclass + SibSp + Parch + Sex + Floor, data=tr
ain_cv, ntree=500,
                   mtry=4, control=tree.control(nobs=668, mincut=7, minsize=14, mindev=0.1))
rf
```

```
##
## Call:
##  randomForest(formula = Survived ~ Age + Fare + Title + Pclass +      SibSp + Parch + Sex + F
loor, data = train_cv, ntree = 500,      mtry = 4, control = tree.control(nobs = 668, mincut =
7,          minsize = 14, mindev = 0.1))
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 15.72%
## Confusion matrix:
##     0   1 class.error
## 0 389  39   0.0911215
## 1  66 174   0.2750000
```

```
pred <- predict(rf,newdata=test_cv)
mean(test_cv[,2] == pred)*100
```

```
## [1] 78.92377
```

The final model included predictors Age, Fare, Title, Pclass, Sex, Floor, SibSp, and Parch. The OOB error rate of the final model was 17.73%, and the actual submission score was 0.77511.

```
rf <- randomForest(Survived ~ Age + Fare + Title + Pclass + SibSp + Parch + Sex + Floor, data=tr
ain, ntree=1000)
rf
```

```
##
## Call:
##  randomForest(formula = Survived ~ Age + Fare + Title + Pclass +      SibSp + Parch + Sex + F
loor, data = train, ntree = 1000)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 17.73%
## Confusion matrix:
##     0   1 class.error
## 0 487  62   0.1129326
## 1  96 246   0.2807018
```

```
pred <- predict(rf,newdata=test)
test$Survived <- pred
write.csv(test[,c(1,2)],file="D:/Documents/Applied Stats MS/Spring 2021/STAT 488/Strom_titanicpr
ed.csv", row.names=FALSE)

#Accuracy score of final submission is 0.77511
```

Discussion

It was surprisingly difficult to find changes that would consistently improve the model accuracy. Without setting the seed to standardize the run, the same code would produce different results from one day of work to the next, sometimes drastic enough to change the decision about what the final model should be. Despite this variability, there were some variable selection steps that were obvious improvements. The continuous variables Age and Fare consistently outperformed their categorical counterparts. Conversely, derived variable Tik never made an improvement to the model. The grid searches returned different optimal hyperparameter values with every run, but the ranger model never outperformed the randomForest model.

One of the most interesting results was the recurring appearance of Title at the top of importance plots. This single piece of derived information is obviously an important predictor of survivorship. This might be because it is a simple way to capture many aspects of an individual's identity and relationship to others on the ship. As mentioned earlier, there appears to be some type of interaction between age and sex as it relates to chances of survival. The Title variable is a way to capture the intersection of age and sex for each passenger. Those with a title “Master” or “Miss” were among those who were prioritized based on age, but the chances of survival were not equal for both titles. The Title category with the highest chance of survival was “Mrs.” There are several possible reasons for this, one being that women were prioritized and “Mrs.” is the only category for adult women. Another could be that a married woman, as indicated by this title, was more likely to have children who she would have accompanied, or to have a spouse who might have helped her secure a spot. The “Other” category had a lower survival percentage than might be expected, given most of these titles indicated wealth. A future direction could be to try to tease apart other factors that might separate those in the other category who were more likely to survive from those who were less likely.

Other future directions could include digging deeper into the feature engineering aspect. Some passengers had multiple cabins listed; could the number of cabins listed be a good predictor of survival? Is there information buried in the Ticket variable that was ignored in this report? The strategy of splitting the training data set for cross-validation could be another topic of future study. Simply changing the seed in the code before randomly splitting the data set caused the cross-validation accuracy to be closer to the actual submission score. What changed in the composition of the cross-validation data sets that led to this change in results? What lessons could be learned from this observation that can be applied to future machine learning projects? It is likely that repeating everything from this project with k-fold cross-validation might reduce the variability in cross-validation error; that would be another potential aspect to test in the future. While the random forest algorithm is powerful, it would also be worthwhile to explore other predictive models.