Bank Marketing

Marzuq Khan and Kartik Vaish

12/07/2019

The Data Set

We used the professor's link; http://archive.ics.uci.edu/ml/index.php, where we found this data set on bank marketing at this link; http://archive.ics.uci.edu/ml/datasets/Bank+Marketing. The summary is that, "The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed". We look at the full data set with 41,188 observations and 21 total variables, ordered by date from May 2008 to November 2010.

```
library(data.table)
library(tidyverse)
library(lubridate)
library(perturb)
library(car)
library(pROC)
library(tree)
library(ISLR)
library(class)
library(MASS)
BankM <- fread("data/bank-additional-full.csv")
glimpse(BankM)
```

```
## Observations: 41,188
## Variables: 21
## $ age            <int> 56, 57, 37, 40, 56, 45, 59, 41, 24, 25, 41, 25,...
## $ job            <chr> "housemaid", "services", "services", "admin.", ...
## $ marital        <chr> "married", "married", "married", "married", "ma...
## $ education      <chr> "basic.4y", "high.school", "high.school", "basi...
## $ default        <chr> "no", "unknown", "no", "no", "no", "unknown", "...
## $ housing        <chr> "no", "no", "yes", "no", "no", "no", "no", "no"...
## $ loan           <chr> "no", "no", "no", "no", "yes", "no", "no", "no"...
## $ contact        <chr> "telephone", "telephone", "telephone", "telepho...
## $ month          <chr> "may", "may", "may", "may", "may", "may", "may"...
## $ day_of_week    <chr> "mon", "mon", "mon", "mon", "mon", "mon", "mon"...
## $ duration       <int> 261, 149, 226, 151, 307, 198, 139, 217, 380, 50...
## $ campaign       <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ pdays          <int> 999, 999, 999, 999, 999, 999, 999, 999, 999, 99...
## $ previous       <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ poutcome       <chr> "nonexistent", "nonexistent", "nonexistent", "n...
## $ emp.var.rate   <dbl> 1.1, 1.1, 1.1, 1.1, 1.1, 1.1, 1.1, 1.1, 1.1, 1....
## $ cons.price.idx <dbl> 93.994, 93.994, 93.994, 93.994, 93.994, 93.994,...
## $ cons.conf.idx  <dbl> -36.4, -36.4, -36.4, -36.4, -36.4, -36.4, -36.4...
## $ euribor3m      <dbl> 4.857, 4.857, 4.857, 4.857, 4.857, 4.857, 4.857...
## $ nr.employed    <dbl> 5191, 5191, 5191, 5191, 5191, 5191, 5191, 5191,...
## $ y              <chr> "no", "no", "no", "no", "no", "no", "no", "no",...
```

7 Variables Related to Bank-Client Data:

1 - age (numeric)

2 - job : type of job (categorical: 'admin.', 'blue-collar', 'entrepreneur', 'housemaid', 'management', 'retired', 'self-employed', 'services', 'student', 'technician', 'unemployed', 'unknown')

3 - marital : marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)

4 - education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unkno

5 - default: has credit in default? (categorical: 'no','yes','unknown')

6 - housing: has housing loan? (categorical: 'no','yes','unknown')

7 - loan: has personal loan? (categorical: 'no','yes','unknown')

4 Variables Related to the Last Contact of the Current Campaign:

8 - contact: contact communication type (categorical: 'cellular','telephone')

9 - month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')

10 - day_of_week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')

11 - duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

4 Variables Related to Other Attributes:

12 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)

13 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)

14 - previous: number of contacts performed before this campaign and for this client (numeric)

15 - poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')

5 Social and Economic Variables

16 - emp.var.rate: employment variation rate - quarterly indicator (numeric)

17 - cons.price.idx: consumer price index - monthly indicator (numeric)

18 - cons.conf.idx: consumer confidence index - monthly indicator (numeric)

19 - euribor3m: euribor 3 month rate - daily indicator (numeric)

20 - nr.employed: number of employees - quarterly indicator (numeric)

Output variable

21 - y - has the client subscribed a term deposit? (binary: 'yes','no')

Cleaning the Data Set

Fixing date values

```
BankM %>%
  mutate(month = str_to_title(month)) -> BankM
parse_factor(BankM$month, levels = month.abb) -> BankM$month
parse_factor(BankM$day_of_week) -> BankM$day_of_week
glimpse(BankM)
```

```
## Observations: 41,188
## Variables: 21
## $ age            <int> 56, 57, 37, 40, 56, 45, 59, 41, 24, 25, 41, 25,...
## $ job            <chr> "housemaid", "services", "services", "admin.", ...
## $ marital        <chr> "married", "married", "married", "married", "ma...
## $ education      <chr> "basic.4y", "high.school", "high.school", "basi...
## $ default        <chr> "no", "unknown", "no", "no", "no", "unknown", "...
## $ housing        <chr> "no", "no", "yes", "no", "no", "no", "no", "no"...
## $ loan           <chr> "no", "no", "no", "no", "yes", "no", "no", "no"...
## $ contact        <chr> "telephone", "telephone", "telephone", "telepho...
## $ month          <fct> May, May, May, May, May, May, May, May, May, Ma...
## $ day_of_week    <fct> mon, mon, mon, mon, mon, mon, mon, mon, mon, mo...
## $ duration       <int> 261, 149, 226, 151, 307, 198, 139, 217, 380, 50...
## $ campaign       <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ pdays          <int> 999, 999, 999, 999, 999, 999, 999, 999, 999, 99...
## $ previous       <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ poutcome       <chr> "nonexistent", "nonexistent", "nonexistent", "n...
## $ emp.var.rate   <dbl> 1.1, 1.1, 1.1, 1.1, 1.1, 1.1, 1.1, 1.1, 1.1, 1....
## $ cons.price.idx <dbl> 93.994, 93.994, 93.994, 93.994, 93.994, 93.994,...
## $ cons.conf.idx  <dbl> -36.4, -36.4, -36.4, -36.4, -36.4, -36.4, -36.4...
## $ euribor3m      <dbl> 4.857, 4.857, 4.857, 4.857, 4.857, 4.857, 4.857...
## $ nr.employed    <dbl> 5191, 5191, 5191, 5191, 5191, 5191, 5191, 5191,...
## $ y              <chr> "no", "no", "no", "no", "no", "no", "no", "no",...
```

Changing characters to factors

```
BankM$job <- as.factor(BankM$job)
BankM$marital <- as.factor(BankM$marital)
BankM$education <- as.factor(BankM$education)
BankM$default <- as.factor(BankM$default)
BankM$housing <- as.factor(BankM$housing)
BankM$loan <- as.factor(BankM$loan)
BankM$contact <- as.factor(BankM$contact)
BankM$poutcome <- as.factor(BankM$poutcome)
BankM$y <- as.factor(BankM$y)
glimpse(BankM)
```

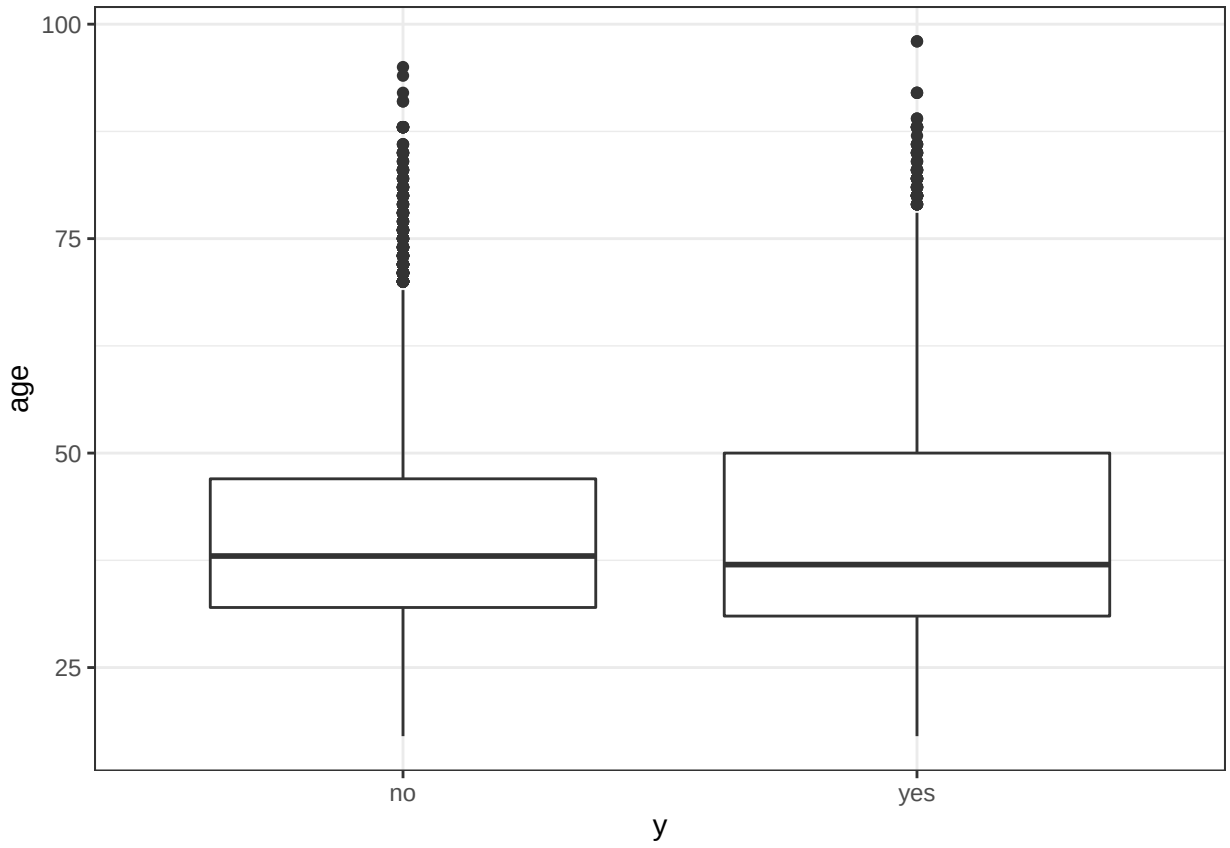
```
## Observations: 41,188
## Variables: 21
```

```
## $ age            <int> 56, 57, 37, 40, 56, 45, 59, 41, 24, 25, 41, 25,...
## $ job            <fct> housemaid, services, services, admin., services...
## $ marital        <fct> married, married, married, married, married, ma...
## $ education      <fct> basic.4y, high.school, high.school, basic.6y, h...
## $ default        <fct> no, unknown, no, no, no, unknown, no, unknown, ...
## $ housing        <fct> no, no, yes, no, no, no, no, no, yes, yes, no, ...
## $ loan           <fct> no, no, no, no, yes, no, no, no, no, no, no, no...
## $ contact        <fct> telephone, telephone, telephone, telephone, tel...
## $ month          <fct> May, May, May, May, May, May, May, May, May, Ma...
## $ day_of_week    <fct> mon, mon, mon, mon, mon, mon, mon, mon, mon, mo...
## $ duration       <int> 261, 149, 226, 151, 307, 198, 139, 217, 380, 50...
## $ campaign       <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ pdays          <int> 999, 999, 999, 999, 999, 999, 999, 999, 999, 99...
## $ previous       <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ poutcome       <fct> nonexistent, nonexistent, nonexistent, nonexist...
## $ emp.var.rate   <dbl> 1.1, 1.1, 1.1, 1.1, 1.1, 1.1, 1.1, 1.1, 1.1, 1....
## $ cons.price.idx <dbl> 93.994, 93.994, 93.994, 93.994, 93.994, 93.994,...
## $ cons.conf.idx  <dbl> -36.4, -36.4, -36.4, -36.4, -36.4, -36.4, -36.4...
## $ euribor3m      <dbl> 4.857, 4.857, 4.857, 4.857, 4.857, 4.857, 4.857...
## $ nr.employed    <dbl> 5191, 5191, 5191, 5191, 5191, 5191, 5191, 5191,...
## $ y              <fct> no, no, no, no, no, no, no, no, no, no, no, no,...
```

Exploratory Data Analysis(EDA)

Firstly, looking at the age difference between clients that subscribe, and don't subscribe to the term deposits

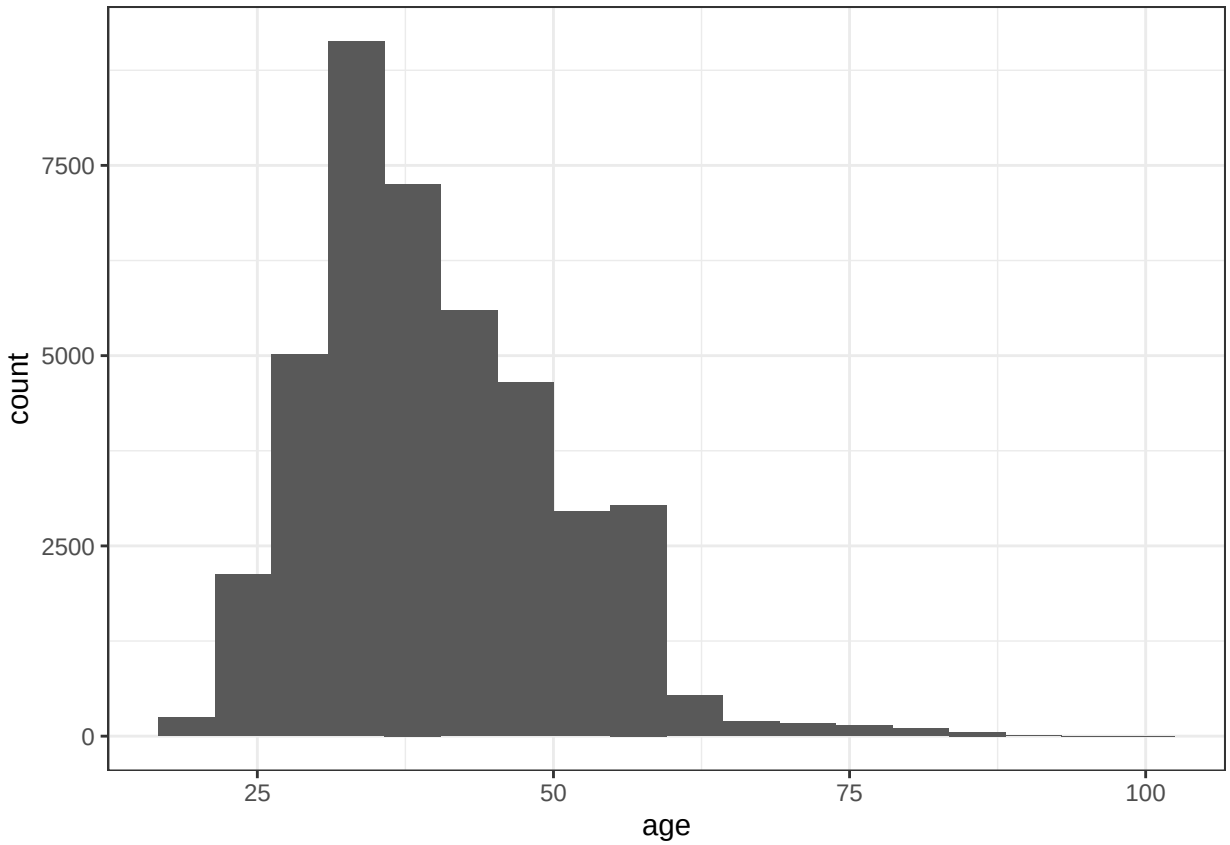
```
BankM %>%
  ggplot(aes(x=y, y=age)) +
  geom_boxplot() +
  theme_bw()
```



Based on the plot above, it is difficult to notice any differences. On a closer look, we can see that range varies for the subcribers, more than the people who do not subscribe.

Let's look at a histogram of age

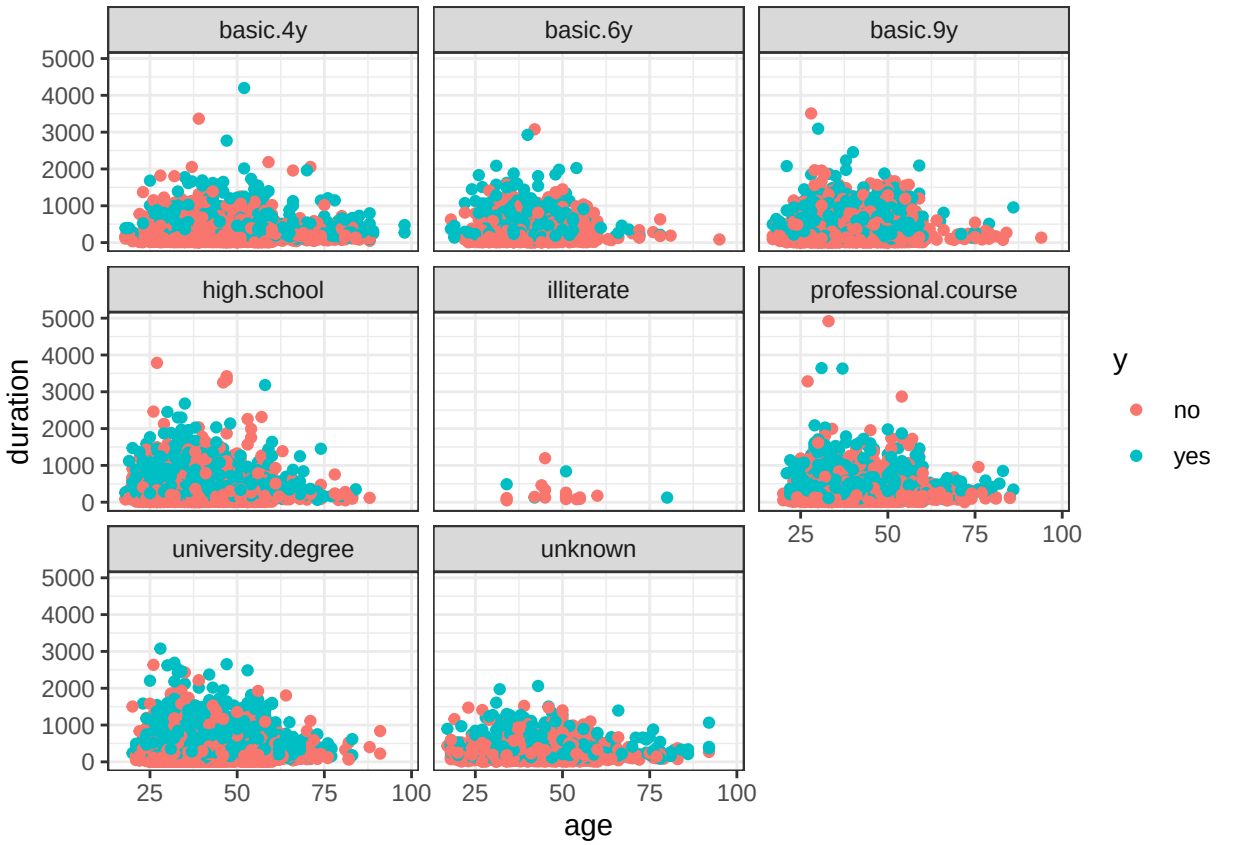
```
BankM %>%
  ggplot(aes(x=age)) +
  geom_histogram(bins = 18) +
  theme_bw()
```



Based on the histogram above, the majority of the clients seem to be aged between 25 and 60.

Scatter Plot on Last Contact Duration and Age, Facetted by Education

```
BankM %>%
  ggplot(aes(x=age, y=duration, color = y)) +
  facet_wrap( ~ education) +
  geom_point() +
  theme_bw()
```



From the plot above, it is difficult to derive any insights, except for the fact that the the majorty of people who subcribe or not, are not illiterate.

Summary of Dataset

```
summary(BankM)
```

```
##       age                  job            marital
##  Min.   :17.00   admin.     :10422   divorced: 4612
##  1st Qu.:32.00   blue-collar: 9254   married :24928
##  Median :38.00   technician : 6743   single  :11568
##  Mean   :40.02   services   : 3969   unknown :   80
##  3rd Qu.:47.00   management : 2924
##  Max.   :98.00   retired    : 1720
##                  (Other)    : 6156
##                education        default         housing
##  university.degree  :12168   no     :32588   no     :18622
##  high.school        : 9515   unknown: 8597   unknown:  990
##  basic.9y           : 6045   yes    :    3   yes    :21576
##  professional.course: 5243
##  basic.4y           : 4176
##  basic.6y           : 2292
##  (Other)            : 1749
##       loan            contact          month       day_of_week
```

```
##  no     :33950   cellular :26144   May    :13769   mon:8514
##  unknown:  990   telephone:15044   Jul    : 7174   tue:8090
##  yes    : 6248                     Aug    : 6178   wed:8134
##                                    Jun    : 5318   thu:8623
##                                    Nov    : 4101   fri:7827
##                                    Apr    : 2632
##                                    (Other): 2016
##     duration         campaign          pdays          previous
##  Min.   :   0.0   Min.   : 1.000   Min.   :  0.0   Min.   :0.000
##  1st Qu.: 102.0   1st Qu.: 1.000   1st Qu.:999.0   1st Qu.:0.000
##  Median : 180.0   Median : 2.000   Median :999.0   Median :0.000
##  Mean   : 258.3   Mean   : 2.568   Mean   :962.5   Mean   :0.173
##  3rd Qu.: 319.0   3rd Qu.: 3.000   3rd Qu.:999.0   3rd Qu.:0.000
##  Max.   :4918.0   Max.   :56.000   Max.   :999.0   Max.   :7.000
##
##         poutcome      emp.var.rate      cons.price.idx  cons.conf.idx
##  failure    : 4252   Min.   :-3.40000   Min.   :92.20   Min.   :-50.8
##  nonexistent:35563   1st Qu.:-1.80000   1st Qu.:93.08   1st Qu.:-42.7
##  success    : 1373   Median : 1.10000   Median :93.75   Median :-41.8
##                      Mean   : 0.08189   Mean   :93.58   Mean   :-40.5
##                      3rd Qu.: 1.40000   3rd Qu.:93.99   3rd Qu.:-36.4
##                      Max.   : 1.40000   Max.   :94.77   Max.   :-26.9
##
##    euribor3m      nr.employed     y
##  Min.   :0.634   Min.   :4964   no :36548
##  1st Qu.:1.344   1st Qu.:5099   yes: 4640
##  Median :4.857   Median :5191
##  Mean   :3.621   Mean   :5167
##  3rd Qu.:4.961   3rd Qu.:5228
##  Max.   :5.045   Max.   :5228
##
```

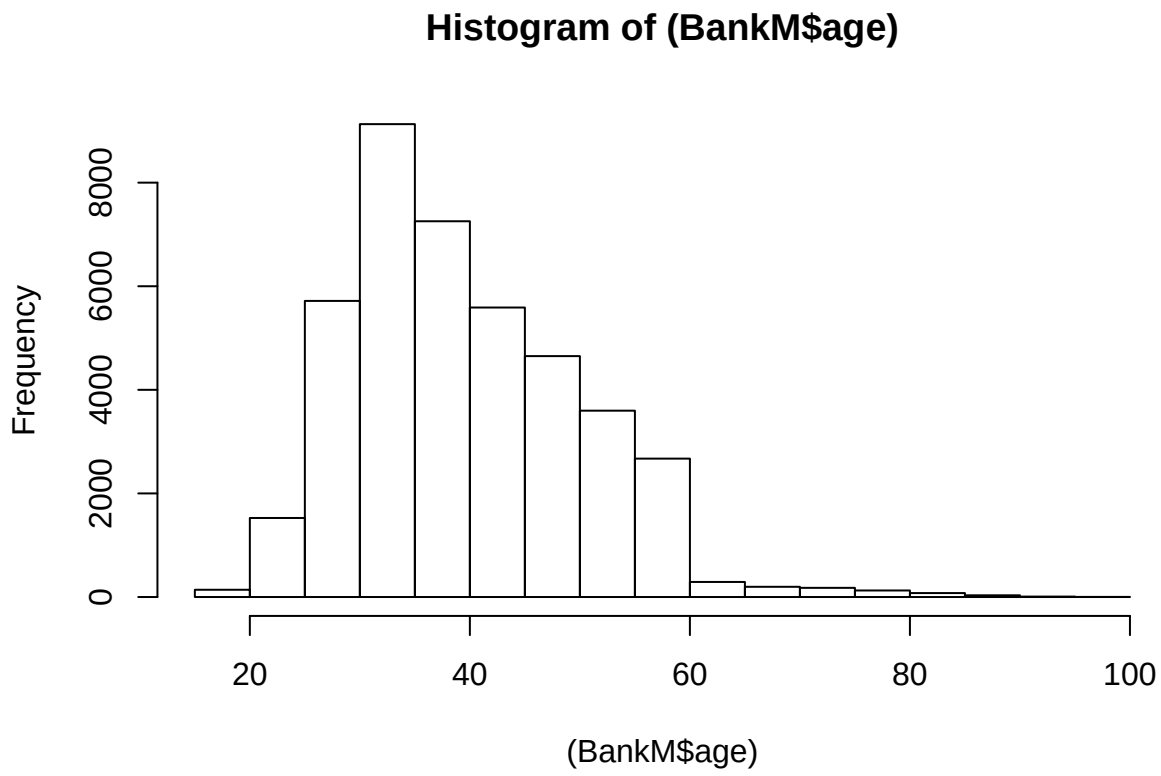
Initial Thoughts

Since the main output variable (y) is binary some of the models we can run are logistic regression, classification trees, linear discriminant analysis, quadratic discriminant analysis, k-nearest neighbor, and support vector machines.

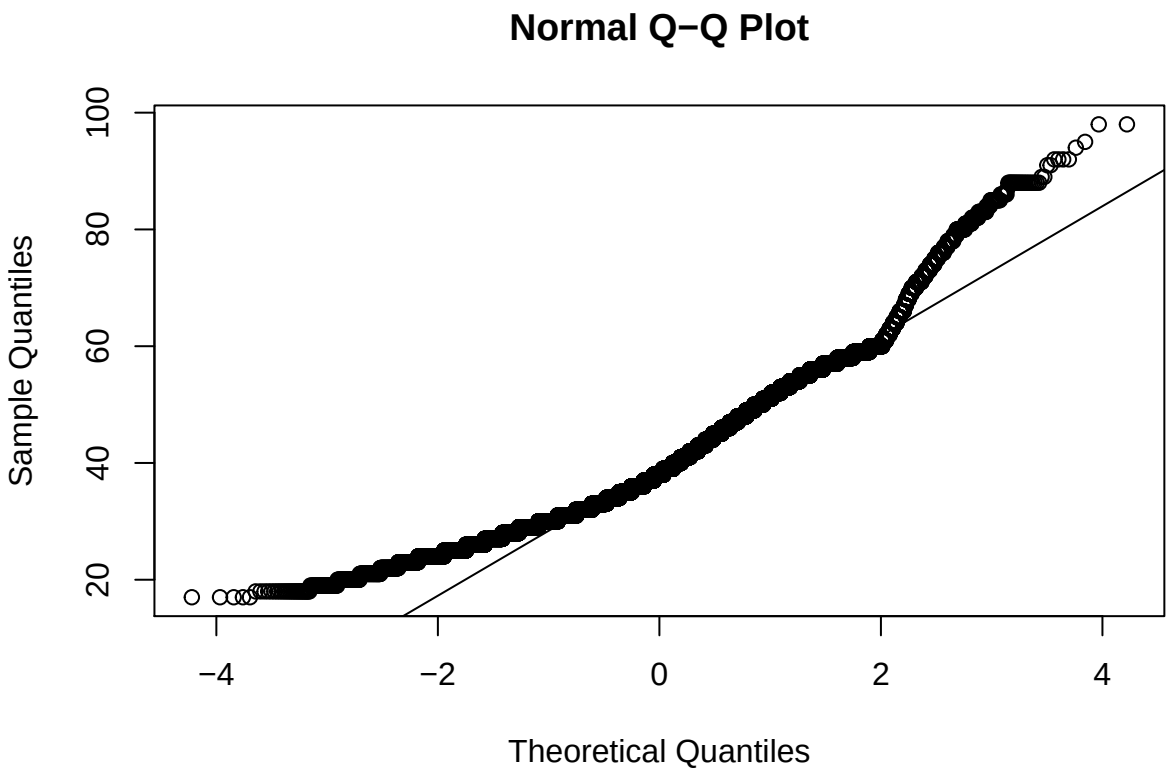
Checking for Normality

A quick note here, is that we realized, we could not run a Shapiro-Wilkes test for normality, because we have well over 5,000 data points in this sample.

```r
hist((BankM$age))
```

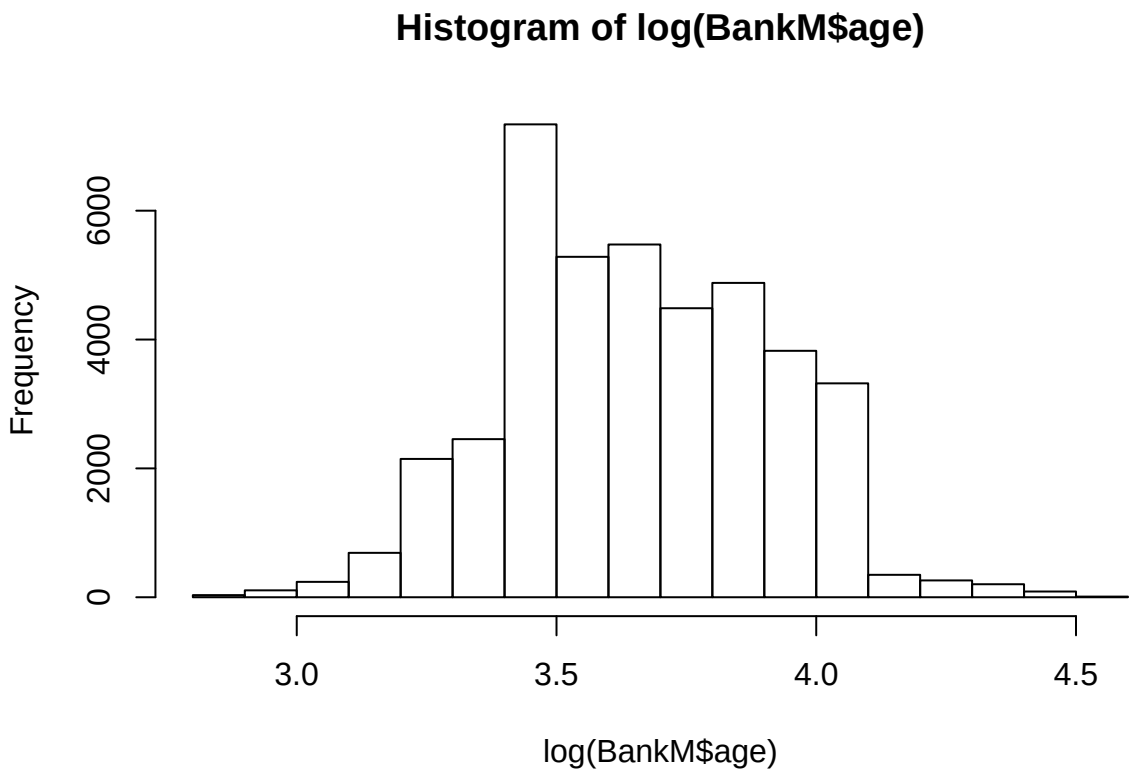



```r
qqnorm((BankM$age))
qqline((BankM$age))
```

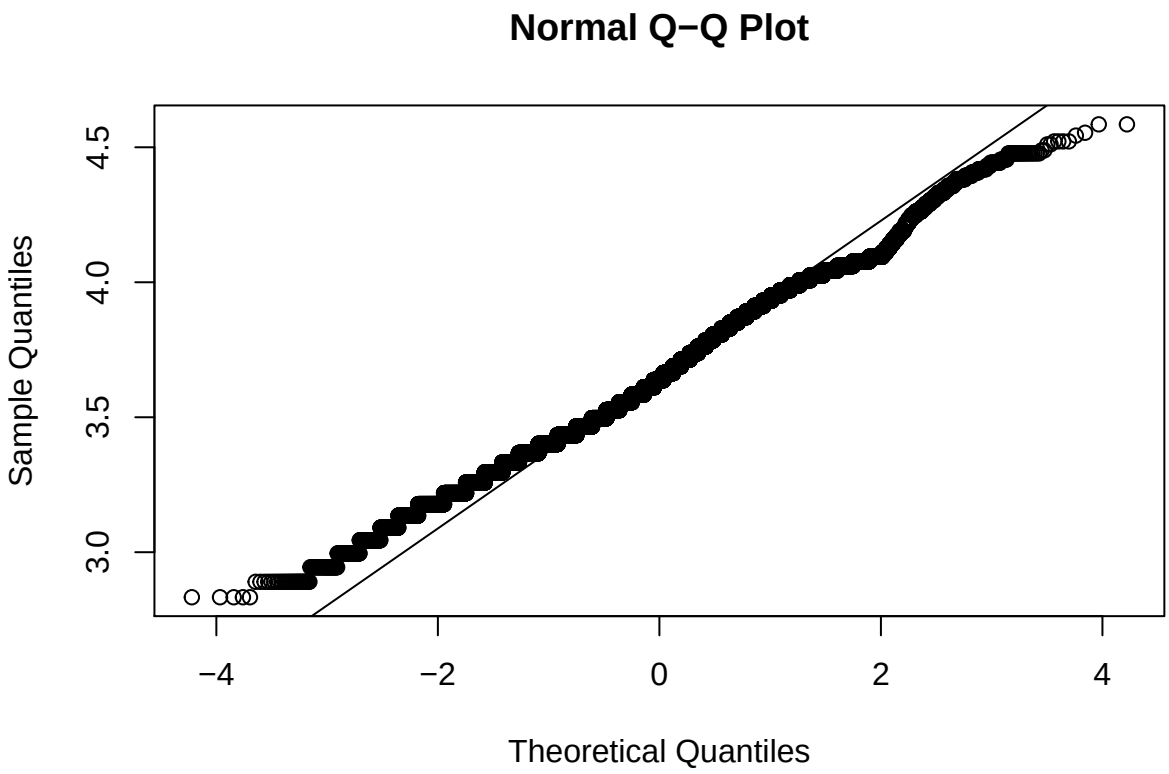


Based on the QQplot above, Age is not normally distributed so we can try doing a log transformation.

```
hist(log(BankM$age))
```



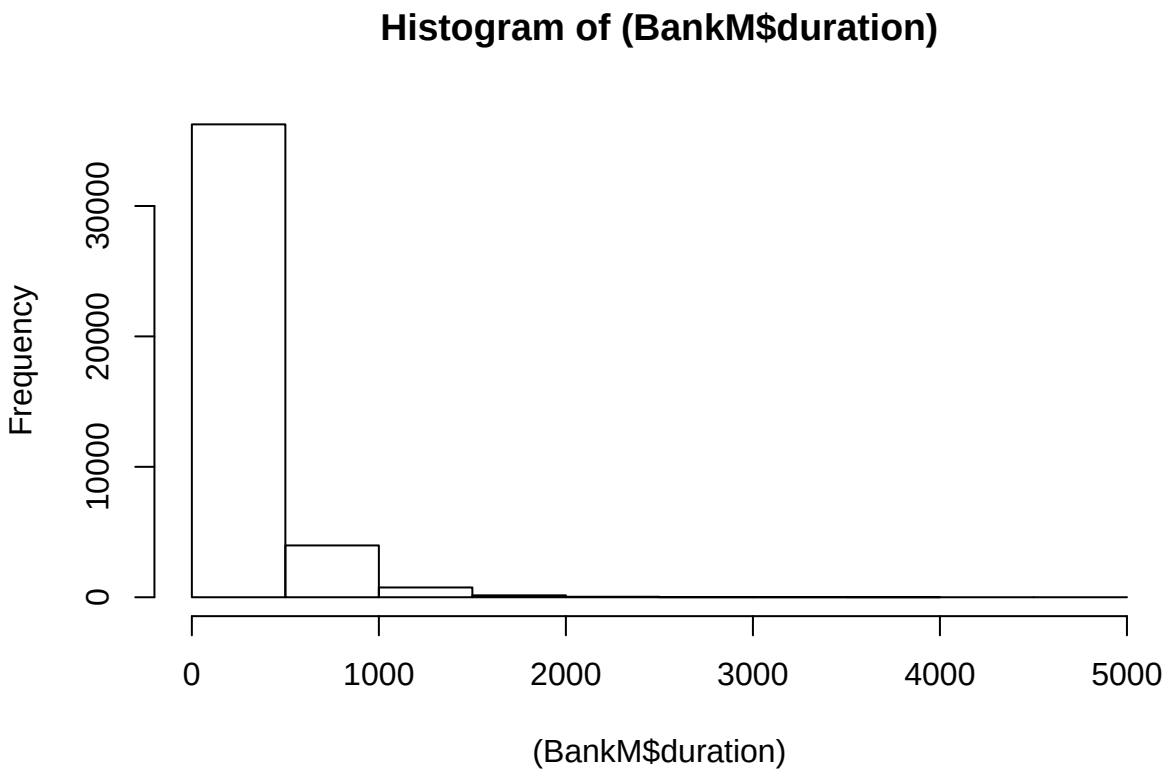

```
qqnorm(log(BankM$age))
qqline(log(BankM$age))
```



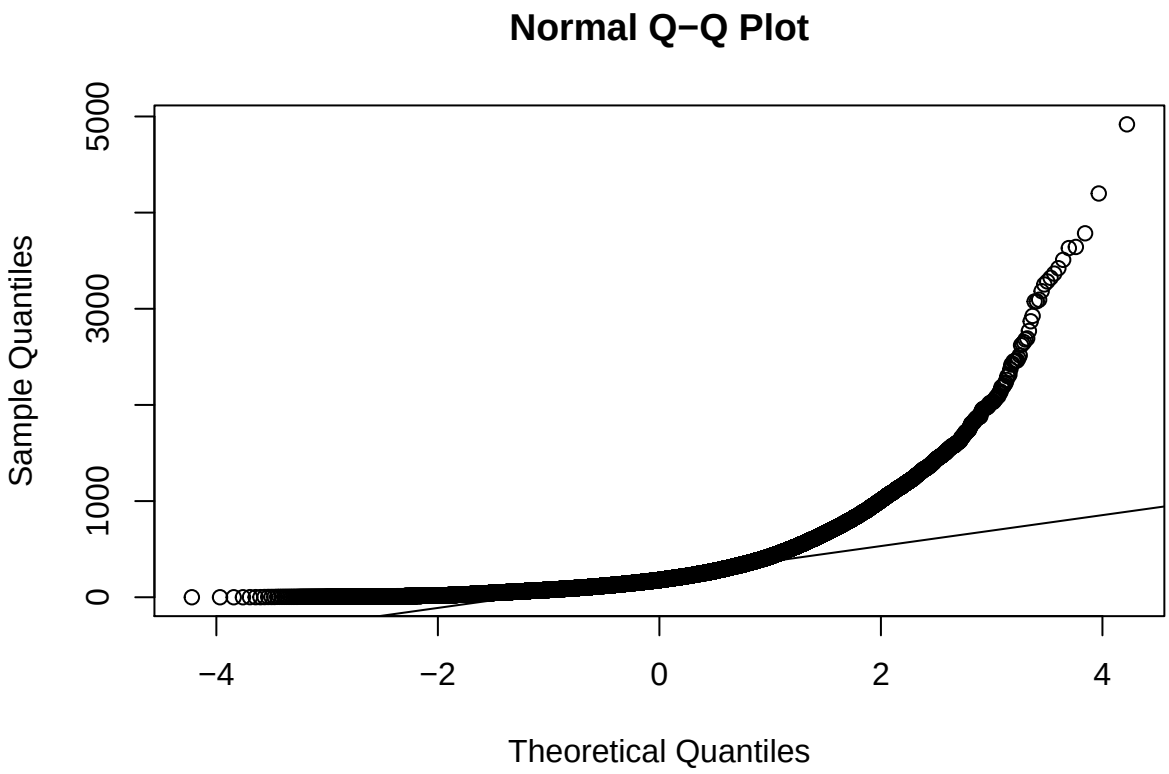
From the QQplot and histogram, age now seems like it is slightly more normally distributed.

Lets take a look at duration below:

```
hist((BankM$duration))
```

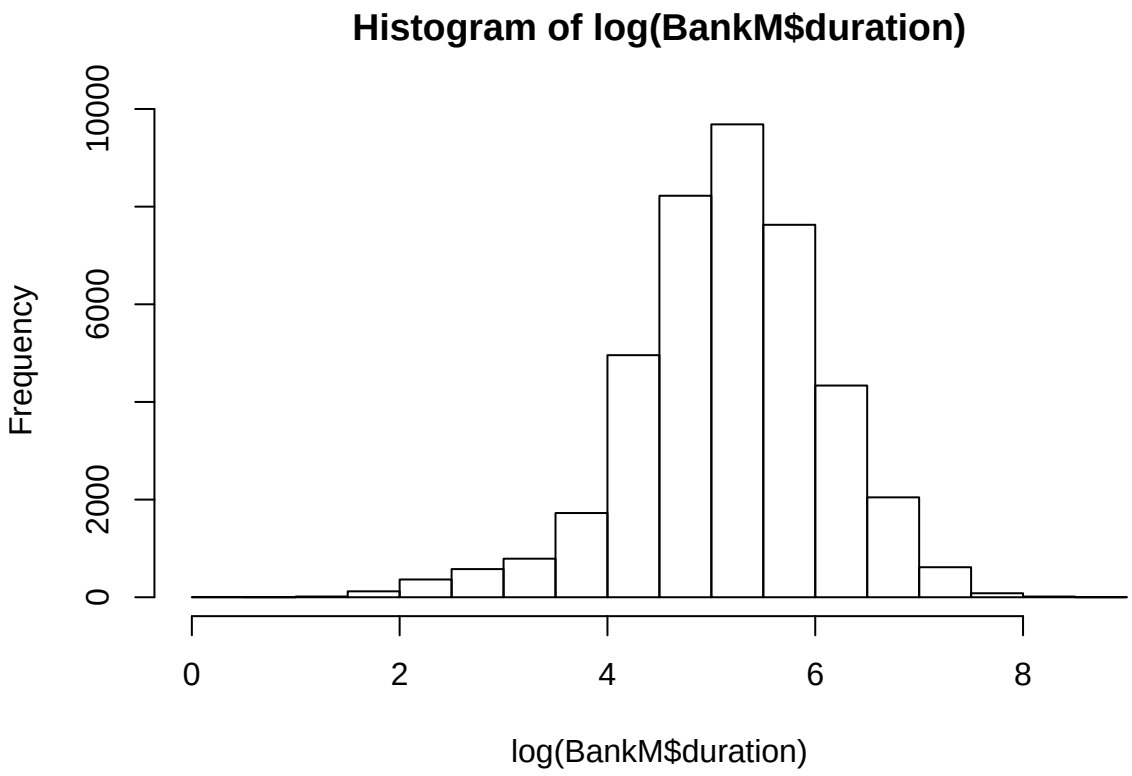



```
qqnorm((BankM$duration))
qqline((BankM$duration))
```



Duration also does not seem normally distributed, so we can try to do a log transformation again.

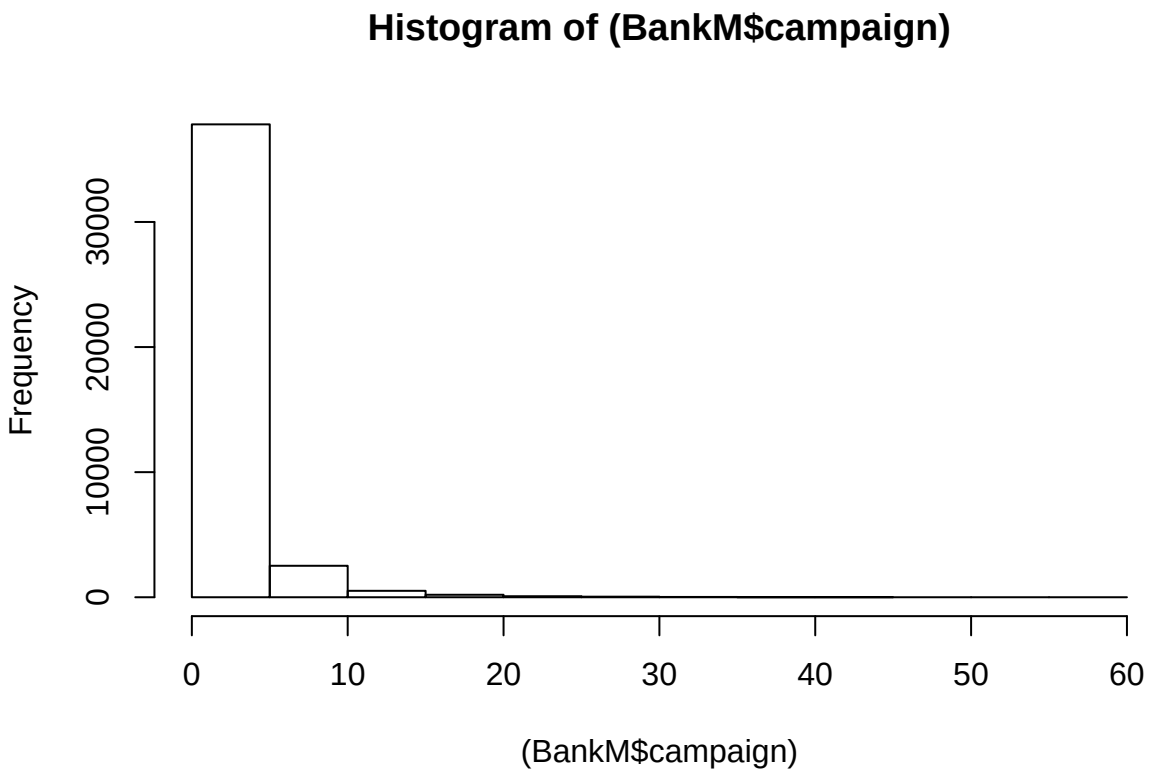
```
hist(log(BankM$duration))
```



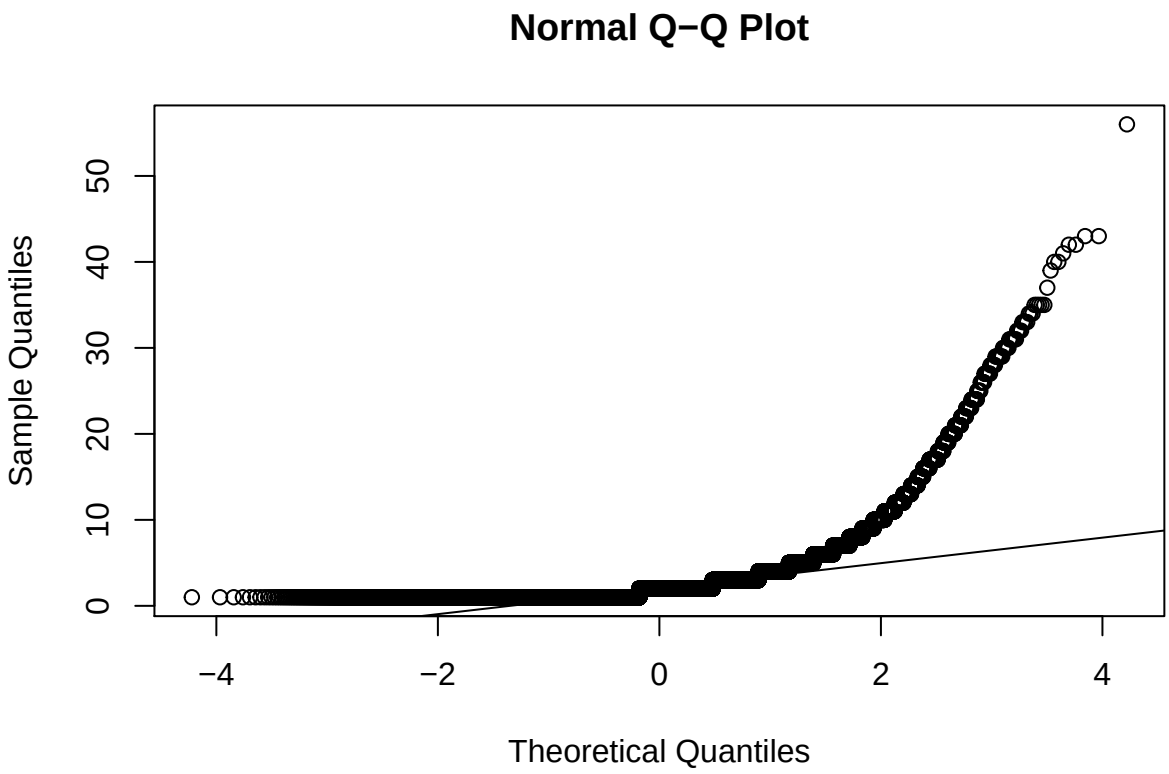
After the log transformation, duration now also seems more normally distributed.

Lets take a look at the Campaign variable below:

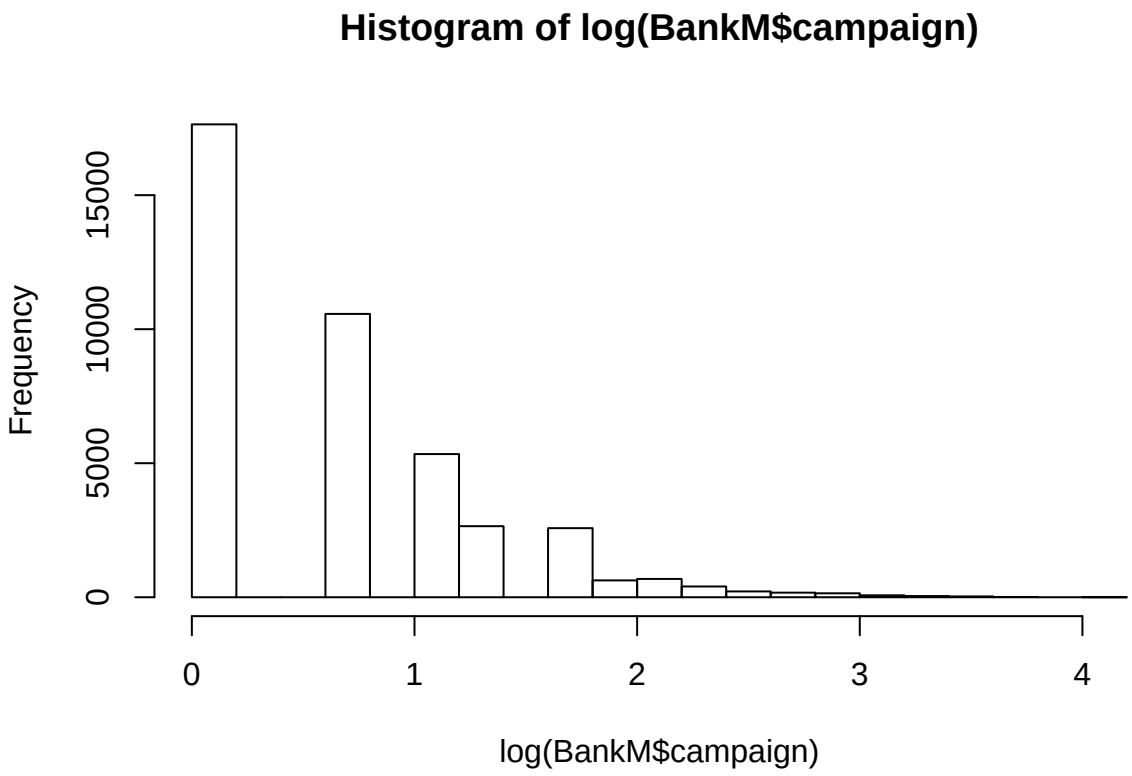
```
hist((BankM$campaign))
```



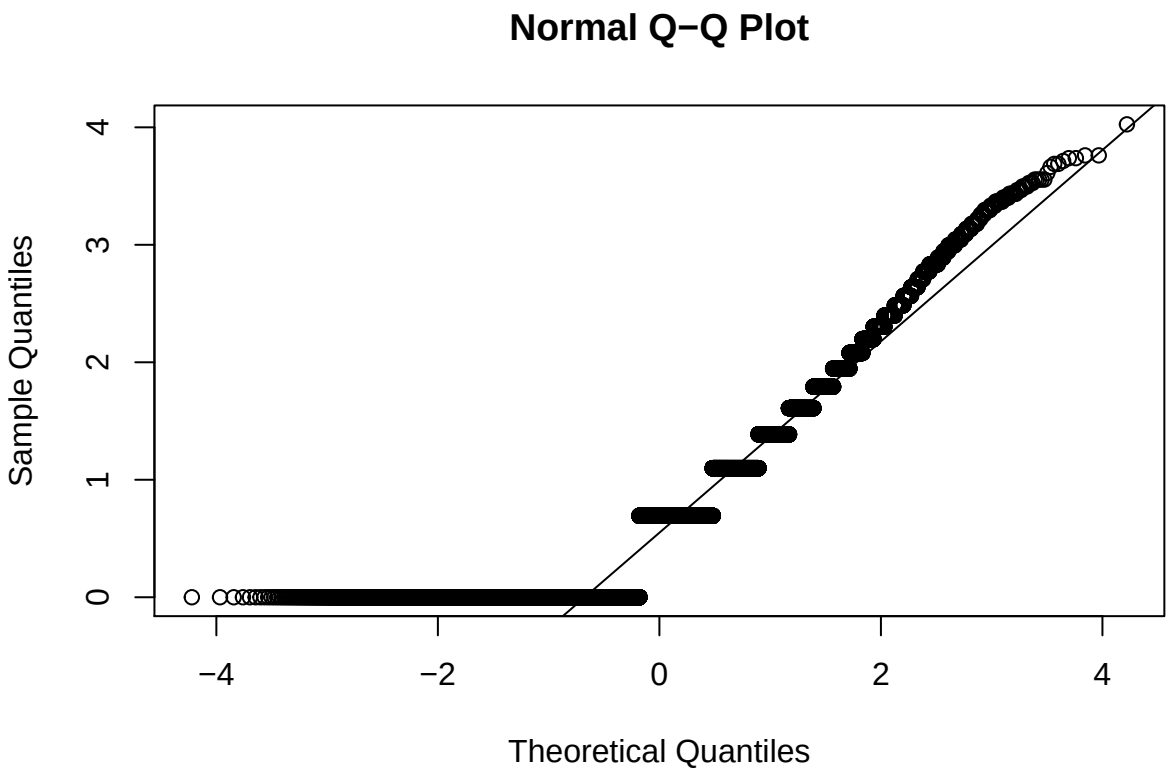

```
qqnorm((BankM$campaign))
qqline((BankM$campaign))
```




```
hist(log(BankM$campaign))
```




```
qqnorm(log(BankM$campaign))
qqline(log(BankM$campaign))
```



After doing a log transformation on this variables, the QQplot seems to worsen, so it is better left not transformed.

Early Logistic Regression Check

```
glm.fit = glm(y ~ . -age + log(age), data=BankM, family=binomial)
summary(glm.fit)
```

```
##
## Call:
## glm(formula = y ~ . - age + log(age), family = binomial, data = BankM)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -6.0019  -0.2985  -0.1856  -0.1342   3.3813
##
## Coefficients: (1 not defined because of singularities)
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -2.359e+02  3.821e+01  -6.174 6.65e-10 ***
## jobblue-collar              -2.388e-01  7.986e-02  -2.991 0.002784 **
## jobentrepreneur             -1.729e-01  1.259e-01  -1.373 0.169856
## jobhousemaid                -7.836e-03  1.474e-01  -0.053 0.957613
## jobmanagement               -4.859e-02  8.537e-02  -0.569 0.569218
```

```
## jobretired                    3.389e-01  1.021e-01   3.321 0.000897 ***
## jobself-employed             -1.570e-01  1.178e-01  -1.332 0.182756
## jobservices                  -1.440e-01  8.614e-02  -1.672 0.094521 .
## jobstudent                    1.661e-01  1.133e-01   1.466 0.142644
## jobtechnician                -1.516e-02  7.115e-02  -0.213 0.831256
## jobunemployed                 2.157e-02  1.279e-01   0.169 0.866066
## jobunknown                   -5.642e-02  2.383e-01  -0.237 0.812870
## maritalmarried               -9.749e-03  6.849e-02  -0.142 0.886802
## maritalsingle                 2.407e-02  7.871e-02   0.306 0.759758
## maritalunknown                1.521e-02  4.165e-01   0.037 0.970865
## educationbasic.6y             1.126e-01  1.204e-01   0.936 0.349495
## educationbasic.9y            -1.265e-02  9.514e-02  -0.133 0.894191
## educationhigh.school          3.435e-02  9.175e-02   0.374 0.708170
## educationilliterate           1.051e+00  7.556e-01   1.391 0.164124
## educationprofessional.course  1.032e-01  1.012e-01   1.020 0.307662
## educationuniversity.degree    1.813e-01  9.179e-02   1.975 0.048228 *
## educationunknown              1.470e-01  1.195e-01   1.231 0.218421
## defaultunknown               -2.917e-01  6.745e-02  -4.324 1.53e-05 ***
## defaultyes                   -7.291e+00  1.135e+02  -0.064 0.948769
## housingunknown               -9.472e-02  1.397e-01  -0.678 0.497880
## housingyes                   -3.913e-03  4.135e-02  -0.095 0.924617
## loanunknown                         NA         NA      NA       NA
## loanyes                      -5.169e-02  5.746e-02  -0.900 0.368354
## contacttelephone             -6.479e-01  7.689e-02  -8.425  < 2e-16 ***
## monthApr                     -2.016e+00  1.443e-01 -13.964  < 2e-16 ***
## monthMay                     -2.463e+00  1.216e-01 -20.254  < 2e-16 ***
## monthJun                     -2.549e+00  2.091e-01 -12.195  < 2e-16 ***
## monthJul                     -1.887e+00  1.523e-01 -12.392  < 2e-16 ***
## monthAug                     -1.149e+00  1.271e-01  -9.035  < 2e-16 ***
## monthSep                     -1.640e+00  1.546e-01 -10.609  < 2e-16 ***
## monthOct                     -1.821e+00  1.503e-01 -12.113  < 2e-16 ***
## monthNov                     -2.434e+00  1.442e-01 -16.875  < 2e-16 ***
## monthDec                     -1.689e+00  2.118e-01  -7.972 1.56e-15 ***
## day_of_weektue                2.134e-01  6.486e-02   3.290 0.001003 **
## day_of_weekwed                2.901e-01  6.489e-02   4.470 7.80e-06 ***
## day_of_weekthu                1.717e-01  6.340e-02   2.708 0.006770 **
## day_of_weekfri                1.160e-01  6.613e-02   1.754 0.079406 .
## duration                      4.707e-03  7.458e-05  63.111  < 2e-16 ***
## campaign                     -4.000e-02  1.156e-02  -3.459 0.000542 ***
## pdays                        -9.352e-04  2.171e-04  -4.307 1.65e-05 ***
## previous                     -6.222e-02  5.911e-02  -1.053 0.292535
## poutcomenonexistent           4.254e-01  9.423e-02   4.515 6.34e-06 ***
## poutcomesuccess               9.635e-01  2.116e-01   4.554 5.26e-06 ***
## emp.var.rate                 -1.763e+00  1.419e-01 -12.421  < 2e-16 ***
## cons.price.idx                2.202e+00  2.523e-01   8.728  < 2e-16 ***
## cons.conf.idx                 2.121e-02  7.765e-03   2.732 0.006294 **
## euribor3m                     3.295e-01  1.300e-01   2.535 0.011231 *
## nr.employed                   5.518e-03  3.115e-03   1.771 0.076487 .
## log(age)                     -1.133e-01  1.002e-01  -1.131 0.257964
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 28999  on 41187  degrees of freedom
## Residual deviance: 17077  on 41135  degrees of freedom
## AIC: 17183
##
## Number of Fisher Scoring iterations: 10
```

We can check for significance by assuming a confidence level of 95%, which would mean that alpha = 0.05, to see if variables should be removed. Lasso and stepwise selection may also be good for feature selection, but first let's try using the significance method. The variables log(age), marital, housing, loan, and previous don't seem to have p-values less than 0.05 so we can try to make a reduced model by removing them.

```
new.glm.fit = glm(y ~ . -age -marital -housing -loan -previous, data=BankM, family=binomial)
summary(new.glm.fit)
```

```
##
## Call:
## glm(formula = y ~ . - age - marital - housing - loan - previous,
##     family = binomial, data = BankM)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.9953  -0.2983  -0.1854  -0.1344   3.3627
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -2.343e+02  3.819e+01  -6.135 8.54e-10 ***
## jobblue-collar               -2.371e-01  7.965e-02  -2.977 0.002910 **
## jobentrepreneur              -1.875e-01  1.254e-01  -1.495 0.134804
## jobhousemaid                 -2.863e-02  1.468e-01  -0.195 0.845332
## jobmanagement                -6.948e-02  8.440e-02  -0.823 0.410383
## jobretired                    2.760e-01  9.149e-02   3.017 0.002557 **
## jobself-employed             -1.601e-01  1.178e-01  -1.359 0.174228
## jobservices                  -1.416e-01  8.602e-02  -1.646 0.099810 .
## jobstudent                    2.321e-01  1.053e-01   2.205 0.027440 *
## jobtechnician                -1.268e-02  7.110e-02  -0.178 0.858475
## jobunemployed                 1.998e-02  1.278e-01   0.156 0.875777
## jobunknown                   -6.818e-02  2.377e-01  -0.287 0.774227
## educationbasic.6y             1.214e-01  1.200e-01   1.011 0.311950
## educationbasic.9y             1.285e-03  9.461e-02   0.014 0.989163
## educationhigh.school          5.569e-02  9.081e-02   0.613 0.539687
## educationilliterate           1.069e+00  7.550e-01   1.416 0.156724
## educationprofessional.course  1.190e-01  1.007e-01   1.182 0.237199
## educationuniversity.degree    2.062e-01  9.048e-02   2.279 0.022697 *
## educationunknown              1.533e-01  1.194e-01   1.284 0.199252
## defaultunknown               -3.029e-01  6.694e-02  -4.525 6.03e-06 ***
## defaultyes                   -7.300e+00  1.134e+02  -0.064 0.948687
## contacttelephone             -6.421e-01  7.674e-02  -8.367  < 2e-16 ***
## monthApr                     -2.018e+00  1.442e-01 -13.996  < 2e-16 ***
## monthMay                     -2.464e+00  1.215e-01 -20.286  < 2e-16 ***
## monthJun                     -2.540e+00  2.089e-01 -12.159  < 2e-16 ***
## monthJul                     -1.880e+00  1.521e-01 -12.364  < 2e-16 ***
## monthAug                     -1.161e+00  1.269e-01  -9.147  < 2e-16 ***
## monthSep                     -1.649e+00  1.545e-01 -10.678  < 2e-16 ***
```

```
## monthOct                    -1.822e+00  1.502e-01 -12.132  < 2e-16 ***
## monthNov                    -2.441e+00  1.441e-01 -16.939  < 2e-16 ***
## monthDec                    -1.709e+00  2.114e-01  -8.083 6.32e-16 ***
## day_of_weektue               2.132e-01  6.481e-02   3.289 0.001006 **
## day_of_weekwed               2.907e-01  6.485e-02   4.483 7.36e-06 ***
## day_of_weekthu               1.723e-01  6.338e-02   2.719 0.006544 **
## day_of_weekfri               1.170e-01  6.609e-02   1.770 0.076660 .
## duration                     4.706e-03  7.457e-05  63.112  < 2e-16 ***
## campaign                    -4.044e-02  1.156e-02  -3.498 0.000469 ***
## pdays                       -8.549e-04  2.035e-04  -4.200 2.67e-05 ***
## poutcomenonexistent          5.011e-01  6.418e-02   7.809 5.79e-15 ***
## poutcomesuccess              1.025e+00  2.040e-01   5.024 5.05e-07 ***
## emp.var.rate                -1.757e+00  1.419e-01 -12.380  < 2e-16 ***
## cons.price.idx               2.181e+00  2.519e-01   8.657  < 2e-16 ***
## cons.conf.idx                2.070e-02  7.747e-03   2.673 0.007528 **
## euribor3m                    3.294e-01  1.299e-01   2.536 0.011225 *
## nr.employed                  5.478e-03  3.113e-03   1.760 0.078436 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 28999  on 41187  degrees of freedom
## Residual deviance: 17082  on 41143  degrees of freedom
## AIC: 17172
##
## Number of Fisher Scoring iterations: 10
```

The AIC for our reduced model is slightly lower at 17172 rather than 17184 which suggests that the reduced model is slightly better.

Testing if Multicollinearity Exists

```
vif(new.glm.fit)
```

```
##                      GVIF Df GVIF^(1/(2*Df))
## job              3.348796 11        1.056473
## education        3.019826  7        1.082142
## default          1.127713  2        1.030504
## contact          2.310272  1        1.519958
## month           61.471396  9        1.257103
## day_of_week      1.064243  4        1.007813
## duration         1.243213  1        1.114995
## campaign         1.052022  1        1.025681
## pdays            9.647132  1        3.105983
## poutcome        10.855893  2        1.815166
## emp.var.rate   142.183546  1       11.924074
## cons.price.idx  67.843474  1        8.236715
## cons.conf.idx    5.308586  1        2.304037
## euribor3m      134.942062  1       11.616457
## nr.employed    171.736403  1       13.104824
```

Since the VIF is greater than 5 in some cases, we should be careful about the variables emp.var.rate, cons.price.idx, euribor3m, and nr.employed. We could try best subset selection, Lasso, PCR, or PLS for variable selection here, but let's just try reducing the model here and assessing the model summaries.

```
small.model = glm(y ~ . -age -marital -housing -loan -previous
                  -emp.var.rate -cons.price.idx -euribor3m -nr.employed, data=BankM, family=binomial)
summary(small.model)
```

```
##
## Call:
## glm(formula = y ~ . - age - marital - housing - loan - previous -
##     emp.var.rate - cons.price.idx - euribor3m - nr.employed,
##     family = binomial, data = BankM)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.6540  -0.3472  -0.2415  -0.1598   3.1920
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    2.870e+00  3.223e-01   8.905  < 2e-16 ***
## jobblue-collar                -3.287e-01  7.696e-02  -4.272 1.94e-05 ***
## jobentrepreneur               -3.144e-01  1.207e-01  -2.605 0.009187 **
## jobhousemaid                  -3.193e-02  1.398e-01  -0.228 0.819291
## jobmanagement                 -1.621e-01  8.151e-02  -1.988 0.046788 *
## jobretired                     6.196e-01  8.808e-02   7.034 2.00e-12 ***
## jobself-employed              -2.142e-01  1.143e-01  -1.875 0.060843 .
## jobservices                   -2.220e-01  8.333e-02  -2.664 0.007717 **
## jobstudent                     7.165e-01  1.031e-01   6.946 3.76e-12 ***
## jobtechnician                 -1.396e-01  6.768e-02  -2.063 0.039095 *
## jobunemployed                  1.329e-01  1.230e-01   1.081 0.279749
## jobunknown                     2.854e-02  2.313e-01   0.123 0.901784
## educationbasic.6y              5.252e-02  1.168e-01   0.450 0.653058
## educationbasic.9y             -7.231e-02  9.136e-02  -0.792 0.428625
## educationhigh.school           1.376e-02  8.781e-02   0.157 0.875489
## educationilliterate            1.186e+00  7.053e-01   1.682 0.092650 .
## educationprofessional.course   9.647e-02  9.743e-02   0.990 0.322109
## educationuniversity.degree     2.124e-01  8.749e-02   2.427 0.015208 *
## educationunknown               2.501e-01  1.153e-01   2.170 0.030031 *
## defaultunknown                -6.505e-01  6.403e-02 -10.159  < 2e-16 ***
## defaultyes                    -8.428e+00  1.136e+02  -0.074 0.940839
## contacttelephone              -1.351e+00  6.435e-02 -21.002  < 2e-16 ***
## monthApr                      -1.484e+00  1.185e-01 -12.525  < 2e-16 ***
## monthMay                      -2.467e+00  1.130e-01 -21.837  < 2e-16 ***
## monthJun                      -1.474e+00  1.203e-01 -12.252  < 2e-16 ***
## monthJul                      -2.645e+00  1.161e-01 -22.789  < 2e-16 ***
## monthAug                      -2.815e+00  1.217e-01 -23.130  < 2e-16 ***
## monthSep                      -1.473e+00  1.547e-01  -9.520  < 2e-16 ***
## monthOct                      -1.188e+00  1.462e-01  -8.124 4.52e-16 ***
## monthNov                      -2.708e+00  1.217e-01 -22.243  < 2e-16 ***
## monthDec                      -1.284e+00  2.163e-01  -5.937 2.90e-09 ***
## day_of_weektue                 1.821e-01  6.236e-02   2.921 0.003490 **
## day_of_weekwed                 2.026e-01  6.242e-02   3.246 0.001171 **
## day_of_weekthu                 1.061e-01  6.119e-02   1.734 0.082979 .
```

```
## day_of_weekfri                1.046e-01  6.389e-02   1.638 0.101449
## duration                      4.335e-03  6.907e-05  62.760  < 2e-16 ***
## campaign                     -7.183e-02  1.131e-02  -6.353 2.11e-10 ***
## pdays                        -1.576e-03  2.067e-04  -7.623 2.49e-14 ***
## poutcomenonexistent          -4.982e-02  6.228e-02  -0.800 0.423741
## poutcomesuccess               7.858e-01  2.076e-01   3.785 0.000154 ***
## cons.conf.idx                 5.729e-02  5.009e-03  11.436  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 28999  on 41187  degrees of freedom
## Residual deviance: 18702  on 41147  degrees of freedom
## AIC: 18784
##
## Number of Fisher Scoring iterations: 10
```

The AIC actually increaded higher than the full model, so this would not be a great model to use.

We run an anova below, to see which is the best model to use.

```
anova(small.model, new.glm.fit, glm.fit)
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ (age + job + marital + education + default + housing + loan +
##     contact + month + day_of_week + duration + campaign + pdays +
##     previous + poutcome + emp.var.rate + cons.price.idx + cons.conf.idx +
##     euribor3m + nr.employed) - age - marital - housing - loan -
##     previous - emp.var.rate - cons.price.idx - euribor3m - nr.employed
## Model 2: y ~ (age + job + marital + education + default + housing + loan +
##     contact + month + day_of_week + duration + campaign + pdays +
##     previous + poutcome + emp.var.rate + cons.price.idx + cons.conf.idx +
##     euribor3m + nr.employed) - age - marital - housing - loan -
##     previous
## Model 3: y ~ (age + job + marital + education + default + housing + loan +
##     contact + month + day_of_week + duration + campaign + pdays +
##     previous + poutcome + emp.var.rate + cons.price.idx + cons.conf.idx +
##     euribor3m + nr.employed) - age + log(age)
##   Resid. Df Resid. Dev Df Deviance
## 1     41147      18702
## 2     41143      17082  4  1620.70
## 3     41135      17077  8     5.06
```

Model 3 has the lowest deviance, and residual deviance, which means that the full model has the lowest variance as expected.

Checking all logistic regression models on the whole data set

Lets look at the accuracy rate of model 3.

```
BankM$y -> y
levels(BankM$y)
```

```
## [1] "no"  "yes"
```

```
prob <- predict(glm.fit, type = "response")
pred1 <- ifelse(prob < 0.5, "no", "yes")
table(pred1, y)
```

```
##       y
## pred1    no   yes
##   no  35561  2668
##   yes   987  1972
```

```
mean(y==pred1)
```

```
## [1] 0.9112606
```

The accuracy rate is 91.12%.

Let take a look at the accuracy rate of the slightly reduced model, which is model 2.

```
prob2 <- predict(new.glm.fit, type = "response")
pred2 <- ifelse(prob < 0.5, "no", "yes")
table(pred2, y)
```

```
##       y
## pred2    no   yes
##   no  35561  2668
##   yes   987  1972
```

```
mean(y==pred2)
```

```
## [1] 0.9112606
```

The accuracy rate is 91.12%

Lets take a look at the accuracy rate of the most reduced model, or model 1.

```
prob3 <- predict(small.model, type = "response")
pred3 <- ifelse(prob < 0.5, "no", "yes")
table(pred3, y)
```

```
##       y
## pred3    no   yes
##   no  35561  2668
##   yes   987  1972
```

```
mean(y==pred3)
```

```
## [1] 0.9112606
```

The accracy rate is 91.12%. The overall fraction of correct prdictions is about 91.13% for all 3 models which is pretty high.

Cross Validation

Setting up the training and testing sets

We set up a random sample of 70% of the data to fit the model then tested it against the remaining 30%.

```
set.seed(1)

n = length(BankM$y)
z = sample(n, n*0.7)
train <- BankM[z, ]
data.test <- BankM[-z, ]
y.test <- BankM$y[-z]
```

Comparing the testing set on the full model

Let's use the large model to make predictions below:

```
large.model = glm(y ~ . -age + log(age), family=binomial, data = train)

large.predict = predict(large.model, data.test, type="response" )
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
## == : prediction from a rank-deficient fit may be misleading
```

```
pred1 <- ifelse(large.predict < 0.5, "no", "yes")
table(pred1, y.test)
```

```
##       y.test
## pred1    no   yes
##   no  10703   793
##   yes   276   585
```

```
mean(y.test==pred1)
```

```
## [1] 0.9134903
```

The accuracy ratae is 91.34%

Comparing Training and testing sets on the slightly reduced model

Lets use the slightly reduced model to make predictions below:

```
med.model = glm(y ~ . -age -marital -housing -loan -previous,
                  family=binomial,
                  data = train)

med.predict = predict(large.model, data.test, type="response" )
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
## == : prediction from a rank-deficient fit may be misleading
```

```
pred2 <- ifelse(med.predict < 0.5, "no", "yes")
table(pred2, y.test)
```

```
##      y.test
## pred2    no   yes
##   no  10703   793
##   yes   276   585
```

```
mean(y.test==pred2)
```

```
## [1] 0.9134903
```

The accuracy ratae is 91.34%

Comparing Training and testing sets on the most reduced model

Lets use the most reduced model to make predictions below:

```
small.model = glm(y ~ . -age -marital -housing -loan -previous
                  -emp.var.rate -cons.price.idx -euribor3m -nr.employed,
                  family=binomial,
                  data = train)

small.predict = predict(large.model, data.test, type="response" )
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
## == : prediction from a rank-deficient fit may be misleading
```

```
pred3 <- ifelse(small.predict < 0.5, "no", "yes")
table(pred3, y.test)
```

```
##      y.test
## pred3    no   yes
##   no  10703   793
##   yes   276   585
```

```
mean(y.test==pred3)
```

```
## [1] 0.9134903
```

The accuracy rate is 91.34%

The overall fraction of correct predictions is about 91.35% on the same test set for all 3 models which is pretty high.

Checking an ROC curve

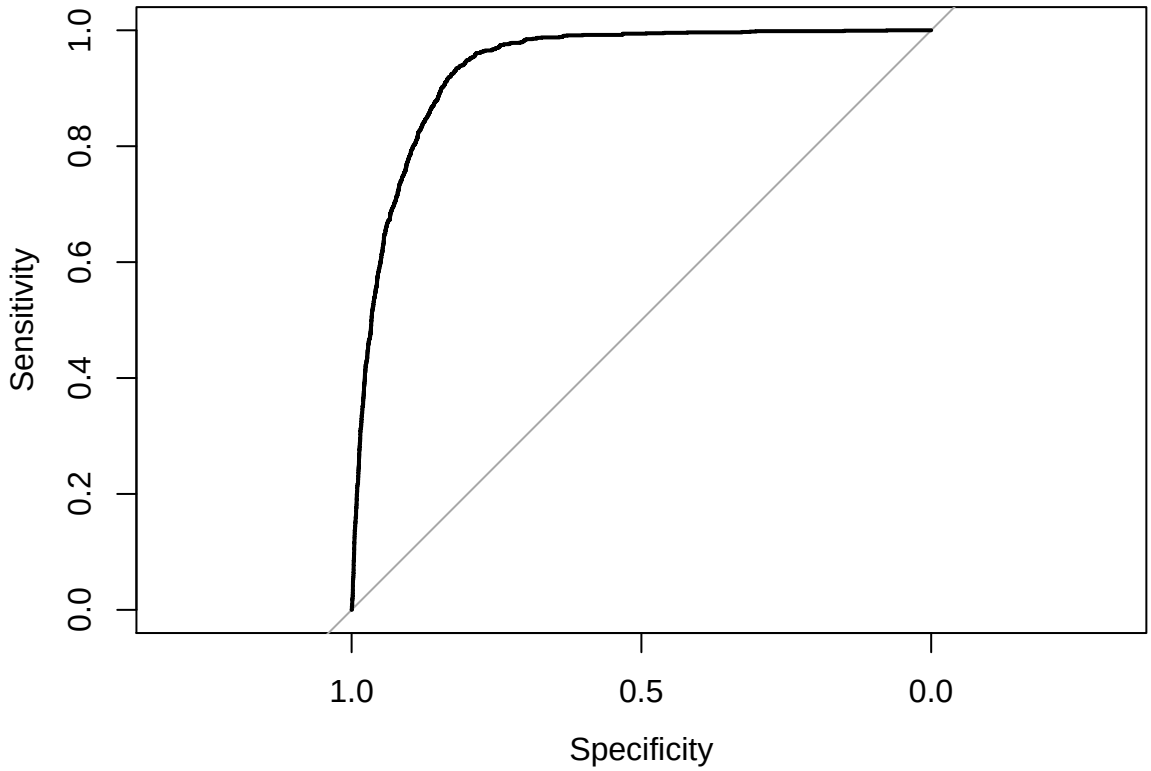
We can build an ROC curve, to check the model. Since, all 3 models have identical accuracy rates, we decided to only build an ROC curve for the largest model.

```
myRoc <- roc(y.test, large.predict)
```

```
## Setting levels: control = no, case = yes
```

```
## Setting direction: controls < cases
```

```
plot(myRoc)
```



This is a strong model since the plot generally stays near 1.0 for both sensitivity and specificity.

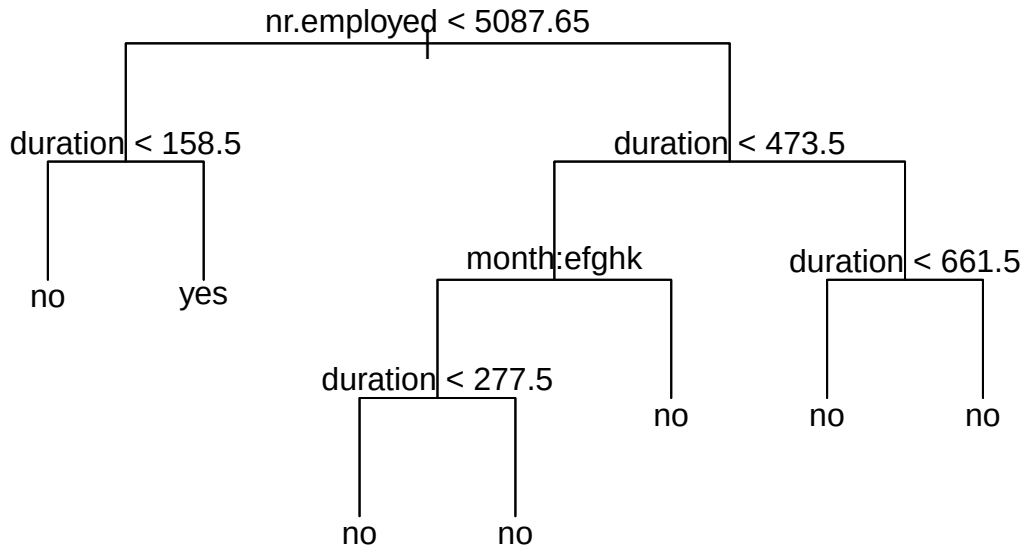
Classification Tree

We choose to simply run the classification tree on the full model because logistic regression showed that not much really changes by reducing the model, so we can let the tree function reduce the model itself.

```
tree.fit = tree(y ~ ., data=train)
tree.fit
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 28831 20360.0 no ( 0.886858 0.113142 )
##    2) nr.employed < 5087.65 3549  4878.0 no ( 0.554241 0.445759 )
##      4) duration < 158.5 1234  1029.0 no ( 0.853323 0.146677 ) *
##      5) duration > 158.5 2315  3106.0 yes ( 0.394816 0.605184 ) *
##    3) nr.employed > 5087.65 25282 12360.0 no ( 0.933550 0.066450 )
##      6) duration < 473.5 21938  4634.0 no ( 0.978029 0.021971 )
##       12) month: May,Jun,Jul,Aug,Nov 20272  2143.0 no ( 0.990677 0.009323 )
##         24) duration < 277.5 16512   714.4 no ( 0.996790 0.003210 ) *
##         25) duration > 277.5 3760  1170.0 no ( 0.963830 0.036170 ) *
##       13) month: Mar,Apr,Oct,Dec 1666  1550.0 no ( 0.824130 0.175870 ) *
##      7) duration > 473.5 3344  4363.0 no ( 0.641746 0.358254 )
##       14) duration < 661.5 1623  1705.0 no ( 0.781269 0.218731 ) *
##       15) duration > 661.5 1721  2385.0 no ( 0.510169 0.489831 ) *
```

```
plot(tree.fit, type="uniform")
text(tree.fit)
```




```
summary(tree.fit)
```

```
##
## Classification tree:
## tree(formula = y ~ ., data = train)
## Variables actually used in tree construction:
## [1] "nr.employed" "duration"    "month"
## Number of terminal nodes:  7
## Residual mean deviance:  0.4045 = 11660 / 28820
## Misclassification error rate: 0.09625 = 2775 / 28831
```

The only variables that are used in the construction of this tree, based on the training set, are "nr.employed", "duration", and "month". This also helped us realize how few "y" end up being yes, since only one terminal mode ends in yes, when nr.employed $<$ 5087.6 and duration $>$ 158.5, it predicts only 2315 values out of 28,831 in the training set will result in a yes. The misclassification error rate of 0.09625 is low for the training set.

```
tree.predict = predict(tree.fit, data.test, type="class" )
table(tree.predict, y.test)
```

```
##             y.test
## tree.predict    no   yes
##          no  10578   824
##          yes   401   554
```

```r
mean(y.test == tree.predict)
```

```
## [1] 0.9008659
```

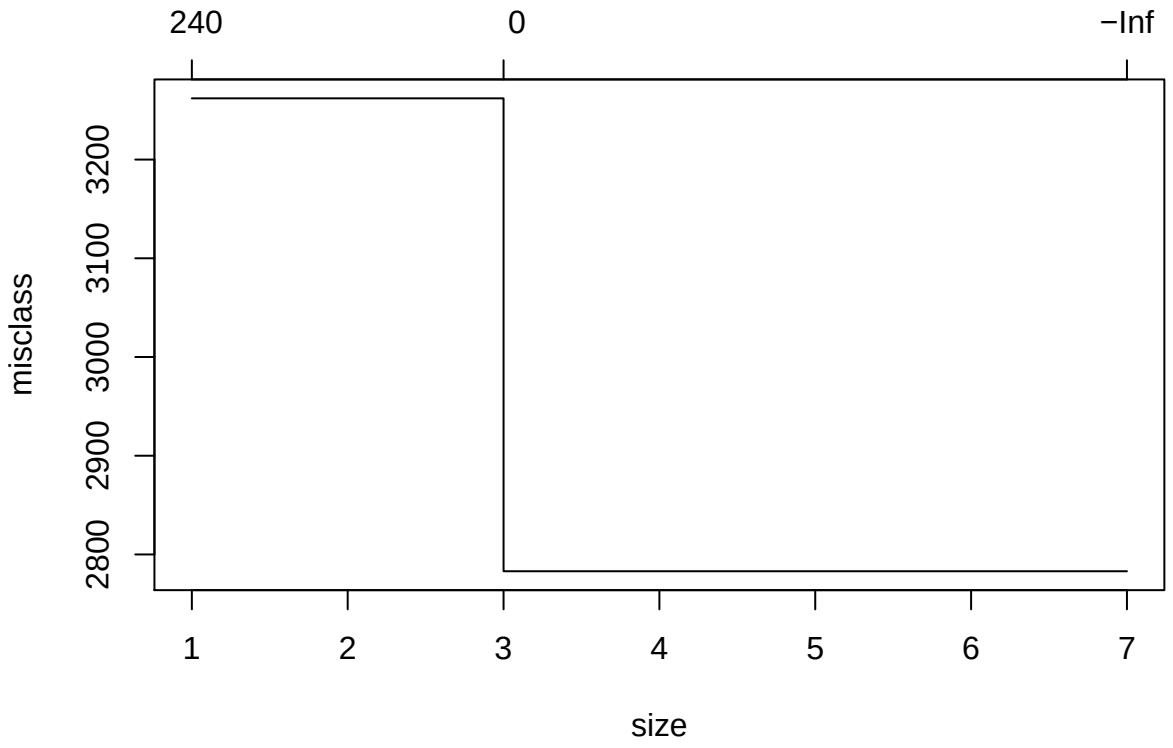
Simply comparing accuracy here of 90.09% we can see that the classification model is slightly less accurate than the logistic regression model for the same test set.

Checking if pruning is necessary

```r
cv=cv.tree(tree.fit, FUN = prune.misclass )
cv
```

```
## $size
## [1] 7 3 1
##
## $dev
## [1] 2783 2783 3262
##
## $k
## [1]  -Inf   0.0 243.5
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```r
plot(cv)
```

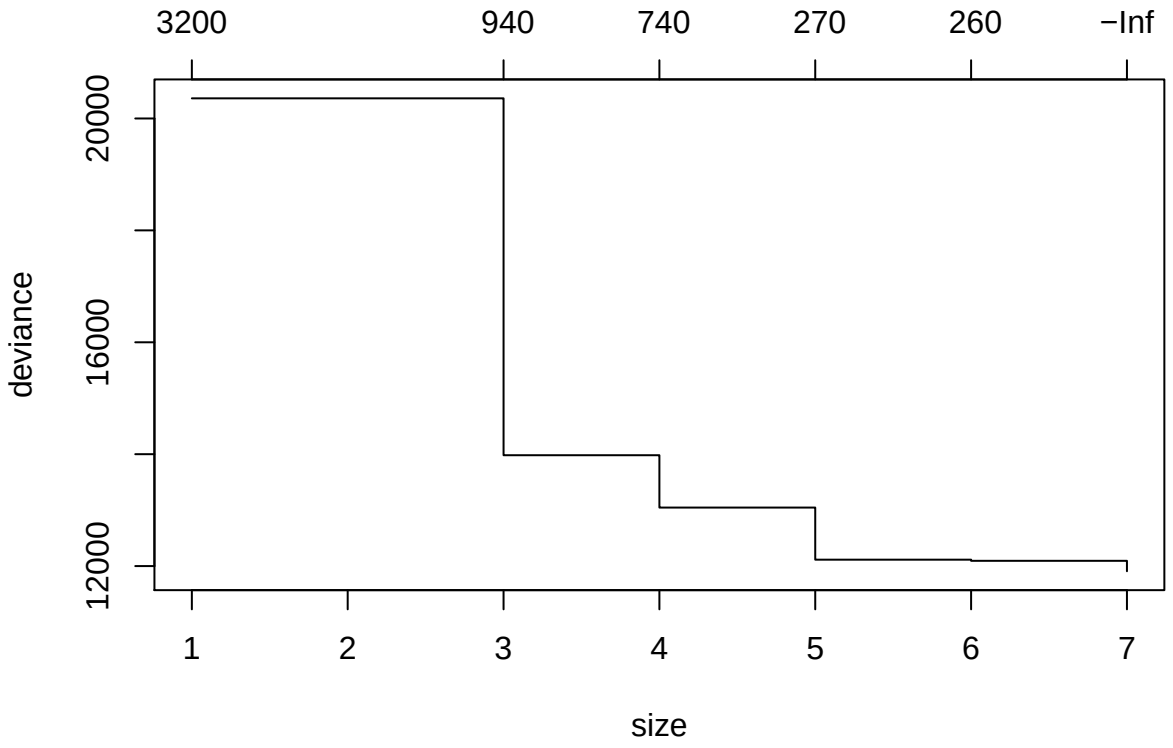


Here we see that 3 terminal nodes leads to the lowest misclassification rate. We can compare this result when basing it off of deviance.

```
cv=cv.tree(tree.fit)
cv
```

```
## $size
## [1] 7 6 5 4 3 1
##
## $dev
## [1] 11907.71 12093.94 12114.46 13045.89 13981.27 20359.75
##
## $k
## [1]      -Inf  259.1127  273.0821  743.1430  940.8129 3240.6815
##
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
plot(cv)
```

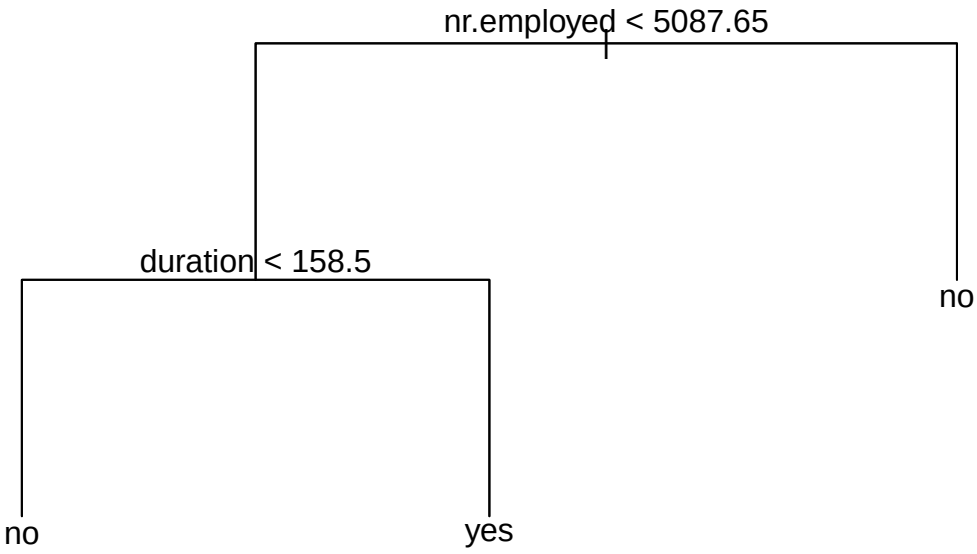


In contrast, deviance is lowest when all 7 terminal nodes remain. We can still test a pruned tree and compare the results.

```
pruned = prune.misclass(tree.fit, best=3)
pruned
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
## 1) root 28831 20360 no ( 0.88686 0.11314 )
##   2) nr.employed < 5087.65 3549  4878 no ( 0.55424 0.44576 )
##     4) duration < 158.5 1234  1029 no ( 0.85332 0.14668 ) *
##     5) duration > 158.5 2315  3106 yes ( 0.39482 0.60518 ) *
##   3) nr.employed > 5087.65 25282 12360 no ( 0.93355 0.06645 ) *
```

```
plot(pruned, type="uniform")
text(pruned)
```




```
summary(pruned)
```

```
##
## Classification tree:
## snip.tree(tree = tree.fit, nodes = 3L)
## Variables actually used in tree construction:
## [1] "nr.employed" "duration"
## Number of terminal nodes:  3
## Residual mean deviance:  0.572 = 16490 / 28830
## Misclassification error rate: 0.09625 = 2775 / 28831
```

```
tree.predict = predict(pruned, data.test, type="class" )
table(tree.predict, y.test)
```

```
##             y.test
## tree.predict    no   yes
##          no  10578   824
##          yes   401   554
```

```
mean(y.test == tree.predict)
```

```
## [1] 0.9008659
```

The misclassification rate and the accuracy are the same as the unpruned tree, however the residual mean deviance has slightly increased to 0.572, so relying on the previous model may help reduce some variance.

K-Nearest Neighbor

Simply to be more efficient, we decided to only use the variables used in the better classification tree model.

```
trainx1 <- as.matrix(BankM$nr.employed[z])
trainx2 <- as.matrix(BankM$duration[z])
trainx3 <- as.matrix(as.numeric(BankM$month[z]))
trainx <- as.matrix(cbind(trainx1, trainx2, trainx3))

testx1 <- as.matrix(BankM$nr.employed[-z])
testx2 <- as.matrix(BankM$duration[-z])
testx3 <- as.matrix(as.numeric(BankM$month[-z]))
testx <- as.matrix(cbind(testx1, testx2, testx3))

train.y <- BankM$y[z]

best.k <- -1
accuracy <- 999999999
best.accuracy <- -1
for (i in 1:50) {
predn <- knn(trainx, testx, train.y, k=i)
accuracy <- mean(y.test==predn)
if (accuracy > best.accuracy) {
best.table <- as.matrix(table(predn, y.test))
best.k <- i
best.accuracy <- accuracy
}
}
print(paste("The optimal value of k is",best.k,"with an overall accuracy of",best.accuracy))
```

```
## [1] "The optimal value of k is 43 with an overall accuracy of 0.90920126244234"
```

```
best.table
```

```
##      y.test
## predn    no   yes
##   no  10563   706
##   yes   416   672
```

We can see that KNN results in an accuracy of 90.92% which is higher than the classification tree model's accuracy of 90.09%, but still slightly less than the logistic regression model's accuracy of 91.35% for the test set.

Linear Discriminant Analysis

```
model.lda <- lda(y ~ nr.employed + duration + as.numeric(month), data = BankM, subset = z, cv=TRUE)
model.lda
```

```
## Call:
```

```
## lda(y ~ nr.employed + duration + as.numeric(month), data = BankM,
##     cv = TRUE, subset = z)
##
## Prior probabilities of groups:
##        no       yes
## 0.8868579 0.1131421
##
## Group means:
##     nr.employed duration as.numeric(month)
## no     5175.991 220.9174          6.578083
## yes    5093.654 549.2520          6.827406
##
## Coefficients of linear discriminants:
##                             LD1
## nr.employed       -0.010770591
## duration           0.003299635
## as.numeric(month)  0.097214422
```

```
pred4 <- predict(model.lda, data.test)
table(pred4$class, y.test)
```

```
##      y.test
##          no   yes
##   no  10604   768
##   yes   375   610
```

```
mean(y.test==pred4$class)
```

```
## [1] 0.9075018
```

LDA seems to produce the second worst model since it has an accuracy of about 90.75%, and it is still not as high as the accuracy of the logistic regression.

Quadratic Discriminant Analysis

```
model.qda <- qda(y ~ nr.employed + duration + as.numeric(month), family = binomial,
                 data = BankM, subset = z, cv=TRUE)
model.qda
```

```
## Call:
## qda(y ~ nr.employed + duration + as.numeric(month), data = BankM,
##     family = binomial, cv = TRUE, subset = z)
##
## Prior probabilities of groups:
##        no       yes
## 0.8868579 0.1131421
##
## Group means:
##     nr.employed duration as.numeric(month)
## no     5175.991 220.9174          6.578083
## yes    5093.654 549.2520          6.827406
```

```
pred5 <- predict(model.qda, data.test)
table(pred5$class, y.test)
```

```
##      y.test
##          no   yes
##   no  10329   688
##   yes   650   690
```

```
mean(y.test==pred4$class)
```

```
## [1] 0.9075018
```

The qda model seems to correctly predict about 90.75% of the test set as well.

Final Logistic Regression

Our classification tree model was very helpful in reducing all the variables down to just 3 critical variables, so it would be worth testing logistic regression again with just those 3 variables.

```
log.reg = glm(y ~ nr.employed + duration + month, family=binomial, data = train)

fin.predict = predict(log.reg, data.test, type="response" )

predf <- ifelse(fin.predict < 0.5, "no", "yes")
table(predf, y.test)
```

```
##      y.test
## predf    no   yes
##   no  10694   876
##   yes   285   502
```

```
mean(y.test==predf)
```

```
## [1] 0.9060452
```

The most reduced Logistic Regression model predicted with an accuracy rate of 90.60% on the test set.

Conclusion

To summarize, the full logistic regression model helped us achieve the highest accuracy rate of 91.34%. The second highest accuracy that we got was 90.92%, using the KNN model. In terms of accuracy, the next best models were the LDA and QDA models which both had an accuracy rate of 90.75%. When we retested the logistic regression model on the most reduced set, it was the next best model at 90.60% accuracy. The least accurate model was the classification tree, which gave us an accuracy rate of 90.09%. We had also tried an SVM model, but processing the full model had taken us almost 7 hours, only to give us a "Warning Limit Exceeded" error. We believe this may be due the the limitation of our processing abilities.

In an overall view, we were able to predict whether a customer would subscribe to a bank term deposit or not, with an accuracy of about 90%, which in general would be considered very high. When we were deciding on cross-validation techniques to use, we decided to simply go with training and testing sets with comparisons on accuracy, rather than k-fold cross validation or leave one out cross validation with comparisons on the Receiver Operating Characteristic (ROC) or brier scores simply because it was a bit less computationally exhaustive on our machines. We also understand that when comparing different sized models, looking for the lowest bayesian information criterion (BIC), akaike information criterion (AIC), Mallow's Cp, or even the largest adjusted R-squared values would have all been viable options, however we simply went with the classifaction tree's variables for model selection.

Based on the source of the data, we see that it was donated to UCI in 2014. In order to reduce biasness in the dataset, we believe that a more updated sample should be combined to the 2014 data. Another way to reduce biasness in the predictions and hopefully even have a higher prediction accuracy is to combine this Portuguese banking institutions data with other banking institutions from multiple sources.

End Results Based on the Test Set

```
## # A tibble: 6 x 3
##    Rank Model                            Accuracy
##   <dbl> <chr>                               <dbl>
## 1     1 Full Logistic Regression            0.913
## 2     2 K-Nearest Neighbor                  0.909
## 3     3 Linear Discriminant Analysis        0.908
## 4     3 Quadratic Discriminant Analysis     0.908
## 5     5 Reduced Logistic Regression         0.906
## 6     6 Classification Tree                 0.901
```