Jin Kweon (3032235207) - HW 5

Jin Kweon

11/15/2017

Problem 1

Link: https://www.kaggle.com/c/titanic/data

Data import

```
getwd()
```

```
## [1] "/Users/yjkweon24/Desktop/Cal/2017 Fall/Stat 151a/HW/HW5"
```

```
train <- read.csv("train.csv", header = T)
test <- read.csv("test.csv", header = T)
dim(train)
```

```
## [1] 891  12
```

```
dim(test)
```

```
## [1] 418  11
```

Data dictionary:

Variable Definition Key survival Survival 0 = No, 1 = Yes pclass Ticket class 1 = 1st, 2 = 2nd, 3 = 3rd sex Sex Age Age in years
sibsp # of siblings / spouses aboard the Titanic
parch # of parents / children aboard the Titanic
ticket Ticket number
fare Passenger fare
cabin Cabin number
embarked Port of Embarkation C = Cherbourg, Q = Queenstown, S = Southampton

EDA (This will only apply to the training set.)

Here are the plans:

1. Find missing value or NA -> use sum(is.na) or use the way I did in pbhlth 245 project... & variable == "" to find empty place -> Think of how to fix these issues.... (i.e. randomly assign, plug in the most frequent values, or use predict function...)
2. Take out the variables that are not interpretable... -> maybe passenger ID as this is repetitve...

3. Do str and summary to find out data structures... (fix if necessary)
4. Find out continuous variables, and transform them into numeric if necessary.
5. Find out categorical variables, and transform them into factor if necessary.
6. scatter plot
7. boxplot
8. pairs plot with smoothing lines
9. co-plot
10. density estimators plot
11. Cook's distance & leverage & residual plots
12. Do str and summary to find out data structures again...

```r
#Training cleaning
na <- c()
for(i in 1:ncol(train)){
  na[i] <- sum(is.na(train[,i]))
}
na
```

```
##  [1]   0   0   0   0   0 177   0   0   0   0   0   0
```

```r
empty <- c()
for(i in 1:ncol(train)){
  empty[i] <- sum(train[,i] == "")
}
empty
```

```
##  [1]   0   0   0   0   0  NA   0   0   0   0 687   2
```

```r
train <- train[,-11]
test <- test[,-10]

table(train$Embarked)
```

```
##
##       C   Q   S
##   2 168  77 644
```

```r
which(train$Embarked == "")
```

```
## [1]  62 830
```

```r
train$Embarked[which(train$Embarked == "")] <- "S"

train$Age[is.na(train$Age)] <- 0
train$Age <- as.numeric(train$Age)
group_by(train, Sex) %>% summarize(m = mean(Age))
```

```
## # A tibble: 2 x 2
##      Sex        m
##   <fctr>    <dbl>
## 1 female 23.20382
## 2   male 24.12334
```

```r
train$Age[which(train$Sex == "female" & train$Age == 0)] <- 23
train$Age[which(train$Sex == "male" & train$Age == 0)] <- 24

train$Fare[train$Fare == 0] <- mean(train$Fare)

#Testing cleaning
na <- c()
for(i in 1:ncol(test)){
  na[i] <- sum(is.na(test[,i]))
}
na
```

```
##  [1]  0  0  0  0 86  0  0  0  1  0
```

```r
empty <- c()
for(i in 1:ncol(test)){
  empty[i] <- sum(test[,i] == "")
}
empty
```

```
##  [1]  0  0  0  0 NA  0  0  0 NA  0
```

```r
test$Fare <- as.numeric(test$Fare)
test$Fare[is.na(test$Fare)] <- 0
sum(test$Fare == 0)
```

```
## [1] 3
```

```r
test$Fare[test$Fare == 0] <- mean(test$Fare)

test$Age <- as.numeric(test$Age)
test$Age[is.na(test$Age)] <- 0
group_by(test, Sex) %>% summarize(m = mean(Age))
```

```
## # A tibble: 2 x 2
##      Sex        m
##   <fctr>    <dbl>
## 1 female 25.29336
## 2   male 23.33049
```

```r
test$Age[which(test$Sex == "female" & test$Age == 0)] <- 25
test$Age[which(test$Sex == "male" & test$Age == 0)] <- 23


#Take out variables not needed
train <- train[,-1]
train <- train[,-3]
train <- train[,-7]

test <- test[,-1]
test <- test[,-2]
test <- test[,-6]

names(train)
```

```
## [1] "Survived" "Pclass"   "Sex"      "Age"      "SibSp"    "Parch"
## [7] "Fare"     "Embarked"
```

```r
names(test)
```

```
## [1] "Pclass"   "Sex"      "Age"      "SibSp"    "Parch"    "Fare"
## [7] "Embarked"
```

```r
#Check the structure
str(train)
```

```
## 'data.frame':    891 obs. of  8 variables:
##  $ Survived: int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass  : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age     : num  22 38 26 35 35 24 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 ...
##  $ Fare    : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Embarked: Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

```r
train$Survived <- as.factor(train$Survived)

str(test)
```

```
## 'data.frame':    418 obs. of  7 variables:
##  $ Pclass  : int  3 3 2 3 3 3 3 2 3 3 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2 ...
##  $ Age     : num  34.5 47 62 27 22 14 30 26 18 21 ...
##  $ SibSp   : int  0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch   : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Fare    : num  7.83 7 9.69 8.66 12.29 ...
##  $ Embarked: Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3 ...
```

```r
#Check the summary and correct wrong information
summary(train)
```

```
##  Survived     Pclass          Sex           Age            SibSp
##  0:549    Min.   :1.000   female:314   Min.   : 0.42   Min.   :0.000
##  1:342    1st Qu.:2.000   male  :577   1st Qu.:22.00   1st Qu.:0.000
##           Median :3.000                Median :24.00   Median :0.000
##           Mean   :2.309                Mean   :28.51   Mean   :0.523
##           3rd Qu.:3.000                3rd Qu.:35.00   3rd Qu.:1.000
##           Max.   :3.000                Max.   :80.00   Max.   :8.000
##      Parch             Fare         Embarked
##  Min.   :0.0000   Min.   :  4.013    :  0
##  1st Qu.:0.0000   1st Qu.:  7.925   C:168
##  Median :0.0000   Median : 15.100   Q: 77
##  Mean   :0.3816   Mean   : 32.746   S:646
##  3rd Qu.:0.0000   3rd Qu.: 32.204
##  Max.   :6.0000   Max.   :512.329
```

```r
summary(test)
```

```
##      Pclass          Sex           Age            SibSp
##  Min.   :1.000   female:152   Min.   : 0.17   Min.   :0.0000
##  1st Qu.:1.000   male  :266   1st Qu.:23.00   1st Qu.:0.0000
##  Median :3.000                Median :25.00   Median :0.0000
##  Mean   :2.266                Mean   :28.90   Mean   :0.4474
##  3rd Qu.:3.000                3rd Qu.:35.75   3rd Qu.:1.0000
```

```
##  Max.   :3.000                  Max.   :76.00   Max.   :8.0000
##      Parch             Fare          Embarked
##  Min.   :0.0000   Min.   :  3.171   C:102
##  1st Qu.:0.0000   1st Qu.:  7.896   Q: 46
##  Median :0.0000   Median : 14.479   S:270
##  Mean   :0.3923   Mean   : 35.797
##  3rd Qu.:0.0000   3rd Qu.: 31.634
##  Max.   :9.0000   Max.   :512.329
```

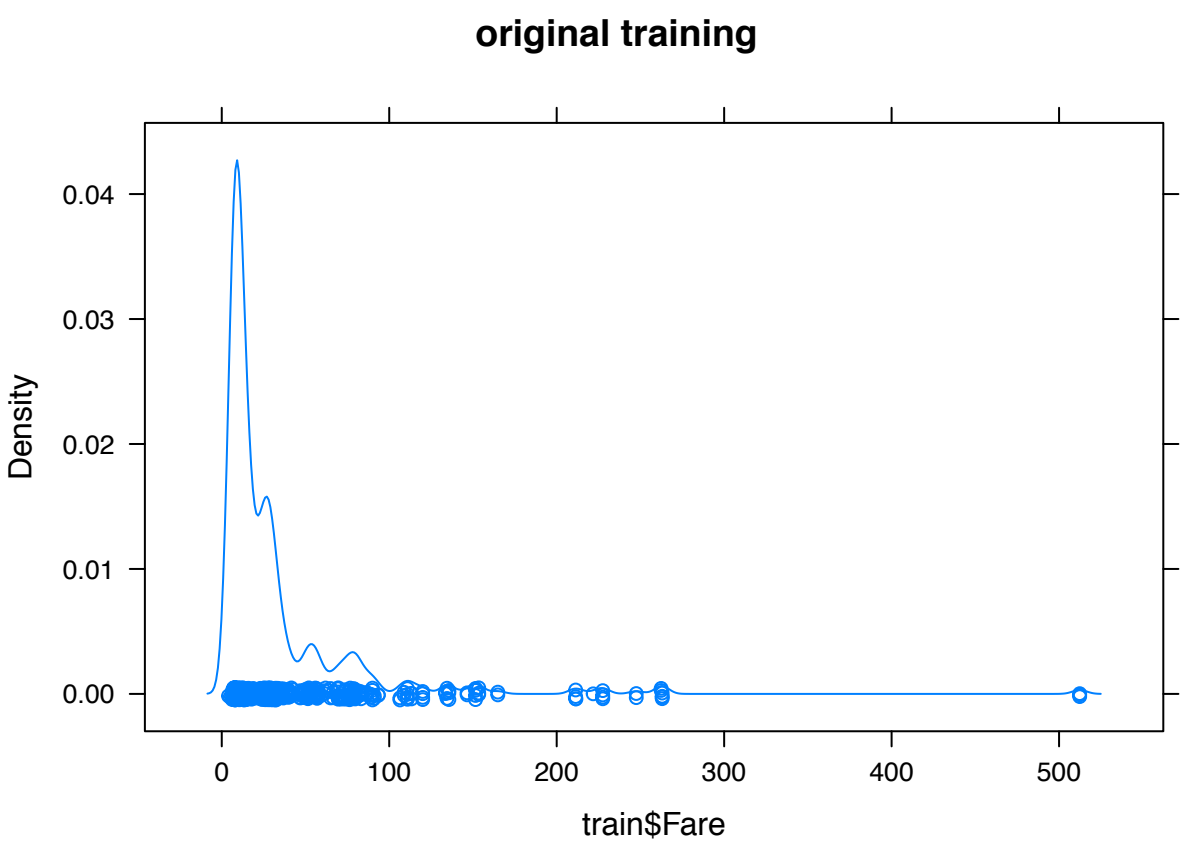
```r
head(sort(train$Age), 20)
```

```
##  [1] 0.42 0.67 0.75 0.75 0.83 0.83 0.92 1.00 1.00 1.00 1.00 1.00 1.00 1.00
## [15] 2.00 2.00 2.00 2.00 2.00 2.00
```

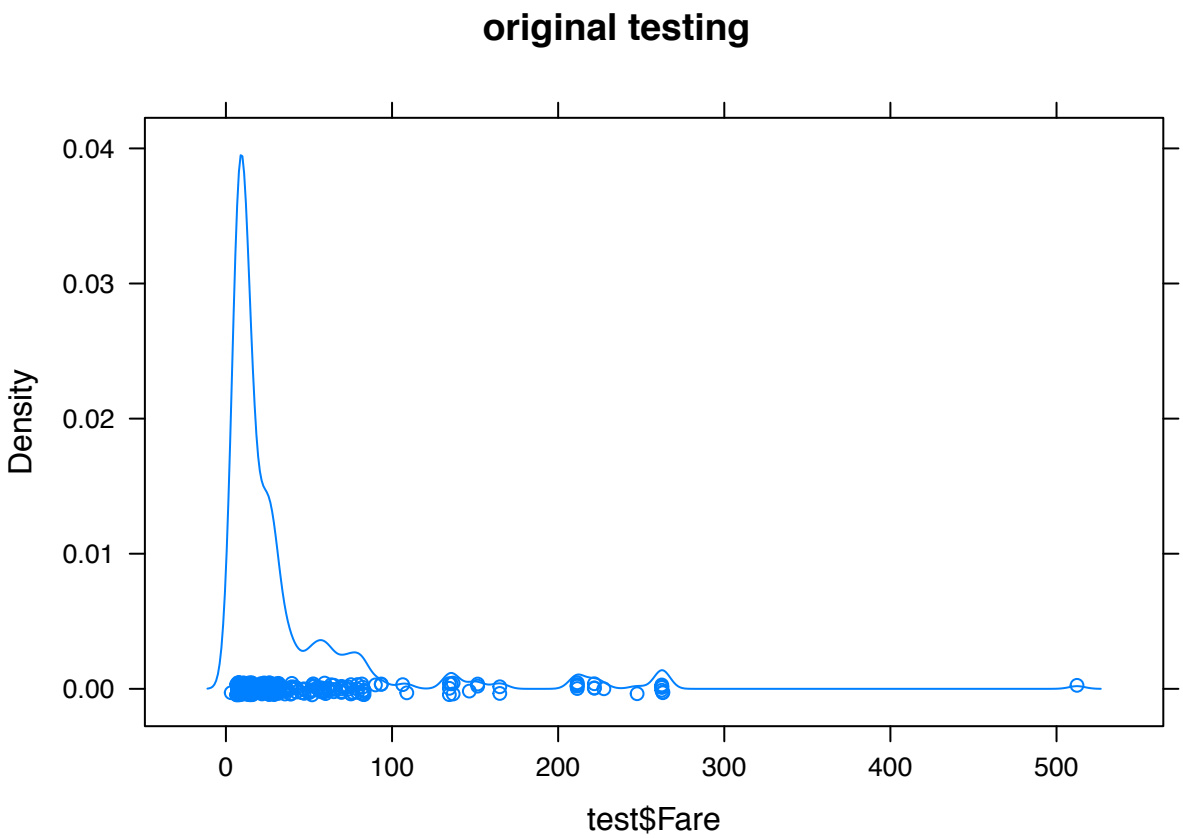
```r
head(sort(test$Age), 20)
```

```
##  [1] 0.17 0.33 0.75 0.83 0.92 1.00 1.00 1.00 2.00 2.00 3.00 5.00 6.00 6.00
## [15] 6.00 7.00 8.00 8.00 9.00 9.00
```

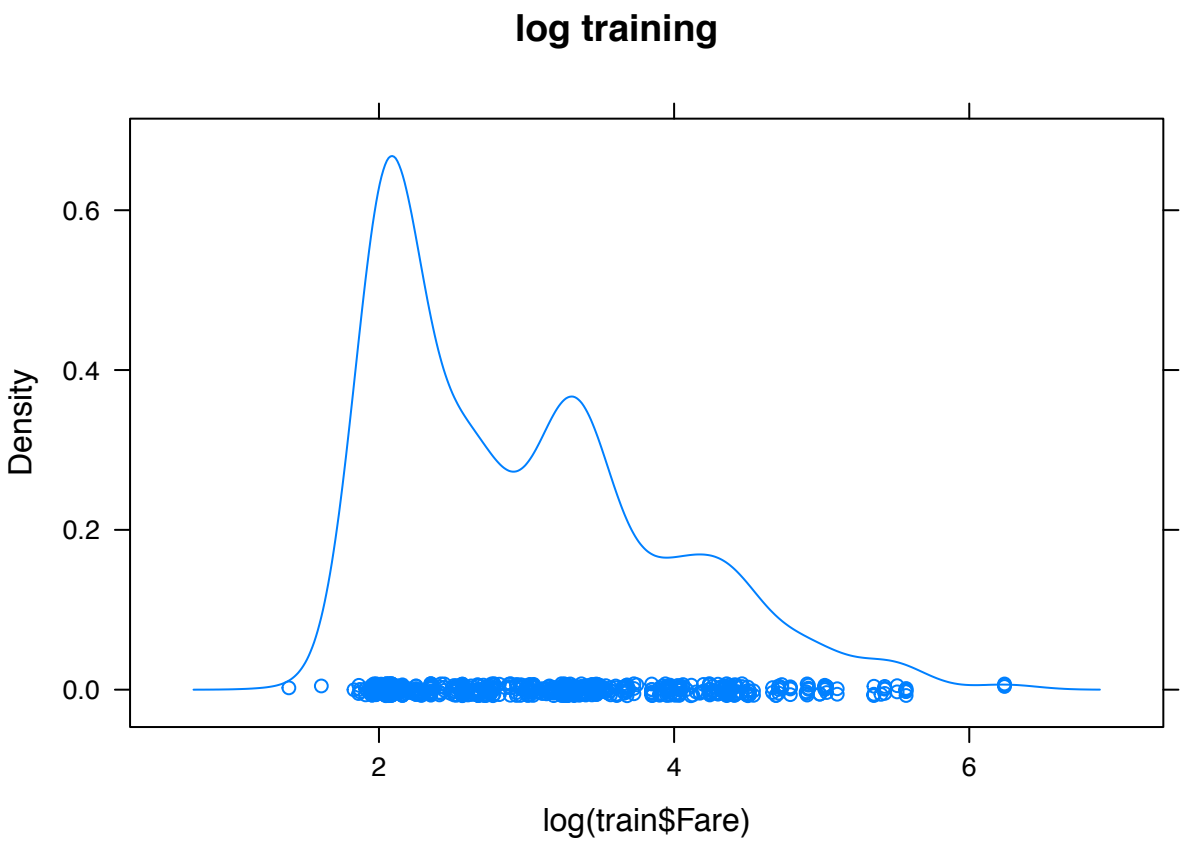
```r
train$Age[train$Age < 1] <- 1
test$Age[test$Age < 1] <- 1

densityplot(train$Fare, main = "original training")
```



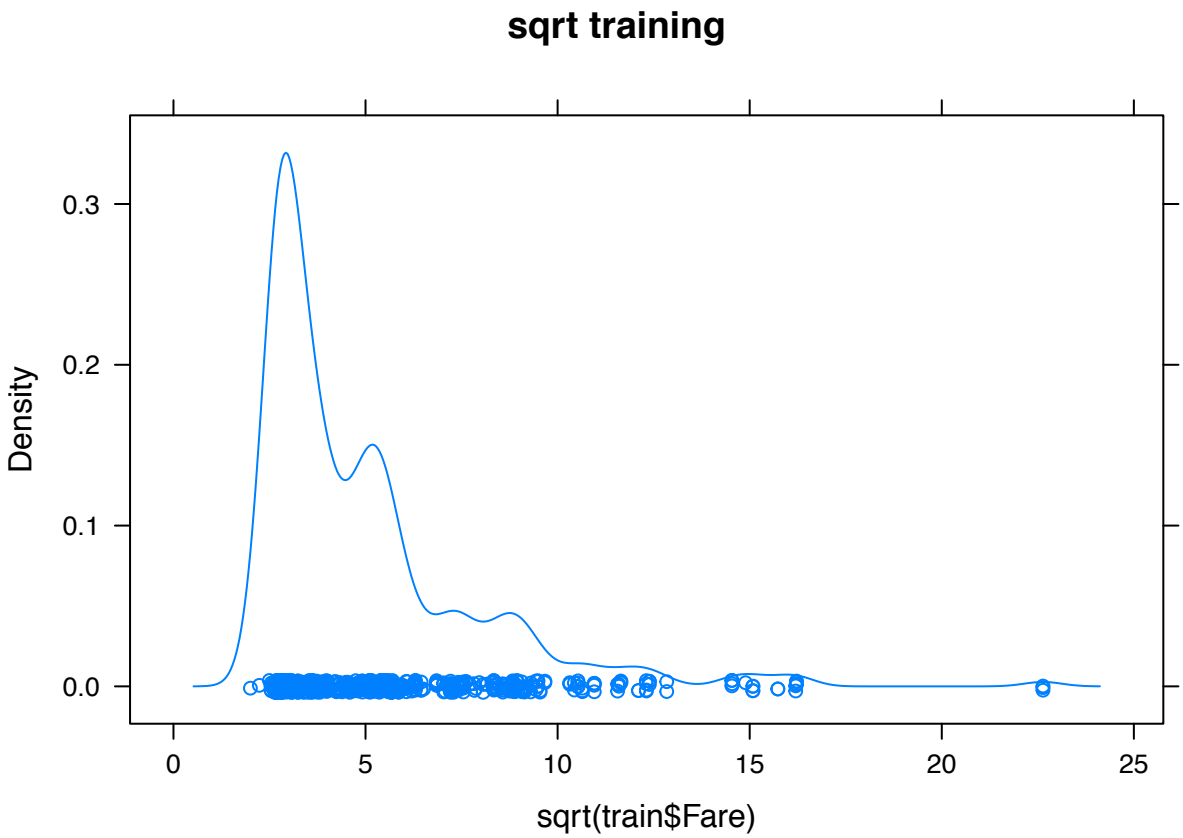

```r
densityplot(test$Fare, main = "original testing")
```



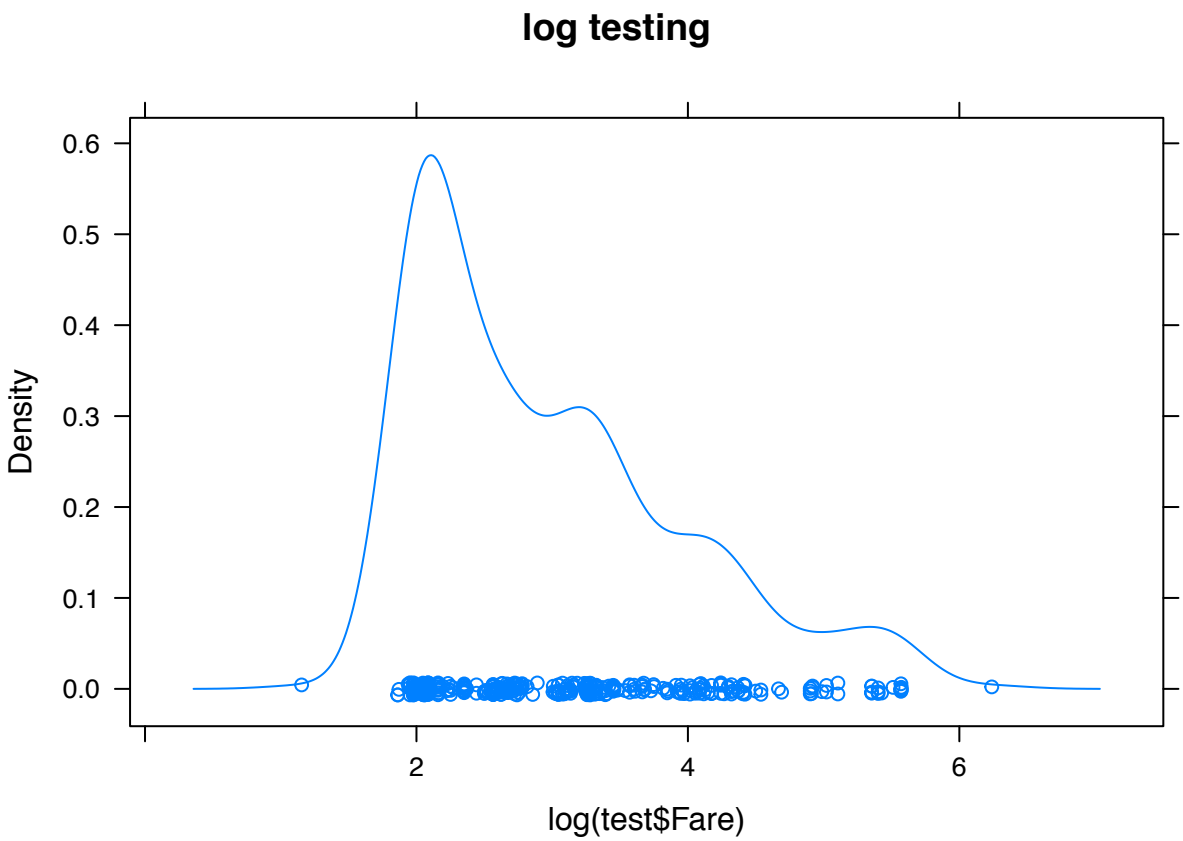

```
densityplot(log(train$Fare), main = "log training")
```



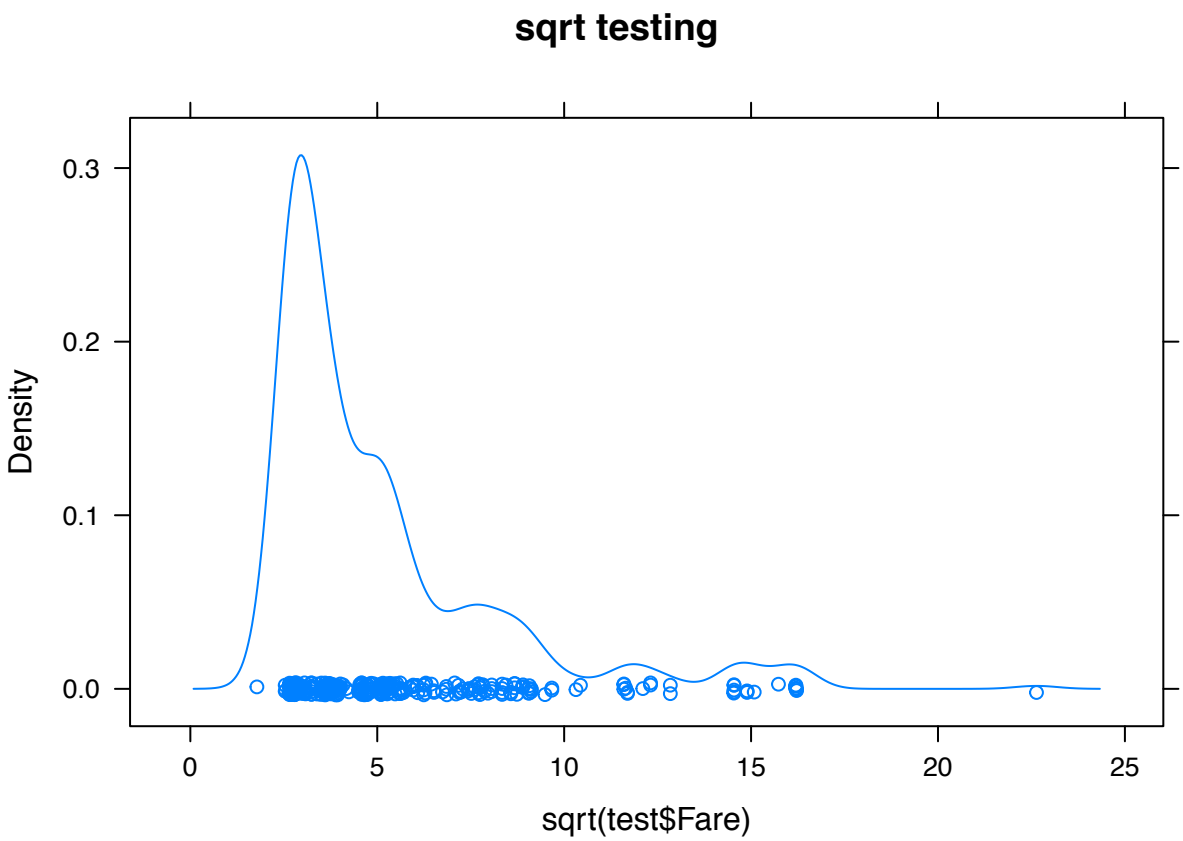
```r
densityplot(sqrt(train$Fare), main = "sqrt training")
```



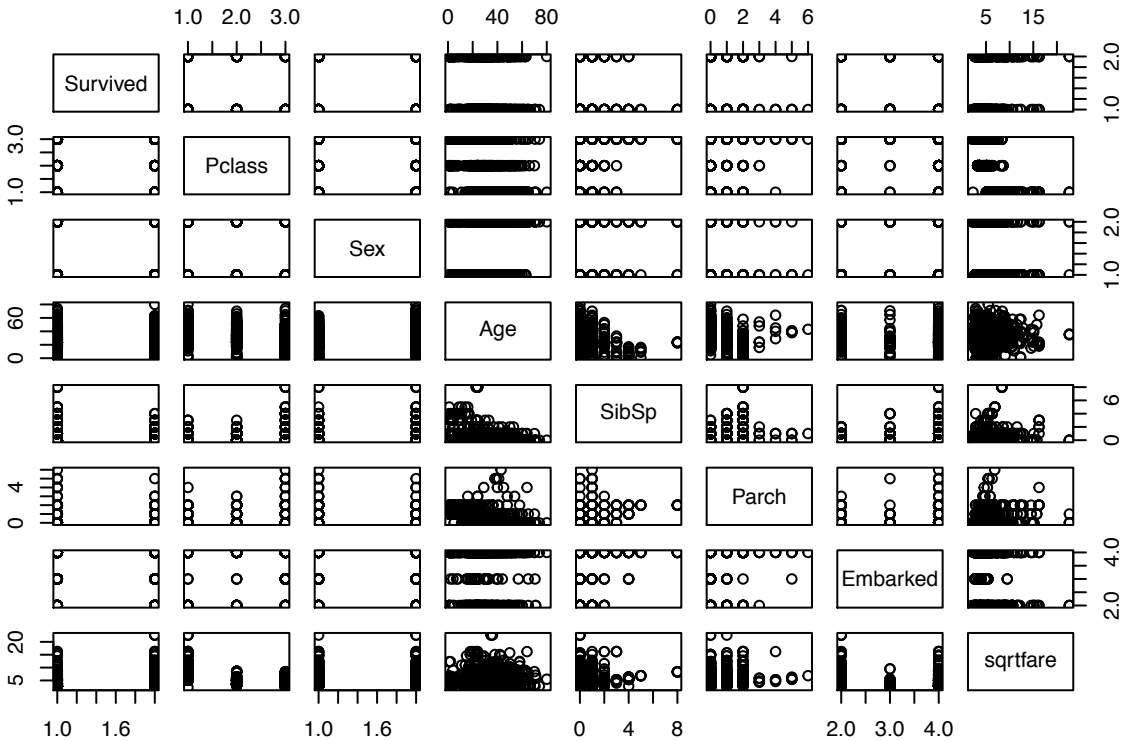

```r
densityplot(log(test$Fare), main = "log testing")
```



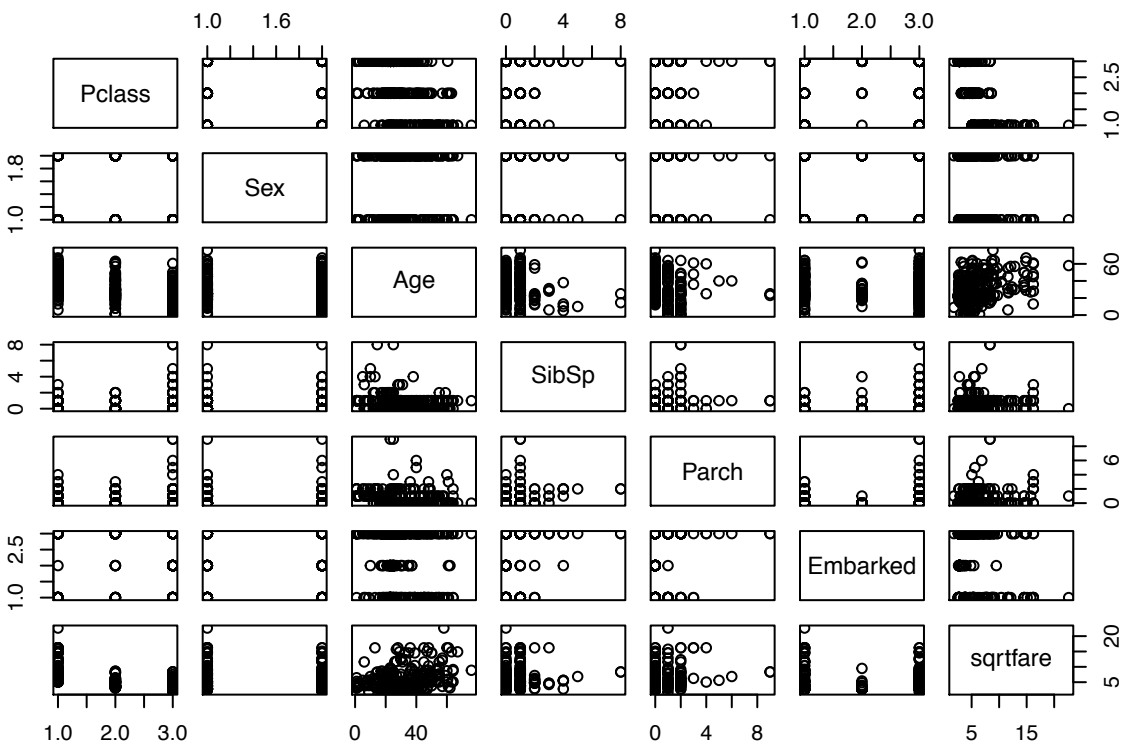

```
densityplot(sqrt(test$Fare), main = "sqrt testing")
```




```
sqrtfare <- sqrt(train$Fare)
train$sqrtfare <- sqrtfare
train <- train[,-7]

sqrtfare <- sqrt(test$Fare)
test$sqrtfare <- sqrtfare
test <- test[,-6]


#Plots
plot(train)
```




```r
plot(test)
```



Comment:

First, I checked the missing value and NA on the training data set. First of all, there are too many missing values for "Cabin" variable, (687 out of 891 rows) and this will not really help us interpret/predict when I include this variable into the model. So, I decided to take out "Cabin" out of from training and testing sets. Also, there are two missing values for "Embarked" variable in training set, and I decided to assign the

most frequent values in it, which is the most reasonable thing I can do at this time when I have no other information. (S assigned as S is the most frequent...) Also, I need to find the way to cover the NA values for "Age" variable. However, I cannot just use prediction based on this, since they are both categorical values and prediction is hard to be run well... So, I decide to find averages for each gender, and assign to the NA. I do the similar things for testing set. (there is no missing value in testing set, but some of them are NA. And, I found some of the fares are zero, which does not make sense, so I assigned mean onto the all these values.) I found one thing weird when I modified the data sets. The women mean age for training set is lower than the men's for training, but higher for testing.... It implies some of the data might be wrong...

After I finish all these cleanings, I have looked at the data and took out some variables (as I took out "Cabin" before): passengerID is not needed as this is just a row number (only need it when I submit it on Kaggle) and Ticket is not needed as this is mixed with character and numeric. Logically, these two variables are not needed to predict survival. Ticket might mean mroe than just ticket itselt; however, we do not have enough information to interpret it. Last but not least, I decide to take out the variable "Name." How does the name contribute predicting survival rate... Personally, I thought "Embarked" variable might not be important (but I just leave the varaible, since this might mean something important later.)

The next step is to check the structure and summary of our data sets. I need to classify numerical and categorical variables into the right groups. Here is how I tranformed in to, below:

Survived should be categorical.

Pclass should be numeric.

Sex should be categorical.

Age should be numeric.

SibSp should be numeric.

Parch should be numeric.

Fare should be numeric.

Embarked should be categorical.

And, when I checked the summary of training and testing sets, there are somethings I found weird. For example, minimum age for both training sets and testing sets look weird (0.42 and 0.17 for training and testing, respectively). So, I decided to assign 1 onto all the ages that are below 1. I also found the distribution of Fare variable is little bit suspicious... When you look at it the maximum of fare for both training and testing are super large. (maximum is over 500 dollars for both training and testing, but minimum is around 3 dollars.) I am sure at that time, the currency was way lower than this... So I made the density plots for log and square root. After I did some research of titanic, I found square root transformation makes more sense to me.

After that, I made bunch of different plots.

Modeling

Here are the plans:

1. Logistic regression with a full basic model, with survival status being response (make sure to include family = "binomial") onto the training data set
2. Perform different model selection methods: adjusted R^2, AIC, BIC, Mallow's Cp, on to the full model -> use either bestglm
3. Include interaction term or remove some variables
4. Perform different model selection methods: adjusted R^2, AIC, BIC, Mallow's Cp, on to the new model -> use either bestglm
5. Get the Best model/model selection methods by comparing misclassification rate with confusion matrix or ROC curve inside of the training set
6. See how residual deviance gets better than the full model

```
glmoriginal <- glm(Survived ~., data = train, family = binomial)
glmoriginal$aic
```

```
## [1] 805.9974
```

```
glmoriginal
```

```
##
## Call:  glm(formula = Survived ~ ., family = binomial, data = train)
##
## Coefficients:
## (Intercept)       Pclass      Sexmale          Age        SibSp
##     4.92835     -1.06313     -2.72465     -0.03687     -0.33194
##       Parch    EmbarkedQ    EmbarkedS     sqrtfare
##    -0.08776     -0.13696     -0.39312      0.03989
##
## Degrees of Freedom: 890 Total (i.e. Null);  882 Residual
## Null Deviance:       1187
## Residual Deviance: 788   AIC: 806
```

```
Xy <- data.frame(train[,-1], Survived = as.numeric(train[,1]) - 1)

aic1 <- bestglm(Xy, family = binomial, IC = "AIC", TopModels = 10)
```

```
## Morgan-Tatar search since family is non-gaussian.
## Note: factors present with more than 2 levels.
```

```
aic1$BestModels
```

```
##    Pclass  Sex  Age SibSp Parch Embarked sqrtfare Criterion
## 1    TRUE TRUE TRUE  TRUE FALSE    FALSE    FALSE  800.7617
## 2    TRUE TRUE TRUE  TRUE FALSE    FALSE     TRUE  801.8553
## 3    TRUE TRUE TRUE  TRUE  TRUE    FALSE    FALSE  802.4656
## 4    TRUE TRUE TRUE  TRUE FALSE     TRUE    FALSE  802.8746
## 5    TRUE TRUE TRUE  TRUE  TRUE    FALSE     TRUE  803.0801
## 6    TRUE TRUE TRUE  TRUE FALSE     TRUE     TRUE  804.5268
## 7    TRUE TRUE TRUE  TRUE  TRUE     TRUE    FALSE  804.6192
## 8    TRUE TRUE TRUE  TRUE  TRUE     TRUE     TRUE  805.9974
## 9    TRUE TRUE TRUE FALSE  TRUE    FALSE    FALSE  811.5270
## 10   TRUE TRUE TRUE FALSE  TRUE     TRUE    FALSE  812.3644
```

```
aic1$BestModel
```

```
##
```

```
## Call:  glm(formula = y ~ ., family = family, data = Xi, weights = weights)
##
## Coefficients:
## (Intercept)       Pclass      Sexmale          Age        SibSp
##     5.09741     -1.17927     -2.73206     -0.03802     -0.34431
##
## Degrees of Freedom: 890 Total (i.e. Null);  886 Residual
## Null Deviance:       1187
## Residual Deviance: 792.8     AIC: 802.8
```

```r
bic1 <- bestglm(Xy, family = binomial, IC = "BIC", TopModels = 10)
```

```
## Morgan-Tatar search since family is non-gaussian.
## Note: factors present with more than 2 levels.
```

```r
bic1$BestModels
```

```
##    Pclass  Sex   Age SibSp Parch Embarked sqrtfare Criterion
## 1    TRUE TRUE  TRUE  TRUE FALSE    FALSE    FALSE  819.9311
## 2    TRUE TRUE  TRUE  TRUE FALSE    FALSE     TRUE  825.8170
## 3    TRUE TRUE  TRUE  TRUE  TRUE    FALSE    FALSE  826.4273
## 4    TRUE TRUE  TRUE FALSE FALSE    FALSE    FALSE  827.0773
## 5    TRUE TRUE  TRUE FALSE  TRUE    FALSE    FALSE  830.6963
## 6    TRUE TRUE  TRUE  TRUE  TRUE    FALSE     TRUE  831.8342
## 7    TRUE TRUE  TRUE FALSE FALSE    FALSE     TRUE  833.5791
## 8    TRUE TRUE  TRUE  TRUE FALSE     TRUE    FALSE  836.4210
## 9    TRUE TRUE  TRUE FALSE  TRUE    FALSE     TRUE  837.4578
## 10   TRUE TRUE FALSE  TRUE FALSE    FALSE    FALSE  839.6952
```

```r
bic1$BestModel
```

```
##
## Call:  glm(formula = y ~ ., family = family, data = Xi, weights = weights)
##
## Coefficients:
## (Intercept)       Pclass      Sexmale          Age        SibSp
##     5.09741     -1.17927     -2.73206     -0.03802     -0.34431
##
## Degrees of Freedom: 890 Total (i.e. Null);  886 Residual
## Null Deviance:       1187
## Residual Deviance: 792.8     AIC: 802.8
```

```r
#Model changes...
glm(Survived ~ Pclass + Sex + Age + SibSp + Parch + sqrtfare, data = train, family = binomial)
```

```
##
## Call:  glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch +
##     sqrtfare, family = binomial, data = train)
##
## Coefficients:
## (Intercept)       Pclass      Sexmale          Age        SibSp
##     4.56193     -1.04449     -2.74846     -0.03770     -0.35953
##       Parch     sqrtfare
##    -0.10481      0.05776
##
## Degrees of Freedom: 890 Total (i.e. Null);  884 Residual
## Null Deviance:       1187
```

```
## Residual Deviance: 791.1     AIC: 805.1
```

```r
glm(Survived ~ Pclass + SibSp + Parch + Embarked + sqrtfare, data = train, family = binomial)
```

```
##
## Call:  glm(formula = Survived ~ Pclass + SibSp + Parch + Embarked +
##     sqrtfare, family = binomial, data = train)
##
## Coefficients:
## (Intercept)       Pclass        SibSp        Parch    EmbarkedQ
##     1.01772     -0.69057     -0.14955      0.27478      0.29792
##   EmbarkedS     sqrtfare
##    -0.51556      0.07648
##
## Degrees of Freedom: 890 Total (i.e. Null);  884 Residual
## Null Deviance:       1187
## Residual Deviance: 1055  AIC: 1069
```

```r
glm(Survived ~ Pclass + Sex + Age + SibSp + Parch + sqrtfare + Sex*Age, data = train, family = binomial
```

```
##
## Call:  glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch +
##     sqrtfare + Sex * Age, family = binomial, data = train)
##
## Coefficients:
## (Intercept)       Pclass      Sexmale          Age        SibSp
##     4.01031     -1.07582     -1.61294     -0.01254     -0.32483
##       Parch     sqrtfare  Sexmale:Age
##    -0.13030      0.05050     -0.04228
##
## Degrees of Freedom: 890 Total (i.e. Null);  883 Residual
## Null Deviance:       1187
## Residual Deviance: 782.6     AIC: 798.6
```

```r
glm(Survived ~ Pclass + Sex + Age + SibSp + Parch + sqrtfare + Sex*Age + Embarked, data = train, family
```

```
##
## Call:  glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch +
##     sqrtfare + Sex * Age + Embarked, family = binomial, data = train)
##
## Coefficients:
## (Intercept)       Pclass      Sexmale          Age        SibSp
##    4.415493    -1.105461    -1.494606    -0.009433    -0.290467
##       Parch     sqrtfare    EmbarkedQ    EmbarkedS  Sexmale:Age
##   -0.111978     0.028869    -0.160783    -0.472988    -0.045798
##
## Degrees of Freedom: 890 Total (i.e. Null);  881 Residual
## Null Deviance:       1187
## Residual Deviance: 778.2     AIC: 798.2
```

```r
glm(Survived ~ Pclass + Sex + Age + SibSp + Parch + sqrtfare + Sex*Age + Embarked + Sex*Pclass, data = 
```

```
##
## Call:  glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch +
##     sqrtfare + Sex * Age + Embarked + Sex * Pclass, family = binomial,
##     data = train)
##
```

```
## Coefficients:
##    (Intercept)          Pclass         Sexmale             Age
##        7.66018        -2.16120        -5.54016        -0.02739
##          SibSp           Parch        sqrtfare       EmbarkedQ
##       -0.29626        -0.04371         0.02218        -0.05127
##      EmbarkedS     Sexmale:Age  Pclass:Sexmale
##       -0.50699        -0.01886         1.33303
##
## Degrees of Freedom: 890 Total (i.e. Null);  880 Residual
## Null Deviance:       1187
## Residual Deviance: 761.2     AIC: 783.2
```

```
glm(Survived ~ Pclass + Sex + Age + SibSp + Parch + sqrtfare + Sex*Age + Embarked + Sex*Pclass + Sex*Pa
```

```
##
## Call:  glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch +
##     sqrtfare + Sex * Age + Embarked + Sex * Pclass + Sex * Parch,
##     family = binomial, data = train)
##
## Coefficients:
##    (Intercept)          Pclass         Sexmale             Age
##      7.9669180      -2.1765017      -5.9626444      -0.0275289
##          SibSp           Parch        sqrtfare       EmbarkedQ
##     -0.3191225      -0.2445326      -0.0004177      -0.1085956
##      EmbarkedS     Sexmale:Age  Pclass:Sexmale   Sexmale:Parch
##     -0.5057462      -0.0138969       1.3447956       0.6444724
##
## Degrees of Freedom: 890 Total (i.e. Null);  879 Residual
## Null Deviance:       1187
## Residual Deviance: 753.7     AIC: 777.7
```

```
sexage <- c() #let male becomes 1 and women becomes 0
sexage <- (as.numeric(train$Sex) - 1) * train$Age

sexpclass <- c()
sexpclass <- (as.numeric(train$Sex) - 1) * train$Pclass

sexparch <- c()
sexparch <- (as.numeric(train$Sex) - 1) * train$Parch

Xynew <- data.frame(train[,-1], sexage = sexage, sexpclass = sexpclass, sexparch = sexparch,
                    Survived = as.numeric(train[,1]) - 1)
glmnew <- glm(Survived~., data = Xynew, family = binomial)


#AIC and BIC on new model
aic2 <- bestglm(Xynew, family = binomial, IC = "AIC", TopModels = 10)
```

```
## Morgan-Tatar search since family is non-gaussian.
## Note: factors present with more than 2 levels.
```

```
aic2$BestModels
```

```
##    Pclass  Sex  Age SibSp Parch Embarked sqrtfare sexage sexpclass
## 1    TRUE TRUE TRUE  TRUE  TRUE    FALSE    FALSE  FALSE      TRUE
## 2    TRUE TRUE TRUE  TRUE  TRUE     TRUE    FALSE  FALSE      TRUE
```

```
## 3    TRUE TRUE TRUE  TRUE FALSE    FALSE    FALSE  FALSE      TRUE
## 4    TRUE TRUE TRUE  TRUE FALSE     TRUE    FALSE  FALSE      TRUE
## 5    TRUE TRUE TRUE  TRUE  TRUE    FALSE    FALSE   TRUE      TRUE
## 6    TRUE TRUE TRUE  TRUE  TRUE    FALSE     TRUE  FALSE      TRUE
## 7    TRUE TRUE TRUE  TRUE  TRUE     TRUE    FALSE   TRUE      TRUE
## 8    TRUE TRUE TRUE  TRUE FALSE    FALSE    FALSE   TRUE      TRUE
## 9    TRUE TRUE TRUE  TRUE  TRUE     TRUE     TRUE  FALSE      TRUE
## 10   TRUE TRUE TRUE  TRUE FALSE     TRUE    FALSE   TRUE      TRUE
##    sexparch Criterion
## 1      TRUE  773.4819
## 2      TRUE  774.3908
## 3      TRUE  774.6375
## 4      TRUE  775.0424
## 5      TRUE  775.1038
## 6      TRUE  775.3214
## 7      TRUE  775.7492
## 8      TRUE  776.3733
## 9      TRUE  776.3908
## 10     TRUE  776.5379
```

```r
aic2$BestModel
```

```
##
## Call:  glm(formula = y ~ ., family = family, data = Xi, weights = weights)
##
## Coefficients:
## (Intercept)       Pclass      Sexmale          Age        SibSp
##     7.98315     -2.20418     -6.42917     -0.03825     -0.35235
##       Parch    sexpclass     sexparch
##    -0.26086      1.36672      0.68516
##
## Degrees of Freedom: 890 Total (i.e. Null);  883 Residual
## Null Deviance:       1187
## Residual Deviance: 759.5     AIC: 775.5
```

```r
bic1 <- bestglm(Xynew, family = binomial, IC = "BIC", TopModels = 10)
```

```
## Morgan-Tatar search since family is non-gaussian.
## Note: factors present with more than 2 levels.
```

```r
bic1$BestModels
```

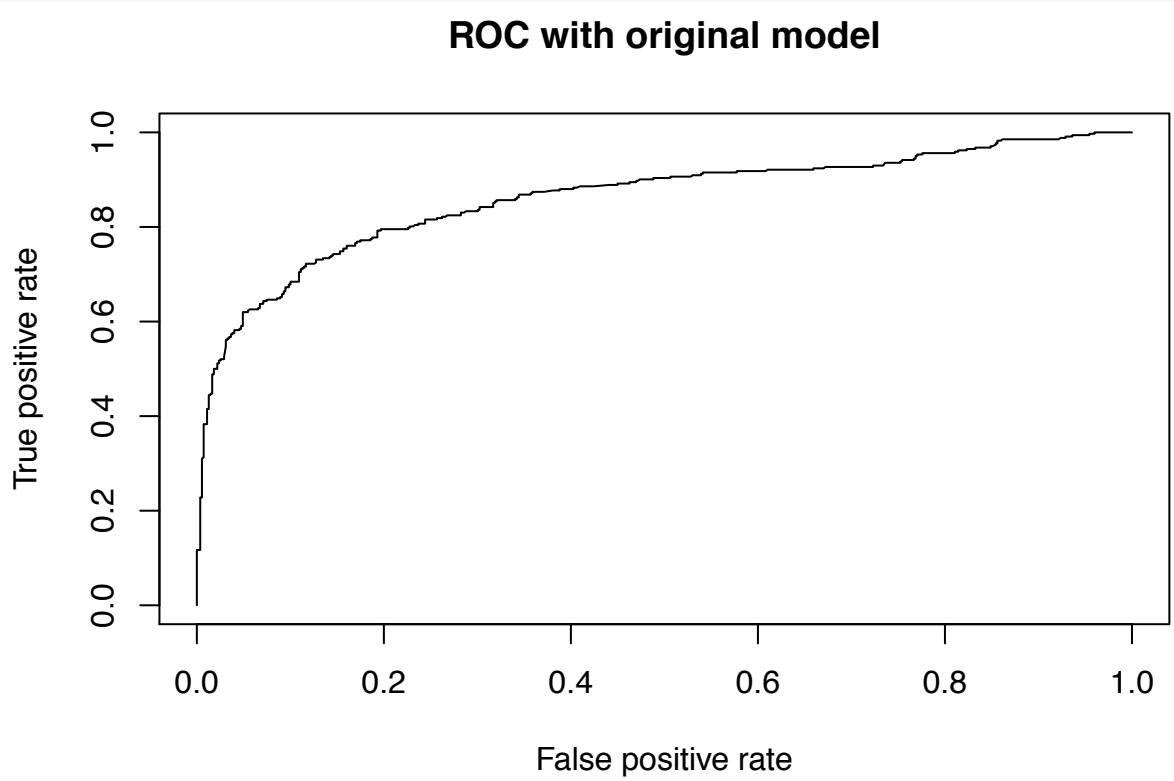
```
##    Pclass  Sex   Age SibSp Parch Embarked sqrtfare sexage sexpclass
## 1    TRUE TRUE  TRUE  TRUE FALSE    FALSE    FALSE  FALSE      TRUE
## 2    TRUE TRUE  TRUE  TRUE FALSE    FALSE    FALSE  FALSE      TRUE
## 3    TRUE TRUE FALSE  TRUE FALSE    FALSE    FALSE   TRUE      TRUE
## 4    TRUE TRUE  TRUE  TRUE  TRUE    FALSE    FALSE  FALSE      TRUE
## 5    TRUE TRUE  TRUE  TRUE FALSE    FALSE    FALSE   TRUE      TRUE
## 6    TRUE TRUE  TRUE FALSE FALSE    FALSE    FALSE  FALSE      TRUE
## 7    TRUE TRUE  TRUE  TRUE FALSE    FALSE     TRUE  FALSE      TRUE
## 8    TRUE TRUE FALSE  TRUE FALSE    FALSE    FALSE   TRUE      TRUE
## 9    TRUE TRUE  TRUE  TRUE  TRUE    FALSE    FALSE  FALSE      TRUE
## 10   TRUE TRUE  TRUE  TRUE FALSE    FALSE    FALSE   TRUE      TRUE
##    sexparch Criterion
## 1     FALSE  802.4603
## 2      TRUE  803.3915
```

```
## 3     FALSE  806.8212
## 4      TRUE  807.0283
## 5     FALSE  808.4873
## 6     FALSE  808.4892
## 7     FALSE  808.6331
## 8      TRUE  808.7969
## 9     FALSE  809.2361
## 10     TRUE  809.9197
```

```
bic1$BestModel
```

```
##
## Call:  glm(formula = y ~ ., family = family, data = Xi, weights = weights)
##
## Coefficients:
## (Intercept)       Pclass      Sexmale          Age        SibSp
##     8.03392     -2.26521     -6.21927     -0.04159     -0.32988
##   sexpclass
##     1.39635
##
## Degrees of Freedom: 890 Total (i.e. Null);  885 Residual
## Null Deviance:       1187
## Residual Deviance: 768.5     AIC: 780.5
```

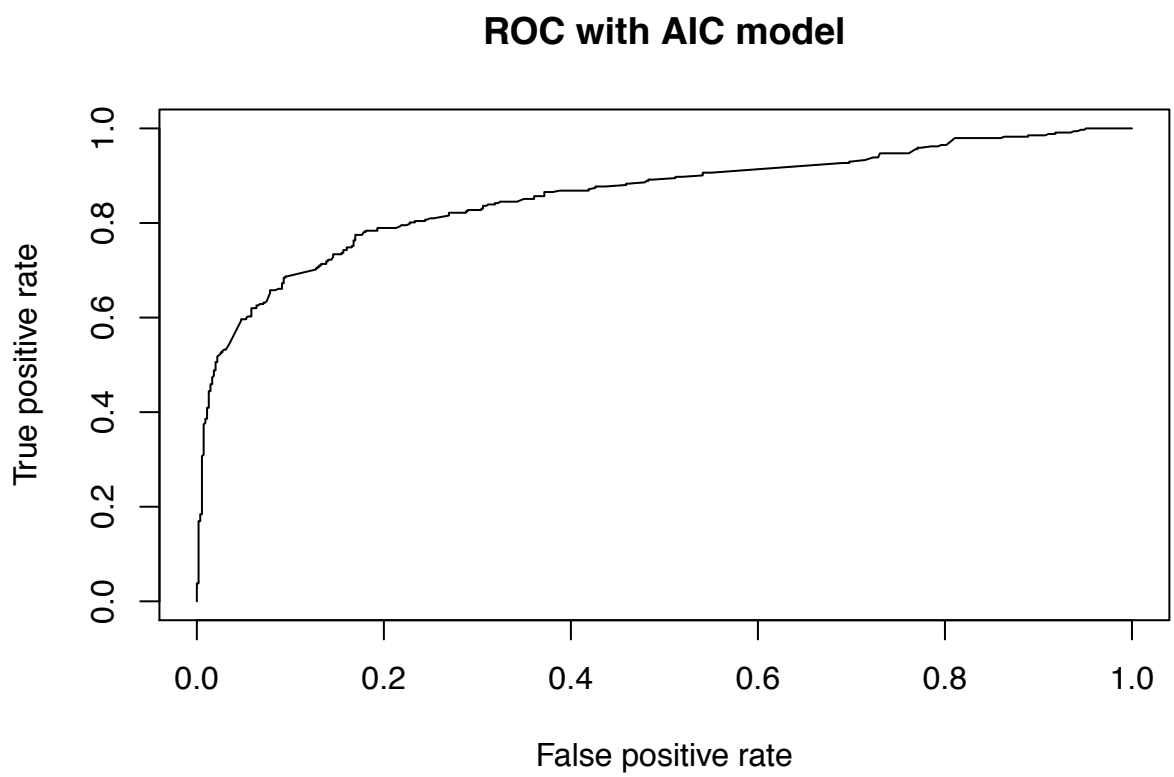
```
#Compare
aicmodel <- Xynew[,c(1,2,3,4,5,9,10)]
bicmodel <- Xynew[,c(1,2,3,4,9)]

predicting <- prediction(predict(glmnew, Xynew[,-11], type = "response"), Xynew[,11])
roc <- performance(predicting, measure = "tpr", x.measure = "fpr")
plot(roc, main = "ROC with original model")
```



```r
performance(predicting, measure = "auc")@y.values #AUC for original
```

```
## [[1]]
## [1] 0.8613827
```

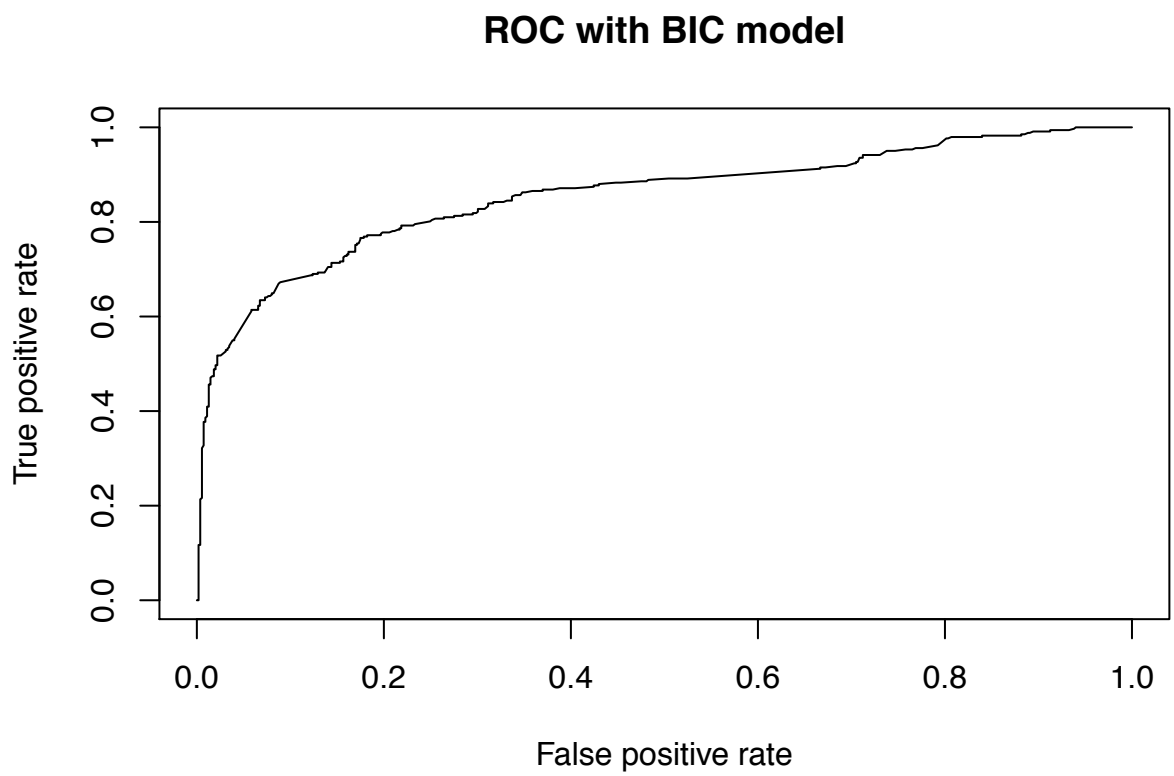
```r
predicting2 <- prediction(predict(aic2$BestModel, aicmodel, type = "response"), Xynew[,11])
roc2 <- performance(predicting2, measure = "tpr", x.measure = "fpr")
plot(roc2, main = "ROC with AIC model")
```




```r
performance(predicting2, measure = "auc")@y.values #AUC for AIC
```

```
## [[1]]
## [1] 0.8579901
```

```r
predicting3 <- prediction(predict(bic1$BestModel, bicmodel, type = "response"), Xynew[,11])
roc3 <- performance(predicting3, measure = "tpr", x.measure = "fpr")
plot(roc3, main = "ROC with BIC model")
```

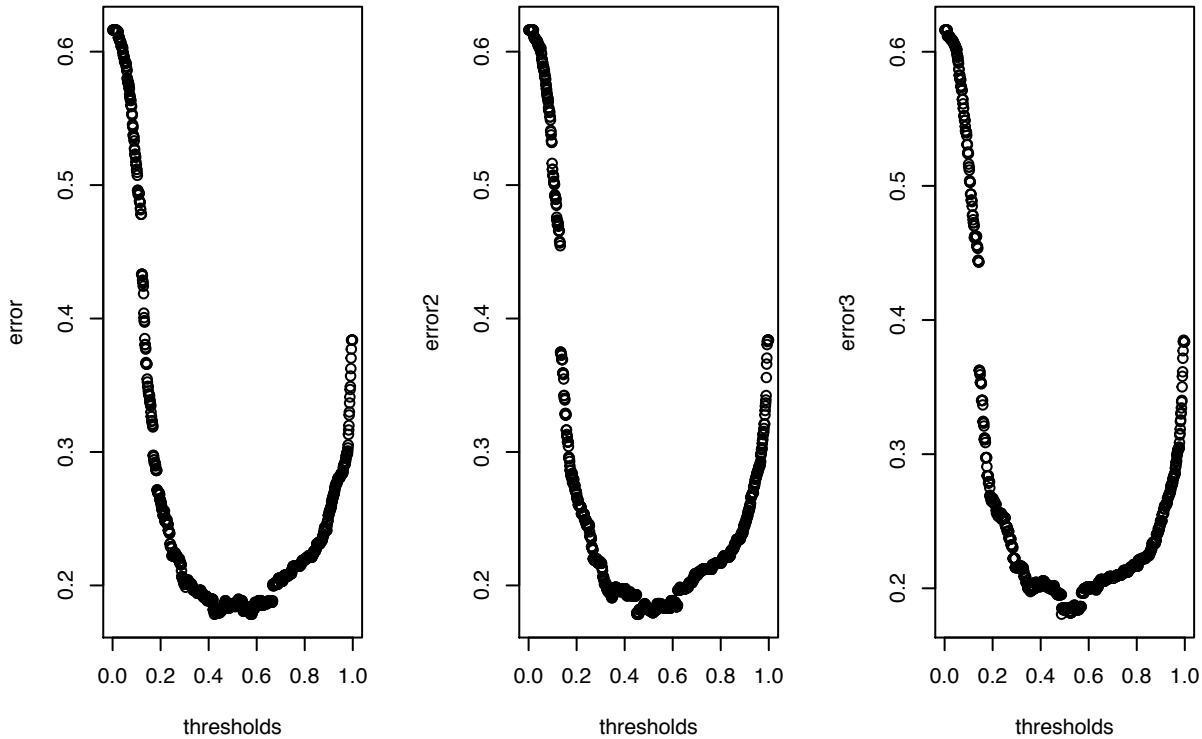



```r
performance(predicting3, measure = "auc")@y.values #AUC for BIC
```

```
## [[1]]
## [1] 0.8545548
```

```r
thresholds <- seq(from = 0.001, 0.999, 0.001)

error <- c() #misclassfication for original
for(i in 1:length(thresholds)){
  pr <- c()
  pr <- ifelse(predict(glmnew, Xynew[,-11], type = "response") > (0.001 * i), 1, 0)
  error[i] <- mean(pr != Xynew[,11])
}


error2 <- c() #misclassfication for AIC
for(i in 1:length(thresholds)){
  pr <- c()
  pr <- ifelse(predict(aic2$BestModel, aicmodel, type = "response") > (0.001 * i), 1, 0)
  error2[i] <- mean(pr != Xynew[,11])
}

error3 <- c() #misclassfication for BIC
for(i in 1:length(thresholds)){
  pr <- c()
  pr <- ifelse(predict(bic1$BestModel, bicmodel, type = "response") > (0.001 * i), 1, 0)
  error3[i] <- mean(pr != Xynew[,11])
}

par(mfrow=c(1,3))
```

```r
plot(thresholds, error)
plot(thresholds, error2)
plot(thresholds, error3)
```




```r
#Reporting
paste("Minimum misclassification rate for original model is", min(error), "at thresholds",
      which.min(error)*0.001, "and average rate is", mean(error), "and AUC is",
      performance(predicting, measure = "auc")@y.values[[1]])
```

```
## [1] "Minimum misclassification rate for original model is 0.178451178451178 at thresholds 0.424 and a
```

```r
paste("Minimum misclassification rate for AIC model is", min(error2), "at thresholds",
      which.min(error2)*0.001, "and average rate is", mean(error2), "and AUC is",
      performance(predicting2, measure = "auc")@y.values[[1]])
```

```
## [1] "Minimum misclassification rate for AIC model is 0.178451178451178 at thresholds 0.451 and averag
```

```r
paste("Minimum misclassification rate for BIC model is", min(error3), "at thresholds",
      which.min(error3)*0.001, "and average rate is", mean(error3), "and AUC is",
      performance(predicting3, measure = "auc")@y.values[[1]])
```

```
## [1] "Minimum misclassification rate for BIC model is 0.180695847362514 at thresholds 0.487 and averag
```

Comment:

I did not use stepwise regression for my variable selection at all, since we learned that it might not be able to select the BEST model. That is why I only peroformed criteria-based variable selections only!!!

Regsubsets does not work logistic function, so I decide to go with bestglm() function; however, it does not support Mallow's Cp or adjusted R^2 for model selection... So, I basically compare BIC with AIC only...

After that, I tried bunch of differnt models and compare residual deviance and AIC to choose which model can be my optimal model. I did not have to try AIC and BIC for original model before I transform model, but I just did it for fun. Basically, my goal is to select the glm() function with smallest deviance and AIC, and

then, perform AIC or BIC for model selection. (As I could find, sex and age are really important variables in this model.) I found that adding three interactions (sex and age, sex and pclass, and sex and parch) give the best model.

From the new model, AIC selected Pclass, Sex, Age, Sibsp, Parch, sexpclass, and sexparch. BIC selected Pclass, Sex, Age, SibSp, and sexpclass. So, I will compare the original model, AIC model, and BIC model. I will try ROC/AUC and misclassification rate. Basically ROC and misclassification rate graph are doing the same thing. They are plotting rate for each different threshold.

In conclusion, my new original model is the BEST. (AIC is the second and the BIC is the third...)

Predict

Here are the plans:

1. Now use the selected model to predict onto the test set
2. Try different times with different thresholds.

```
#Optimal model
glmnew
```

```
##
## Call:  glm(formula = Survived ~ ., family = binomial, data = Xynew)
##
## Coefficients:
## (Intercept)       Pclass      Sexmale          Age        SibSp
##   7.9669180   -2.1765017   -5.9626444   -0.0275289   -0.3191225
##       Parch    EmbarkedQ    EmbarkedS     sqrtfare       sexage
##  -0.2445326   -0.1085956   -0.5057462   -0.0004177   -0.0138969
##   sexpclass     sexparch
##   1.3447956    0.6444724
##
## Degrees of Freedom: 890 Total (i.e. Null);  879 Residual
## Null Deviance:       1187
## Residual Deviance: 753.7     AIC: 777.7
```

```
#Tranform test data set according to my optimal model
sexage <- c() #let male becomes 1 and women becomes 0
sexage <- (as.numeric(test$Sex) - 1) * test$Age

sexpclass <- c()
sexpclass <- (as.numeric(test$Sex) - 1) * test$Pclass

sexparch <- c()
sexparch <- (as.numeric(test$Sex) - 1) * test$Parch

newtest <- data.frame(test, sexage = sexage, sexpclass = sexpclass, sexparch = sexparch)
```

```r
#Predict
conclusion <- c()
conclusion <- ifelse(predict(glmnew, newtest, type = "response") > which.min(error)*0.001, 1, 0)

answerbarca <- data.frame(PassengerId = seq(nrow(train) + 1, nrow(train) + nrow(test), 1),
                          Survived = conclusion)
head(answerbarca)
```

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

```r
write.csv(answerbarca, file = "JinKweon.csv", row.names = F, col.names = T)
```

```
## Warning in write.csv(answerbarca, file = "JinKweon.csv", row.names = F, :
## attempt to set 'col.names' ignored
```

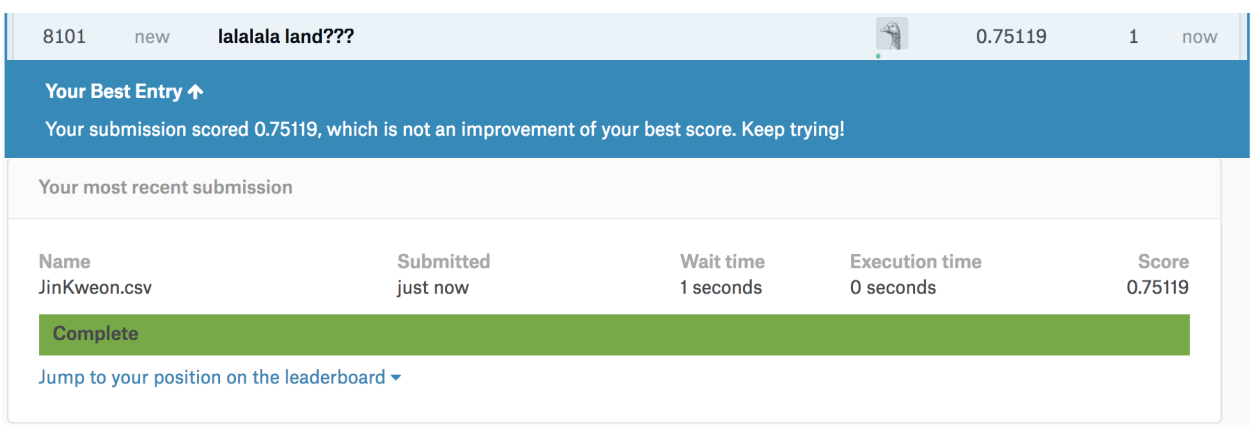
Comment:

For prediction, I assume training and testing sets are similar (from EDA at the beginning), I think the minimum misclassfication rate for testing set will occur at the similar point as the training set.

Prediction accuracy score

Comment:

I got 0.75119. . . .



Jin Kweon
SID: 3032235207
HW 5
Stat 151A

#2 This is the similar thing that we did before.

In linear regression, we use maximum likelihood and get first grdient being zero, and get a normal equation. For example, we get $X^T X \hat{\beta} = X^T y$, and $X^T(\hat{y} - y) = 0$. Thus, say that X and $\hat{e}$ are orthogonal, as the inner product is equal to zero.

I am going to do the similar thing.

The likelihood of y_i, for $i = 1, \cdots, n$ is $\prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i}$ where $p_i = \frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}}$.

Then $\log(\beta) = \sum_{i=1}^{n} \left[y_i \cdot (x_i^T \beta) - \log(1 + e^{x_i^T \beta}) \right]$.

Then, $\frac{d \log(\beta)}{d\beta_0} = \sum \left(y_i - \frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}} \right)$

$$\frac{d \log(\beta)}{d\beta_1} = \sum \left(y_i x_{i1} - \frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}} x_{i1} \right) = \sum \left[\left(y_i - \frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}} \right) x_{i1} \right]$$

$\vdots$

So, $\nabla \log(\beta) = \sum \left(y_i - \frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}} \right) (1, x_{i1}, \cdots, x_{ip})^T$

$$= \sum \underbrace{(y_i - p_i)}_{\text{number}} \underbrace{(1, \cdots, x_{ip})^T}_{\text{vector}}$$

$$= X^T(y - p)$$

So, from there, $X^T(y - \hat{p}) = 0 \Rightarrow \langle X, Y - \hat{p} \rangle = 0$.

And, $Y - \hat{p}$ is orthogonal to columns of the X matrix.

$\therefore$ proved

#3.)

a) we have proved in the class that $cov(\hat{\beta}) \approx (X^T W X)^{-1}$.
where X and W are both $n \times n$ matrix.

So, for Std. Error of Intercept is $\sqrt{4.8038479} \approx$

$\approx$ ① $\boxed{2.1918}$, as $(X^T W X)^{-1}$ is given.

So, for Std. Error of log(distance) is $\sqrt{0.043313307} \approx$

$\approx$ ② $\boxed{0.20812}$.

$AIC(m) = -2 \log$ (maximum value of likelihood in m)
$+2$ (# of parameters in m)

$= 222.18$.

Since there are four parameters including an intercept,

$-2 \log$ (max likelihood in m) $=$ ⑤ $\boxed{214.18}$ = Residual deviance

Residual deviance is $-2\left[\frac{n}{2}\log\left(\frac{RSS(m)}{n}\right) - \frac{n}{2}\log(2\pi e)\right]$ ⇒ (intuitively similar to linear)
and thus it is generalization of RSS(m).

and remember that its degree of freedom is $n-p-1$.

So, $n-p-1 = 212-3-1 =$ ⑥ $\boxed{208}$

Null deviance's d.f. is basically $n-1 = 212-1 =$ ④ $\boxed{211}$.

The Null deviance is $-2\sum_{i=1}^{n}[y_i \log \bar{y} + (1-y_i)\log(1-\bar{y})] = -2n[\bar{y}\log\bar{y} + (1-\bar{y})\log(1-\bar{y})]$

and we know that $y_i \sim Ber(p_i)$, So $E(y) = \bar{y} = \hat{\bar{p}} = \frac{79}{212}$.

So, if we plug $\bar{y} = \frac{79}{212}$, I got ③ $\boxed{279.987}$

c.t.
⟹

C.t.

$\Rightarrow$

The diagonal (of 4th) of $(X^T W X)^{-1} \approx cov(\hat{\beta})$,

and we already have std. Error for meanmin, 0.3131.

So, $(0.3131)^2 \approx$ ⑧ $\boxed{0.0980}$

The last one is $cov(\hat{\beta}_0, \hat{\beta}_2)$.

And since $(X^T W X)^{-1}$ is covariance matrix, it is symmetric.

So, it should be ⑦ $\boxed{-0.255928180}$

3-#b) So, what the question is asking, is to get the predicted probability, $\hat{p}_i$.

$$\hat{p}_i = \frac{e^{x_i^T \hat{\beta}}}{1 + e^{x_i^T \hat{\beta}}}$$ And we know that

$$\hat{\beta} = \begin{bmatrix} 0.6864 \\ -0.9050 \\ 0.5027 \\ 1.1153 \end{bmatrix}$$ and the new $x_i = [1 \quad \log(265) \quad \log(26) \quad 3.5]$.

So, $e^{x_i^T \hat{\beta}} \approx 3.2483255$, and thus, $\boxed{\hat{p}_i \approx 0.7646}$

3-#c)

First of all $\boxed{\text{Null deviance will stay the same}}$ (df. stays the same as well.). Null deviance is the model with the intercept only, so, it does not matter with how many other explanatory variables we have !!!

Second, $\boxed{\text{residual deviance should either decrease/stay the same}}$ (d.f. will be 207)

Intuitively residual deviance for linear model is $n \log\left(\frac{RSS(m)}{n}\right) + n \log(2\pi e)$, and when variable number increases, as RSS(m) goes down, residual deviance decreases....

$\Rightarrow$

C.t.

c.t.
$\Rightarrow$ we learned from AIC, that it is why we penalize by adding $2(1+p(m))$.

It is same for logistic model, as residual deviance do not penalize of adding variables, the adding variables always decreases / stay the same. is guaranteed, for residual deviance.

☆Also remember that residual deviance formula is $-2\sum[y_i \log \hat{p}_i + (1-y_i)\log(1-\hat{p}_i)]$

and y_i and $1-y_i$ is either 0 or 1,
and $\hat{p}_i$ and $1-\hat{p}_i$ is between 0 and 1,
and $\log \hat{p}_i$ and $\log(1-\hat{p}_i)$ is between $-\infty$ and 0.
so, as variable is added the model can never be worse
so when adding variables, as $\begin{cases} y_i \approx 1, \log \hat{p}_i \approx 0 \\ (1-y_i) \approx 0, \log(1-\hat{p}_i) \approx -\infty \end{cases}$ are more likely.

what I am trying to say is we try to find $\hat{\beta}$ where maximizing $\prod p_i^{y_i}(1-p_i)^{1-y_i}$ and as variable is added we have better chance to maximize it as p_i is a function of β.

4)

a) $y_i \overset{ind}{\sim} Ber(p_i)$ where $p_i = \dfrac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}}$.

So the likelihood of $y_1, \dots, y_n$ is $\prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i}$.

Then the log-likelihood is $l(\beta) = \sum (y_i \log p_i + (1-y_i)\log(1-p_i))$.

Since we cannot get maximizer of $l(\beta)$ in closed form, we use Newton-Raphson iteration: $\beta^{(m+1)} = \beta^{(m)} - (Hl(\beta^{(m)}))^{-1}\nabla l(\beta^{(m)})$.

And, we got $Hl(\beta) = -X^TWX$ and $\nabla l(\beta) = X^T(Y-P)$ in the lecture.

So, I ended up getting $\beta^{(m+1)} = \beta^{(m)} + (X^TWX)^{-1}X^TY - (X^TWX)^{-1}X^TP$

And, we learned in the class that we re-iterate it till it converges!!!

And so basically estimates of $\hat{\beta} = (X^TWX)^{-1}X^TWZ$ where

$$Z = X\beta^{(m)} + W^{-1}(Y-P)$$

$\therefore$ So, $\hat{\beta}$ depending on Y is only $(X^TWX)^{-1}X^TY$ term.

So it's only through X^TY.

So, we don't need the vector Y itself to find $\hat{\beta}$, as long as we have X^TY.

b) we know that $\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$ for our link function.

So, $p_i = \frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}}$ where $x_i^T = [1 \quad x_{i1} \quad \cdots \quad x_{ip}]$

So, $\hat{p}_i = \frac{e^{x_i^T\hat{\beta}}}{1+e^{x_i^T\hat{\beta}}} = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ip}}}{1+e^{\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ip}}}$

c) They are asking to prove $\sum_{i=1}^{n} \hat{p}_i =$ # of response values $y_1, \cdots, y_n$ that are equal to 1. $= \sum_{i=1}^{n} I(y_i = 1)$

So, basically, they are asking us to prove average of fitted probabilities/values $(= \bar{\hat{p}})$ equals the proportion (= average of # of y_i being 1) of ones in the response variable.

The best way is to use the similar proof I did/used for #1.

I proved in #1 that $\nabla l(\beta) = X^T(Y - \hat{p}) = 0$.

As from the model $\log\frac{p_i}{1-p_i} = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$,

I can tell that $\boxed{\text{intercept exists!!!}}$

So, I can say that $X^T Y = X^T \hat{p}$ implies that $\sum_{i=1}^{n} y_i = \sum_{i=1}^{n} \hat{p}_i$. ☆Remember that y_i is binary (0 or 1), so, $\sum y_i$ implies $\sum I(y_i = 1)$. Thus, $\boxed{\sum \hat{p}_i = \sum I(y_i = 1)}$ ☆

∴ proved

d) The definition of residual deviance is

$-2\log$(maximized value of likelihood in the model).

I found in part a) that $l(\beta) = \sum_{i=1}^{n} (y_i \log p_i + (1-y_i)\log(1-p_i))$.

So, residual deviance is $-2l(\hat{\beta})$ as we need to find the maximized value!!! (that is why we use a hat beta)

And, $-2l(\hat{\beta}) = -2\sum(y_i \log \hat{p}_i + (1-y_i)\log(1-\hat{p}_i))$

$$= -2\{[y_1 \log \hat{p}_1 + (1-y_1)\log(1-\hat{p}_1)] + [y_2 \log \hat{p}_2 + (1-y_2)\log(1-\hat{p}_2)] + \cdots [y_n \log \hat{p}_n + (1-y_n)\log(1-\hat{p}_n)]\}$$

#5.)

a) z-value is checking $\frac{\hat{\beta}_0 - 0}{\text{std.err}(\hat{\beta}_0)} = \frac{4.11947 - 0}{0.36342} \approx \boxed{11.335}$ ①

Similarly, $\frac{\text{xxxxxx} - 0}{\text{S.E.}(\hat{\beta}_4)} = \frac{\text{xxxxxx} - 0}{0.02800} \approx 12.345.$

So, it will be $\boxed{0.3456}$ ②

Again, similar with #3), the null deviance is based on the logistic model with the intercept only.

So, d.f. for Null deviance will be $n-1 = 4601 - 1 = \boxed{4600}$ ④

Also, d.f. for Residual deviance is $n-p-1 = 4601-6-1 =$

$= \boxed{4594}$ ⑤

Also, null deviance is $-2\sum(y_i \log \bar{y} + (1-y_i)\log(1-\bar{y})) =$

$= -2n\bar{y}\log\bar{y} - 2n\log(1-\bar{y}) + 2n\bar{y}\log(1-\bar{y})$

$= -2n[\bar{y}\log\bar{y} + (1-\bar{y})\log(1-\bar{y})]$

$\approx \boxed{6170.1528}$ ③, as $n = 4601$ and $\bar{y} = \frac{1813}{4601}$

Also, AIC $= -2\log$(max value of likelihood in model) $+ 2(1+p(m))$

and we know $-2\log$(max value of likelihood in model) $=$

$=$ Residual deviance $= 3245.1$

Also, $2(1+p(m)) = 14$, so, AIC $= 3245.1 + 14 = \boxed{3259.1}$ ⑥

#5.

b) Again, similar with #3), prediction can be computed with $\hat{p}_i = \dfrac{e^{x_i^T \hat{\beta}}}{1+e^{x_i^T \hat{\beta}}}$ for

$$x_i^T = [1 \quad \log(157) \quad \log(0.868+0.001) \quad \log(2.894+0.001) \quad \cdots \quad \log(0+0.001) \quad \log(0+0.001) \quad \log(0+0.001)].$$

and $\hat{\beta} = \begin{bmatrix} 4.11947 \\ 0.30228 \\ 0.32586 \\ 0.40984 \\ 0.34563 \\ 0.18947 \\ -0.11418 \end{bmatrix}$.

So, $e^{x_i^T \hat{\beta}}$ will be around 22.8779,

So $\boxed{\hat{p}_i \approx 0.95812}$

c)

☆ If I do this, I will get a warning message saying "fitted probability 0 or 1 occured" as I have a variable which separates zero and one perfectly. ☆

It means that iteration # will go increase and then, its model becomes really inefficient. (because they need much more iteration of Newton method to get beta hat....)

Similar to AIC, the smaller residual deviance, the better the model is.

(Also, M1 and M2 have the same # of variables, residual deviance can be interpreted as comparing AICs of two models.)

Thus, $\boxed{\text{Model } M_1 \text{ is better.}}$ ☆

Problem 3

As what instruction says, we are not allowed to use R for this problem. I am just using R to compare whether my answers are matched with what R outputs.

```r
data(frogs)
head(frogs)
```

```
##   pres.abs northing easting altitude distance NoOfPools NoOfSites   avrain
## 2        1      115    1047     1500      500       232         3 155.0000
## 3        1      110    1042     1520      250        66         5 157.6667
## 4        1      112    1040     1540      250        32         5 159.6667
## 5        1      109    1033     1590      250         9         5 165.0000
## 6        1      109    1032     1590      250        67         5 165.0000
## 7        1      106    1018     1600      500        12         4 167.3333
##    meanmin  meanmax
## 2 3.566667 14.00000
## 3 3.466667 13.80000
## 4 3.400000 13.60000
## 5 3.200000 13.16667
## 6 3.200000 13.16667
## 7 3.133333 13.06667
```

```r
frogs.glm <- glm(formula = pres.abs ~ log(distance) + log(NoOfPools) + meanmin,
                 family = binomial, data = frogs)
summary(frogs.glm)
```

```
##
## Call:
## glm(formula = pres.abs ~ log(distance) + log(NoOfPools) + meanmin,
##     family = binomial, data = frogs)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9642  -0.7657  -0.4619   0.8728   2.3219
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)      0.6864     2.1918   0.313 0.754146
## log(distance)   -0.9050     0.2081  -4.349 1.37e-05 ***
## log(NoOfPools)   0.5027     0.2004   2.509 0.012102 *
## meanmin          1.1153     0.3131   3.562 0.000369 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 279.99  on 211  degrees of freedom
## Residual deviance: 214.18  on 208  degrees of freedom
## AIC: 222.18
##
## Number of Fisher Scoring iterations: 5
```

```r
X = model.matrix(frogs.glm)
W = diag(frogs.glm$fitted.values*(1 - frogs.glm$fitted.values))
solve(t(X) %*% W %*% X)
```

```
##                (Intercept) log(distance) log(NoOfPools)     meanmin
## (Intercept)      4.8038479  -0.363947754   -0.255928180 -0.49698440
## log(distance)   -0.3639478   0.043313307    0.008053415  0.01562971
## log(NoOfPools)  -0.2559282   0.008053415    0.040141698  0.02678507
## meanmin         -0.4969844   0.015629708    0.026785069  0.09805609
```

```r
round(mean(frogs$pres.abs),6) == round(mean(frogs.glm$fitted.values),6)
```

```
## [1] TRUE
```

```r
x1 = 265
x2 = 26
x3 = 3.5
rrg = c(1, log(x1), log(x2), x3)
eta = sum(rrg*frogs.glm$coefficients)
prr = exp(eta)/(1 + exp(eta))
prr
```

```
## [1] 0.7645755
```

```r
frogs.glm2 <- glm(formula = pres.abs ~ log(distance) + log(NoOfPools) + meanmin + altitude,
                  family = binomial, data = frogs)
summary(frogs.glm2)
```

```
##
## Call:
## glm(formula = pres.abs ~ log(distance) + log(NoOfPools) + meanmin +
##     altitude, family = binomial, data = frogs)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9879  -0.7533  -0.2974   0.8049   2.6869
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -51.57596   14.48915  -3.560 0.000371 ***
## log(distance)   -0.76187    0.22759  -3.347 0.000815 ***
## log(NoOfPools)   0.55983    0.21126   2.650 0.008051 **
```

```
## meanmin          5.95163    1.39844   4.256 2.08e-05 ***
## altitude         0.02314    0.00633   3.655 0.000257 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 279.99  on 211  degrees of freedom
## Residual deviance: 198.74  on 207  degrees of freedom
## AIC: 208.74
##
## Number of Fisher Scoring iterations: 5
```

Problem 5

As what instruction says, we are not allowed to use R for this problem. I am just using R to compare whether my answers are matched with what R outputs.

```r
data(spam7)
head(spam7)
```

```
##   crl.tot dollar  bang money n000 make yesno
## 1     278  0.000 0.778  0.00 0.00 0.00     y
## 2    1028  0.180 0.372  0.43 0.43 0.21     y
## 3    2259  0.184 0.276  0.06 1.16 0.06     y
## 4     191  0.000 0.137  0.00 0.00 0.00     y
## 5     191  0.000 0.135  0.00 0.00 0.00     y
## 6      54  0.000 0.000  0.00 0.00 0.00     y
```

```r
s = 0.001
M1 <- glm(yesno~ log(crl.tot) + log(dollar+s) + log(bang+s)
          +log(money+s) + log(n000+s) + log(make+s),
          family=binomial, data=spam7)
summary(M1)
```

```
##
## Call:
## glm(formula = yesno ~ log(crl.tot) + log(dollar + s) + log(bang +
##     s) + log(money + s) + log(n000 + s) + log(make + s), family = binomial,
##     data = spam7)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1657  -0.4367  -0.2863   0.3609   2.7152
##
## Coefficients:
```

```
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)       4.11947    0.36342  11.335  < 2e-16 ***
## log(crl.tot)      0.30228    0.03693   8.185 2.71e-16 ***
## log(dollar + s)   0.32586    0.02365  13.777  < 2e-16 ***
## log(bang + s)     0.40984    0.01597  25.661  < 2e-16 ***
## log(money + s)    0.34563    0.02800  12.345  < 2e-16 ***
## log(n000 + s)     0.18947    0.02931   6.463 1.02e-10 ***
## log(make + s)    -0.11418    0.02206  -5.177 2.25e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6170.2  on 4600  degrees of freedom
## Residual deviance: 3245.1  on 4594  degrees of freedom
## AIC: 3259.1
##
## Number of Fisher Scoring iterations: 6
```

```r
x1 = 157
x2 = 0.868
x3 = 2.894
x4 = 0
x5 = 0
x6 = 0
rrg = c(1, log(x1), log(x2+s), log(x3+s),  log(x4+s), log(x5+s), log(x6+s))
eta = sum(rrg*M1$coefficients)
prr = exp(eta)/(1 + exp(eta))
prr
```

```
## [1] 0.9581203
```

```r
M2 <- glm(yesno ~ crl.tot + dollar + bang + money + n000 + make,
         family=binomial, data=spam7)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
summary(M2)
```

```
##
## Call:
## glm(formula = yesno ~ crl.tot + dollar + bang + money + n000 +
##     make, family = binomial, data = spam7)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -8.4904  -0.6153  -0.5816   0.4439   1.9323
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.700e+00  5.361e-02 -31.717  < 2e-16 ***
## crl.tot      6.917e-04  9.745e-05   7.098 1.27e-12 ***
## dollar       8.013e+00  6.175e-01  12.976  < 2e-16 ***
## bang         1.572e+00  1.115e-01  14.096  < 2e-16 ***
## money        2.142e+00  2.418e-01   8.859  < 2e-16 ***
## n000         4.149e+00  4.371e-01   9.492  < 2e-16 ***
```

```
## make         1.698e-02  1.434e-01   0.118    0.906
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6170.2  on 4600  degrees of freedom
## Residual deviance: 4058.8  on 4594  degrees of freedom
## AIC: 4072.8
##
## Number of Fisher Scoring iterations: 16
```