# STA4990 Titanic

Laura Hu

2023-03-29

```
#Load related packages
library(ggplot2)
library(ggfortify)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(dslabs)
library(caret)
```

```
## Loading required package: lattice
```

```
library(modelr)
```

```
##
## Attaching package: 'modelr'

## The following object is masked from 'package:dslabs':
##
##     heights
```

```
library(yardstick)
```

```
## For binary classification, the first factor level is assumed to be the event.
## Use the argument `event_level = "second"` to alter this as needed.

##
## Attaching package: 'yardstick'

## The following objects are masked from 'package:modelr':
##
##     mae, mape, rmse

## The following objects are masked from 'package:caret':
##
##     precision, recall, sensitivity, specificity
```

```
library(dplyr)
```

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

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

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

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

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

```
ship <- read.csv("~/Documents/School /titanic/ship_train.csv")
```

## Initial feature engineering to clean up the raw data

### Name

- First regarding the name title, we found that '**Master**' and '**Miss**' refers to kids below the age 12
- After the age of 12, most people are referred as 'Miss' '**Mrs**' and '**Mr**'
- We found that there are also titles like '**Rev**'
- There are other titles like 'Lady', 'Sir' and 'Countess' and more, we are going to name them all '**Others**'

```
for(i in 1:nrow(ship)){
  name <- ship$Name[i]
  has_master <- grepl("Master", name)
  has_rev <- grepl("Rev.", name)
  has_miss <- grepl("Miss.", name)
  has_mrs <- grepl("Mrs.", name)
  has_mr <- grepl("Mr.", name)


  if(has_master == TRUE){
    ship$Name[i] <- 'Master'
  } else if(has_rev == TRUE){
    ship$Name[i] <- 'Rev'
  } else if (has_miss == TRUE){
    ship$Name[i] <- 'Miss'
```

```
  } else if (has_mrs == TRUE){
    ship$Name[i] <- 'Mrs'
  } else if (has_mr == TRUE){
    ship$Name[i] <- 'Mr'
  } else{
    ship$Name[i] <- 'Other'
  }
}
```

#### Age

There are a lot of 'Age' missing from the data set, what we want to do is fill in all the NA values with the average age in the data set

```
ship$Age[is.na(ship$Age)] <- mean(ship$Age, na.rm = TRUE)
```

#### Ticket

- Variable names: WEP, W./C, STON/O2, SOTON/OQ, SC/AH, SOC, S.O./P.P., SC/, PC, LINE, F.C.C, CA, C, A5, 1,2,3,4,5,6,7,8,9
- Singles: SW/PP 751,PP 4348, S.P. 3464, SO/C 14885

I didn't do any feature engineering on this one, I didn't really bother.

## Remove Variables

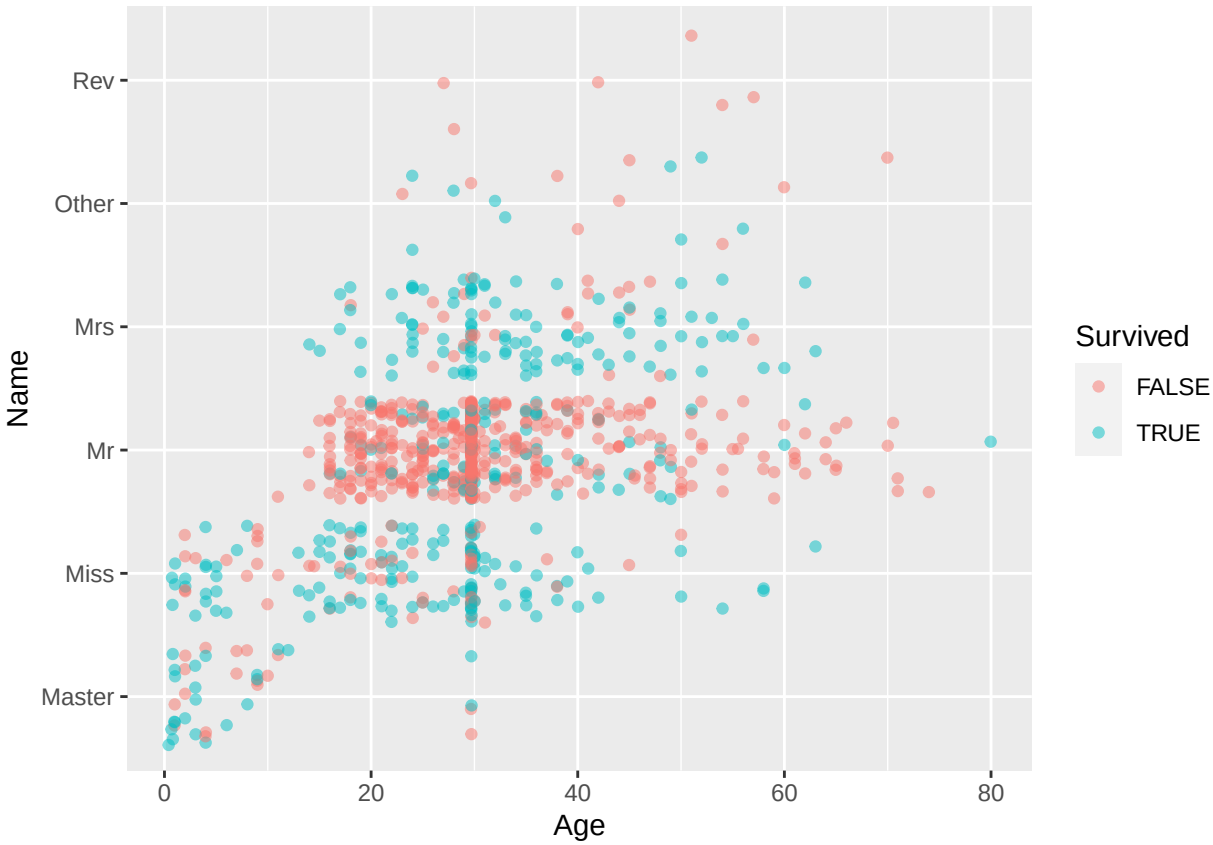
```
ship <- ship %>%
  select(-Ticket)%>%
  select(-Cabin)%>%
  select(-PassengerId) %>%
  mutate(Survived = (Survived > 0))
```

## Initial Observation

```
#ggpairs(ship, aes(color = Survived))
```

#### 'Age'

```
ggplot(ship, aes(x = Age, y = Name, color = Survived)) + geom_jitter(alpha = 0.5)
```

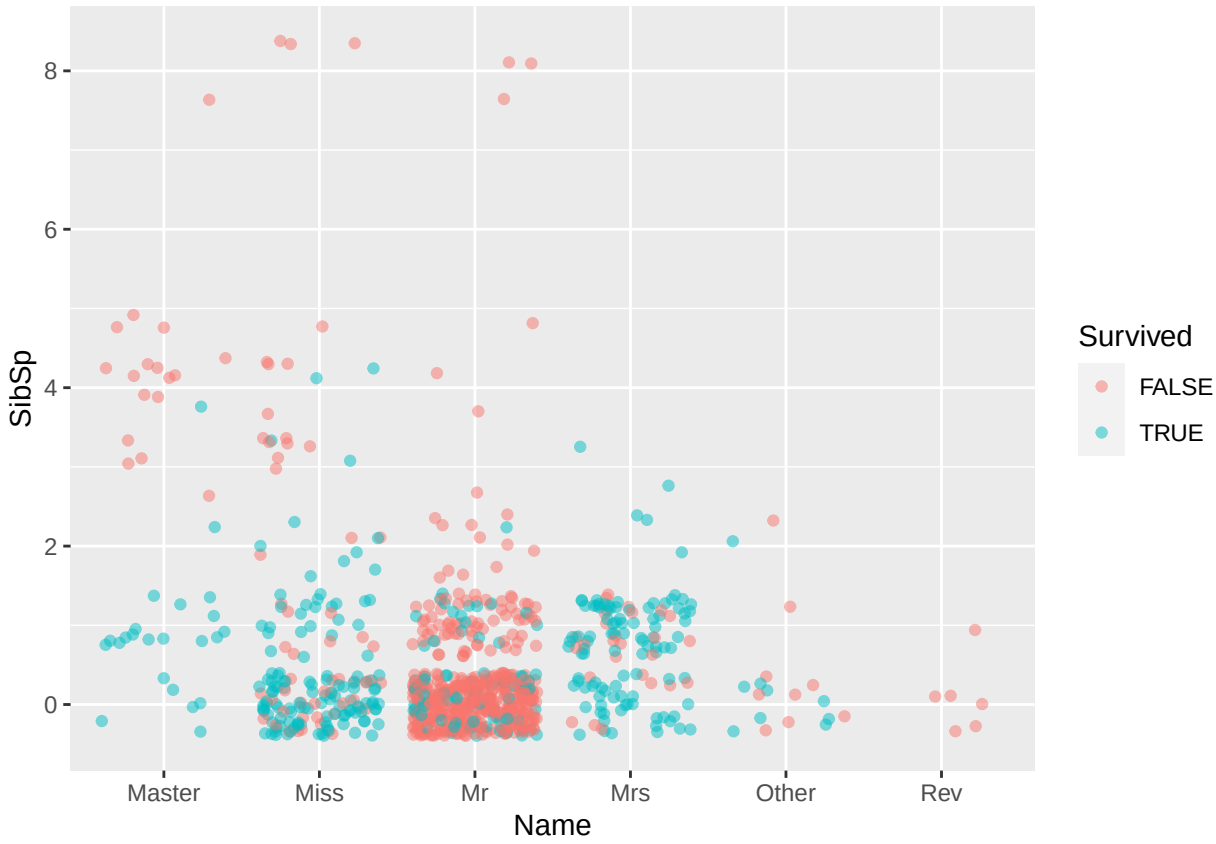



The age cutoff for 'Master' is around 12, we should probably do further investigation on the variable 'Miss'

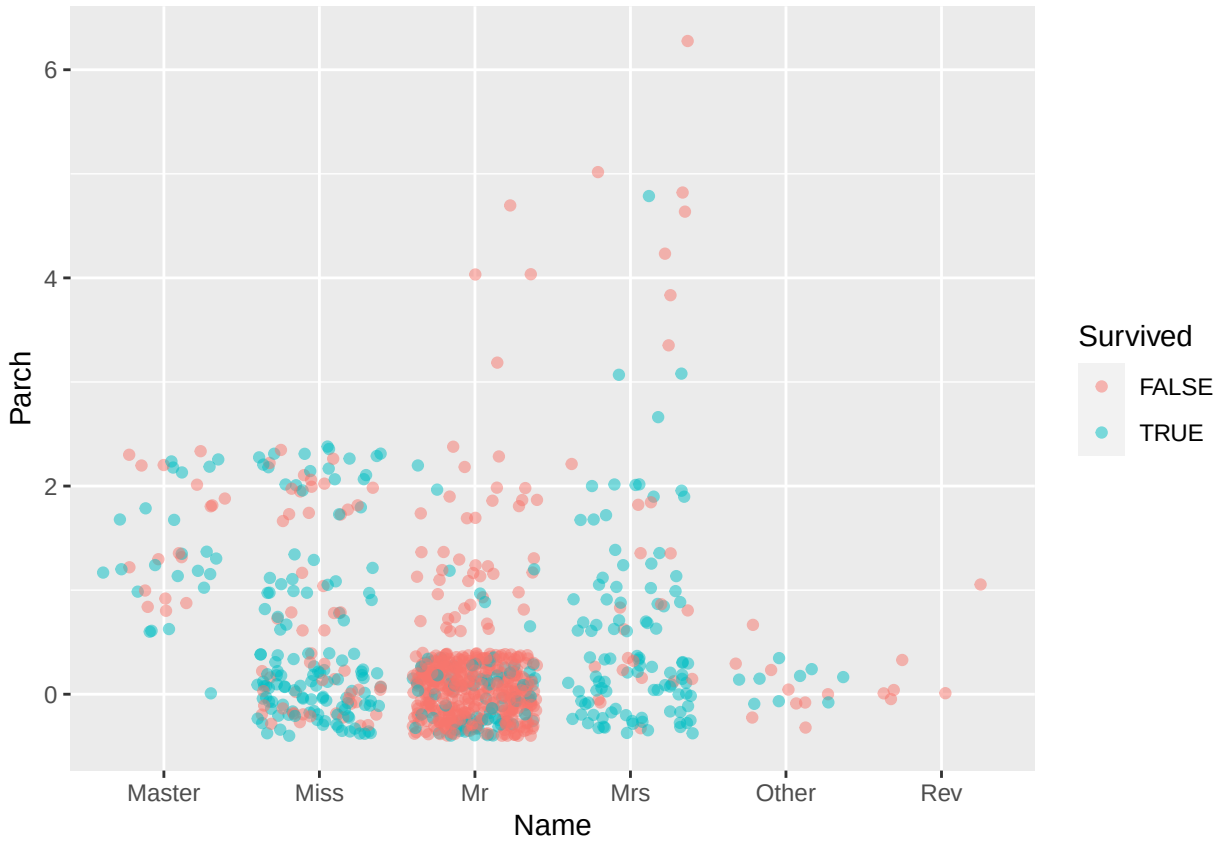
**'Name'**

```
ggplot(ship, aes(x = Name, y = Pclass, color = Survived)) + geom_jitter(alpha = 0.5)
```

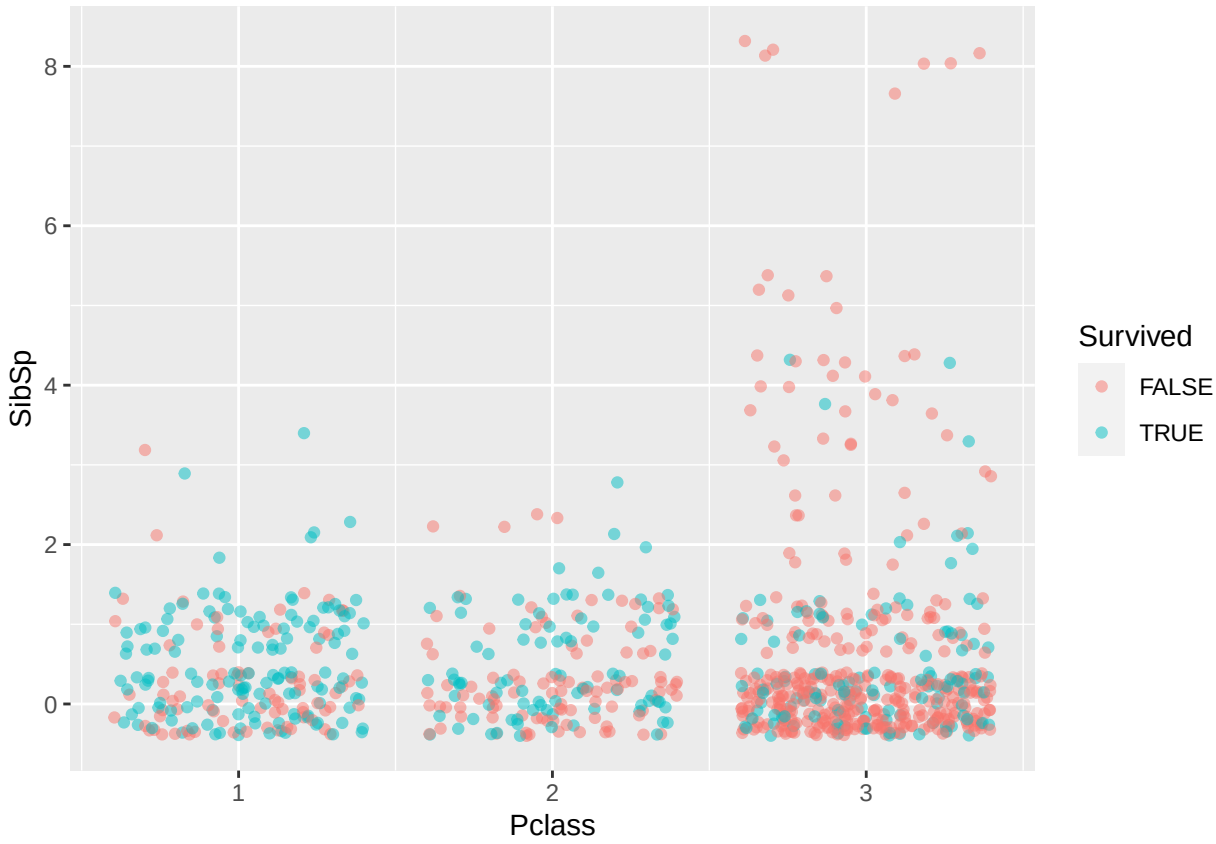

```
ggplot(ship, aes(x = Name, y = SibSp, color = Survived)) + geom_jitter(alpha = 0.5)
```




```
ggplot(ship, aes(x = Name, y = Parch, color = Survived)) + geom_jitter(alpha = 0.5)
```




```
ggplot(ship, aes(x = Pclass, y = SibSp, color = Survived)) + geom_jitter(alpha = 0.5)
```



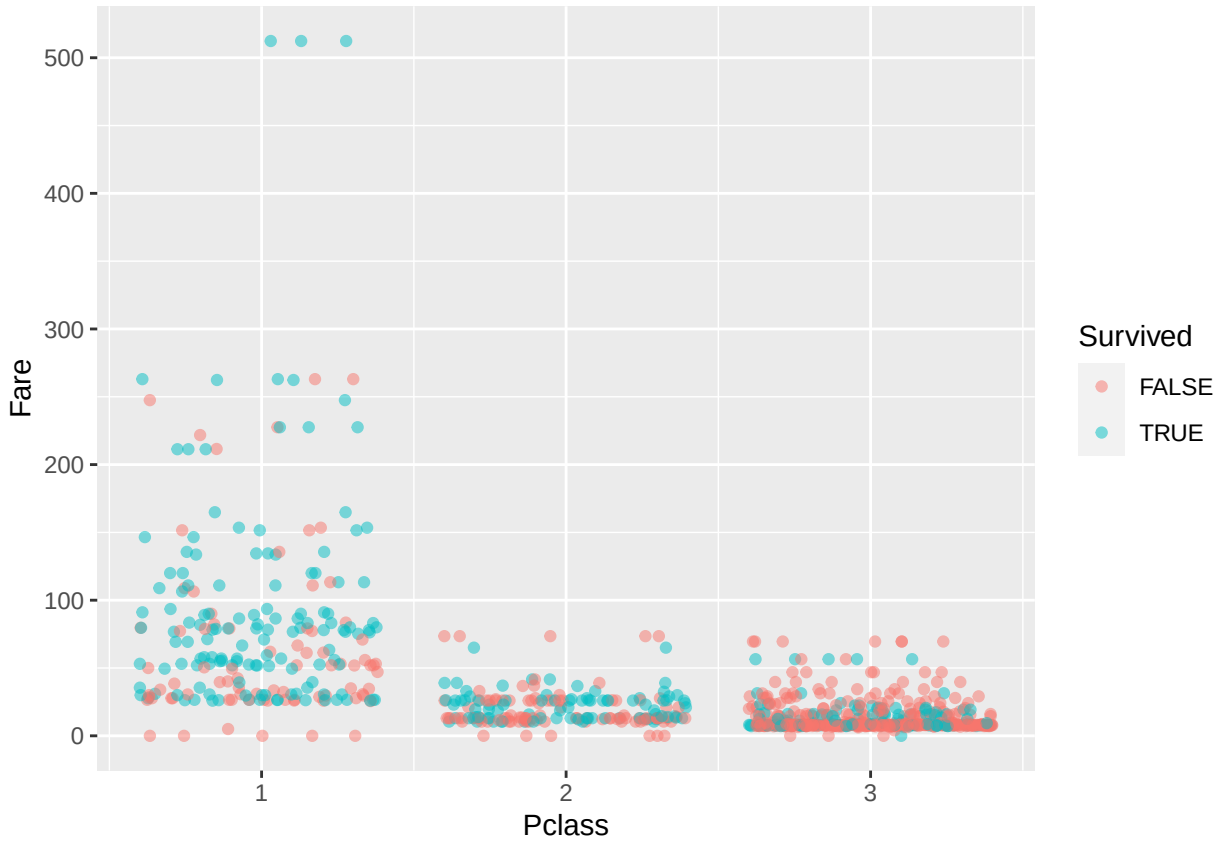

```
ggplot(ship, aes(x = Pclass, y = Parch, color = Survived)) + geom_jitter(alpha = 0.5)
```


'Master', 'Miss' and 'Mrs' in class 1,2 with less than 3 siblings & Parch are more likely to survive.

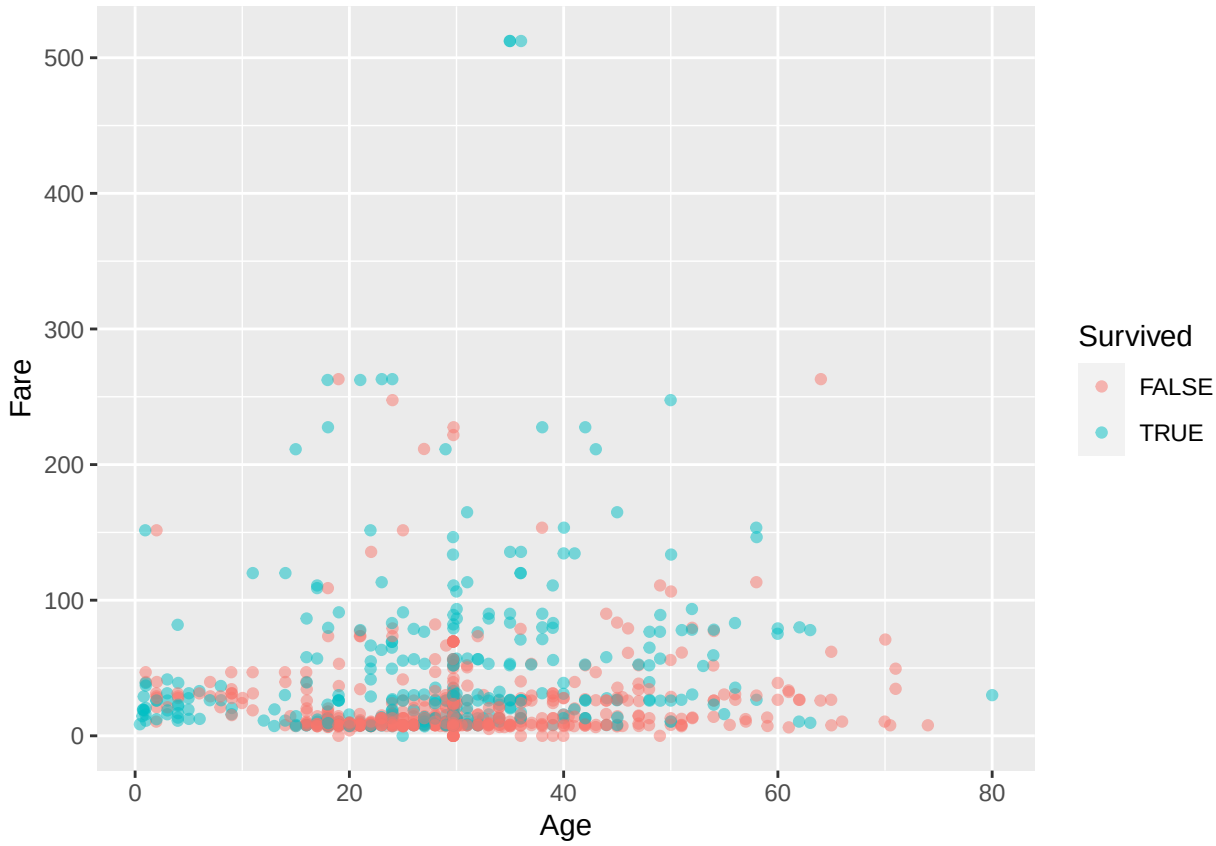
Additionally We can see that almost all class 1,2 didn't bring more than 2 siblings. So Class 1,2 implies SibSp $< 3$ & Parch $< 3$

**'Fare'**

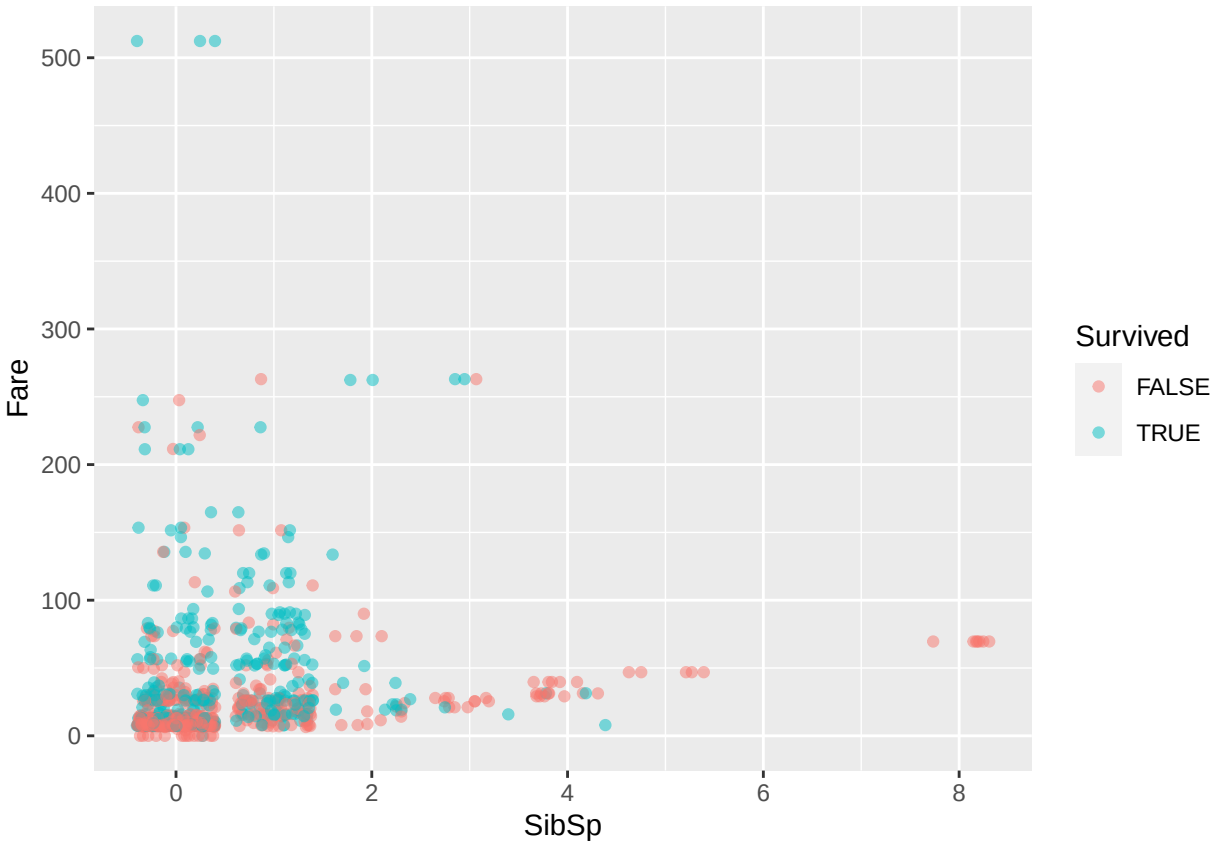
```
ggplot(ship, aes(x = Pclass, y = Fare, color = Survived)) + geom_jitter(alpha = 0.5)
```




```
ggplot(ship, aes(x = Age, y = Fare, color = Survived)) + geom_jitter(alpha = 0.5)
```




```
ggplot(ship, aes(x = SibSp, y = Fare, color = Survived)) + geom_jitter(alpha = 0.5)
```




People that are already in class 1 that paid more than 50 are more likely to survive.

## Additional Feature Engineering

### 'Companion'

```
ship$Companion <- ship$SibSp + ship$Parch
```

### 'Rich'

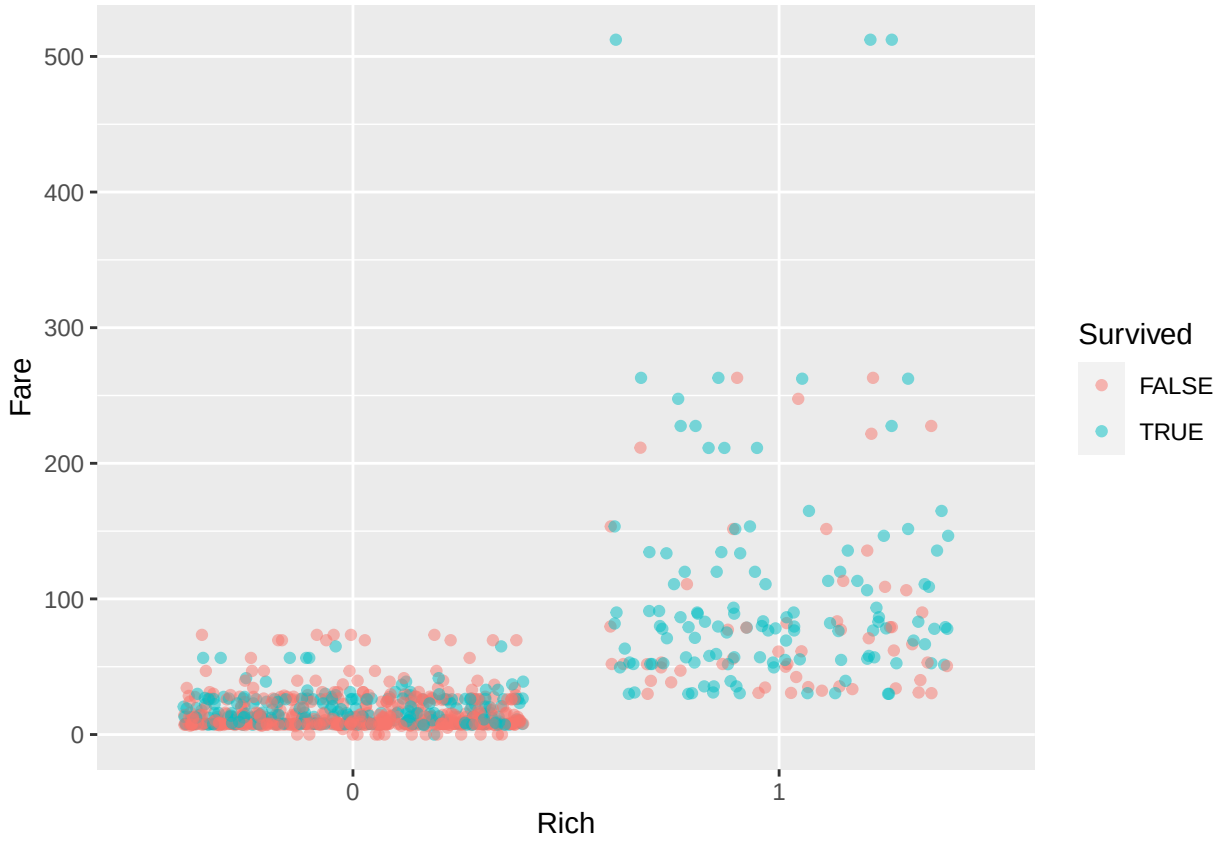
Add variables called 'Rich' for people that are in class 1 and paid more than $50

Add variable called 'Prime' for 'Masters', 'Miss' and 'Mrs' in class 1,2

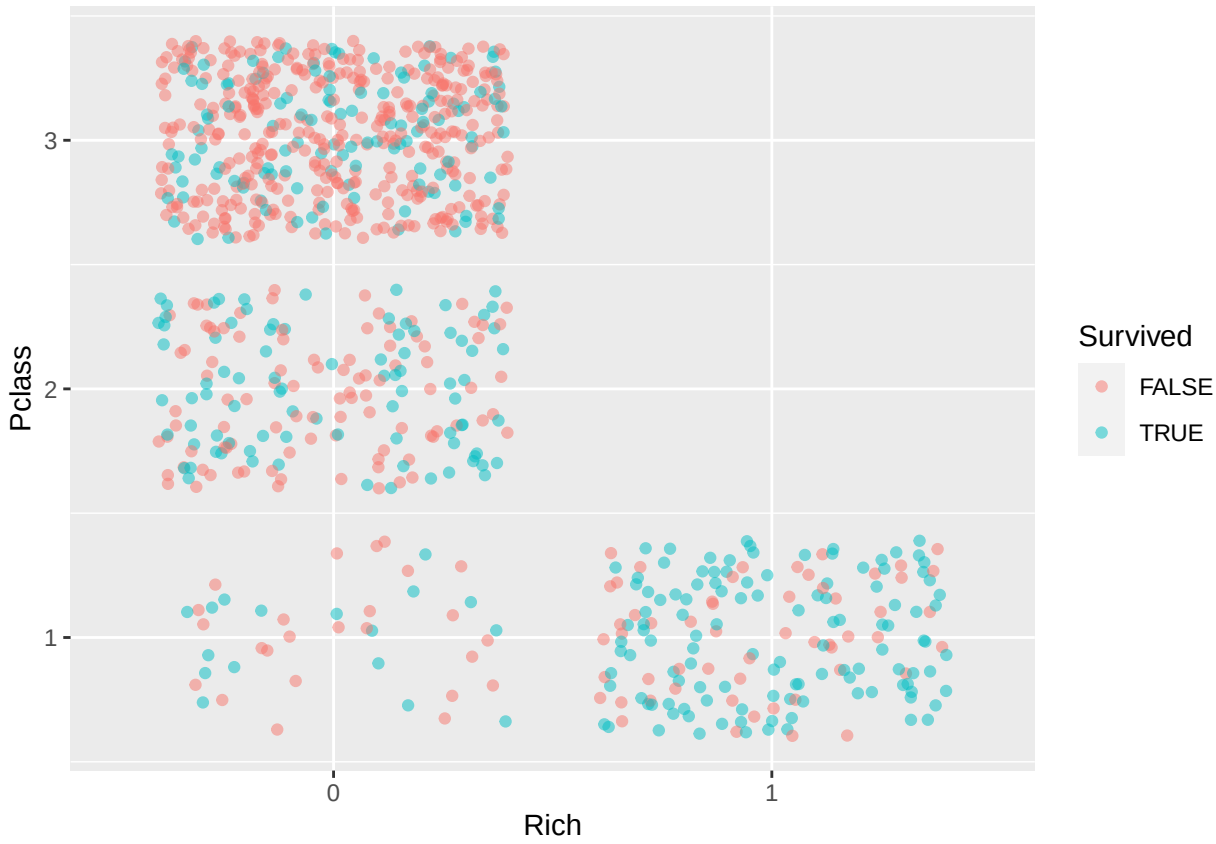
```
ship$Rich <- as.factor(ifelse(ship$Pclass < 2 & ship$Fare < 30, 0,
                              ifelse(ship$Pclass < 2 & ship$Fare< 600, 1,
                                     ifelse(ship$Pclass < 4 & ship$Fare < 600, 0))))
```

**Checking to see if it is significant**

```
ggplot(ship, aes(x = Rich, y = Fare, color = Survived)) + geom_jitter(alpha = 0.5)
```




```
ggplot(ship, aes(x = Rich, y = Pclass, color = Survived)) + geom_jitter(alpha = 0.5)
```




### 'Prime'

'Master', 'Miss' and 'Mrs' in class 1,2 with less than 3 siblings & Parch are more likely to survive.

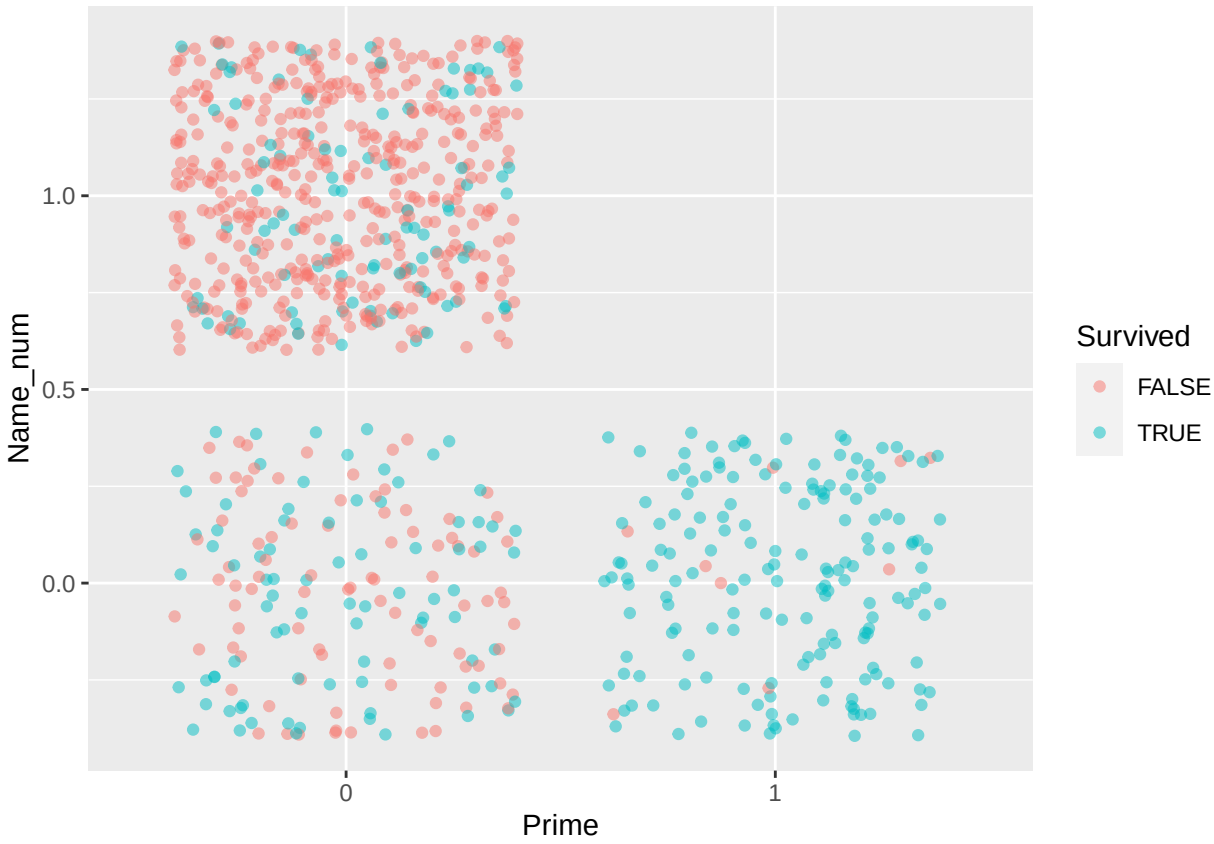
```
#First add a new varaible that converts all the names into numeric titles

ship$Name_num <- c(Master = 0, Miss = 0, Mrs = 0, Mr = 1, Rev = 1, Other = 1)[ship$Name]

#Then Use the new Variable to create the varaible 'Prime'
ship$Prime <- as.factor(ifelse(ship$Name_num < 1 & ship$Pclass < 3, 1,
                      ifelse(ship$Name_num < 2 & ship$Pclass < 4, 0)))
```

#### Graph to check significance

```
ggplot(ship, aes(x = Prime, y = Name_num, color = Survived)) + geom_jitter(alpha = 0.5)
```




### Split data into training set and test set

```
#Create partition (0.7 to 0.3)
set.seed(165)
trainIndex <- createDataPartition(ship$Survived, p = 0.7,
                                  list = FALSE,
                                  time = 1)

ship$train <- FALSE
ship$train[trainIndex] <- TRUE

ship_train <- ship %>% filter(train == TRUE)
ship_test_1 <- ship%>%filter(train == FALSE)
```

## Fit Models

### QDA

```
fit_qda <- train(as.factor(Survived) ~ Prime + Rich + Age + Companion,
                 data = ship_train,
```

```
                  method = "qda",
                  trace = FALSE)
```

### LDA

```
fit_lda <- train(as.factor(Survived) ~ Prime + Rich + Age + Companion,
                  data = ship_train,
                  method = "lda",
                  trace = FALSE)
```

### KNN

```
fit_knn <- train(as.factor(Survived) ~ Prime + Rich + Age + Companion,
                  data = ship_train,
                  method = "knn")
```

### Naive Bayes

```
fit_nb <- train(as.factor(Survived) ~ Prime + Rich + Age + Companion,
                  data = ship_train,
                  method = "naive_bayes")
```

### Logistic Regression

```
fit_log <- train(as.factor(Survived) ~ Prime + Rich + Age + Companion,
                  data = ship_train,
                  method = "glmnet",
                  family = "multinomial",
                  trace = FALSE)
```

## Checking accuracy by fitting into the test set

```
ship_test_1 <- ship_test_1 %>%
  mutate(Survived_qda = predict(fit_qda, newdata = ., type = "raw"))%>%
  mutate(qda_prob = predict(fit_qda, newdata = ., type = "prob")$'TRUE')
```

```
ship_test_1 <- ship_test_1 %>%
  mutate(Survived_lda = predict(fit_lda, newdata = ., type = "raw"))%>%
  mutate(lda_prob = predict(fit_lda, newdata = ., type = "prob")$'TRUE')
```

```
ship_test_1 <- ship_test_1 %>%
  mutate(Survived_knn = predict(fit_knn, newdata = ., type = "raw"))%>%
  mutate(knn_prob = predict(fit_qda, newdata = ., type = "prob")$'TRUE')
```

```
ship_test_1 <- ship_test_1 %>%
  mutate(Survived_nb = predict(fit_nb, newdata = ., type = "raw"))%>%
  mutate(nb_prob = predict(fit_nb, newdata = ., type = "prob")$'TRUE')
```

```
ship_test_1 <- ship_test_1 %>%
  mutate(Survived_log = predict(fit_log, newdata = ., type = "raw"))
#%>%mutate(log_prob = predict(fit_log, newdata = ., type = "prob")$'TRUE')
```

Accuracy

```
ship_test_1 %>% accuracy(truth = as.factor(Survived),
                         estimate = Survived_qda)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.786
```

```
ship_test_1 %>% accuracy(truth = as.factor(Survived),
                         estimate = Survived_lda)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.808
```

```
ship_test_1 %>% accuracy(truth = as.factor(Survived),
                         estimate = Survived_knn)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.729
```

```
ship_test_1 %>% accuracy(truth = as.factor(Survived),
                         estimate = Survived_nb)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.782
```

```
ship_test_1 %>% accuracy(truth = as.factor(Survived),
                         estimate = Survived_log)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.808
```

### Fit to Test Set

```
ship_test <- read.csv("~/Documents/School /titanic/ship_test.csv")
```

```
for(i in 1:nrow(ship_test)){
  name <- ship_test$Name[i]
  has_master <- grepl("Master", name)
  has_rev <- grepl("Rev.", name)
  has_miss <- grepl("Miss.", name)
  has_mrs <- grepl("Mrs.", name)
  has_mr <- grepl("Mr.", name)


  if(has_master == TRUE){
    ship_test$Name[i] <- 'Master'
  } else if(has_rev == TRUE){
    ship_test$Name[i] <- 'Rev'
  } else if (has_miss == TRUE){
    ship_test$Name[i] <- 'Miss'
  } else if (has_mrs == TRUE){
    ship_test$Name[i] <- 'Mrs'
  } else if (has_mr == TRUE){
    ship_test$Name[i] <- 'Mr'
  } else{
    ship_test$Name[i] <- 'Other'
  }
}
```

```
ship_test$Age[is.na(ship_test$Age)] <- mean(ship_test$Age, na.rm = TRUE)

ship_test$Fare[is.na(ship_test$Fare)] <- mean(ship_test$Fare, na.rm = TRUE)
```

```
ship_test$Companion <- ship_test$SibSp + ship_test$Parch
```

```
ship_test$Rich <- as.factor(ifelse(ship_test$Pclass < 2 & ship_test$Fare < 30, 0,
                                   ifelse(ship_test$Pclass < 2 & ship_test$Fare< 600, 1,
                                          ifelse(ship_test$Pclass < 4 & ship_test$Fare < 600, 0))))
```

```
#First add a new varaible that converts all the names into numeric titles

ship_test$Name_num <- c(Master = 0, Miss = 0, Mrs = 0, Mr = 1, Rev = 1, Other = 1)[ship_test$Name]

#Then Use the new Variable to create the varaible 'Prime'
ship_test$Prime <- as.factor(ifelse(ship_test$Name_num < 1 & ship_test$Pclass < 3, 1,
                        ifelse(ship_test$Name_num < 2 & ship_test$Pclass < 4, 0)))
```

```
ship_test <- ship_test %>%
  mutate(Survived_log = predict(fit_log, newdata = ., type = "raw"))
```

```
ship_test <- ship_test %>%
  mutate(Survived = (Survived_log == TRUE))%>%
  select(-Ticket)%>%
  select(-Cabin)%>%
  select(-Pclass)%>%
  select(-Name)%>%
  select(-Sex)%>%
  select(-Age)%>%
  select(-SibSp)%>%
  select(-Parch)%>%
  select(-Fare)%>%
  select(-Embarked)%>%
  select(-Companion)%>%
  select(-Rich)%>%
  select(-Prime)%>%
  select(-Name_num)
```

```
ship_test$Survived <- as.integer(as.logical(ship_test$Survived))
ship_test <- ship_test %>%
  select(-Survived_log)
```

```
write.csv(ship_test, "~/Documents/School /titanic/titanic_pred_1.csv", row.names=FALSE)
```