Witlie Kaggle project DSCI 478

2025-02-19

Data Cleaning

LOAD DATA SET

```
# Read Titanic data set
train_df <- read.csv("train.csv", stringsAsFactors = FALSE)
test_df <- read.csv("test.csv", stringsAsFactors = FALSE)

head(train_df)
```

```
##   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 Parch
## 1                            Braund, Mr. Owen Harris   male  22     1     0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1     0
## 3                             Heikkinen, Miss. Laina female  26     0     0
## 4       Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1     0
## 5                           Allen, Mr. William Henry   male  35     0     0
## 6                                   Moran, Mr. James   male  NA     0     0
##             Ticket    Fare Cabin Embarked
## 1        A/5 21171  7.2500              S
## 2         PC 17599 71.2833   C85        C
## 3 STON/O2. 3101282  7.9250              S
## 4           113803 53.1000  C123        S
## 5           373450  8.0500              S
## 6           330877  8.4583              Q
```

```
head(test_df)
```

```
##   PassengerId Pclass                                          Name    Sex  Age
## 1         892      3                              Kelly, Mr. James   male 34.5
## 2         893      3                  Wilkes, Mrs. James (Ellen Needs) female 47.0
## 3         894      2                         Myles, Mr. Thomas Francis   male 62.0
## 4         895      3                                 Wirz, Mr. Albert   male 27.0
## 5         896      3 Hirvonen, Mrs. Alexander (Helga E Lindqvist) female 22.0
## 6         897      3                       Svensson, Mr. Johan Cervin   male 14.0
##   SibSp Parch  Ticket    Fare Cabin Embarked
## 1     0     0  330911  7.8292              Q
## 2     1     0  363272  7.0000              S
## 3     0     0  240276  9.6875              Q
## 4     0     0  315154  8.6625              S
## 5     1     1 3101298 12.2875              S
```

```
## 6     0     0    7538  9.2250              S
```

CHECK FOR MISSING VALUES

```
# Count missing values in each column
colSums(is.na(train_df))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0         177
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0           0           0
```

```
# Remove rows where Age is missing
train_df <- train_df[!is.na(train_df$Age), ]
```

DOUBLE CHECK TO MAKE SURE THERE ARE NO MORE MISSING VALUES AFTER REMOVING ROWS TO BE DONE CORRECTLY

```
# Count missing values in each column
colSums(is.na(train_df))
```

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

CONVERT CATEGORICAL VARIABLES TO NUMERIC

```
# Sex: Male -> 0; Females -> 1
train_df$Sex <- ifelse(train_df$Sex == "male", 0, 1)
test_df$Sex <- ifelse(test_df$Sex == "male", 0, 1)

# Embarked: "C" (Cherbourg) -> 0; "Q" (Queenstown) -> 1; "S" (Southampton) -> 2
train_df$Embarked <- ifelse(train_df$Embarked == "C", 0,
                            ifelse(train_df$Embarked == "Q", 1, 2))
test_df$Embarked <- ifelse(test_df$Embarked == "C", 0,
                           ifelse(test_df$Embarked == "Q", 1, 2))

# Fare: Round to two decimal places
train_df$Fare <- round(train_df$Fare, 2)
test_df$Fare <- round(test_df$Fare, 2)

# Check the first few rows to confirm changes
head(train_df[, c("PassengerId",
                  "Survived",
                  "Pclass",
                  "Sex",
                  "Age",
                  "SibSp",
                  "Parch",
                  "Fare",
                  "Embarked")])
```

```
##   PassengerId Survived Pclass Sex Age SibSp Parch  Fare Embarked
## 1           1        0      3   0  22     1     0  7.25        2
## 2           2        1      1   1  38     1     0 71.28        0
## 3           3        1      3   1  26     0     0  7.92        2
## 4           4        1      1   1  35     1     0 53.10        2
```

```
## 5           5        0      3   0  35     0     0  8.05        2
## 7           7        0      1   0  54     0     0 51.86        2
```

SAVE CLEAN DATASET

```
write.csv(train_df, "clean_train.csv", row.names = FALSE)
write.csv(test_df, "clean_test.csv", row.names = FALSE)

head(train_df)
```

```
##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      3
## 4           4        1      1
## 5           5        0      3
## 7           7        0      1
##                                                 Name Sex Age SibSp Parch
## 1                             Braund, Mr. Owen Harris   0  22     1     0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer)   1  38     1     0
## 3                              Heikkinen, Miss. Laina   1  26     0     0
## 4        Futrelle, Mrs. Jacques Heath (Lily May Peel)   1  35     1     0
## 5                            Allen, Mr. William Henry   0  35     0     0
## 7                             McCarthy, Mr. Timothy J   0  54     0     0
##             Ticket  Fare Cabin Embarked
## 1        A/5 21171  7.25              2
## 2         PC 17599 71.28   C85        0
## 3 STON/O2. 3101282  7.92              2
## 4           113803 53.10  C123        2
## 5           373450  8.05              2
## 7            17463 51.86   E46        2
```

LOADING CLEAN DATASET

```
# Load the cleaned dataset
train_df <- read.csv("clean_train.csv", stringsAsFactors = FALSE)
test_df <- read.csv("clean_test.csv", , stringsAsFactors = FALSE)

# Check the first few rows to verify correctness
head(train_df)
```

```
##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      3
## 4           4        1      1
## 5           5        0      3
## 6           7        0      1
##                                                 Name Sex Age SibSp Parch
## 1                             Braund, Mr. Owen Harris   0  22     1     0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer)   1  38     1     0
## 3                              Heikkinen, Miss. Laina   1  26     0     0
## 4        Futrelle, Mrs. Jacques Heath (Lily May Peel)   1  35     1     0
## 5                            Allen, Mr. William Henry   0  35     0     0
## 6                             McCarthy, Mr. Timothy J   0  54     0     0
##             Ticket  Fare Cabin Embarked
## 1        A/5 21171  7.25              2
```

```
## 2         PC 17599 71.28   C85        0
## 3 STON/O2. 3101282  7.92              2
## 4           113803 53.10  C123        2
## 5           373450  8.05              2
## 6            17463 51.86   E46        2
```

```
colSums(is.na(train_df))
```

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

Exploratory Data Analysis

For my exploratory data analysis, I focused on variables SibSp and Parch. Sibsp describes the number of siblings or spouses a passenger was accompanied by aboard the ship. Parch describes the number of parents or children a passenger was accompanied by aboard the ship.

```
# Summarize siblings/spouses aboard and parents/children aboard
summary(train_df$SibSp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.5126  1.0000  5.0000
```

```
summary(train_df$Parch)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.4314  1.0000  6.0000
```

```
# Distribution of siblings/spouses aboard by survival
table(train_df$SibSp, train_df$Survived)
```

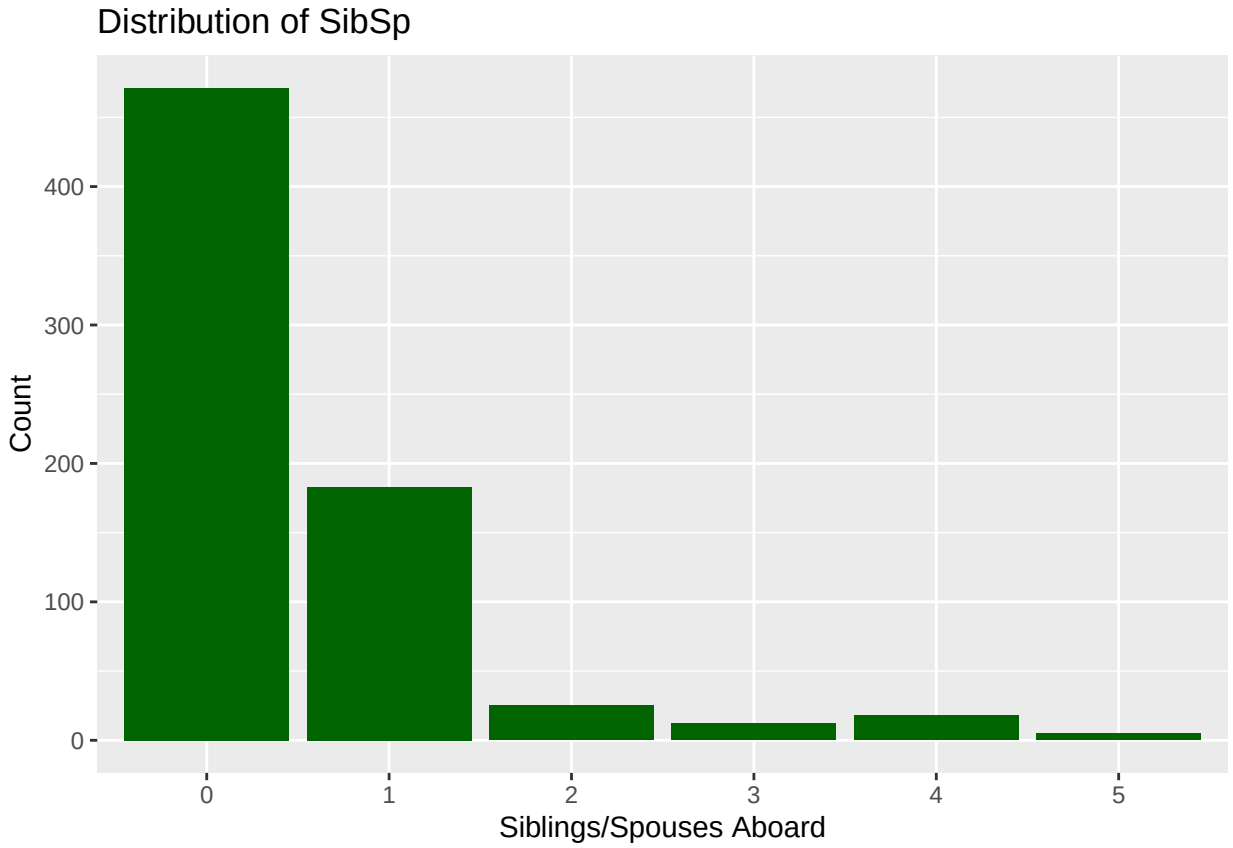
```
##
##       0   1
##   0 296 175
##   1  86  97
##   2  14  11
##   3   8   4
##   4  15   3
##   5   5   0
```

```
# Distribution of parents/children aboard by survival
table(train_df$Parch, train_df$Survived)
```

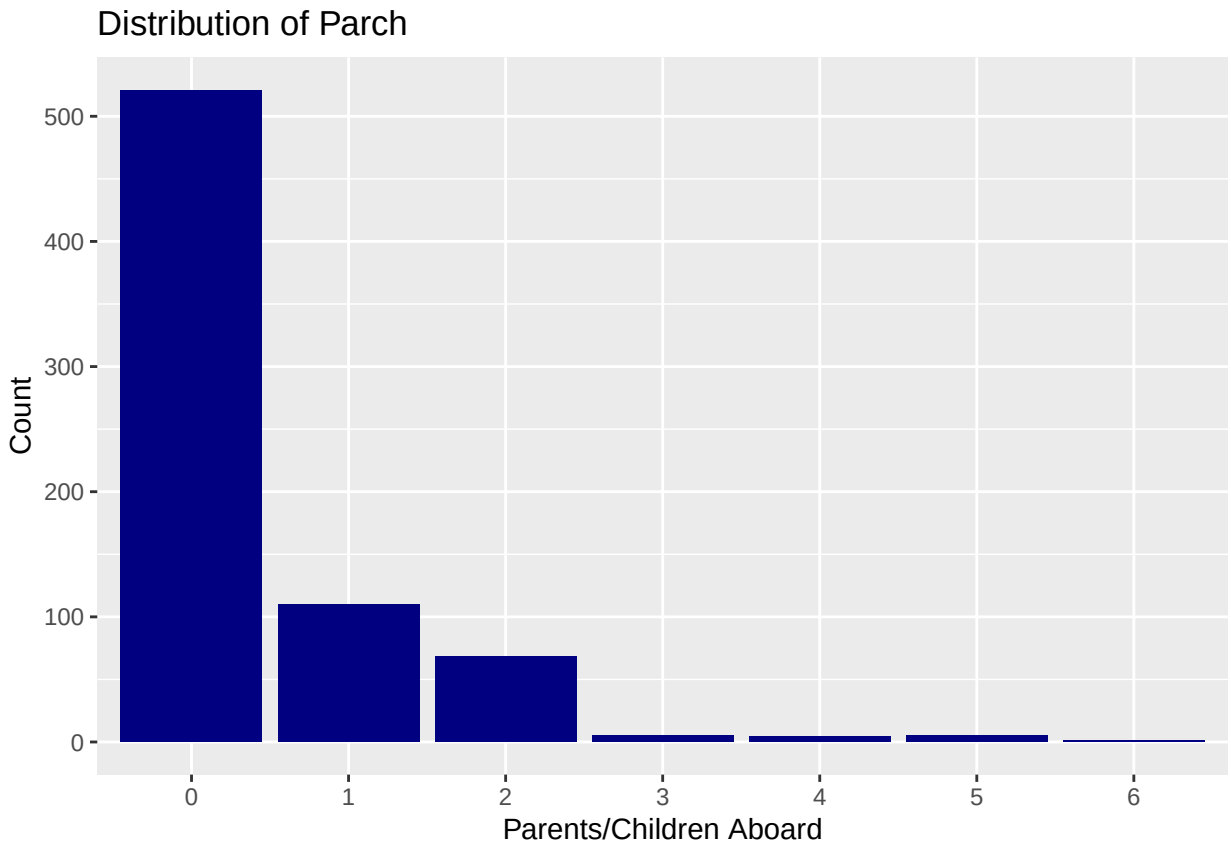
```
##
##       0   1
##   0 335 186
##   1  49  61
##   2  29  39
##   3   2   3
##   4   4   0
##   5   4   1
##   6   1   0
```

```
# Visualize distribution of number of siblings and spouses aboard
ggplot(data = train_df, mapping = aes(x = factor(SibSp))) +
  geom_bar(fill = "darkgreen") +
```

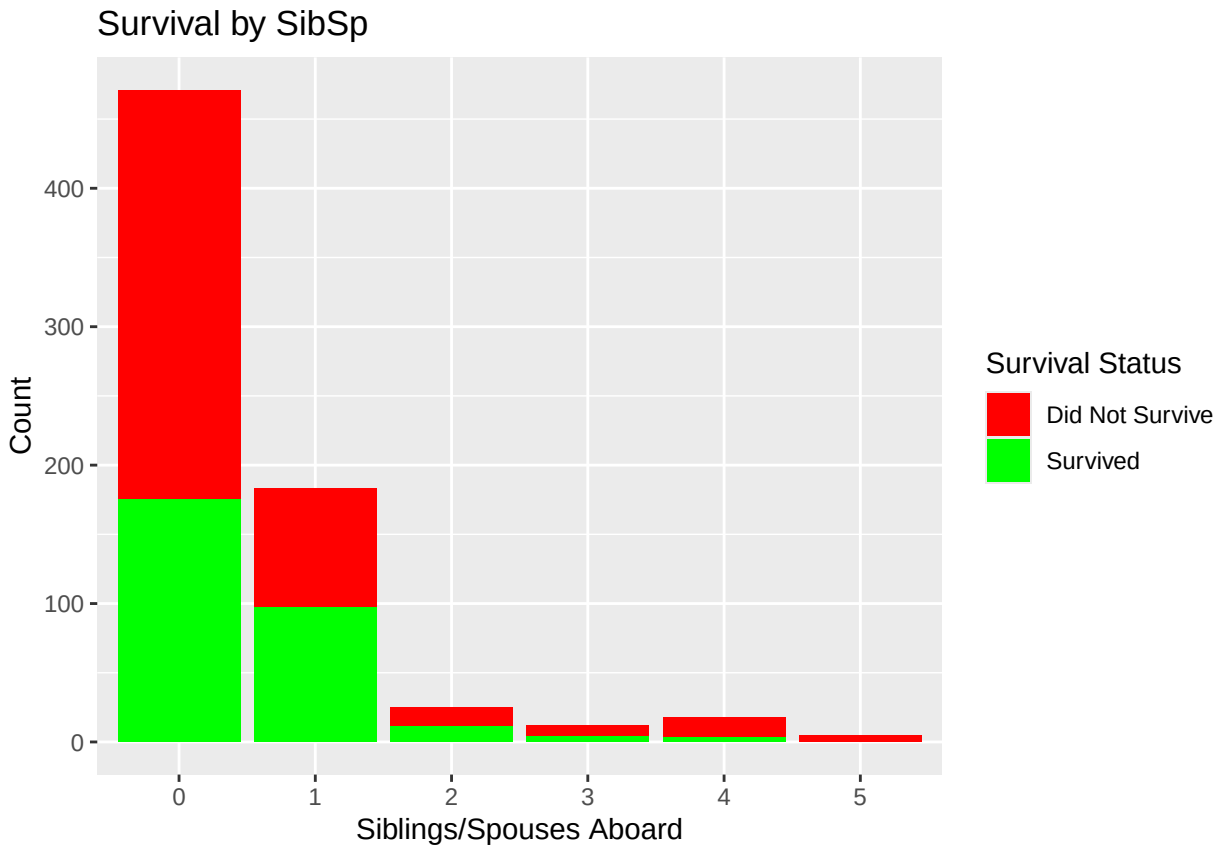
```
  labs(title = "Distribution of SibSp",
       x = "Siblings/Spouses Aboard",
       y = "Count")
```



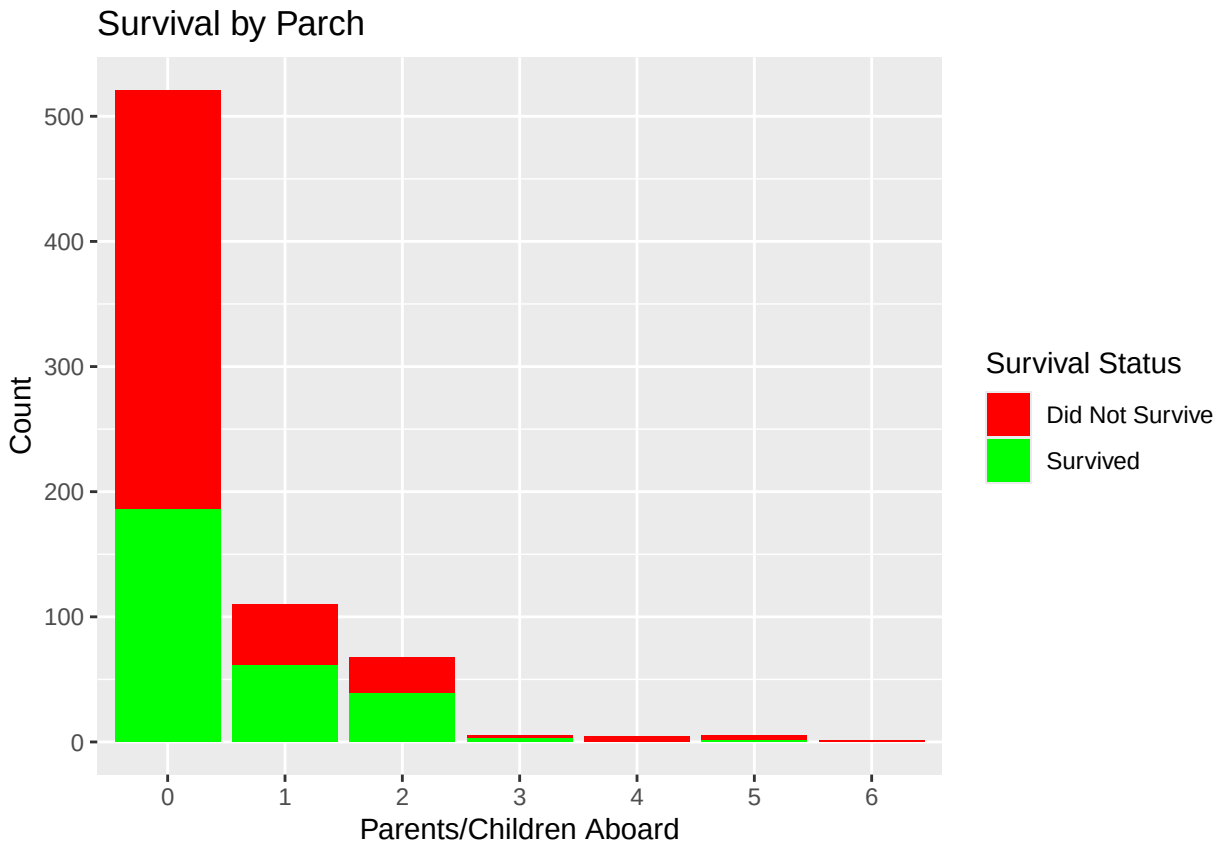

```
# Visualize distribution of number of parents and children aboard
ggplot(data = train_df, mapping = aes(x = factor(Parch))) +
  geom_bar(fill = "navyblue") +
  labs(title = "Distribution of Parch",
       x = "Parents/Children Aboard",
       y = "Count")
```




```
# Visualize survival by number of siblings/spouses aboard
ggplot(data = train_df,
       mapping = aes(x = factor(SibSp), fill = factor(Survived))) +
  geom_bar() +
  labs(title = "Survival by SibSp",
       x = "Siblings/Spouses Aboard",
       y = "Count",
       fill = "Survival Status") +
  scale_fill_manual(values = c("red", "green"),
                    labels = c("Did Not Survive", "Survived"))
```

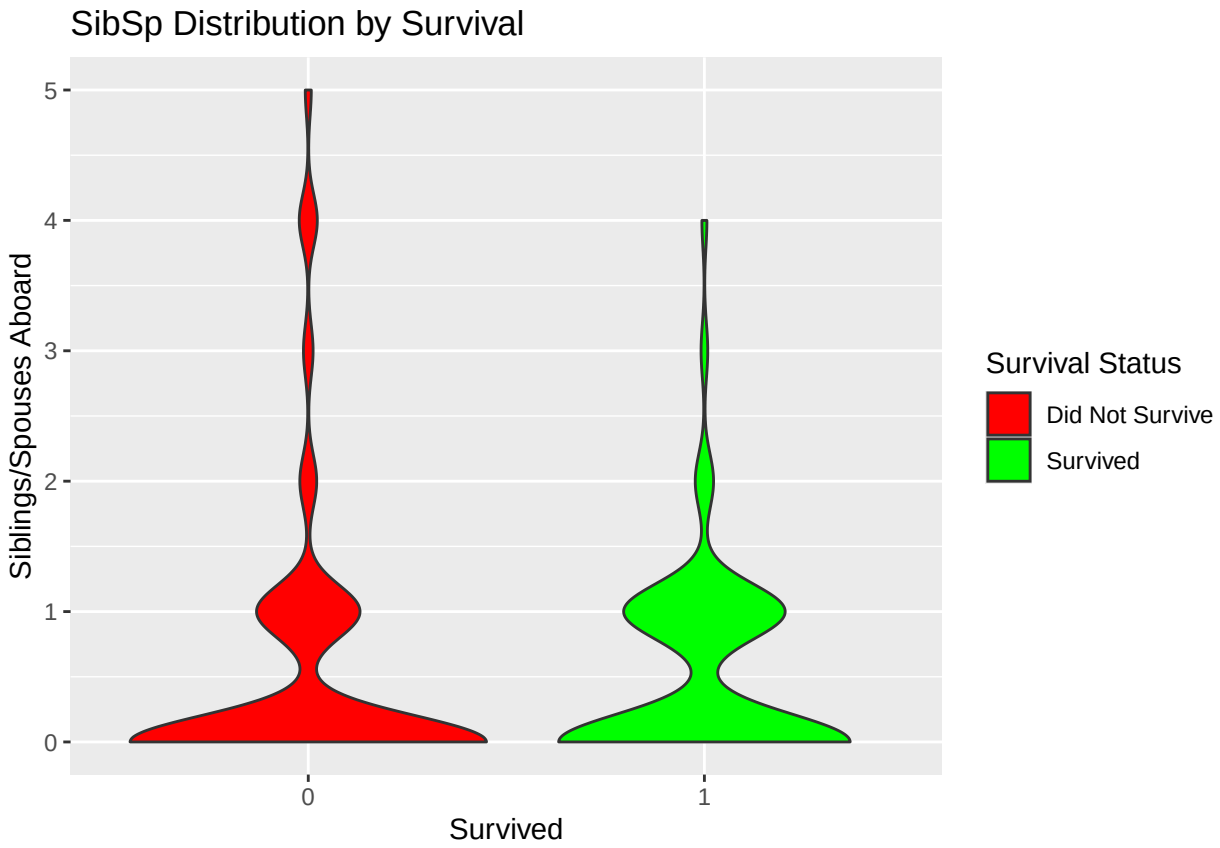



```
# Visualize survival by number of parents/children aboard
ggplot(data = train_df,
       mapping = aes(x = factor(Parch), fill = factor(Survived))) +
  geom_bar() +
  labs(title = "Survival by Parch",
       x = "Parents/Children Aboard",
       y = "Count",
       fill = "Survival Status") +
  scale_fill_manual(values = c("red", "green"),
                    labels = c("Did Not Survive", "Survived"))
```

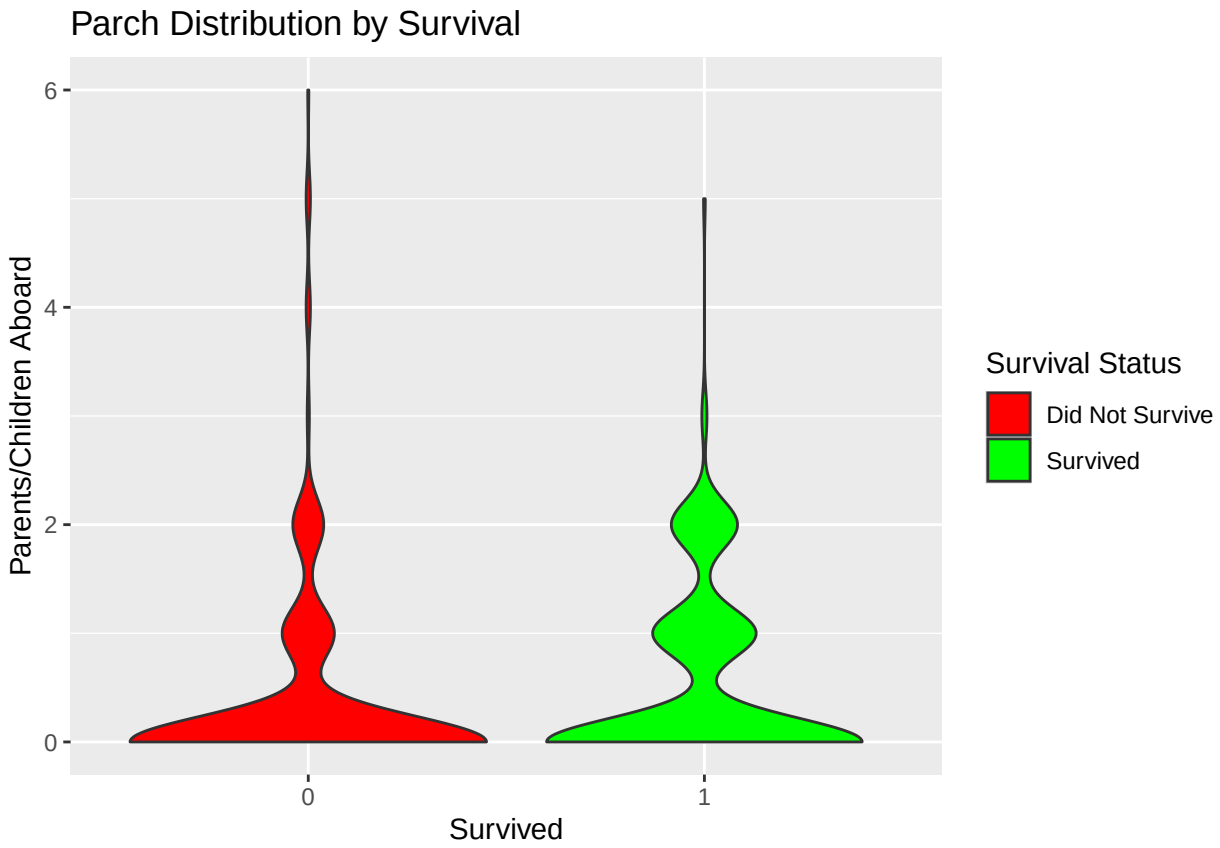


A smaller proportion of passengers survived than not, but the class imbalance between survival statuses does not appear to be excessive.

```
# Visualize distribution of siblings/spouses aboard by survival
ggplot(data = train_df,
       mapping = aes(x = factor(Survived), y = SibSp, fill = factor(Survived))) +
  geom_violin() +
  labs(title = "SibSp Distribution by Survival",
       x = "Survived",
       y = "Siblings/Spouses Aboard",
       fill = "Survival Status") +
  scale_fill_manual(values = c("red", "green"),
                    labels = c("Did Not Survive", "Survived"))
```

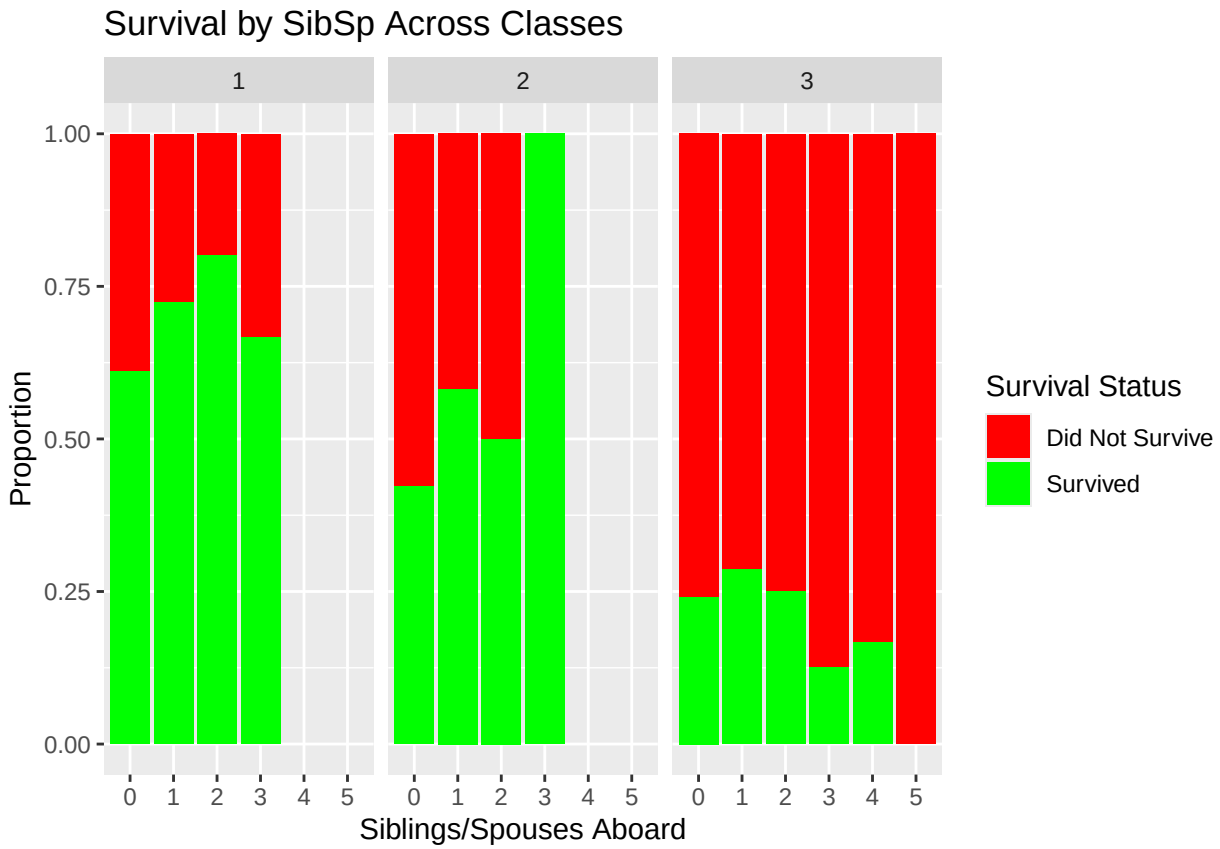



```
# Visualize distribution of parents/children aboard by survival
ggplot(data = train_df,
       mapping = aes(x = factor(Survived), y = Parch, fill = factor(Survived))) +
  geom_violin() +
  labs(title = "Parch Distribution by Survival",
       x = "Survived",
       y = "Parents/Children Aboard",
       fill = "Survival Status") +
  scale_fill_manual(values = c("red", "green"),
                    labels = c("Did Not Survive", "Survived"))
```

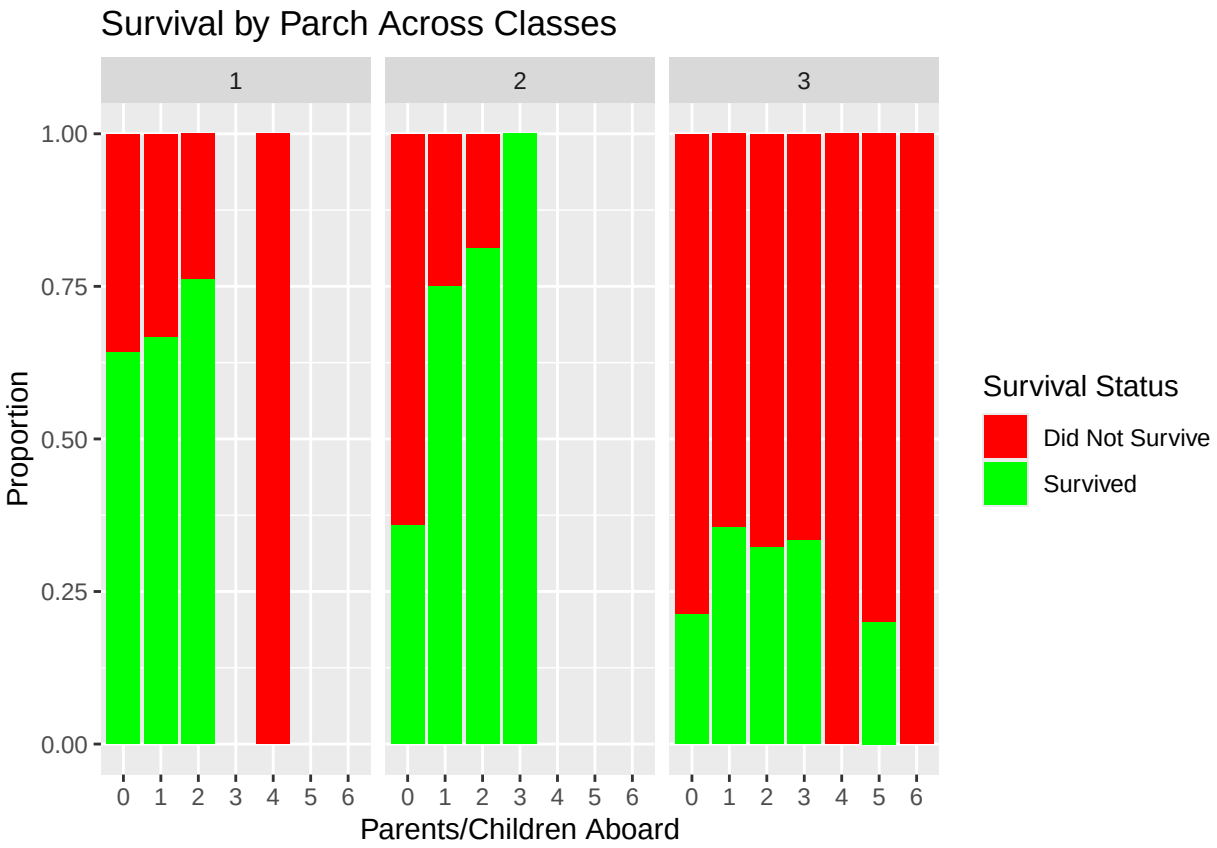


The vast majority of passengers did not have any family on board with them. Passengers with more than 2 family members aboard were relatively rare. No passengers with familial relations greater than 4 siblings/spouses or 5 parents/children survived.

```
# Visualize survival distribution of siblings/spouses aboard by class
ggplot(data = train_df,
       mapping = aes(x = factor(SibSp), fill = factor(Survived))) +
  geom_bar(position = "fill") +
  facet_wrap(~ Pclass) +
  labs(title = "Survival by SibSp Across Classes",
       x = "Siblings/Spouses Aboard",
       y = "Proportion",
       fill = "Survival Status") +
  scale_fill_manual(values = c("red", "green"),
                    labels = c("Did Not Survive", "Survived"))
```




```
# Visualize survival distribution of parents/children aboard by class
ggplot(data = train_df,
       mapping = aes(x = factor(Parch), fill = factor(Survived))) +
  geom_bar(position = "fill") +
  facet_wrap(~ Pclass) +
  labs(title = "Survival by Parch Across Classes",
       x = "Parents/Children Aboard",
       y = "Proportion",
       fill = "Survival Status") +
  scale_fill_manual(values = c("red", "green"),
                    labels = c("Did Not Survive", "Survived"))
```



After visualizing the distribution of surviving and non-surviving passengers by class and familial relations, it was clear to see that larger families were more common in the lowest class (class 3). The proportion of surviving passengers was much higher for those belonging to the higher classes (class 1 and 2) than for class 3.

Random Forest

One approach to a binary classification problem such as this Titanic challenge is a random forest. A random forest is a type of classification model that trains multiple decision trees on randomized subsets of the data and assigns a classification based on the combined predictions of each tree. Random forests do not assume linearity or normality, as they are a type of non-parametric model that is able to capture non-linear relationships in the data.

```
str(train_df)
```

```
## 'data.frame':    714 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 7 8 9 10 11 ...
##  $ Survived   : int  0 1 1 1 0 0 0 1 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 1 3 3 2 3 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ Sex        : int  0 1 1 1 0 0 0 1 1 1 ...
##  $ Age        : num  22 38 26 35 35 54 2 27 14 4 ...
##  $ SibSp      : int  1 1 0 1 0 0 3 0 1 1 ...
##  $ Parch      : int  0 0 0 0 0 0 1 2 0 1 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : int  2 0 2 2 2 2 2 2 0 2 ...
```

```r
# Set factor variables as factors

train_df$Survived <- as.factor(train_df$Survived)
train_df$Pclass <- as.factor(train_df$Pclass)
train_df$Sex <- as.factor(train_df$Sex)
train_df$SibSp <- as.factor(train_df$SibSp)
train_df$Parch <- as.factor(train_df$Parch)
train_df$Embarked <- as.factor(train_df$Embarked)

str(train_df)
```

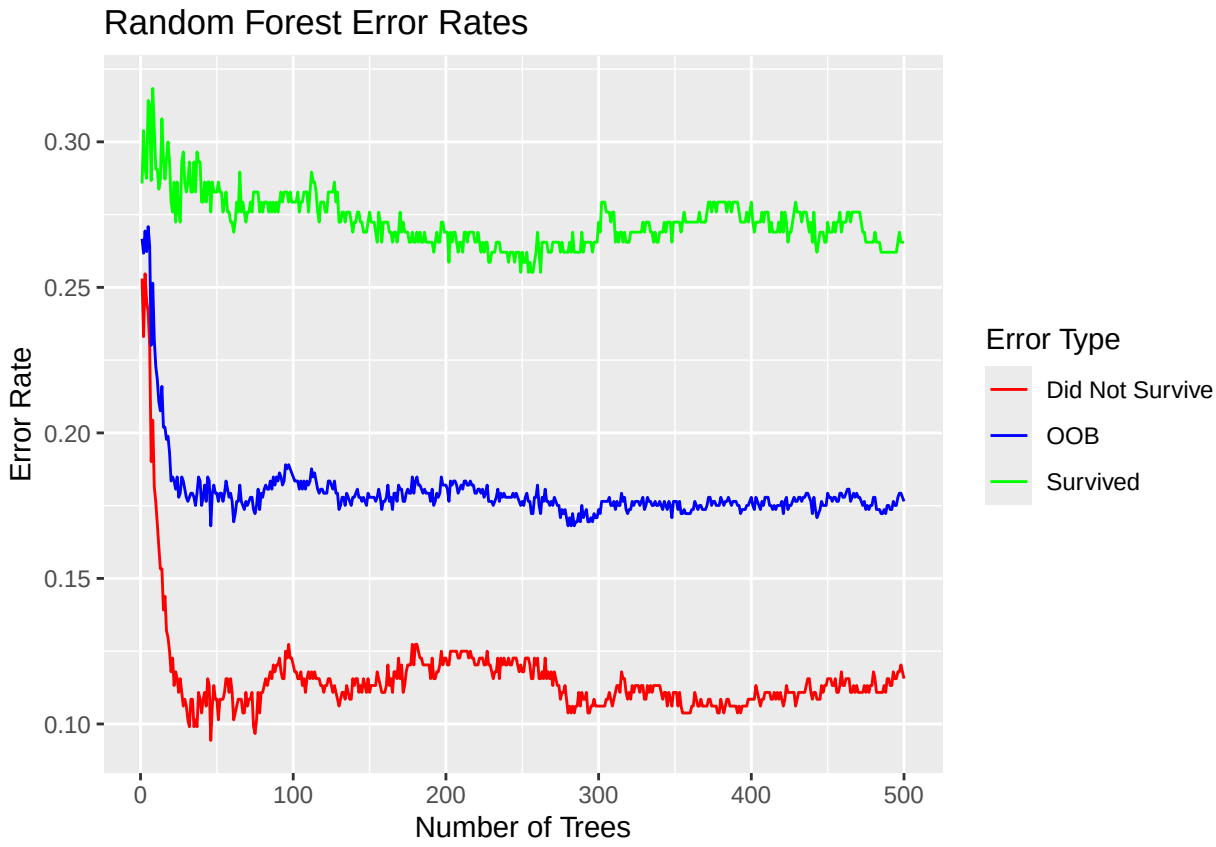
```
## 'data.frame':    714 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 7 8 9 10 11 ...
##  $ Survived   : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 2 2 2 ...
##  $ Pclass     : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 1 3 3 2 3 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ Sex        : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 2 2 2 ...
##  $ Age        : num  22 38 26 35 35 54 2 27 14 4 ...
##  $ SibSp      : Factor w/ 6 levels "0","1","2","3",..: 2 2 1 2 1 1 4 1 2 2 ...
##  $ Parch      : Factor w/ 7 levels "0","1","2","3",..: 1 1 1 1 1 1 2 3 1 2 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : Factor w/ 3 levels "0","1","2": 3 1 3 3 3 3 3 3 1 3 ...
```

```r
# Create random forest model
set.seed(478)

rfmodel <- randomForest(Survived ~ ., data = train_df, proximity = T)

rfmodel
```

```
##
## Call:
##  randomForest(formula = Survived ~ ., data = train_df, proximity = T)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 17.65%
## Confusion matrix:
##     0   1 class.error
## 0 375  49   0.1155660
## 1  77 213   0.2655172
```

```r
# Create error rate data frame
oob_error_data <- data.frame(
  Trees = rep(1:nrow(rfmodel$err.rate), times = 3),
  Type = rep(c("OOB", "Did Not Survive", "Survived"), each = nrow(rfmodel$err.rate)),
  Error = c(rfmodel$err.rate[, "OOB"],
            rfmodel$err.rate[, "0"],
            rfmodel$err.rate[, "1"])
)

# Visualize error rates
```

```
ggplot(data = oob_error_data, mapping = aes(x = Trees, y = Error)) +
  geom_line(mapping = aes(color = Type)) +
  labs(title = "Random Forest Error Rates",
       x = "Number of Trees",
       y = "Error Rate",
       color = "Error Type") +
  scale_color_manual(values = c("red", "blue" ,"green"),
                     labels = c("Did Not Survive", "OOB" ,"Survived"))
```



The default parameters for a random forest in R is 500 trees and 3 variables tried at each split. This default model produced an out-of-bag (OOB) error rate of 17.65%. The "survived" class suffered a greater frequency of erroneous predictions (class error 26.55%) than the "did not survive" class (class error 11.56%). This means that the model is most often committing Type 1 error, incorrectly predicting that passengers survived when they did not.

The graph above visualizes the OOB error rate, "survived" class error rate, and "did not survive" class error rate for each addition of a decision tree.

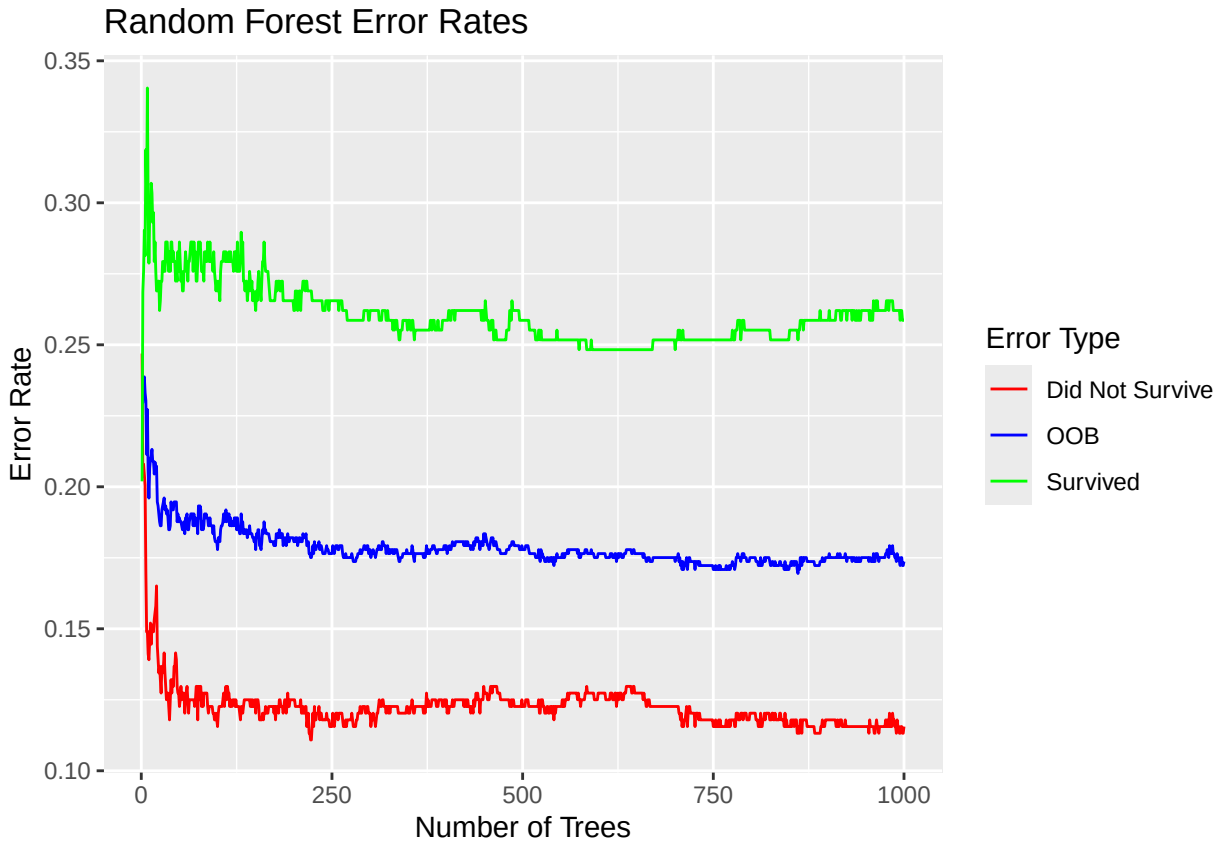
In order to improve these error rates, I tried increasing the number of trees included in the model from 500 to 1000.

```
# Create model with increased number of trees (1000)
rfmodel2 <- randomForest(Survived ~ ., data = train_df, ntree = 1000, proximity = T)

# Check error rates
rfmodel2
```

```
##
## Call:
```

```
##  randomForest(formula = Survived ~ ., data = train_df, ntree = 1000,      proximity = T)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 17.37%
## Confusion matrix:
##     0   1 class.error
## 0 375  49   0.1155660
## 1  75 215   0.2586207
```

```r
# Create new error data frame
oob_error_data2 <- data.frame(
  Trees = rep(1:nrow(rfmodel2$err.rate), times = 3),
  Type = rep(c("OOB", "Did Not Survive", "Survived"), each = nrow(rfmodel2$err.rate)),
  Error = c(rfmodel2$err.rate[, "OOB"],
            rfmodel2$err.rate[, "0"],
            rfmodel2$err.rate[, "1"])
)

# Visualize error rates
ggplot(data = oob_error_data2, mapping = aes(x = Trees, y = Error)) +
  geom_line(mapping = aes(color = Type)) +
  labs(title = "Random Forest Error Rates",
       x = "Number of Trees",
       y = "Error Rate",
       color = "Error Type") +
  scale_color_manual(values = c("red", "blue" ,"green"),
                     labels = c("Did Not Survive", "OOB" ,"Survived"))
```



Increasing the number of decision trees only slightly improved the OOB error rate to 17.37% and the "survived" class error rate to 25.86%, while the "did not survive" class error remained the same (11.56%).

In the graph above, you can see that the error rates plateau as the number of trees increases, indicating that we should not expect to see an improvement in error rates by the addition of more decision trees.

Next, I checked which number of variables checked at each split would result in the lowest OOB error rate. The default is 3 variables, and I checked from 1 to 10 variables.

```
# Determine best number of variables to try at each split

oob_values <- vector(length = 10)

for(i in 1:10) {
  temp_model <- randomForest(Survived ~ ., data = train_df, mtry = i, ntree = 1000)
  oob_values[i] <- temp_model$err.rate[nrow(temp_model$err.rate), 1]
}

oob_values
```

```
##  [1] 0.1848739 0.1862745 0.1764706 0.1722689 0.1764706 0.1876751 0.1778711
##  [8] 0.1890756 0.1848739 0.1848739
```

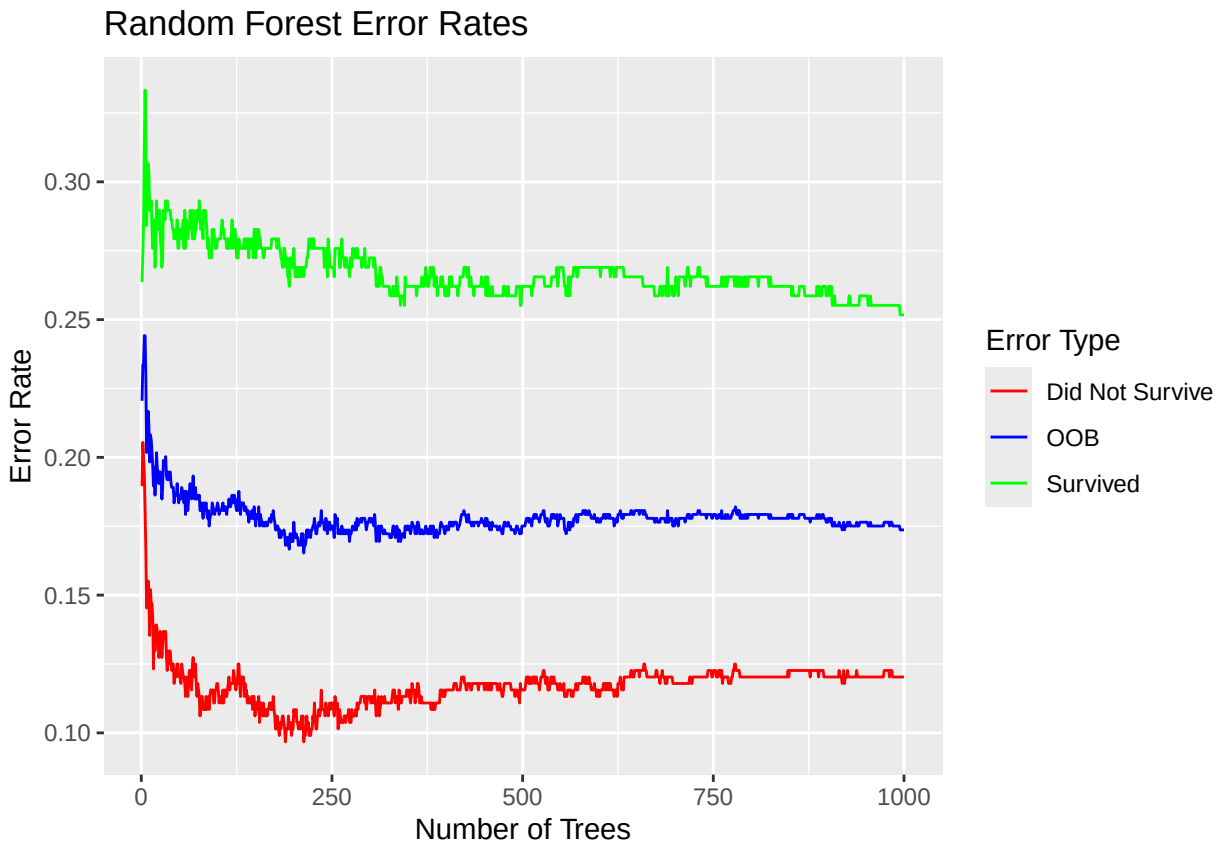
The number of variables that produced the lowest OOB error rate (17.23%) was 4 variables.

```
# Update model
rfmodel3 <- randomForest(Survived ~ .,
                         data = train_df,
                         mtry = 4,
                         ntree = 1000,
```

```
                        proximity = T)

# Check error rates
rfmodel3
```

```
##
## Call:
##  randomForest(formula = Survived ~ ., data = train_df, mtry = 4,      ntree = 1000, proximity = T)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 17.37%
## Confusion matrix:
##     0   1 class.error
## 0 373  51   0.1202830
## 1  73 217   0.2517241
```

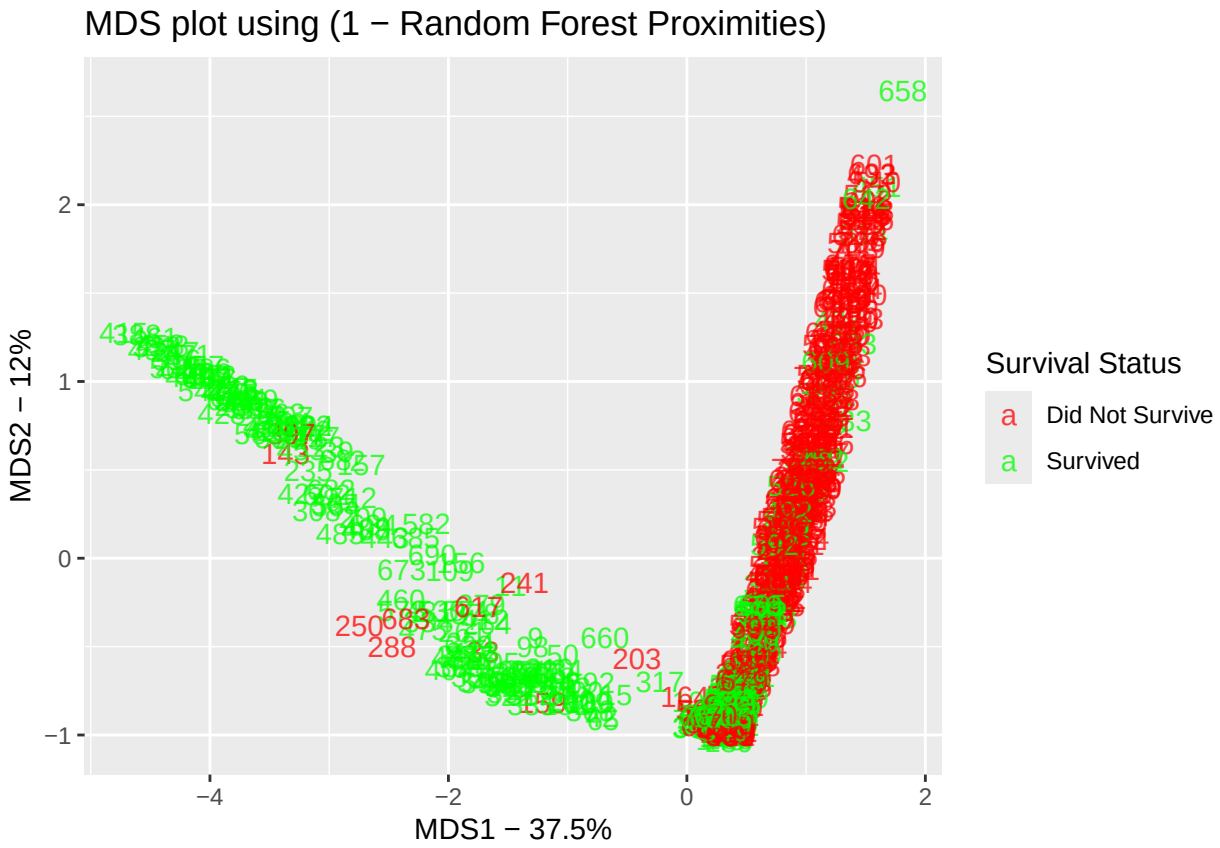
```
# Create new error data frame
oob_error_data3 <- data.frame(
  Trees = rep(1:nrow(rfmodel3$err.rate), times = 3),
  Type = rep(c("OOB", "Did Not Survive", "Survived"), each = nrow(rfmodel3$err.rate)),
  Error = c(rfmodel3$err.rate[, "OOB"],
            rfmodel3$err.rate[, "0"],
            rfmodel3$err.rate[, "1"])
)

# Visualize error rates
ggplot(data = oob_error_data3, mapping = aes(x = Trees, y = Error)) +
  geom_line(mapping = aes(color = Type)) +
  labs(title = "Random Forest Error Rates",
       x = "Number of Trees",
       y = "Error Rate",
       color = "Error Type") +
  scale_color_manual(values = c("red", "blue" ,"green"),
                     labels = c("Did Not Survive", "OOB" ,"Survived"))
```



The new OOB error rate did not improve from the last model using 1000 trees and 3 variables per split and instead remained the same (17.37%). This new model with 1000 trees and 4 variables per split does show a slight improvement in the "survived" class error (now 25.17%), but also a slight increase in the "did not survive" class error (12.03%).

To visualize classification of this model, I created a multidimensional scaling (MDS) plot.

```
# Create multidimensional scaling plot
distance_matrix <- dist(1 - rfmodel3$proximity)

mds_info <- cmdscale(distance_matrix, eig = T, x.ret = T)

mds_var_per <- round(mds_info$eig / sum(mds_info$eig) * 100, 1)

mds_values <- mds_info$points
mds_data <- data.frame(Sample = rownames(mds_values),
                       X = mds_values[, 1],
                       Y = mds_values[, 2],
                       Status = train_df$Survived)

ggplot(data = mds_data, mapping = aes(x = X, y = Y, label = Sample)) +
  geom_text(mapping = aes(color = Status), alpha = 0.75) +
  labs(title = "MDS plot using (1 - Random Forest Proximities)",
       color = "Survival Status",
       x = paste("MDS1 - ", mds_var_per[1], "%", sep = ""),
       y = paste("MDS2 - ", mds_var_per[2], "%", sep = "")) +
  scale_color_manual(values = c("red", "green"),
                     labels = c("Did Not Survive", "Survived"))
```



MDS plots visualize the relative similarities between points in a data set, with closer points denoting greater similarity and further points denoting greater dissimilarity. This plot shows the individuals classified as "survived" in green on the left and individuals classified as "did not survive" in red on the right. Visually it is clear that many data points appear to have been misclassfied.

The percentages included on the axes of the graph denote the amount of variation in the distance matrix that each dimension accounts for. The Y axis accounts for 12% of variation in the distance matrix, and the X axis accounts for 37.5%.

Generating Predictions

```
# Add empty "Survived" column to test_df
library(tibble)
test_df <- add_column(test_df, Survived = rep(NA, nrow(test_df)), .after = "PassengerId")
```

```
# Ensure variables are same type in both data sets
for(col in names(train_df)) {
  if (is.factor(train_df[[col]])) {
    test_df[[col]] <- factor(test_df[[col]], levels = levels(train_df[[col]]))
  }
}

# Create predictions using random forest model
test_predictions <- predict(rfmodel3, test_df)

predictions <- data.frame(PassengerId = test_df$PassengerId, Survived = test_predictions)

head(predictions)
```

```
##   PassengerId Survived
## 1         892        0
## 2         893        0
## 3         894        0
## 4         895        0
## 5         896        1
## 6         897        0
```

Conclusion

The random forest model performed decently well, with an accuracy rate of 82.63% (1 - OOB error). Due to the model's black box nature, it does not lend much interpretability or insight into feature importance.

The "survived" class error rate was 12.03%, and the "did not survive" class error rate was 25.17%. This model is prone to greater Type 1 error than Type 2, meaning that it more often incorrectly classifies non-surviving passengers as "survived" rather than surviving passengers as "did not survive".

My exploratory analysis suggested that larger families were more common in the lowest class (class 3), and that members of the third class survived at a much lower frequency than members of the first and second classes. This corroborates historical understanding. Due to limited access to lifeboats, ship layout, and physical barriers such as gates separating classes aboard the ship, third class passengers are much more likely to have lost their lives on the Titanic.

Bowdoin. (n.d.). Disproportionate Devastation. Titanic. https://courses.bowdoin.edu/history-2203-fall-2020-kmoyniha/reflection/#:~:text=There%20is%20no%20doubt%20that,passengers%20was%20not%20necessarily%20surprising.