Project 2

Importing the data and saving it

```
bank <- read.csv(file = '/Users/trishalvarma/Desktop/bank-full.csv')
```

Data Cleaning

Website: https://archive.ics.uci.edu/ml/datasets/bank+marketing From: "UCI Machine Learning Repositor" — Used any(is.na(data)) to see if there are any data missing from the columns. Then used na.omit, because data can be filled with NA and not show up as a blank data. Eliminates missing values. Provided the link from where the data is downloaded.

Steps taken are listed above, to remove any blank/na values. #Data Cleanup

```
bank$default <- NULL
bank$day <- NULL

bank$duration <- NULL
bank$pdays <- NULL
bank$previous <- NULL
bank$poutcome <- NULL

bank$y <- NULL
bank$contact <- NULL
str(bank)

## 'data.frame':    45211 obs. of  9 variables:
##  $ age      : int  58 44 33 47 33 35 28 42 58 43 ...
##  $ job      : chr  "management" "technician" "entrepreneur" "blue-collar"
...
##  $ marital  : chr  "married" "single" "married" "married" ...
##  $ education: chr  "tertiary" "secondary" "secondary" "unknown" ...
##  $ balance  : int  2143 29 2 1506 1 231 447 2 121 593 ...
##  $ housing  : chr  "yes" "yes" "yes" "yes" ...
##  $ loan     : chr  "no" "no" "yes" "no" ...
##  $ month    : chr  "may" "may" "may" "may" ...
##  $ campaign : int  1 1 1 1 1 1 1 1 1 1 ...
```

Grouping bank and bank job so we can use it as combined.

```
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
```

```r
group_by(bank, bank$job)
```

```
## # A tibble: 45,211 x 10
## # Groups:   bank$job [12]
##      age job   marital education balance housing loan  month campaign
`bank$job`
##    <int> <chr> <chr>   <chr>       <int> <chr>   <chr> <chr>    <int>
<chr>
##  1    58 mana… married tertiary     2143 yes     no    may          1
management
##  2    44 tech… single  secondary      29 yes     no    may          1
technician
##  3    33 entr… married secondary       2 yes     yes   may          1
entrepren…
##  4    47 blue… married unknown      1506 yes     no    may          1
blue-coll…
##  5    33 unkn… single  unknown         1 no      no    may          1
unknown
##  6    35 mana… married tertiary      231 yes     no    may          1
management
##  7    28 mana… single  tertiary      447 yes     yes   may          1
management
##  8    42 entr… divorc… tertiary        2 yes     no    may          1
entrepren…
##  9    58 reti… married primary       121 yes     no    may          1
retired
## 10    43 tech… single  secondary     593 yes     no    may          1
technician
## # … with 45,201 more rows
```

```r
bank$campaign <- NULL
```

Creating factors of the variables we will be using.

We also removed the outliers from the bank dataset.

```r
bank$education <- as.factor(bank$education)
bank$marital <- as.factor(bank$marital)
bank$job <- as.factor(bank$job)
bank$housing <- as.factor(bank$housing)
bank$loan <- as.factor(bank$loan)

outliers <- boxplot(bank$age, plot=FALSE)$out
```

```
bank <- bank[-which(bank$age %in% outliers),]
nrow(bank)

## [1] 44724

outliers1 <- boxplot(bank$balance, plot=FALSE)$out
bank <- bank[-which(bank$balance %in% outliers1),]
nrow(bank)

## [1] 40028
```

Data Exploration

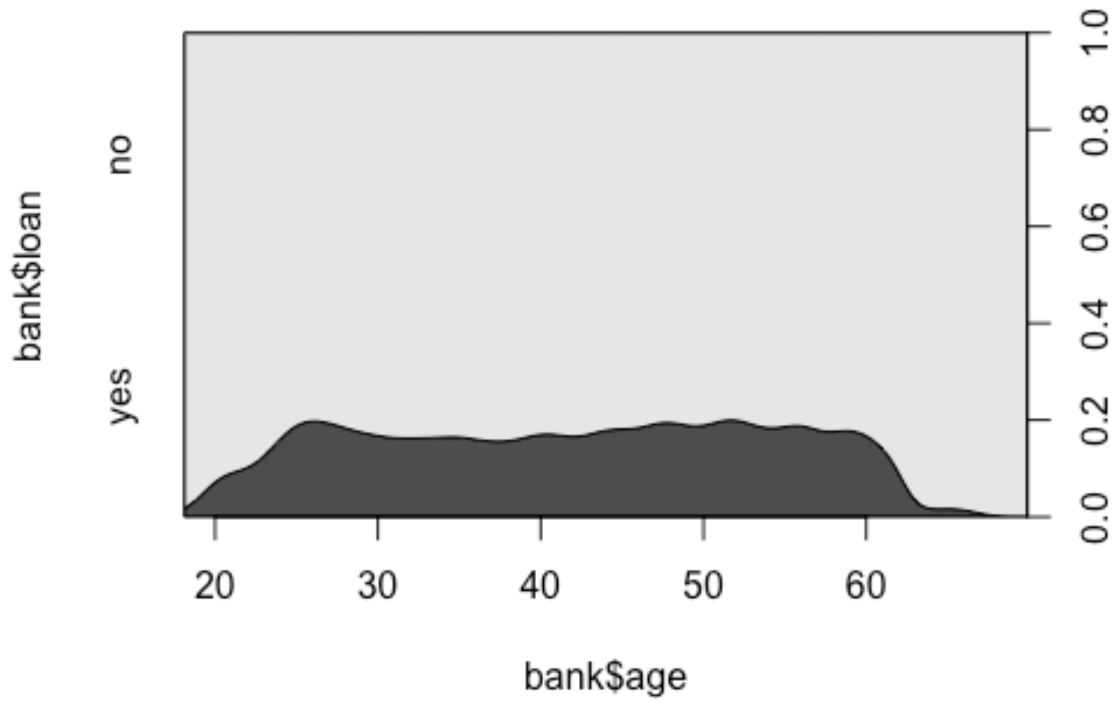
Using more than 5 functions to look at the Air Quality data, then providing a informative R graphs using a historgram, and a plot chart to show the just the sheer amount of data that was collected.

```
tapply(bank$age, bank$job, mean)

##        admin.   blue-collar  entrepreneur     housemaid    management
##      38.95821      39.82201      41.78245      45.95375      40.12708
##       retired self-employed      services       student    technician
##      57.76081      40.02168      38.54024      26.46659      39.17989
##    unemployed       unknown
##      40.68031      46.92946

# then we also check on balance, and education
tapply(bank$balance, bank$education, mean)

##   primary secondary  tertiary   unknown
##  601.0054  589.1953  713.5273  698.7475

# we can create a table and combine the martial status and the job they have.
library(gmodels)
bn.table <- table(bank$marital, bank$job)
bn.table

##
##             admin. blue-collar entrepreneur housemaid management retired
##   divorced    706         702          166       154        966     287
##   married    2412        6395          944       800       4591    1197
##   single     1596        1836          223       127       2556      88
##
##             self-employed services student technician unemployed unknown
##   divorced            113      515       6        865        154      13
##   married             873     2200      49       3602        640     168
##   single              398     1124     783       2365        354      60
```

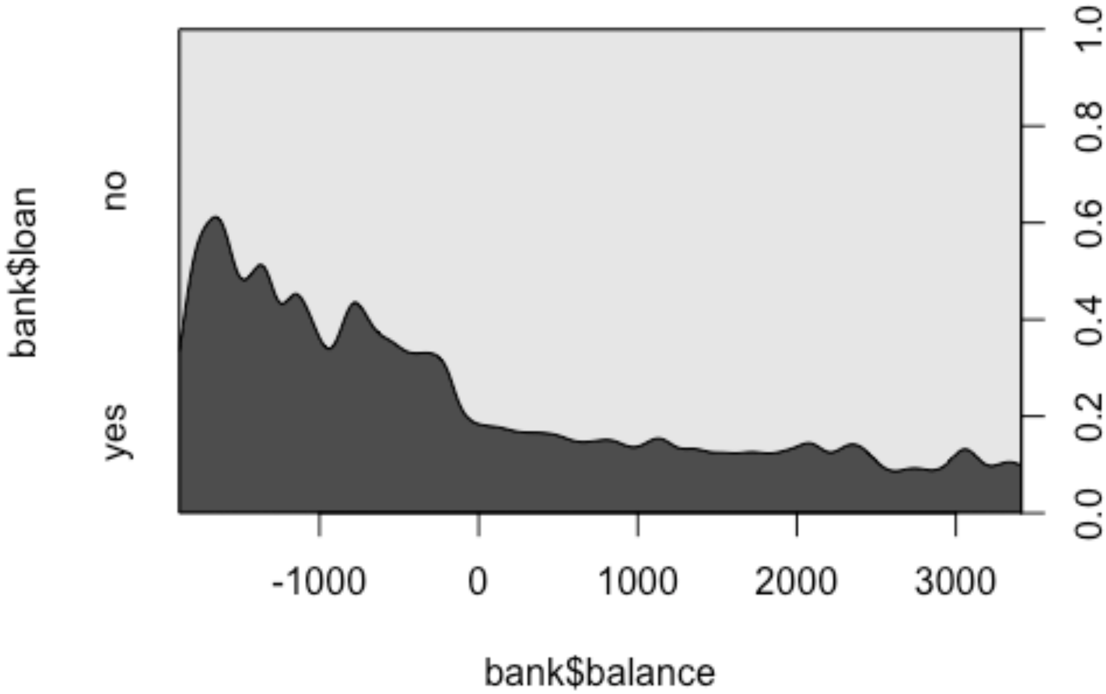
Now we can see visual data exploration.

Here we are able to see the loan and age taken, and the loan and balance takent.

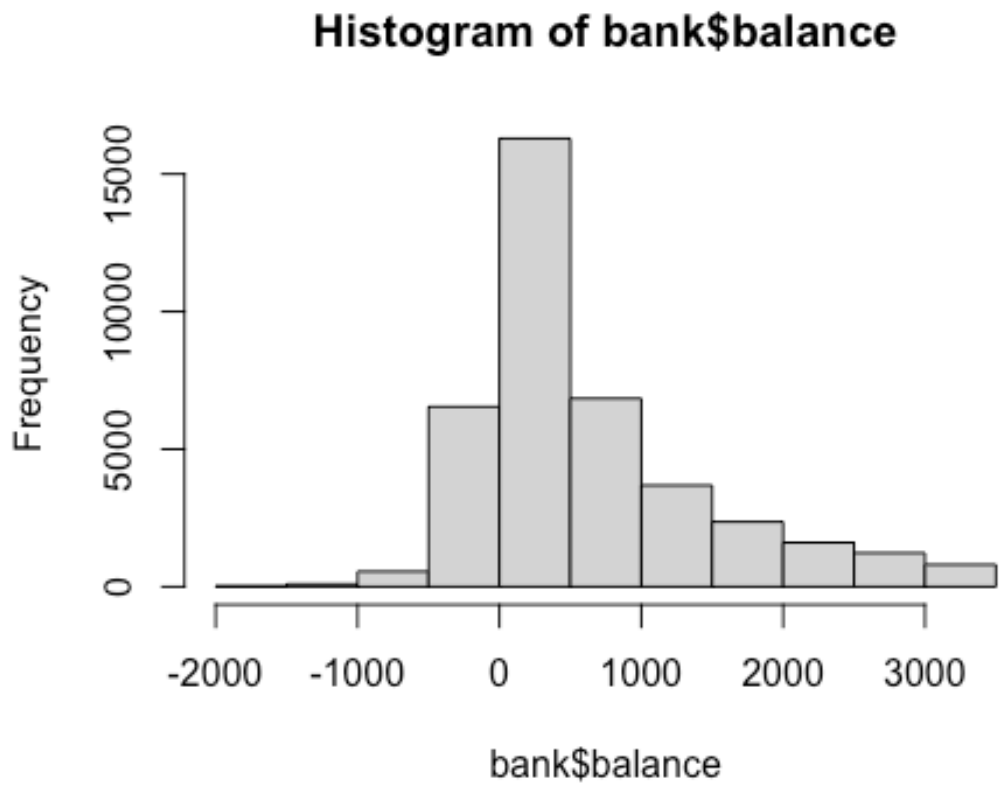
```r
cdplot(bank$loan~bank$age)
```




```r
cdplot(bank$loan~ bank$balance)
```




```r
# second exploration visually is the histogram

hist(bank$balance) #Here we notice that there are a few that shows negative
balance, and could effect the algorithms we will be running.
```



Lets create a test and a training data set we create 2 training and 2 test, just in case.

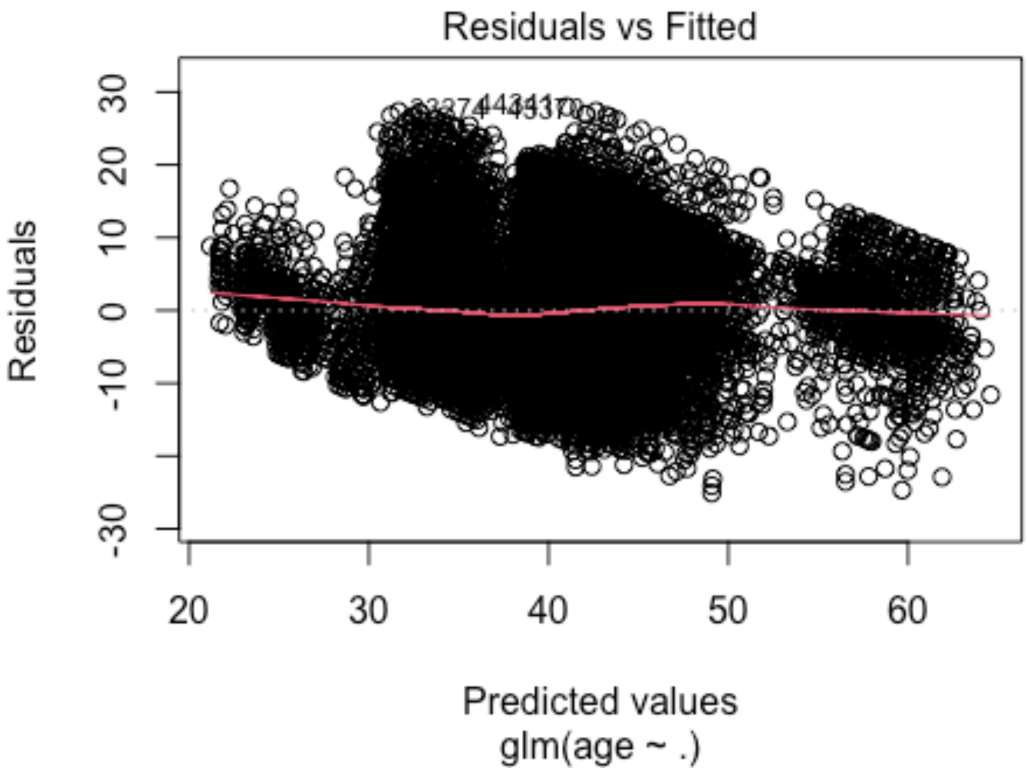
```
set.seed(1234)
bank1 <- bank
bank1 <- na.omit(bank1)

sample_i <- sample(1:nrow(bank1), .80*nrow(bank1), replace=FALSE)
train <- bank1[sample_i,]
test <- bank1[-sample_i,]

train1 <- bank1[sample_i,]
test1 <- bank1[-sample_i,]
```

For classification algorithm, we will run a Logistical regression. Even though it is a regression, it is actually a clasficiation algorithm.

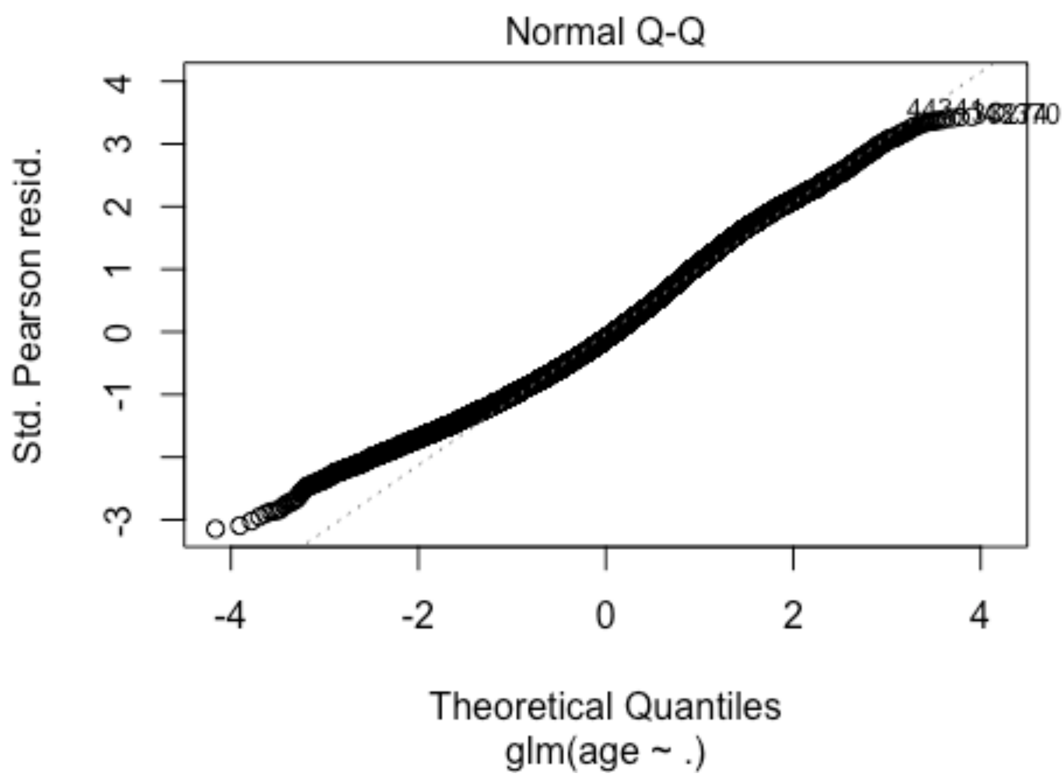
```
gg <- glm(age~., data = train)
summary(gg)

##
## Call:
## glm(formula = age ~ ., data = train)
##
## Deviance Residuals:
```

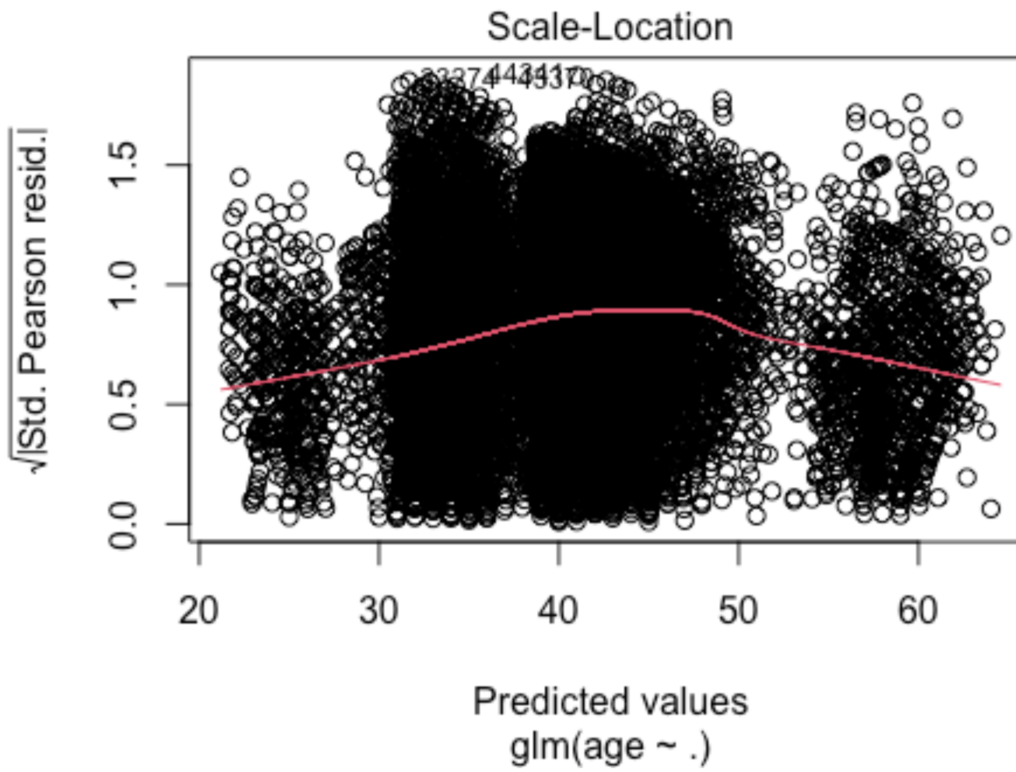
```
##      Min        1Q    Median        3Q       Max
## -25.0823   -5.9086   -0.8489    5.3417   27.9973
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          4.680e+01  2.941e-01 159.139  < 2e-16 ***
## jobblue-collar      -9.911e-01  1.693e-01  -5.855 4.82e-09 ***
## jobentrepreneur      1.508e+00  2.846e-01   5.299 1.17e-07 ***
## jobhousemaid         2.754e+00  3.107e-01   8.865  < 2e-16 ***
## jobmanagement        1.422e+00  1.902e-01   7.474 8.01e-14 ***
## jobretired           1.460e+01  2.664e-01  54.799  < 2e-16 ***
## jobself-employed     5.632e-01  2.808e-01   2.005   0.0449 *
## jobservices         -9.835e-01  1.947e-01  -5.052 4.39e-07 ***
## jobstudent          -9.244e+00  3.473e-01 -26.620  < 2e-16 ***
## jobtechnician        7.339e-02  1.714e-01   0.428   0.6686
## jobunemployed        3.260e-01  2.966e-01   1.099   0.2717
## jobunknown           3.866e+00  5.852e-01   6.607 4.00e-11 ***
## maritalmarried      -2.138e+00  1.445e-01 -14.794  < 2e-16 ***
## maritalsingle       -9.605e+00  1.581e-01 -60.747  < 2e-16 ***
## educationsecondary -3.024e+00  1.432e-01 -21.120  < 2e-16 ***
## educationtertiary  -4.365e+00  1.802e-01 -24.223  < 2e-16 ***
## educationunknown     3.319e-01  2.597e-01   1.278   0.2012
## balance              6.935e-04  5.485e-05  12.643  < 2e-16 ***
## housingyes          -2.103e+00  1.067e-01 -19.705  < 2e-16 ***
## loanyes             -1.168e-01  1.215e-01  -0.961   0.3365
## monthaug             1.347e+00  2.236e-01   6.025 1.71e-09 ***
## monthdec             1.592e-01  7.241e-01   0.220   0.8259
## monthfeb             3.761e-01  2.572e-01   1.463   0.1436
## monthjan             4.482e-01  3.064e-01   1.462   0.1436
## monthjul             2.936e-01  2.119e-01   1.386   0.1658
## monthjun             1.115e+00  2.224e-01   5.016 5.31e-07 ***
## monthmar             2.088e-02  5.029e-01   0.042   0.9669
## monthmay             1.706e-01  1.940e-01   0.880   0.3791
## monthnov             1.352e+00  2.413e-01   5.604 2.12e-08 ***
## monthoct             1.704e+00  4.230e-01   4.028 5.63e-05 ***
## monthsep            -1.072e-01  4.640e-01  -0.231   0.8173
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 63.64874)
##
##     Null deviance: 3152156  on 32021  degrees of freedom
## Residual deviance: 2036187  on 31991  degrees of freedom
## AIC: 223907
##
## Number of Fisher Scoring iterations: 2
```

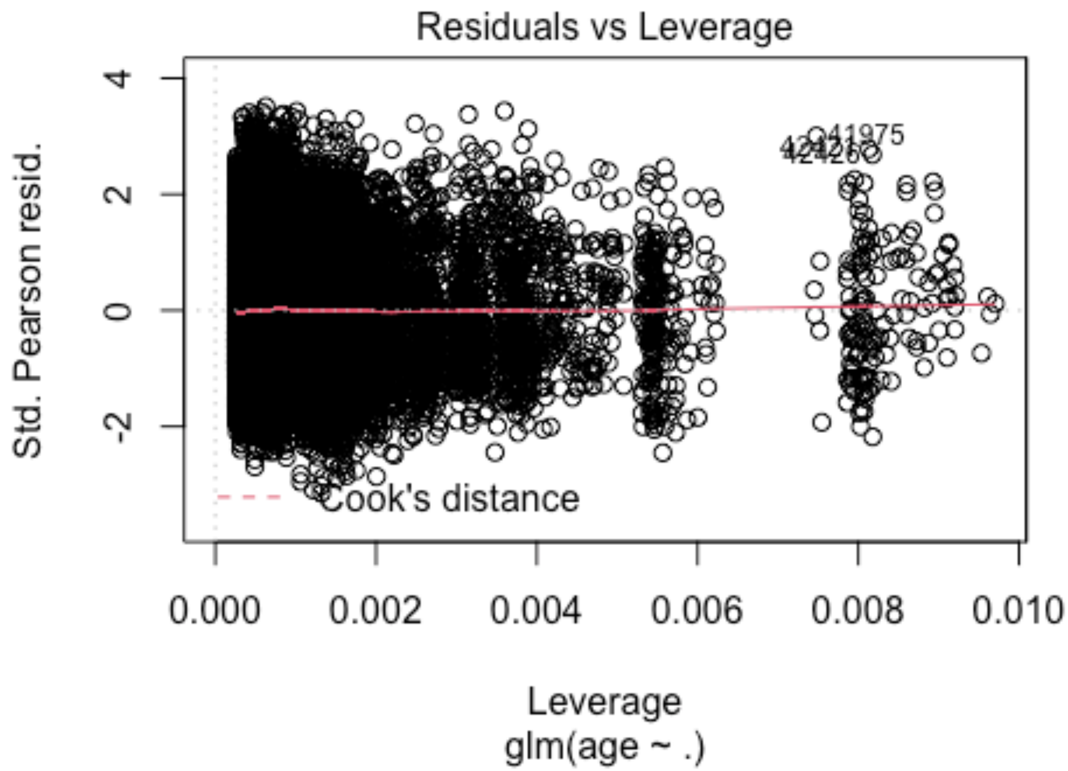
```
plot(gg)
```



Residuals vs Fitted
Residuals
Predicted values
glm(age ~ .)


Normal Q-Q
Std. Pearson resid.
Theoretical Quantiles
glm(age ~ .)


Scale-Location
√|Std. Pearson resid.|
0.0
0.5
1.0
1.5
20
30
40
50
60
Predicted values
glm(age ~ .)




```
pred1 <- predict(gg, newdata = test)

cor1 <- cor(pred1, test$age)
mse1 <- mean((pred1 - test$age)^2)

print(paste("MSE:  ", mse1))

## [1] "MSE:   61.6497266522616"

print(paste("Corrleation: ", cor1))

## [1] "Corrleation:  0.595206875117885"
```

Not bad, but not good, we have a MSE of 61, and a correlation of 59.52 but almost 60. So we'll keep that in mind for later.

#Let's run our second algorithm kNN, and this time using test1 and train1.

```
test1$housing <- as.numeric(test1$housing)
train1$housing <- as.numeric(train1$housing)

test1$age <- as.numeric(test1$age)
train1$age <- as.numeric(train1$age)
```

```
test1$job <- as.numeric(test1$job)
train1$job <- as.numeric(train1$job)

test1$marital <- as.numeric(test1$marital)
train1$marital <- as.numeric(train1$marital)

test1$education <- as.numeric(test1$education)
train1$education <- as.numeric(train1$education)

test1$balance <- as.numeric(test1$balance)
train1$balance <- as.numeric(train1$balance)

test1$loan <- as.numeric(test1$loan)
train1$loan <-as.numeric(train1$loan)

library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(lattice)
library(ggplot2)

knn_fit <- knnreg(age~., data = test1, k = 5)
pred_knn <- predict(knn_fit, test1)

cor_knn <- cor(pred_knn, test1$age)
mse_knn <- mean((pred_knn - test1$age)^2)

print(paste("Correlation for KNN: ", knn_fit))

## [1] "Correlation for KNN:  list(y = c(`3` = 33, `5` = 33, `20` = 33, `21`
= 28, `22` = 56, `28` = 52, `29` = 46, `30` = 36, `37` = 25, `40` = 37, `41`
= 44, `49` = 55, `59` = 40, `63` = 57, `68` = 59, `72` = 31, `75` = 43, `78`
= 55, `91` = 42, `94` = 60, `96` = 36, `98` = 60, `102` = 53, `103` = 52,
`104` = 59, `109` = 59, `112` = 46, `116` = 44, `117` = 41, `118` = 33, `120`
= 57, `122` = 51, `126` = 33, `131` = 55, `132` = 32, `133` = 38, `141` = 53,
`149` = 43, `151` = 51, `158` = 60, `159` = 52, `176` = 53, `178` = 34,
\n`185` = 36, `191` = 51, `197` = 38, `203` = 44, `213` = 59, `220` = 39,
`226` = 48, `229` = 36, `236` = 45, `248` = 40, `249` = 39, `252` = 53, `258`
= 30, `270` = 40, `271` = 42, `274` = 56, `279` = 38, `281` = 50, `289` = 32,
`290` = 40, `295` = 34, `302` = 51, `304` = 36, `313` = 55, `317` = 36, `318`
= 51, `324` = 32, `333` = 30, `337` = 42, `341` = 41, `347` = 45, `350` = 55,
`354` = 36, `361` = 48, `362` = 42, `363` = 27, `372` = 38, `374` = 25, `379`
= 33, `390` = 58, `394` = 27, `400` = 47, \n`402` = 48, `405` = 52, `411` =
32, `414` = 48, `421` = 49, `428` = 54, `432` = 42, `435` = 32, `446` = 51,
`451` = 50, `460` = 35, `462` = 39, `477` = 50, `481` = 51, `482` = 41, `486`
= 54, `489` = 47, `495` = 30, `504` = 29, `507` = 57, `512` = 34, `518` = 44,
= 27, `8696` = 34, `8698` = 26, `8718` = 29, `8722` = 38, `8728` = 34, `8729`
```

```
= 42, `8738` = 54, `8742` = 41, `8746` = 33, `8752` = 59, `8759` = 28, `8760`
= 29, `8767` = 29, `8780` = 34, `8783` = 29, `8784` = 26, `8789` = 22, `8801`
= 21, `8809` = 26, `8811` = 24, `8816` = 52, `8822` = 29, `8823` = 29, `8834`
= 32, `8835` = 36, `8836` = 47, `8839` = 40, `8843` = 57, `8847` = 36, `8848`
= 45, `8849` = 30, `8862` = 35, `8876` = 39, `8879` = 53, `8891` = 46, `8894`
## [2] "Correlation for KNN:  5"
## [3] "Correlation for KNN:  age ~ job + marital + education + balance +
housing + loan + month"
## [4] "Correlation for KNN:  list()"
## [5] "Correlation for KNN:  list()"
```

```
print(paste("MSE for KNN: ", mse_knn))
```

```
## [1] "MSE for KNN:  67.6595729200846"
```

This prints out a long data, but we can see that the mse is 67, adn the correlation is 5. kNN might not be the best sutated for this data.

Lets run a random forest and see our tree

```
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## 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
```

```
set.seed(1234)
tree <- randomForest(age~., data=test, importance = TRUE)
tree
```

```
##
## Call:
##  randomForest(formula = age ~ ., data = test, importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 60.52472
##                     % Var explained: 36.6
```

#Analysis

Ranking of algorithms

1) random Forest
2) Linear Regression
3) kNN

In this case as well, we have random Forest being top, because of the number of trees it can create. We can increase the number of tries, but 1 is sufficient to know that it performed well. The mean residuals. Lieanr regression also gave us very good number, as commented above, and kNN did give us a better MSE, however it was very messy to deal with. #the data for kNN was edited so it could fit in the pdf and not be long. The time it took for the kNN as well on such a large data also effected it perfomrance.

We were able to look at the age and loan amount for people and in this case the random forrest would be the best to analysize the data.

What we learned; EX: If people older age have higher balanace, then it is likely they would be approved for loans, whereas, a lower balance made it a risk for loans. It was defintely a good data to work with, the more we learn, the more we can work with.