Titanic Disaster Prediction 2022

Salisu Mohammed

Created November 21, 2022

Predicting Survivability Using RandomForest

First thing first, let's load the readr and dplyr package

```
library(readr)
library(dplyr)
library(ggplot2)
```

The Train data and Test Data

We begin by importing the datasets needed for this analysis

```
# import the train data

train <- read_csv("train.csv")

# import the test data
test <- read_csv("test.csv")
```

We are going to be training the `train` data and then run the prediction on the `test` data.

```
head(train)
```

Predicting Survivability Using RandomForest

```
## # A tibble: 6 × 12
##   PassengerId Survived Pclass Name    Sex     Age SibSp Parch Ticket  Fare Cabin
##         <dbl>    <dbl>  <dbl> <chr>   <chr> <dbl> <dbl> <dbl> <chr>  <dbl> <chr>
## 1           1        0      3 Braund… male     22     1     0 A/5 2…  7.25 <NA>
## 2           2        1      1 Cuming… fema…    38     1     0 PC 17… 71.3  C85
## 3           3        1      3 Heikki… fema…    26     0     0 STON/…  7.92 <NA>
## 4           4        1      1 Futrel… fema…    35     1     0 113803 53.1  C123
## 5           5        0      3 Allen,… male     35     0     0 373450  8.05 <NA>
## 6           6        0      3 Moran,… male     NA     0     0 330877  8.46 <NA>
## # … with 1 more variable: Embarked <chr>
```

```
# examine the structure

glimpse(train)
```

```
## Rows: 891
## Columns: 12
## $ PassengerId <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,…
## $ Survived    <dbl> 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1…
## $ Pclass      <dbl> 3, 1, 3, 1, 3, 3, 1, 3, 3, 2, 3, 1, 3, 3, 3, 2, 3, 2, 3, 3…
## $ Name        <chr> "Braund, Mr. Owen Harris", "Cumings, Mrs. John Bradley (Fl…
## $ Sex         <chr> "male", "female", "female", "female", "male", "male", "mal…
## $ Age         <dbl> 22, 38, 26, 35, 35, NA, 54, 2, 27, 14, 4, 58, 20, 39, 14, …
## $ SibSp       <dbl> 1, 1, 0, 1, 0, 0, 0, 3, 0, 1, 1, 0, 0, 1, 0, 0, 4, 0, 1, 0…
## $ Parch       <dbl> 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 1, 0, 0, 5, 0, 0, 1, 0, 0, 0…
## $ Ticket      <chr> "A/5 21171", "PC 17599", "STON/O2. 3101282", "113803", "37…
## $ Fare        <dbl> 7.2500, 71.2833, 7.9250, 53.1000, 8.0500, 8.4583, 51.8625,…
## $ Cabin       <chr> NA, "C85", NA, "C123", NA, NA, "E46", NA, NA, NA, "G6", "C…
## $ Embarked    <chr> "S", "C", "S", "S", "S", "Q", "S", "S", "S", "C", "S", "S"…
```

Add a new Survived column to the `test` data.

Predicting Survivability Using RandomForest

```
# add a column using mutate and save as a new variable

test.survived <- test %>%
  mutate(
    Survived = rep("none", nrow(test))
)

head(test.survived)
```

```
## # A tibble: 6 × 12
##   PassengerId Pclass Name     Sex     Age SibSp Parch Ticket  Fare Cabin Embar…¹
##         <dbl>  <dbl> <chr>    <chr> <dbl> <dbl> <dbl> <chr>  <dbl> <chr> <chr>
## 1         892      3 Kelly, … male   34.5     0     0 330911  7.83 <NA>  Q
## 2         893      3 Wilkes,… fema…  47       1     0 363272  7    <NA>  S
## 3         894      2 Myles, … male   62       0     0 240276  9.69 <NA>  Q
## 4         895      3 Wirz, M… male   27       0     0 315154  8.66 <NA>  S
## 5         896      3 Hirvone… fema…  22       1     1 31012… 12.3  <NA>  S
## 6         897      3 Svensso… male   14       0     0 7538    9.22 <NA>  S
## # … with 1 more variable: Survived <chr>, and abbreviated variable name
## #   ¹Embarked
```

Join the `train` and `test.survived` datasets

```
train.test <- rbind(train, test.survived)

head(train.test)
```

Predicting Survivability Using RandomForest

```
## # A tibble: 6 × 12
##   PassengerId Survived Pclass Name    Sex     Age SibSp Parch Ticket  Fare Cabin
##         <dbl> <chr>     <dbl> <chr>   <chr> <dbl> <dbl> <dbl> <chr>  <dbl> <chr>
## 1           1 0             3 Braund… male     22     1     0 A/5 2…  7.25 <NA>
## 2           2 1             1 Cuming… fema…    38     1     0 PC 17… 71.3  C85
## 3           3 1             3 Heikki… fema…    26     0     0 STON/…  7.92 <NA>
## 4           4 1             1 Futrel… fema…    35     1     0 113803 53.1  C123
## 5           5 0             3 Allen,… male     35     0     0 373450  8.05 <NA>
## 6           6 0             3 Moran,… male     NA     0     0 330877  8.46 <NA>
## # … with 1 more variable: Embarked <chr>
```

Converting columns to the appropriate class.

```
# Convert Sex, Pclass, Sex, Survived, SibSp, Parch, Embrked to factors

# check the structure

# create a function to convert variable class

convert_func <- function(x) {
  x <- as.factor(x)
}

glimpse(train.test)
```

Predicting Survivability Using RandomForest

```
## Rows: 1,309
## Columns: 12
## $ PassengerId <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,…
## $ Survived    <chr> "0", "1", "1", "1", "0", "0", "0", "0", "1", "1", "1", "1"…
## $ Pclass      <dbl> 3, 1, 3, 1, 3, 3, 1, 3, 3, 2, 3, 1, 3, 3, 3, 2, 3, 2, 3, 3…
## $ Name        <chr> "Braund, Mr. Owen Harris", "Cumings, Mrs. John Bradley (Fl…
## $ Sex         <chr> "male", "female", "female", "female", "male", "male", "mal…
## $ Age         <dbl> 22, 38, 26, 35, 35, NA, 54, 2, 27, 14, 4, 58, 20, 39, 14, …
## $ SibSp       <dbl> 1, 1, 0, 1, 0, 0, 0, 3, 0, 1, 1, 0, 0, 1, 0, 0, 4, 0, 1, 0…
## $ Parch       <dbl> 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 1, 0, 0, 5, 0, 0, 1, 0, 0, 0…
## $ Ticket      <chr> "A/5 21171", "PC 17599", "STON/O2. 3101282", "113803", "37…
## $ Fare        <dbl> 7.2500, 71.2833, 7.9250, 53.1000, 8.0500, 8.4583, 51.8625,…
## $ Cabin       <chr> NA, "C85", NA, "C123", NA, NA, "E46", NA, NA, NA, "G6", "C…
## $ Embarked    <chr> "S", "C", "S", "S", "S", "Q", "S", "S", "S", "C", "S", "S"…
```

```
train.test <- train.test %>%
  mutate(
    Pclass = convert_func(Pclass),
    Sex = convert_func(Sex),
    Survived = convert_func(Survived),
    Parch = convert_func(Parch),
    SibSp = convert_func(SibSp),
    Embarked = convert_func(Embarked)
  )

glimpse(train.test)
```

Predicting Survivability Using RandomForest

```
## Rows: 1,309
## Columns: 12
## $ PassengerId <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,…
## $ Survived    <fct> 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1…
## $ Pclass      <fct> 3, 1, 3, 1, 3, 3, 1, 3, 3, 2, 3, 1, 3, 3, 3, 2, 3, 2, 3, 3…
## $ Name        <chr> "Braund, Mr. Owen Harris", "Cumings, Mrs. John Bradley (Fl…
## $ Sex         <fct> male, female, female, female, male, male, male, male, fema…
## $ Age         <dbl> 22, 38, 26, 35, 35, NA, 54, 2, 27, 14, 4, 58, 20, 39, 14, …
## $ SibSp       <fct> 1, 1, 0, 1, 0, 0, 0, 3, 0, 1, 1, 0, 0, 1, 0, 0, 4, 0, 1, 0…
## $ Parch       <fct> 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 1, 0, 0, 5, 0, 0, 1, 0, 0, 0…
## $ Ticket      <chr> "A/5 21171", "PC 17599", "STON/O2. 3101282", "113803", "37…
## $ Fare        <dbl> 7.2500, 71.2833, 7.9250, 53.1000, 8.0500, 8.4583, 51.8625,…
## $ Cabin       <chr> NA, "C85", NA, "C123", NA, NA, "E46", NA, NA, NA, "G6", "C…
## $ Embarked    <fct> S, C, S, S, S, Q, S, S, S, C, S, S, S, S, S, S, Q, S, S, C…
```

Exploratory Data Analysis

Let's see the total number of people that survived or perish from the `train.test` data.

```
train.test %>%
  count(Survived)
```

```
## # A tibble: 3 × 2
##   Survived     n
##   <fct>    <int>
## 1 0          549
## 2 1          342
## 3 none       418
```

They are 549 victims and 342 survivor from the `train` data. the "none" paramenter is from the newly created survived column in the `test` data.

Let's explore the data some more

Predicting Survivability Using RandomForest

```
train.test %>%
  count(Pclass)
```

```
## # A tibble: 3 × 2
##   Pclass     n
##   <fct>  <int>
## 1 1        323
## 2 2        277
## 3 3        709
```

```
summary(train.test$Sex)
```

```
## female   male
##    466    843
```

```
summary(train.test$SibSp)
```

```
##   0   1   2   3   4   5   8
## 891 319  42  20  22   6   9
```

```
summary(train.test$Parch)
```

```
##    0    1    2    3    4    5    6    9
## 1002  170  113    8    6    6    2    2
```

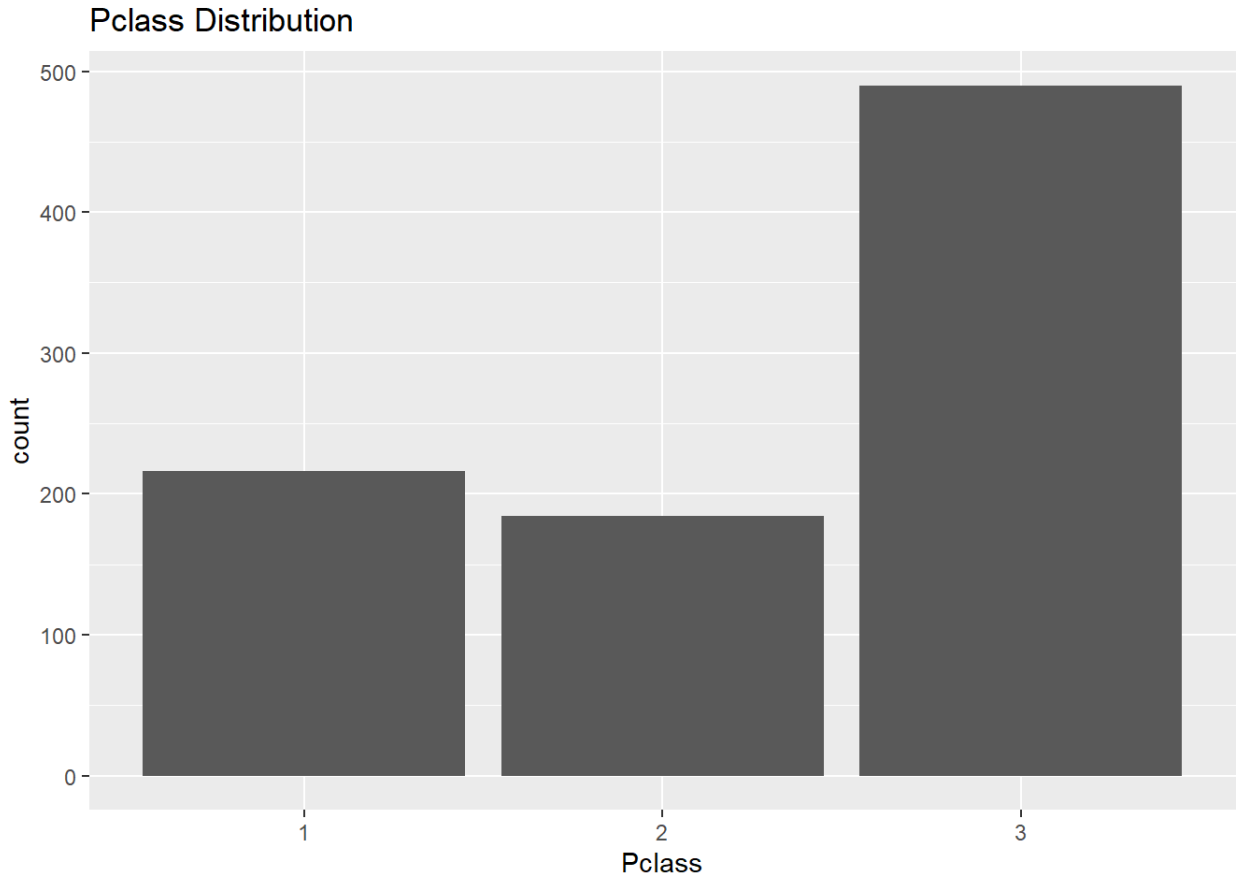
```
summary(train.test$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.17   21.00   28.00   29.88   39.00   80.00     263
```

Predicting Survivability Using RandomForest

Visualising the predictive behavior of each variable

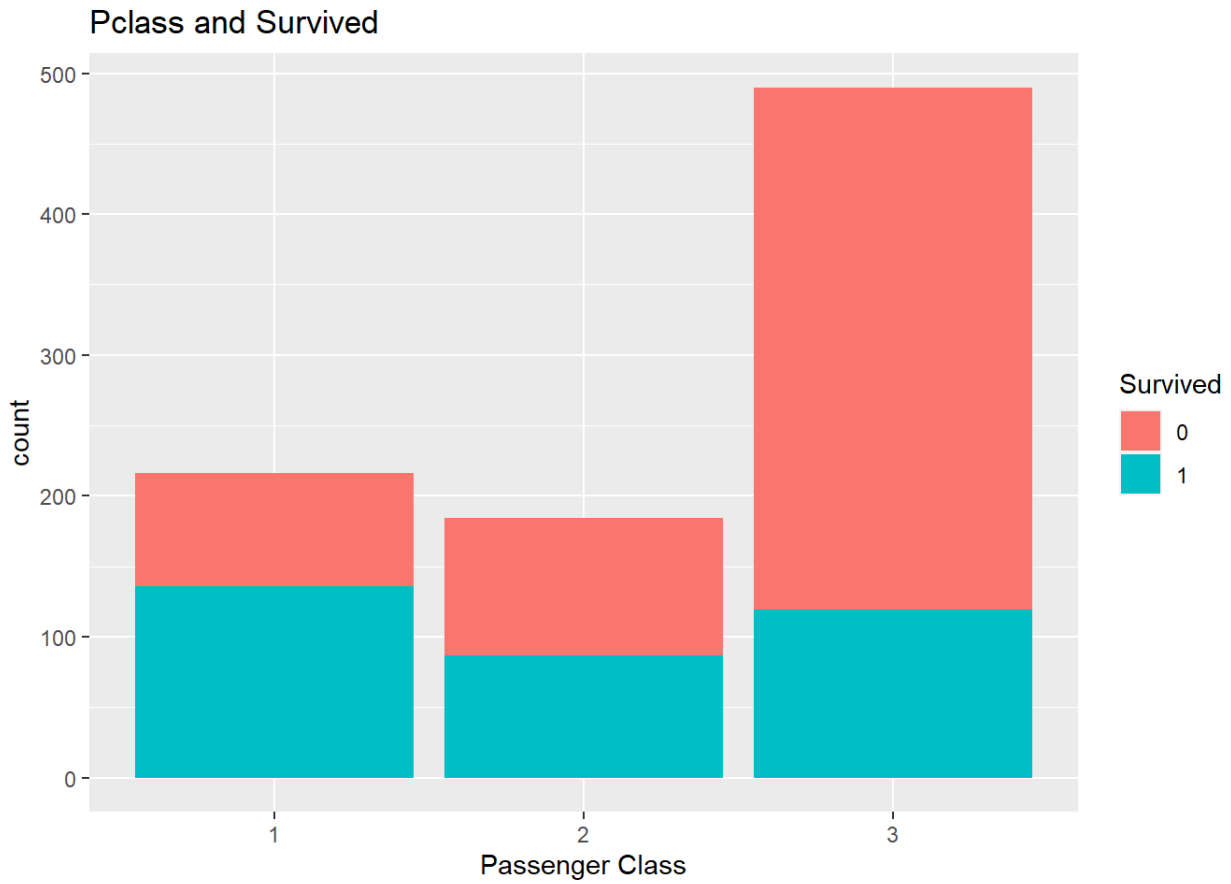
```
ggplot(train.test[1:890, ], aes(Pclass)) +
  geom_bar() +
  ggtitle("Pclass Distribution")
```



There are more passengers in third class than any other class

Predicting Survivability Using RandomForest

```
ggplot(train.test[1:890, ], aes(Pclass, fill = factor(Survived))) +
  geom_bar() +
  xlab("Passenger Class") +
  labs(fill = "Survived") +
  ggtitle("Pclass and Survived")
```



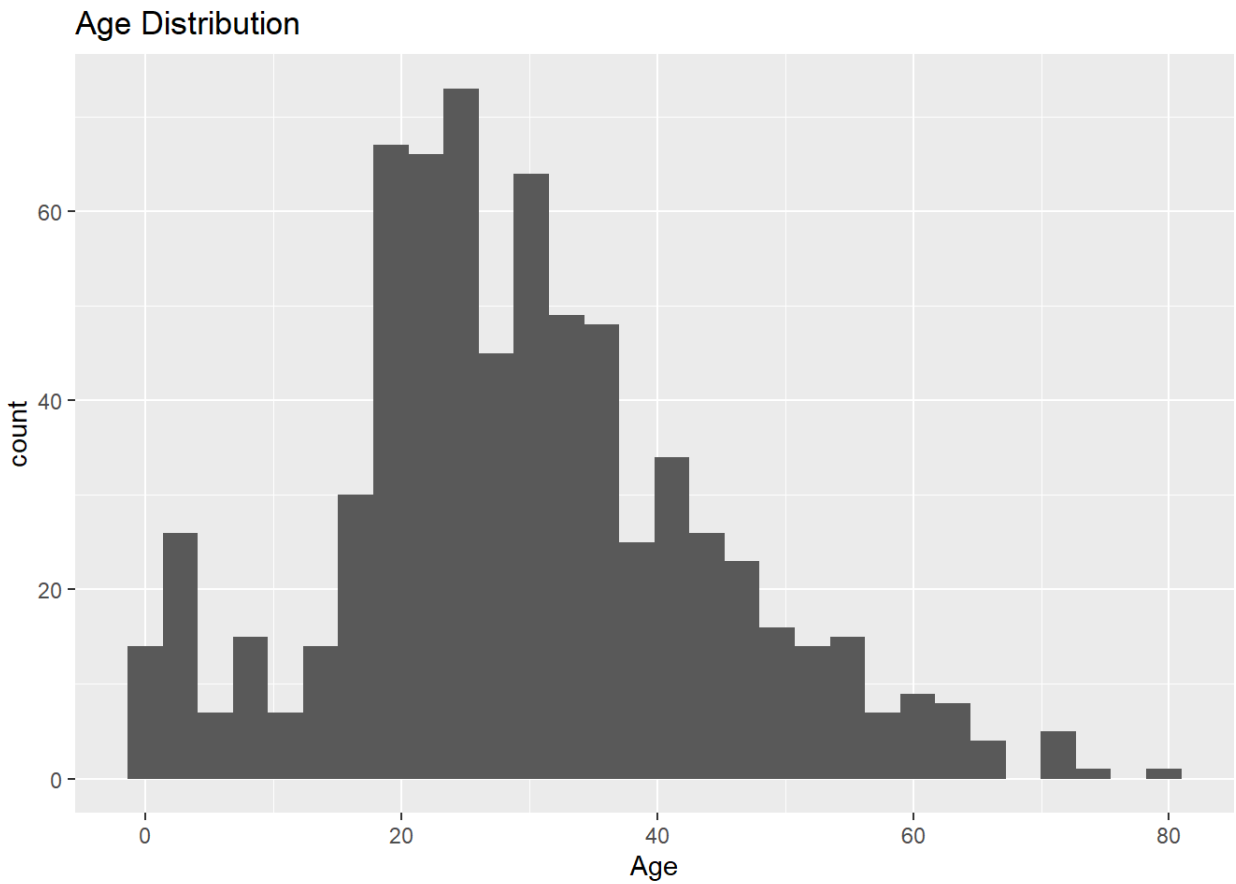
Third class has the highest number of casualties.

```
ggplot(train.test[1:890, ], aes(Age))+
  geom_histogram() +
  ggtitle("Age Distribution")
```

Predicting Survivability Using RandomForest

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

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



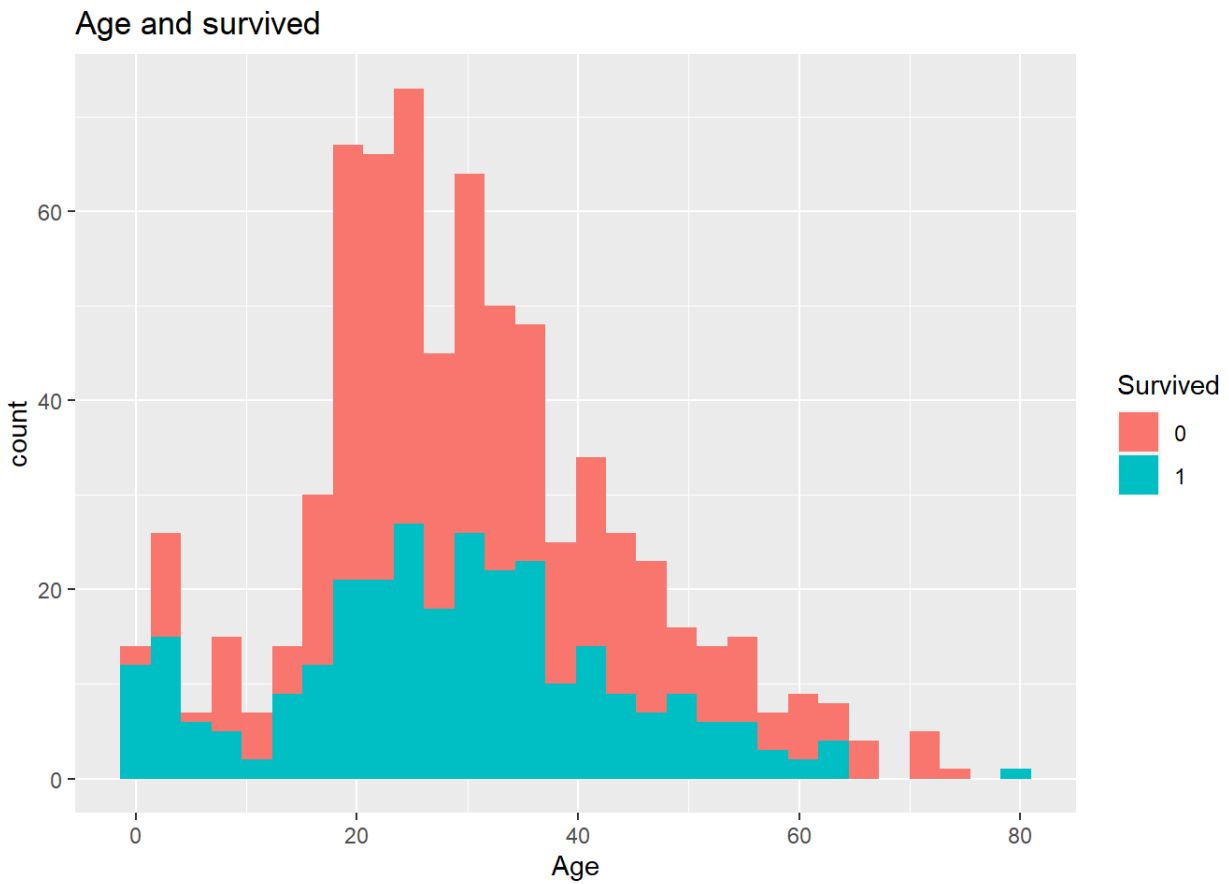
The Age distribution is more clusterd around 20-40 years. According to the data, children are more than likely to survive than adult. We will visualise this hypothesis next but only for exploration.

```
ggplot(train.test[1:891, ], aes(Age, fill = Survived)) +
  geom_histogram() +
  ggtitle("Age and survived")
```

Predicting Survivability Using RandomForest

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

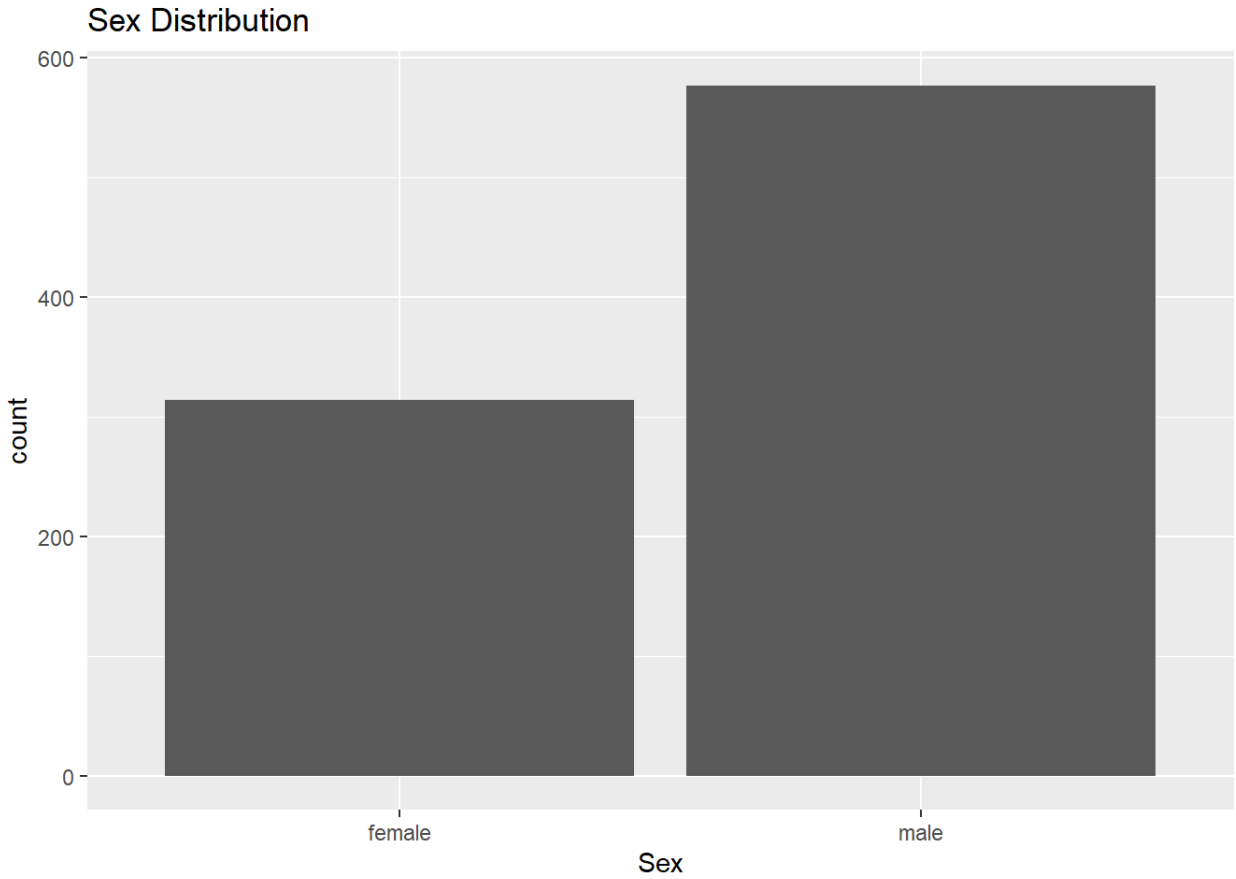
```
## Warning: Removed 177 rows containing non-finite values (stat_bin).
```



Aha! Age is a good predictor. In addition to age, we want to look at the distribution of sex and whether it is also a good predictor.

```
ggplot(train.test[1:891, ], aes(Sex)) +
  geom_bar() +
  ggtitle("Sex Distribution")
```

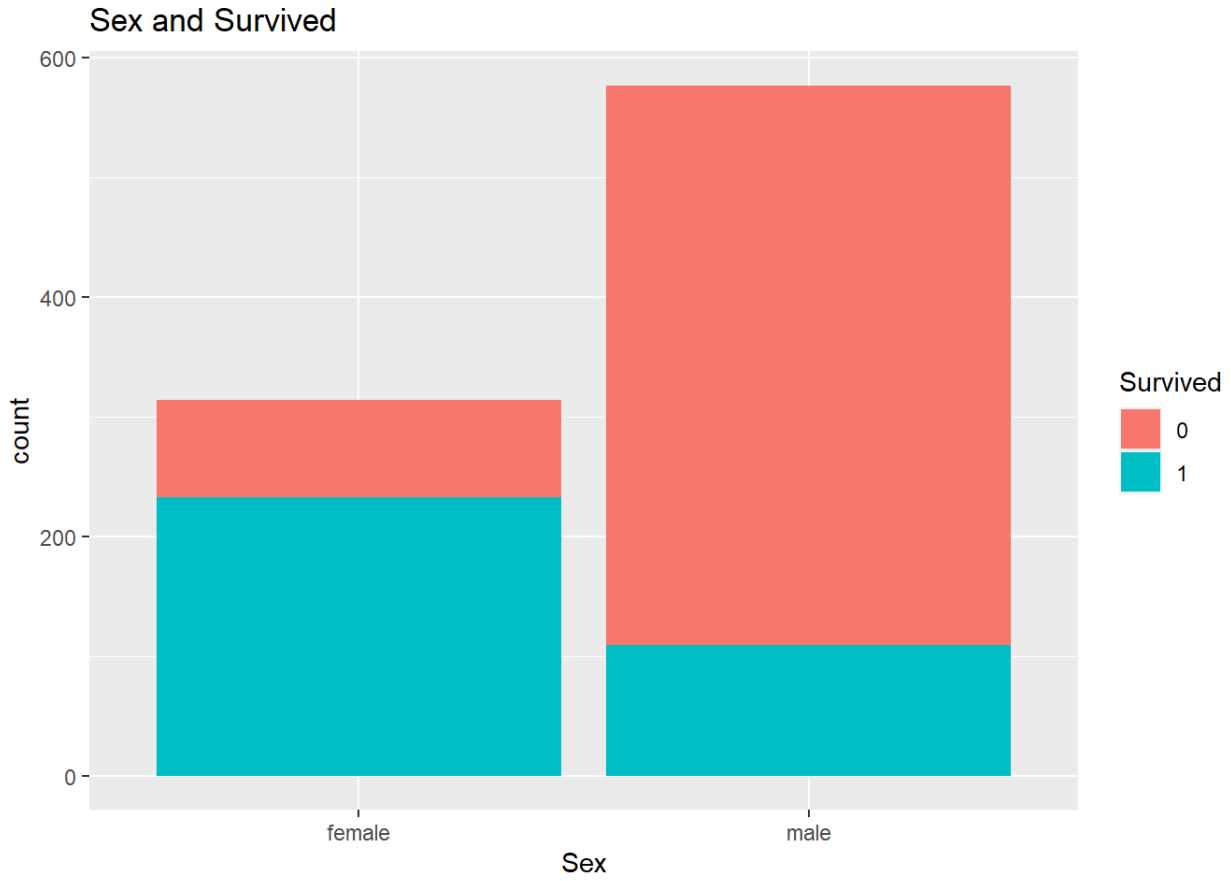
Predicting Survivability Using RandomForest



More males onboard than female.

```
ggplot(train.test[1:891, ], aes(Sex, fill = Survived)) +
  geom_bar() +
  ggtitle("Sex and Survived")
```

Predicting Survivability Using RandomForest



Oops! A large chunk of males onboard perished.

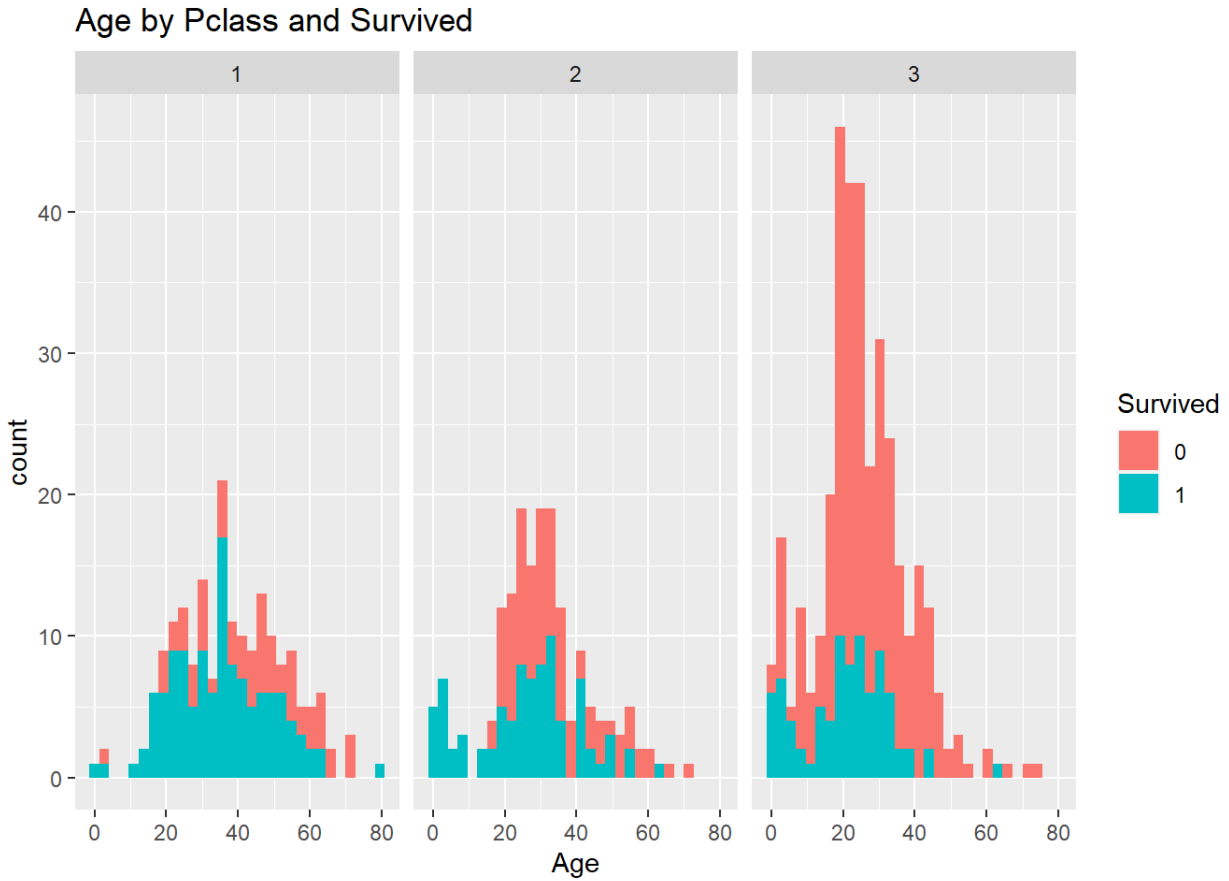
Multi-variate exploration

```
plot1 <- ggplot(train.test[1:891, ], aes(Age, fill = Survived)) +
  geom_histogram()

plot1 +
  facet_wrap(~ Pclass) +
  ggtitle("Age by Pclass and Survived")
```

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

Predicting Survivability Using RandomForest

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



Interesting plot! survival decreases as age increases.

The SibSp and Parch Variable

SipSp is the number of siblings or spouses traveling together. Parch shows whether a passenger is travelling with a parent or partner.

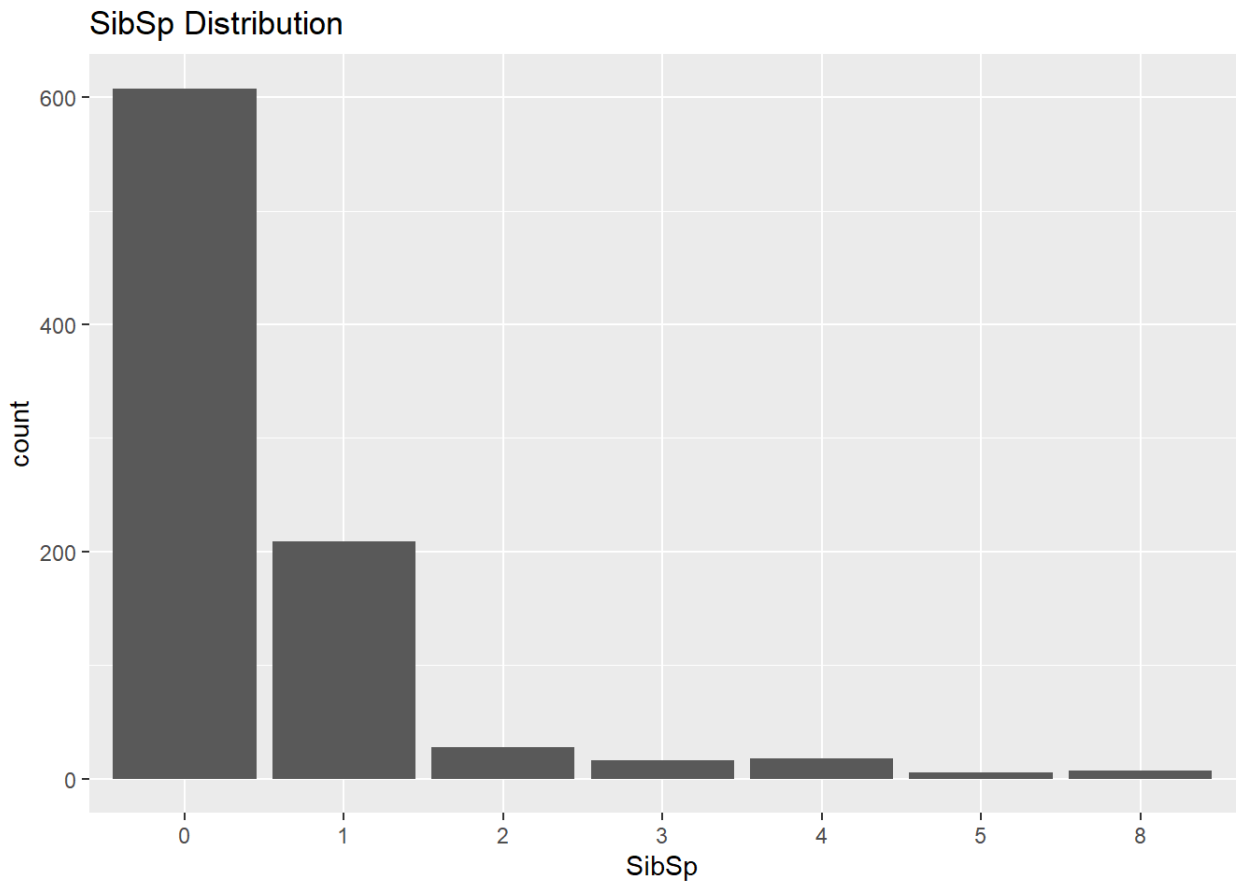
Question: Does a passenger travelling alone has a higher chance of survival?

To answer this, we will be looking at the distribution of each variable.

We will later transfrom this to a new variable later.

Predicting Survivability Using RandomForest

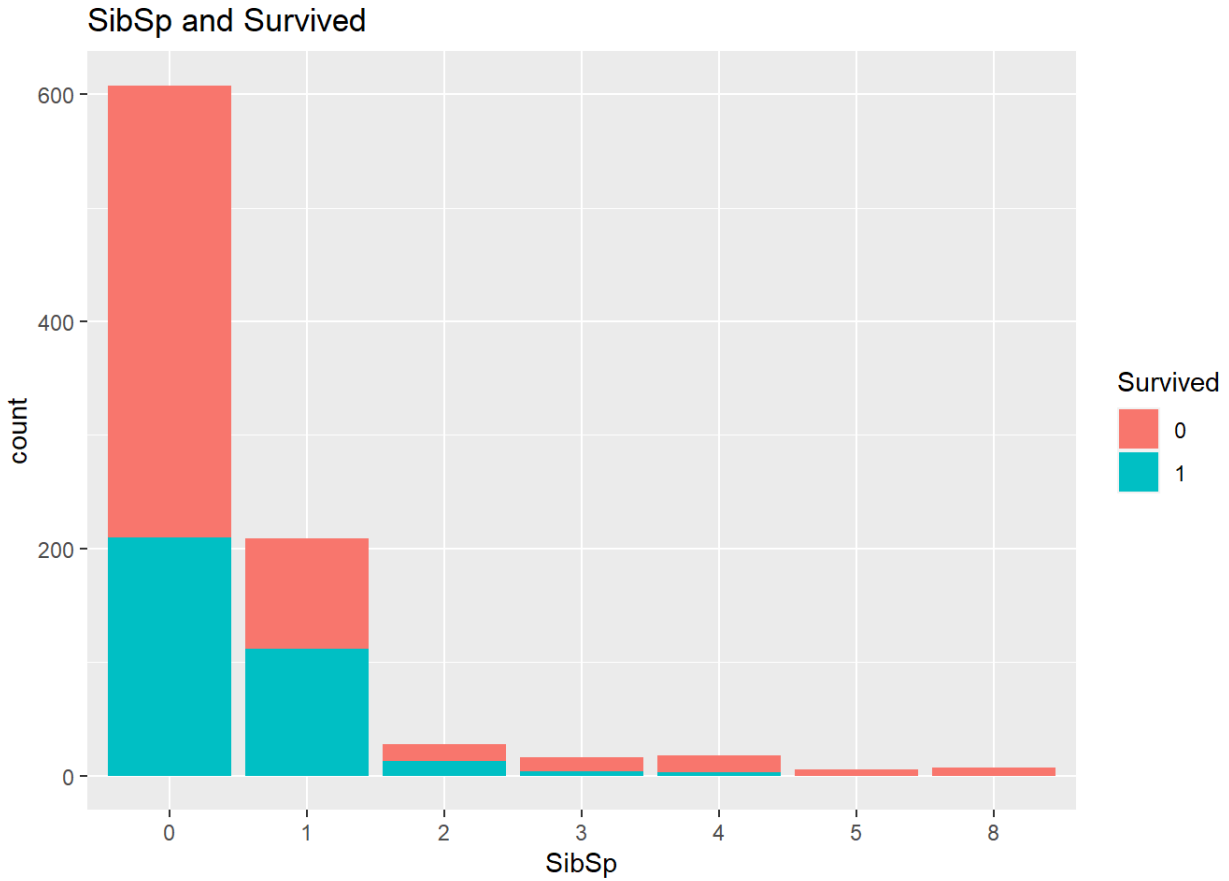
```
ggplot(train.test[1:891, ], aes(SibSp)) +
  geom_bar() +
  ggtitle("SibSp Distribution")
```



Let's add the survival rate

```
ggplot(train.test[1:891, ], aes(SibSp, fill = Survived)) +
  geom_bar() +
  ggtitle("SibSp and Survived")
```

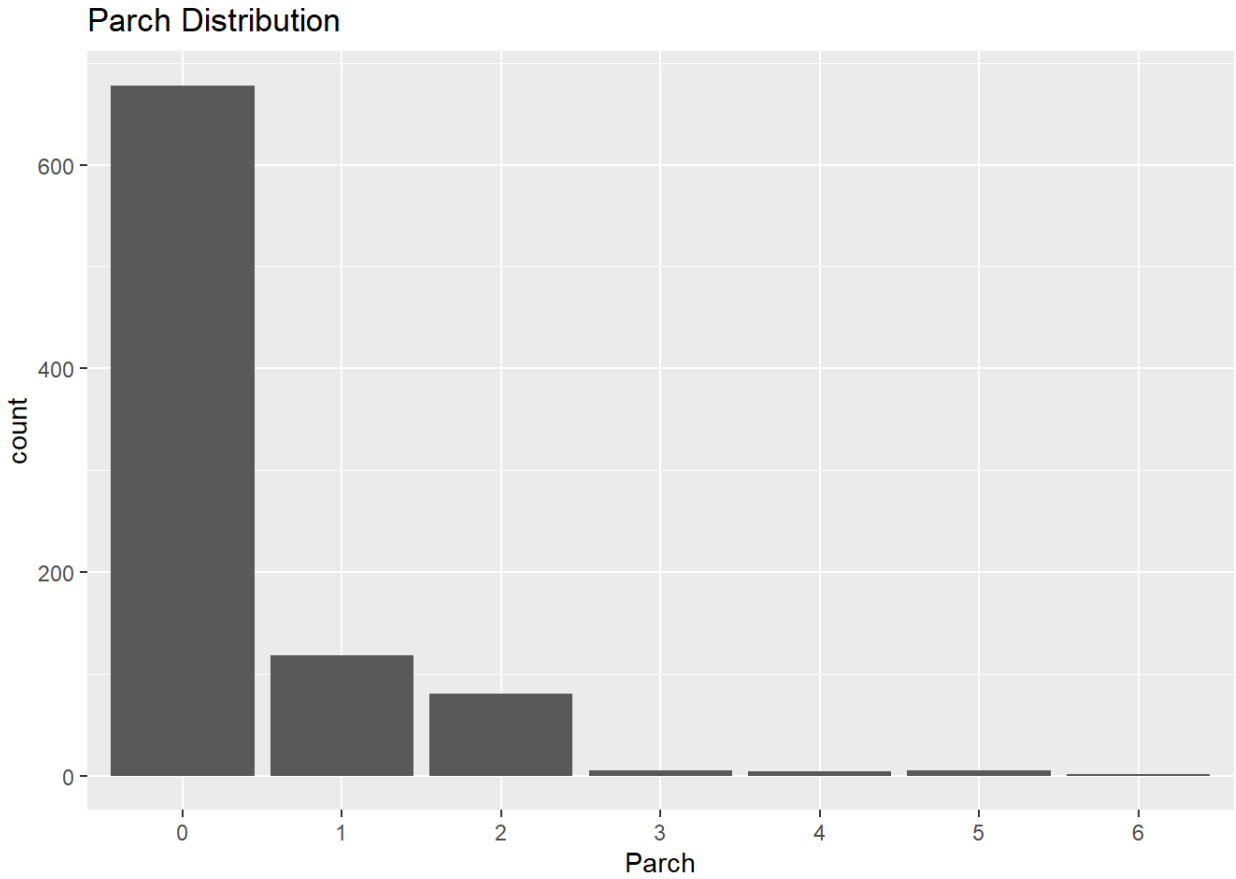
Predicting Survivability Using RandomForest



Survival rate decreases as number of people travelling together increases.

```
ggplot(train.test[1:891, ], aes(Parch)) +
  geom_bar() +
  ggtitle("Parch Distribution")
```

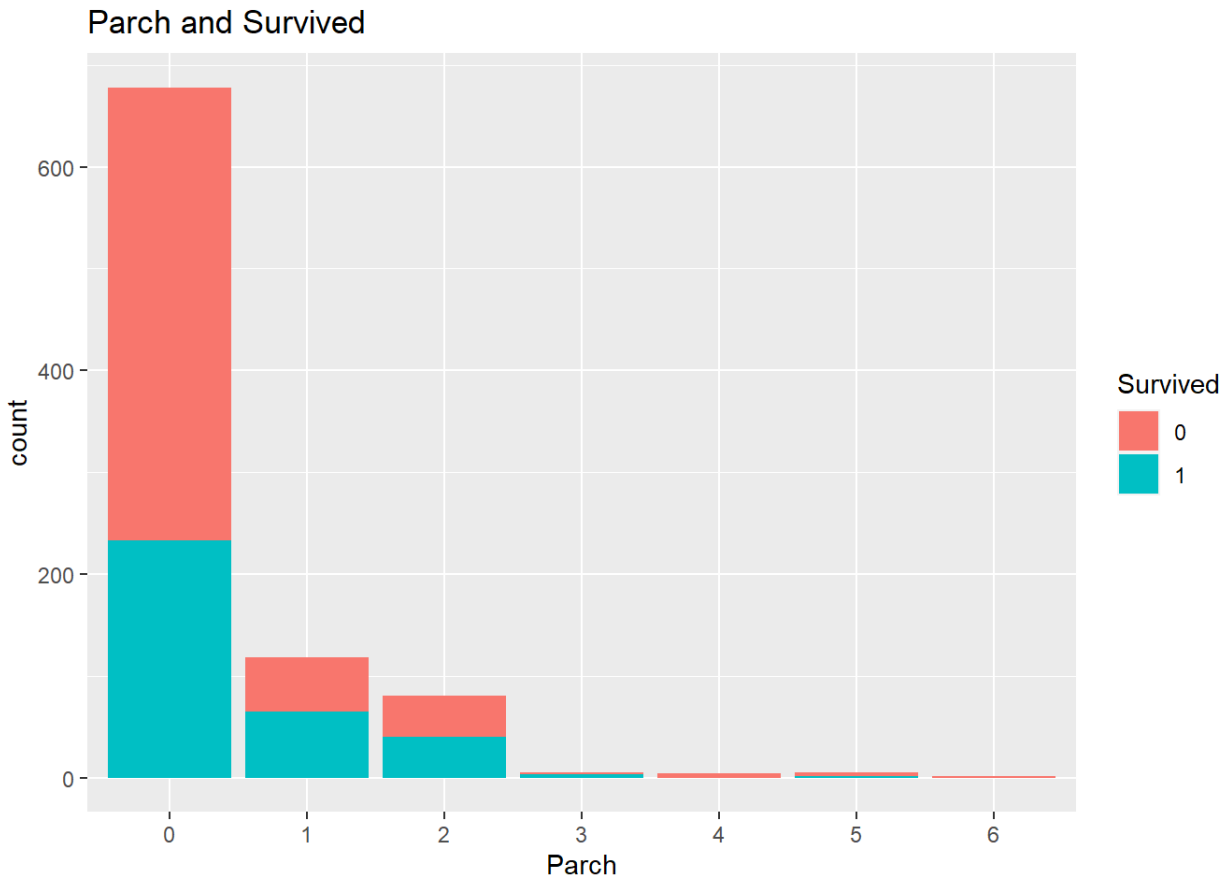
Predicting Survivability Using RandomForest



Hmm! looks like the same pattern. next…

```
ggplot(train.test[1:891, ], aes(Parch, fill = Survived)) +
  geom_bar() +
  ggtitle("Parch and Survived")
```

Predicting Survivability Using RandomForest



Feature Engineering

Adult Child Age Group

Creating a Child and Adult Variable from the `train.test` data.

Predicting Survivability Using RandomForest

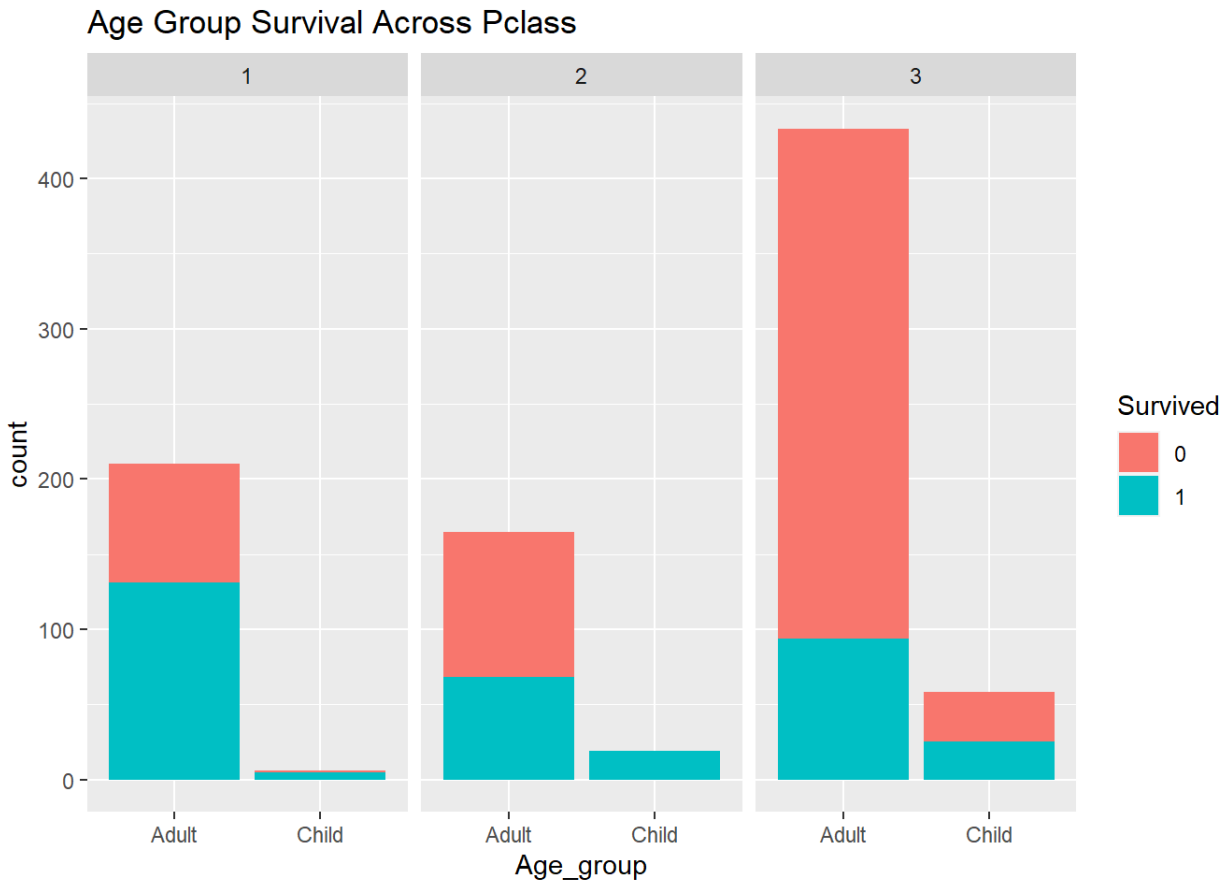
```
# create a childadult_function

chAdult_func <- function(x) {
  case_when(
    x <= 15 ~ "Child", TRUE ~ "Adult"
  )
}

Adult_Child <- NULL #create a null file

for (i in 1:nrow(train.test)) {
  Adult_Child <- c(Adult_Child, chAdult_func(train.test[i, "Age"]))
}

# create a new column and add the Adult_Child vector

train.test <- train.test %>%
  mutate(Age_group = Adult_Child)

# call summary on the new variable

table(train.test$Age_group)
```

```
##
## Adult Child
##  1194   115
```

Visualise the distribution across Pclass

```
ggplot(train.test[1:891, ], aes(Age_group, fill = Survived)) +
  geom_bar() +
  facet_wrap(~ Pclass) +
  ggtitle("Age Group Survival Across Pclass")
```

Predicting Survivability Using RandomForest

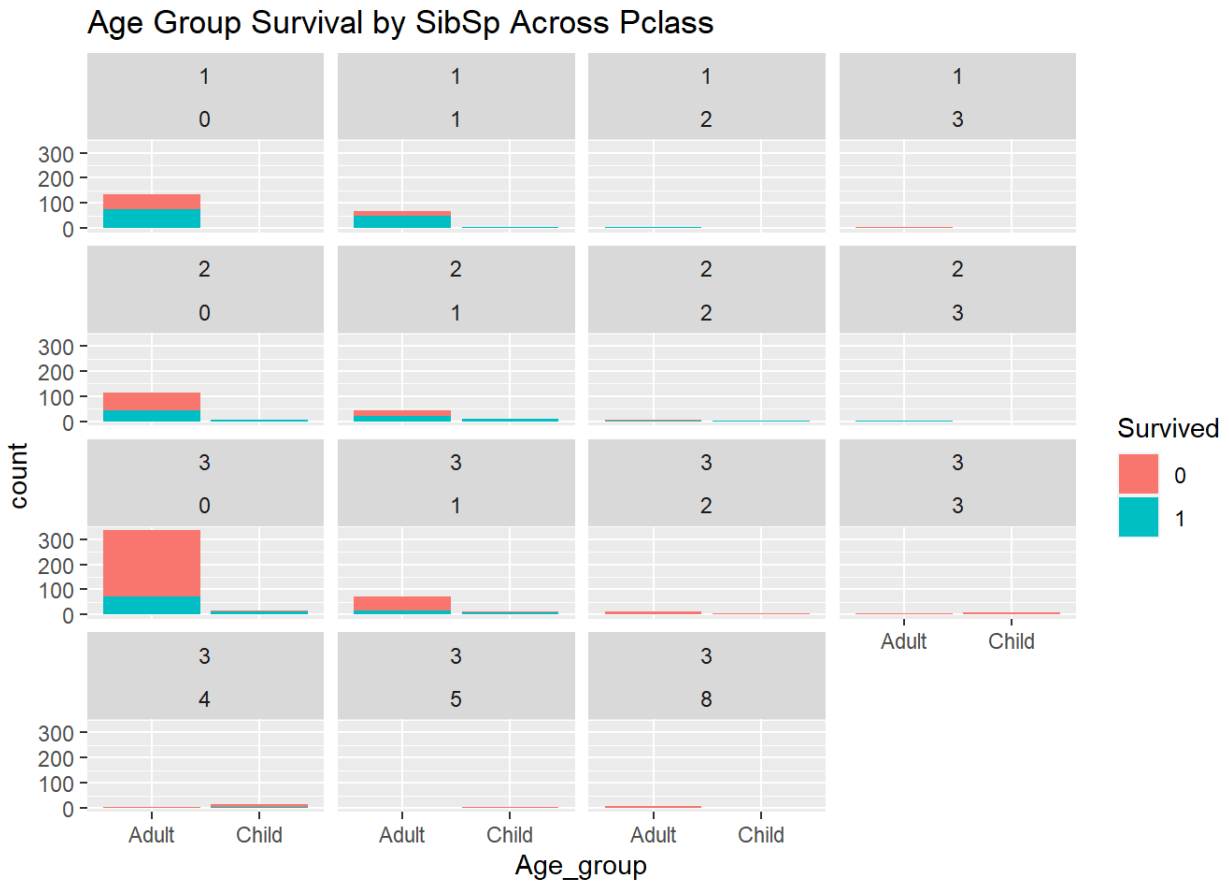


Wow! this shows us more prediction than the age.

Let's see the above chart with the Parch Var before we move on to create another variables.

```
ggplot(train.test[1:891, ], aes(Age_group, fill = Survived)) +
  geom_bar() +
  facet_wrap(~ Pclass ~ SibSp) +
  ggtitle("Age Group Survival by SibSp Across Pclass")
```

Predicting Survivability Using RandomForest



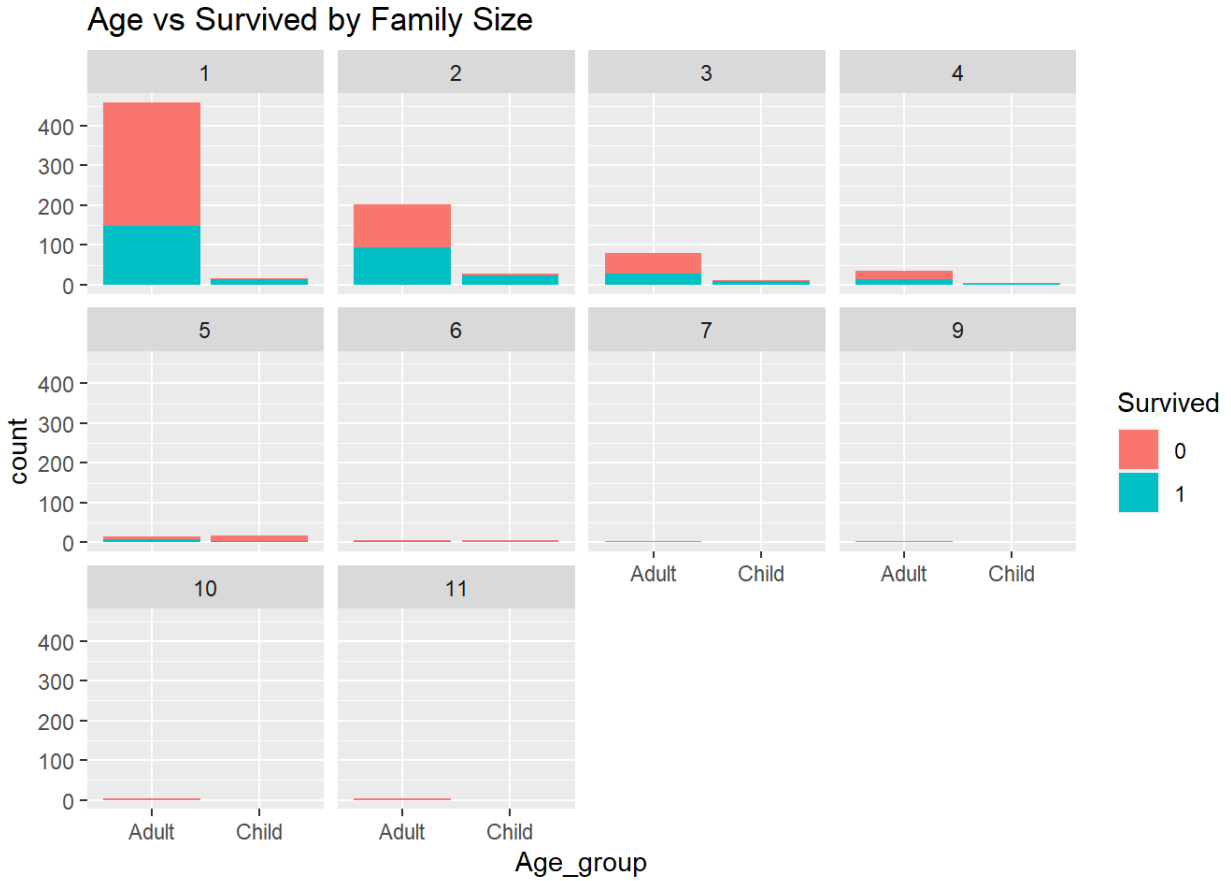
Ouch! the pattern seems to be hard to read. Since both SIbSp and Parch shows the number of dependent/spouse/siblings/parents/guardians traveling together, we will use this information to create a new **Family_Size** variable.

```
# create a vector of the sibsp and parch variable and add them together

train.temp <- c(train$SibSp, test$SibSp)
test.temp <- c(test$Parch, train$Parch)

train.test <- train.test %>%
  mutate(Family_size = as.factor(train.temp + test.temp + 1))
```

Relationship between Age group, Family Size and Survival rate

Predicting Survivability Using RandomForest

```
ggplot(train.test[1:891, ], aes(Age_group, fill = Survived)) +
  geom_bar() +
  facet_wrap(~ Family_size) +
  ggtitle("Age vs Survived by Family Size")
```



Training the RandomForest Algorithm

Now that we have looked at the varibles with the most predictive nature, we will be training our `RandomForest` algorithm with them.

Predicting Survivability Using RandomForest

```
# load the randomforest package

library(randomForest)
```

```
## randomForest 4.7-1.1
```

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

```
##
## Attaching package: 'randomForest'
```

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

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

Let's create our label data for the `randomforest` model and also our train data.

Predicting Survivability Using RandomForest

```
rf.label <- as.factor(train$Survived)

rf.train1 <- train.test[1:891, c("Sex", "Pclass", "Age_group", "Family_size")]

rf.train2 <- train.test[1:891, c("Sex", "Pclass", "Age_group")]

# set the seed to 2022

set.seed(2022)

# run the model

model <- randomForest(rf.train1, rf.label, importance = TRUE, ntree = 2022)

model2 <- randomForest(rf.train2, rf.label, importance = TRUE, ntree = 2022)

# view the model accuracy

model
```

```
##
## Call:
##  randomForest(x = rf.train1, y = rf.label, ntree = 2022, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 2022
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 18.63%
## Confusion matrix:
##     0   1 class.error
## 0 486  63   0.1147541
## 1 103 239   0.3011696
```

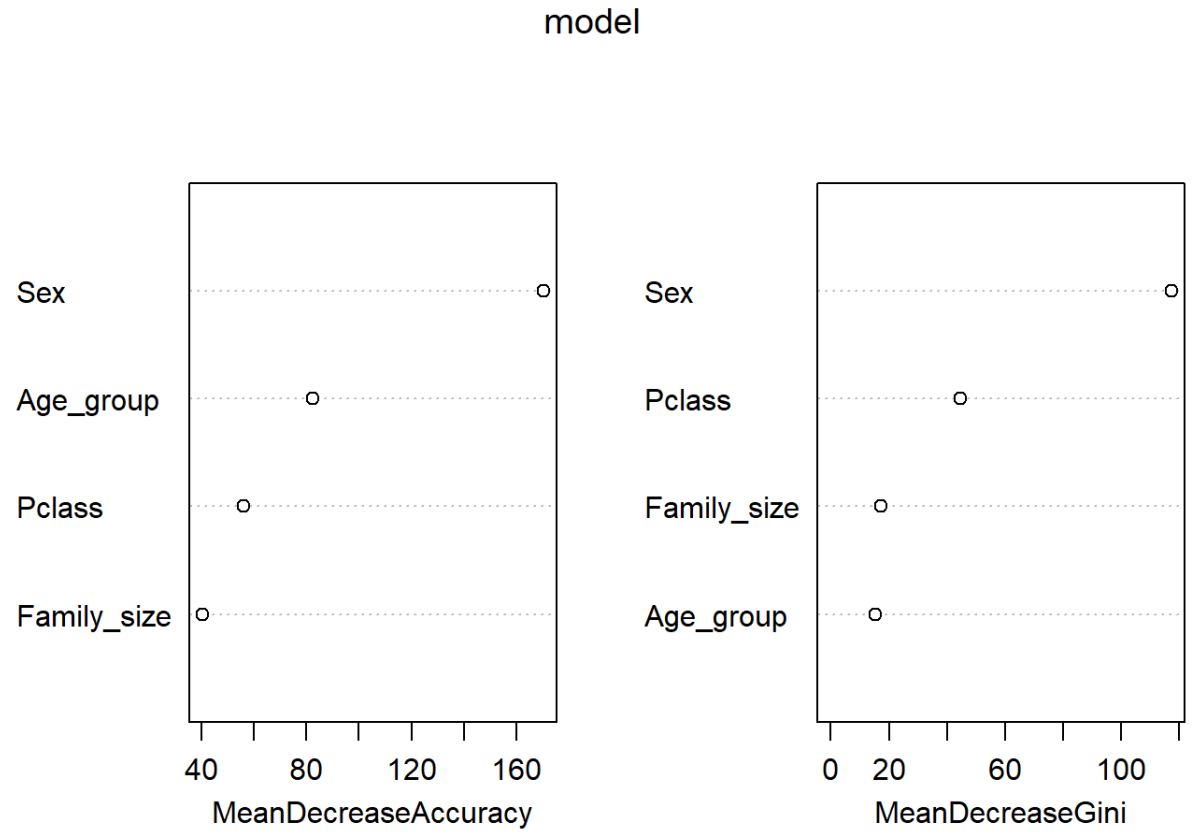
```
model2
```

Predicting Survivability Using RandomForest

```
##
## Call:
##  randomForest(x = rf.train2, y = rf.label, ntree = 2022, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 2022
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 19.75%
## Confusion matrix:
##     0   1 class.error
## 0 526  23  0.04189435
## 1 153 189  0.44736842
```

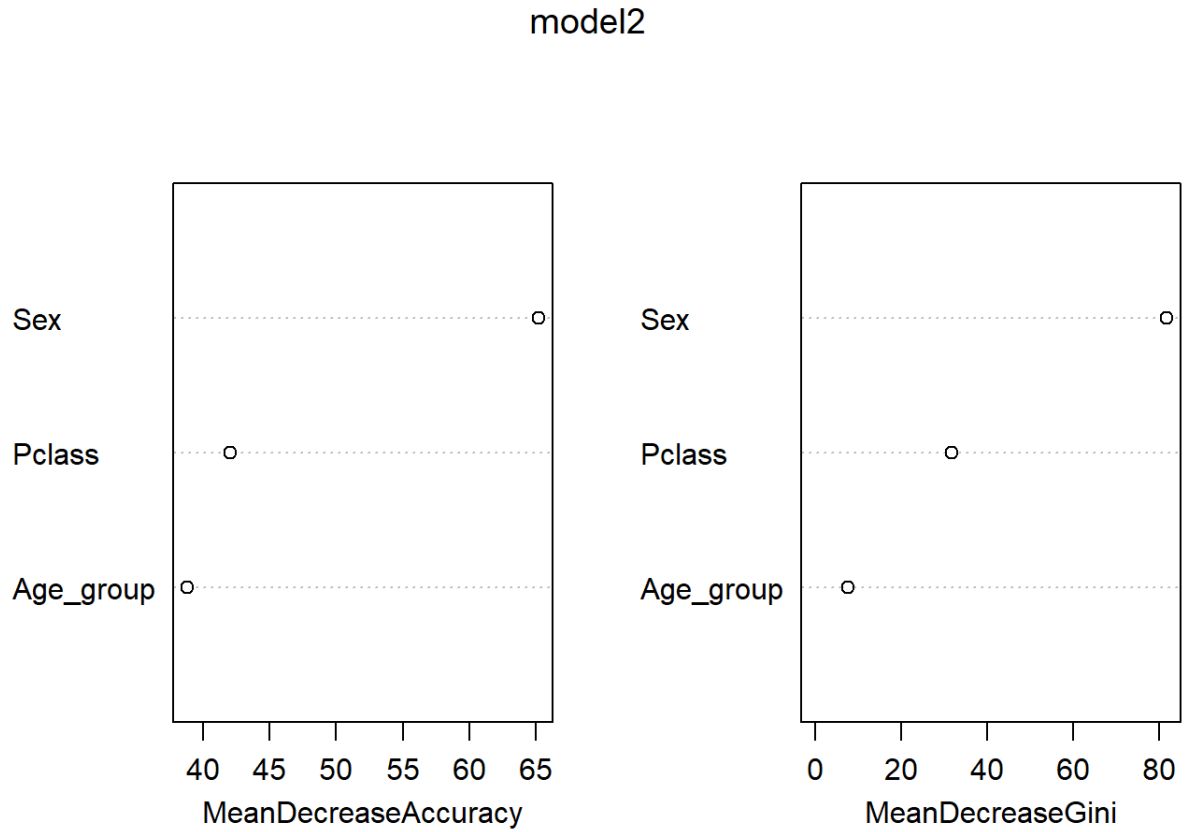
```
# view the model plot

varImpPlot(model) # this has more accuracy
```

Predicting Survivability Using RandomForest




```
varImpPlot(model2)
```

Predicting Survivability Using RandomForest



Running Our Model on the Test Data

Predicting Survivability Using RandomForest

```
set.seed(2022)

# get nrow of test data

rf.test <- train.test[892:1309, c("Sex", "Pclass", "Age_group", "Family_size")]

# run prediction on test data

prediction <- predict(model, rf.test)

table(prediction)
```

```
## prediction
##   0   1
## 268 150
```

```
# create a dataframe of PassengerId and Survived
# file to be submitted on Kaggle

submission.file <- data.frame(PassengerId = rep(892:1309), Survived = prediction)

write_csv(submission.file, "Titanic-Randomforest-prediction-2022.csv")
```

Thank you for reading. please leave me a comment. I am currently looking for collaboration on real world projects to add to my portfolio. email me write.ethereal@gmail.com (mailto:write.ethereal@gmail.com)