CPTR330 – Final Project

Ryan Rabello

May 31, 2017

1 Titanic Data

1.1 Importing data.

First things first we need to import our data to take a look at it.

```
# train <- read.csv("https://cs.wallawalla.edu/~carmpr/cptr330/titanic/train.csv", stringsAsFactors = T
# test <- read.csv("https://cs.wallawalla.edu/~carmpr/cptr330/titanic/test.csv", stringsAsFactors = TRU
# test_final <- read.csv("https://cs.wallawalla.edu/~carmpr/cptr330/titanic/test_final.csv", stringsAsF
# test_results <- read.csv("https://cs.wallawalla.edu/~carmpr/cptr330/titanic/test_results.csv", string
# test_results_final <- read.csv("https://cs.wallawalla.edu/~carmpr/cptr330/titanic/test_results_final.

# For testing purposes.
train <- read.csv("./data/train.csv", stringsAsFactors = TRUE)
test_final <- read.csv("./data/test.csv", stringsAsFactors = TRUE)
test_results_final <- read.csv("./data/gender_submission.csv", stringsAsFactors = TRUE)
str(train)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : Factor w/ 891 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520 629 417 58
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : Factor w/ 681 levels "110152","110413",..: 524 597 670 50 473 276 86 396 345 133 ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : Factor w/ 148 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 1 1 ...
##  $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

1.2 Explaining data

The following list briefly explains what each variable is and what type (categorical or regression) it is.

- **Survived** A Boolean (0 or 1) indicating the survival of this particular passenger.
- **pclass** Ticket class 1 = 1st, 2 = 2nd, 3 = 3rd (A numerical representation of the)
- **sex** Gender of the individule.
- **Age** Age in floating point years.
- **sibsp** # of siblings / spouses aboard the Titanic
- **parch** # of parents / children aboard the Titanic

- **ticket** Ticket number
- **fare** Passenger fare (the cost of the ticket).
- **cabin** Cabin number (String ex “A15” and “B12”)

- **embarked** Port of Embarkation C = Cherbourg, Q = Queenstown, S = Southampton

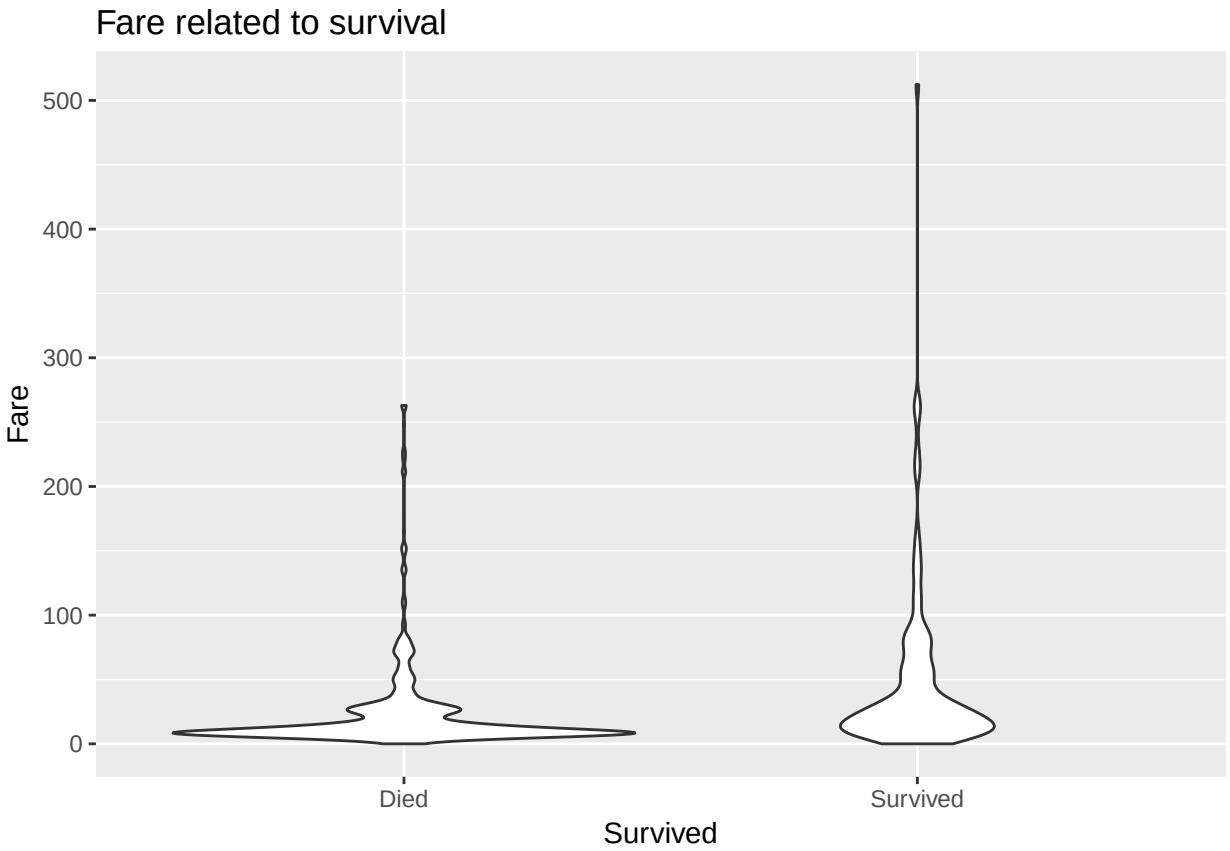
Because `PassengerId`, `Name` and `ticket` are mostly unique we they will not be useful in any machine learning calculation and are therefore nullified. I'm also changing some of the names of vairables so that everything works okay. Lastly I'm simplifying the cabin variable to just the first letter of the cabin. I believe this corresponds to the floor of the cabin.

```
process <- function (titanic) {
  if ("Survived" %in% names(titanic)){
    titanic$Survived <- factor(titanic$Survived, c(0, 1), c("Died", "Survived"))
  }
  titanic$PassengerId <- NULL
  titanic$Name <- NULL
  titanic$Ticket <- NULL
  levels(titanic$Embarked) <- c("Missing", "Cherbourg", "Queenstown",  "Southampton")

  # Plot only cabins with more than 3 people.
  cabinLev <- levels(titanic$Cabin)
  cabinLev <- substr(cabinLev, 1,1)
  levels(titanic$Cabin) <- cabinLev
  levels(titanic$Cabin) <- c("?","A", "B", "C", "D", "E", "F", "G", "T")
  return(titanic)
}
train_titanic <- process(train)
test_titanic <- process(test_final)
test_titanic_labels <- factor(test_results_final$Survived, c(0, 1), c("Died", "Survived"))

test_titanic_all <- cbind(Survived = test_titanic_labels, test_titanic)

titanic <- rbind(train_titanic, test_titanic_all)
```

1.3 Analyzing some data.

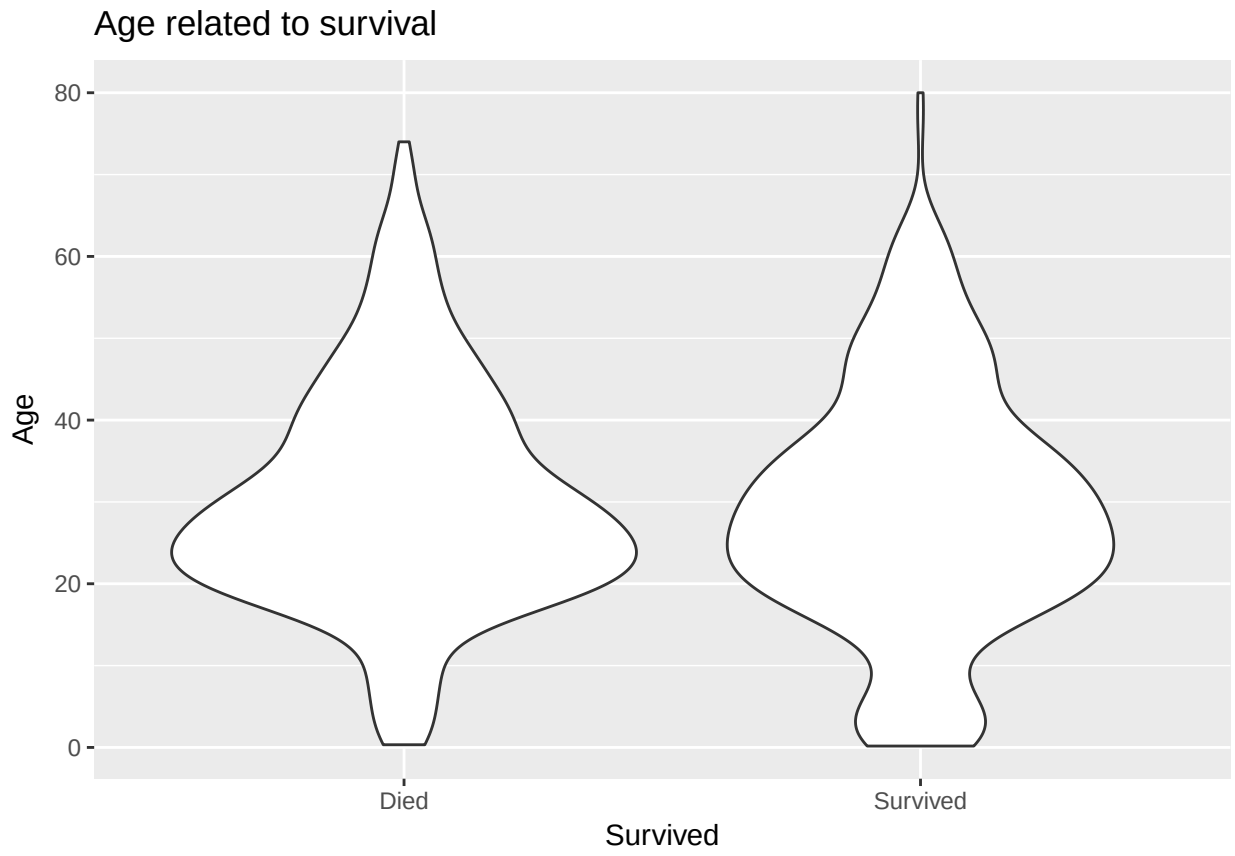
```
# Dynamically load `ggplot2`
package <- 'ggplot2'
if(package %in% rownames(installed.packages())) {
  do.call('library', list(package))
} else {
  install.packages(package)
  do.call("library", list(package))
}
package <- 'ggthemes'
if(package %in% rownames(installed.packages())) {
  do.call('library', list(package))
} else {
  install.packages(package)
  do.call("library", list(package))
}


#ggplot <- function(...) ggplot2::ggplot(...) + scale_color_fivethirtyeight("cyl") + theme_fivethirtyei
# ggplot <- function(...) ggplot2::ggplot(...) + theme_hc() + scale_colour_hc()
ggplot(titanic, aes(Survived, Fare)) + geom_violin(scale = "area") + ggtitle("Fare related to survival"
```

```
## Warning: Removed 1 rows containing non-finite values (stat_ydensity).
```

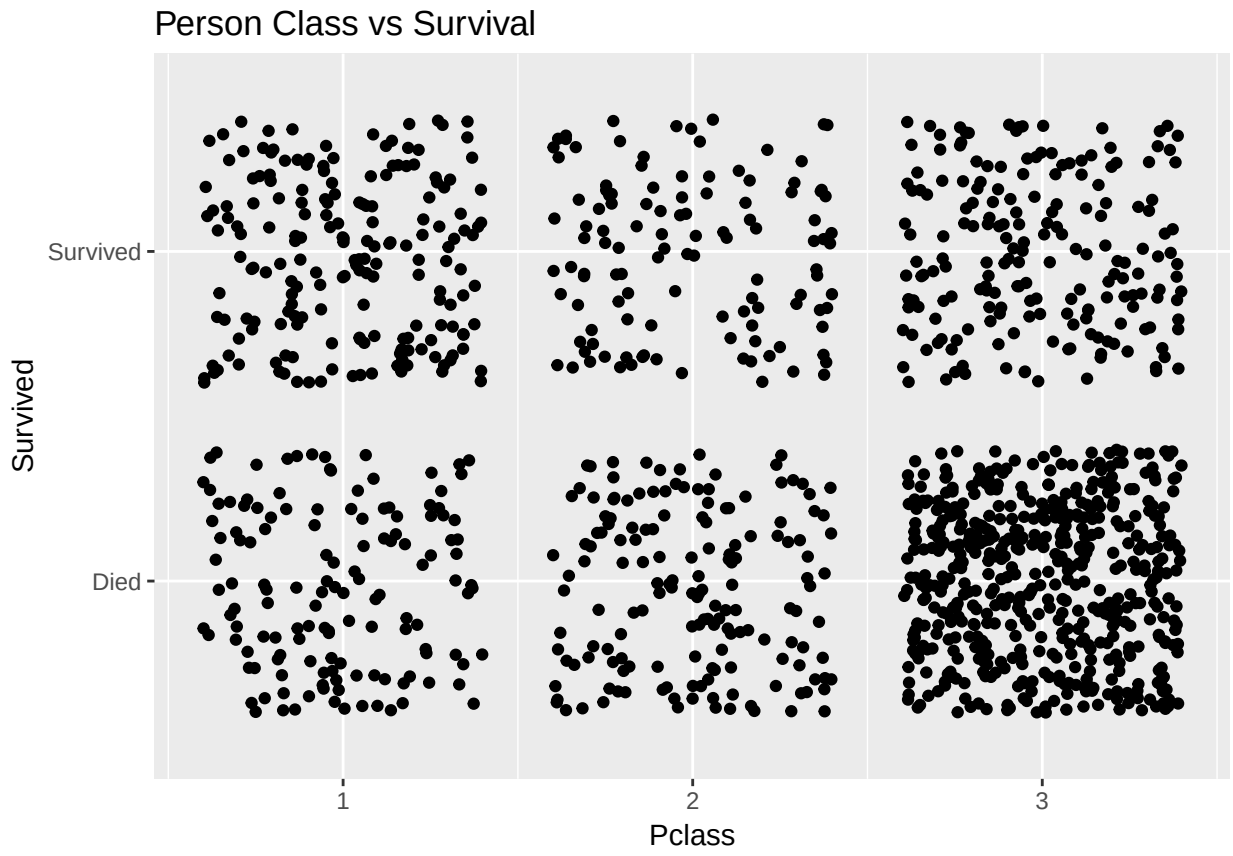



```
ggplot(titanic, aes(Survived, Age)) + geom_violin(scale = "area") + ggtitle("Age related to survival")
```

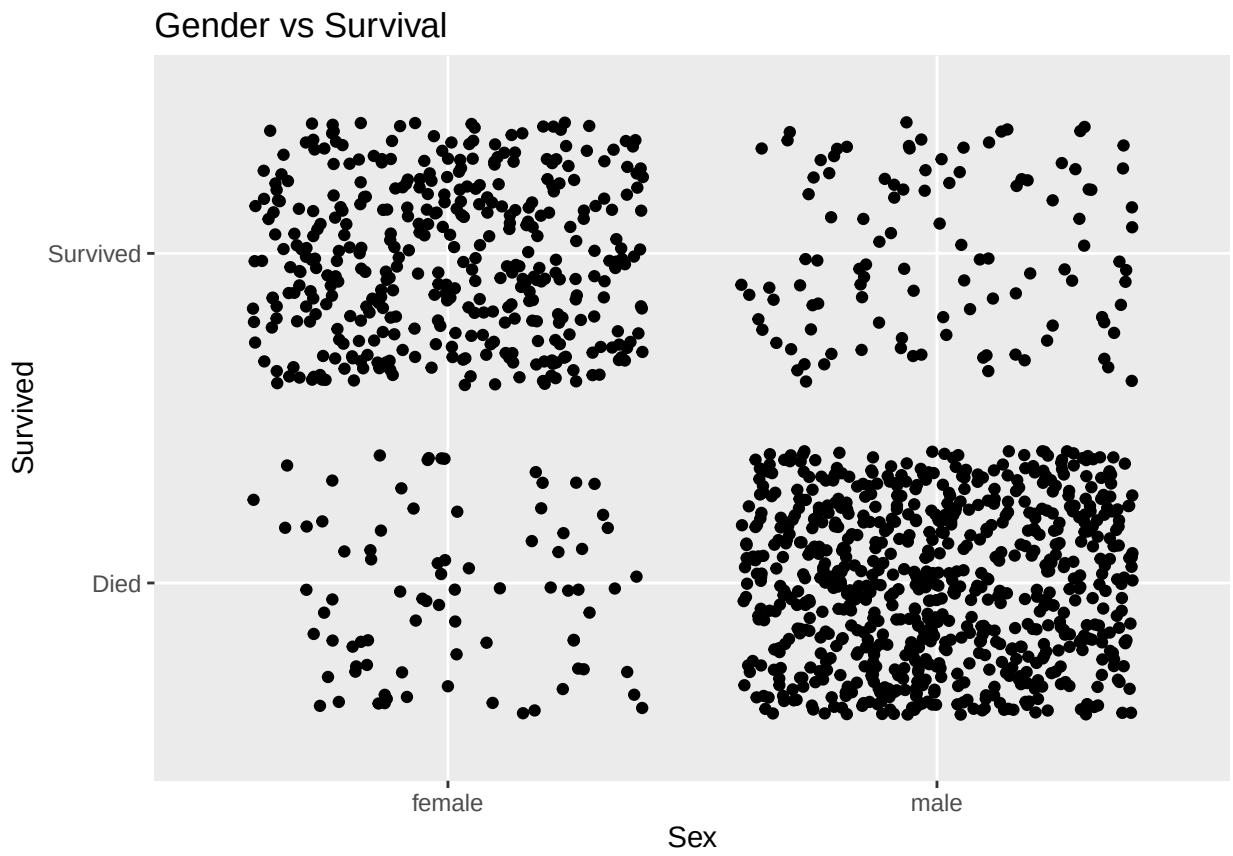
```
## Warning: Removed 263 rows containing non-finite values (stat_ydensity).
```



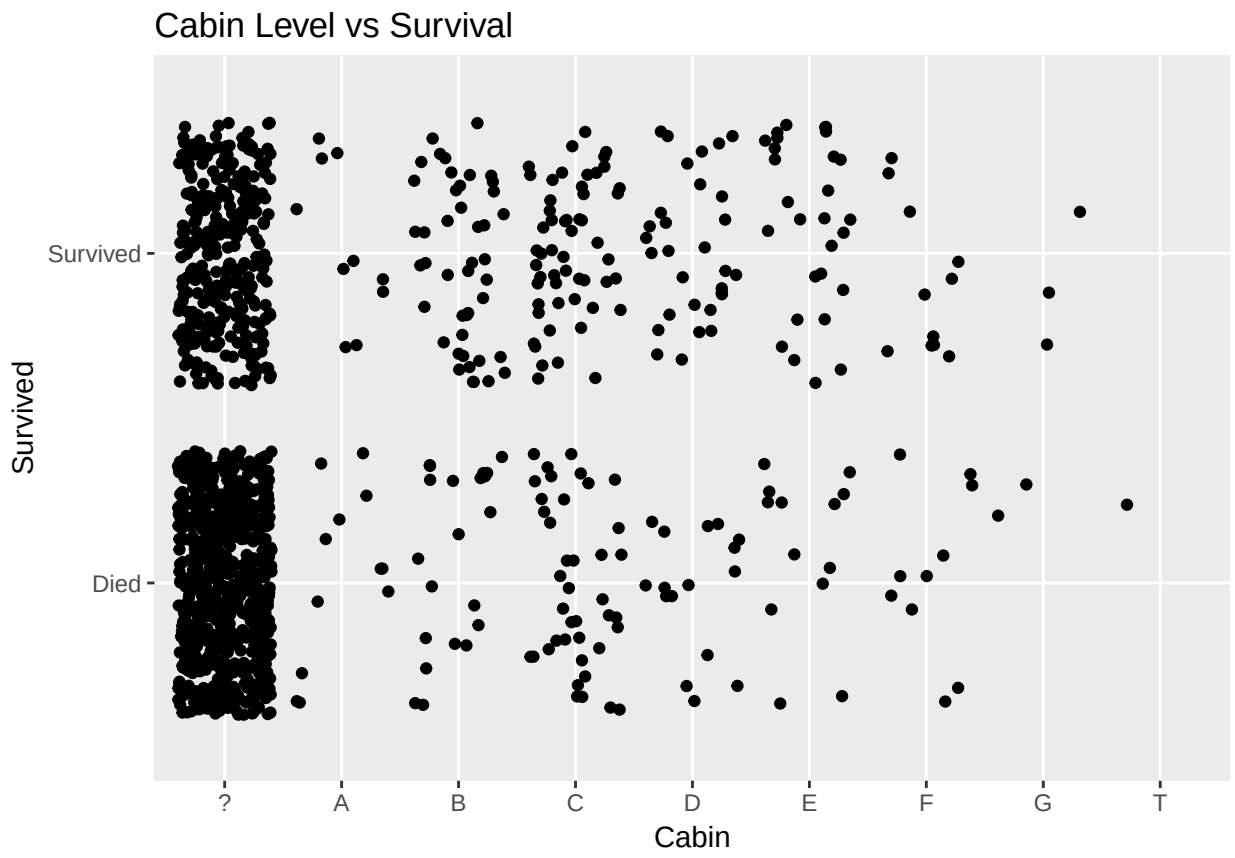

```
# Plot a jitter graph
ggplot(titanic, aes(Pclass, Survived)) + geom_jitter() + ggtitle("Person Class vs Survival")
```



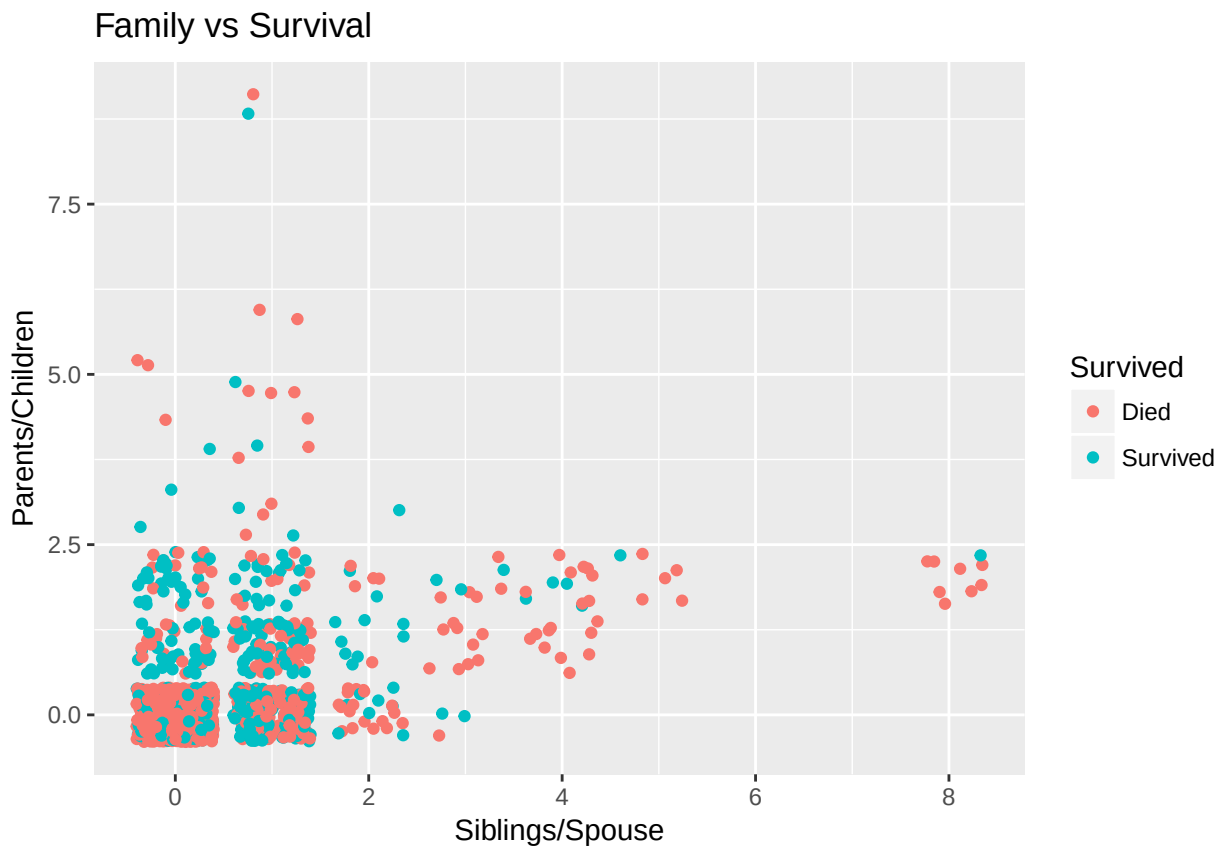

```
ggplot(titanic, aes(Sex, Survived)) + geom_jitter() + ggtitle("Gender vs Survival")
```




```
ggplot(titanic, aes(Cabin, Survived)) + geom_jitter() + ggtitle("Cabin Level vs Survival")
```




```
ggplot(titanic, aes(x=SibSp, y=Parch, colour = Survived)) + geom_jitter() + ggtitle("Family vs Survival
```



2 Descision Trees

2.1 Training

```
# Dynamically load `C50`
package <- 'C50'
if(package %in% rownames(installed.packages())) {
  do.call('library', list(package))
} else {
  install.packages(package)
  do.call("library", list(package))
}

# Train the model.
tree <- C5.0(train_titanic[,-1], train_titanic$`Survived`)

summary(tree)
```

```
##
## Call:
## C5.0.default(x = train_titanic[, -1], y = train_titanic$Survived)
##
##
## C5.0 [Release 2.07 GPL Edition]      Thu Jun 01 11:20:58 2017
```

```
## ------------------------------
##
## Class specified by attribute `outcome'
##
## Read 891 cases (9 attributes) from undefined.data
##
## Decision tree:
##
## Sex = male:
## :...Age > 9: Died (536.2/88.9)
## :   Age <= 9:
## :   :...SibSp > 2: Died (14.4/1)
## :       SibSp <= 2:
## :       :...Parch <= 0: Died (8.2/1)
## :           Parch > 0: Survived (18.2/0.1)
## Sex = female:
## :...Pclass <= 2: Survived (170/9)
##     Pclass > 2:
##     :...Embarked = Missing: Survived (0)
##         Embarked = Queenstown:
##         :...Parch <= 0: Survived (30/6)
##         :   Parch > 0: Died (3)
##         Embarked = Cherbourg:
##         :...Fare > 15.2458: Survived (7)
##         :   Fare <= 15.2458:
##         :   :...Fare <= 13.8625: Survived (6)
##         :       Fare > 13.8625: Died (10/2)
##         Embarked = Southampton:
##         :...Fare > 20.575: Died (29/3)
##             Fare <= 20.575:
##             :...Parch <= 0: Died (45/20)
##                 Parch > 0: Survived (14/4)
##
##
## Evaluation on training data (891 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##      13  135(15.2%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##     530    19    (a): class Died
##     116   226    (b): class Survived
##
##
##  Attribute usage:
##
##  100.00% Sex
##   50.84% Age
##   35.24% Pclass
```

```
##   25.70% Parch
##   17.51% SibSp
##   16.16% Embarked
##   12.46% Fare
##
##
## Time: 0.0 secs
```

```
# plot(tree)
```

2.2 Testing

```
tree_pred <- predict(tree, test_titanic)

table(tree_pred, test_titanic_labels)
```

```
##           test_titanic_labels
## tree_pred  Died Survived
##   Died      257       26
##   Survived    9      126
```

```
# Calculate the percentage of correct guesses.
tree_correct <- tree_pred == test_titanic_labels
table(tree_correct)/length(tree_correct) * 100
```

```
## tree_correct
##     FALSE      TRUE
##  8.373206 91.626794
```

3 Naive Bayes

```
# Load e1071 if it's not installed install it.
package <- 'e1071'
if(package %in% rownames(installed.packages())) {
  do.call('library', list(package))
} else {
  install.packages(package)
  do.call("library", list(package))
}
# Trains the data set.
nb1 <- naiveBayes(Survived~., data=train_titanic)
#nb1
```

3.1 Testing

```
nb_guess <- predict(nb1, test_titanic[,c("Pclass", "Sex", "Age", "SibSp", "Parch", "Fare", "Cabin")])
table(nb_guess, test_titanic_labels)
```

```
##           test_titanic_labels
## nb_guess   Died Survived
##   Died      226       68
```

```
##   Survived   40       84
```

```
# Calculate the percentage of correct guesses.
nb_correct <- nb_guess == test_titanic_labels
table(nb_correct)/length(nb_correct) * 100
```

```
## nb_correct
##    FALSE     TRUE
## 25.83732 74.16268
```

4 Neural Networks

Neural networks only work with numerical data so I'm going to for now, remove the categorical data. An option to do later would be to associate a category to a set of nodes and activate the corresponding inpute node for the specific category.

```
nn_process <- function(titanic){
  # Covert Everything to a numeric.
  titanic <- lapply(titanic, function(x) as.numeric(x))

  # Define our Normalize function
  normalize <- function(x) {
    if (anyNA(x)) {
      x <- scale(x, center = FALSE, scale = TRUE)
      x[is.na(x)] <- 0
      return(x)
    } else {
      return ((x - min(x)) / (max(x) - min(x)))
    }
  }

  titanic <- lapply(titanic, normalize)
  titanic <- data.frame(titanic)
  return(titanic)
}

nn_train <- nn_process(train_titanic)
nn_test <- nn_process(test_titanic)
```

4.1 Training the neural network

```
# Dynamically load `neuralnet`
package <- 'neuralnet'
if(package %in% rownames(installed.packages())) {
  do.call('library', list(package))
} else {
  install.packages(package)
  do.call("library", list(package))
}

set.seed(1234)
```

```r
# Create a formula in the form "Survived ~ V1 + V2 + ... + V33 + V34"
fmla <- as.formula(paste("Survived ~ ", paste(names(nn_test), collapse= "+")))

# Train the model
nn <- neuralnet(fmla , data = nn_train, hidden = c(3))
plot(nn)
```

```r
# Predict some data
nn_predict <- compute(nn, nn_test)
cor(nn_predict$net.result, as.numeric(test_titanic_labels) - 1)
```

```
##              [,1]
## [1,] 0.7144509261
```

5 knn?

6