Data analysis project

Sam Maximilian Licke Agdur

b) & c)

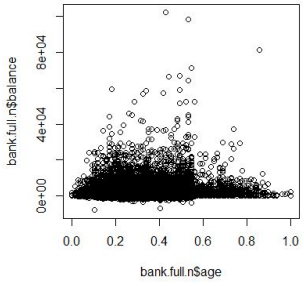
The advantage of direct marketing is that you get a response directly, and because of that you can measure the response of the offer / the interest of the offer /interest of the product directly. In comparison with for example an normal marketing campaign putting up a poster, where you are marketing something, perhaps making the presumptive consumer have an interest in the product but you cannot measure it as with direct marketing(calls,email etc).

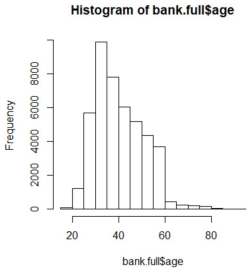
Each row in the dataset represent a client of the bank, the columns represent the different attributes of each client such age, balance, education level etc. They Y variable shows if the client subscribed to a term deposit. This data is very valuable as it can be used to train regression models and neural networks to predict future success of a bank-telemarketing C)

d)

Visualization analysis:

We choose to analyze the age and balance with a plot and also the age frequencies in a histogram





Numeric analysis:

Here is a numeric analysis of the data in the data.

```
      age                job          marital          education     default
 Min.   :18.00   blue-collar:9732   divorced: 5207   primary  : 6851   no :44396
 1st Qu.:33.00   management :9458   married :27214   secondary:23202   yes:  815
 Median :39.00   technician :7597   single  :12790   tertiary :13301
 Mean   :40.94   admin.     :5171                    unknown  : 1857
 3rd Qu.:48.00   services   :4154
 Max.   :95.00   retired    :2264
                 (Other)    :6835
    balance       housing      loan            contact           day              month
 Min.   : -8019   no :20081   no :37967   cellular :29285   Min.   : 1.00   may    :13766
 1st Qu.:    72   yes:25130   yes: 7244   telephone: 2906   1st Qu.: 8.00   jul    : 6895
 Median :   448                           unknown  :13020   Median :16.00   aug    : 6247
 Mean   :  1362                                             Mean   :15.81   jun    : 5341
 3rd Qu.:  1428                                             3rd Qu.:21.00   nov    : 3970
 Max.   :102127                                             Max.   :31.00   apr    : 2932
                                                                            (Other): 6060
    duration         campaign          pdays          previous          poutcome
 Min.   :   0.0   Min.   : 1.000   Min.   : -1.0   Min.   :  0.0000   failure: 4901
 1st Qu.: 103.0   1st Qu.: 1.000   1st Qu.: -1.0   1st Qu.:  0.0000   other  : 1840
 Median : 180.0   Median : 2.000   Median : -1.0   Median :  0.0000   success: 1511
 Mean   : 258.2   Mean   : 2.764   Mean   : 40.2   Mean   :  0.5803   unknown:36959
 3rd Qu.: 319.0   3rd Qu.: 3.000   3rd Qu.: -1.0   3rd Qu.:  0.0000
 Max.   :4918.0   Max.   :63.000   Max.   :871.0   Max.   :275.0000

   y
 no :39922
 yes: 5289
```

With the following functions we can see that there are around 37369 rows that have missing or “unknown” values.

```
> #Finding null values
> bank.full[bank.full=="unknown"] <- NA
> sum(!complete.cases(bank.full))
[1] 37369
```

e)

We converted the data to numeric and normalized it so it could be used in our Pearson correlation analysis and in our trainset for our regression model.

```
bank.full -> bank.full.n
bank.full.n$marital <- as.numeric(bank.full.n$job)
bank.full.n$job <- as.numeric(bank.full.n$marital)
bank.full.n$month <- as.numeric(bank.full.n$month)
bank.full.n$education <- as.numeric(bank.full.n$education)
bank.full.n$loan <- as.numeric(bank.full.n$loan)
bank.full.n$contact <- as.numeric(bank.full.n$contact)
bank.full.n$housing <- as.numeric(bank.full.n$housing)
bank.full.n$default <- as.numeric(bank.full.n$default)
bank.full.n$age <- normalize(bank.full.n$age)
bank.full.n$poutcome <- as.numeric(bank.full.n$poutcome)
bank.full.n$y <- as.numeric(bank.full.n$y)
```

```
#Normalization function
bank.full.norm$balance <- normalize(bank.full.n$balance)
normalize <- function(x) {
  return ((x- min(x)) / (max(x) - min(x)))
}

#Normalizing the data
bank.full -> bank.full.norm
bank.full.n$balance <- normalize(bank.full.n$balance)
bank.full.n$age <- normalize(bank.full.n$age)
bank.full.n$loan <- normalize(bank.full.n$marital)
bank.full.n$duration <- normalize(bank.full.n$duration)
```

f)

The following Pearson correlation matrix shows that there are very few correlations in this data set. The strongest correlation here is with poutcome and pdays and previous. The Y variable seems to have the strongest correlation to duration which is 0.39, a very weak correlation. Because of this we can't should be careful with removing too many variables. Day and month seem to have the lowest correlation to y so according to this analysis they should be disregarded.

```
> bank.full.matrix
              age    job marital education default balance housing   loan contact    day month
age          1.00 -0.02   -0.02     -0.11   -0.02    0.10   -0.19 -0.02    0.03 -0.01 -0.04
job         -0.02  1.00    1.00      0.17   -0.01    0.02   -0.13  1.00   -0.08  0.02 -0.09
marital     -0.02  1.00    1.00      0.17   -0.01    0.02   -0.13  1.00   -0.08  0.02 -0.09
education   -0.11  0.17    0.17      1.00   -0.01    0.06   -0.09  0.17   -0.11  0.02 -0.06
default     -0.02 -0.01   -0.01     -0.01    1.00   -0.07   -0.01 -0.01    0.02  0.01  0.01
balance      0.10  0.02    0.02      0.06   -0.07    1.00   -0.07  0.02   -0.03  0.00  0.02
housing     -0.19 -0.13   -0.13     -0.09   -0.01   -0.07    1.00 -0.13    0.19 -0.03  0.27
loan        -0.02  1.00    1.00      0.17   -0.01    0.02   -0.13  1.00   -0.08  0.02 -0.09
contact      0.03 -0.08   -0.08     -0.11    0.02   -0.03    0.19 -0.08    1.00 -0.03  0.36
day         -0.01  0.02    0.02      0.02    0.01    0.00   -0.03  0.02   -0.03  1.00 -0.01
month       -0.04 -0.09   -0.09     -0.06    0.01    0.02    0.27 -0.09    0.36 -0.01  1.00
duration     0.00  0.00    0.00      0.00   -0.01    0.02    0.01  0.00   -0.02 -0.03  0.01
campaign     0.00  0.01    0.01      0.01    0.02   -0.01   -0.02  0.01    0.02  0.16 -0.11
pdays       -0.02 -0.02   -0.02      0.00   -0.03    0.00    0.12 -0.02   -0.24 -0.09  0.03
previous     0.00  0.00    0.00      0.02   -0.02    0.02    0.04  0.00   -0.15 -0.05  0.02
poutcome     0.01  0.01    0.01     -0.02    0.03   -0.02   -0.10  0.01    0.27  0.08 -0.03
y            0.03  0.04    0.04      0.07   -0.02    0.05   -0.14  0.04   -0.15 -0.03 -0.02
             duration campaign pdays previous poutcome     y
age              0.00     0.00 -0.02     0.00     0.01  0.03
job              0.00     0.01 -0.02     0.00     0.01  0.04
marital          0.00     0.01 -0.02     0.00     0.01  0.04
education        0.00     0.01  0.00     0.02    -0.02  0.07
default         -0.01     0.02 -0.03    -0.02     0.03 -0.02
balance          0.02    -0.01  0.00     0.02    -0.02  0.05
housing          0.01    -0.02  0.12     0.04    -0.10 -0.14
loan             0.00     0.01 -0.02     0.00     0.01  0.04
contact         -0.02     0.02 -0.24    -0.15     0.27 -0.15
day             -0.03     0.16 -0.09    -0.05     0.08 -0.03
month            0.01    -0.11  0.03     0.02    -0.03 -0.02
duration         1.00    -0.08  0.00     0.00     0.01  0.39
campaign        -0.08     1.00 -0.09    -0.03     0.10 -0.07
pdays            0.00    -0.09  1.00     0.45    -0.86  0.10
previous         0.00    -0.03  0.45     1.00    -0.49  0.09
poutcome         0.01     0.10 -0.86    -0.49     1.00 -0.08
y                0.39    -0.07  0.10     0.09    -0.08  1.00
```

g)

```
#Data Partitioning
sz <- dim(bank.full.n)[1]
set.seed(10)
r <- order(runif(sz))
bank.full.shuffled <- bank.full.n[r, ]
trainData <- bank.full.shuffled[1:floor(0.80*sz), ]
testData <- bank.full.shuffled[(floor(0.80*sz)+1):(sz), ]
```

h)

```
Coefficients: (2 not defined because of singularities)
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.0138823  0.0201212   0.690  0.49024
age          0.0180893  0.0112899   1.602  0.10911
job          0.0010155  0.0004707   2.157  0.03098 *
marital             NA         NA      NA       NA
education    0.0190099  0.0020875   9.107  < 2e-16 ***
default     -0.0312099  0.0113934  -2.739  0.00616 **
balance      0.2952721  0.0539303   5.475 4.40e-08 ***
housing     -0.0866121  0.0032985 -26.258  < 2e-16 ***
loan                NA         NA      NA       NA
contact     -0.0380279  0.0019268 -19.737  < 2e-16 ***
day         -0.0004398  0.0001849  -2.379  0.01737 *
month        0.0048550  0.0005592   8.682  < 2e-16 ***
duration     2.3393575  0.0286200  81.739  < 2e-16 ***
campaign    -0.0030854  0.0004981  -6.194 5.92e-10 ***
pdays        0.0004248  0.0000299  14.210  < 2e-16 ***
previous     0.0070251  0.0007090   9.908  < 2e-16 ***
poutcome     0.0261444  0.0030848   8.475  < 2e-16 ***
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.192e-02  1.949e-02   1.125  0.26062
job          9.971e-04  4.706e-04   2.119  0.03411 *
education    1.859e-02  2.071e-03   8.977  < 2e-16 ***
default     -3.153e-02  1.139e-02  -2.768  0.00565 **
balance      3.032e-01  5.370e-02   5.646 1.66e-08 ***
housing     -8.764e-02  3.236e-03 -27.087  < 2e-16 ***
contact     -3.785e-02  1.924e-03 -19.676  < 2e-16 ***
day         -4.426e-04  1.849e-04  -2.394  0.01668 *
month        4.839e-03  5.591e-04   8.654  < 2e-16 ***
duration     2.339e+00  2.862e-02  81.741  < 2e-16 ***
campaign    -3.083e-03  4.981e-04  -6.189 6.13e-10 ***
pdays        4.239e-04  2.989e-05  14.181  < 2e-16 ***
previous     7.036e-03  7.090e-04   9.923  < 2e-16 ***
poutcome     2.600e-02  3.084e-03   8.432  < 2e-16 ***
---
```

After doing a Linear regression model with all the variables included we can see that loan and marital has singularity issues and age seems to have a very low p value. Because of this i

remove these variables and create a new model.

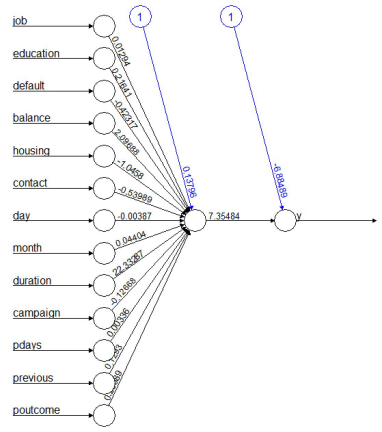
i) Here you can see that the accuracy of the model is at 0.88 which is very high.

```
> prediction1 <- ifelse(prediction1<0.5, 0, 1)
> summary(prediction1)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00000 0.00000 0.00000 0.01979 0.00000 1.00000
> error <- mean(prediction1 != testData$y)
> print(paste('Accuracy of model',1-error))
[1] "Accuracy of model 0.888864314939732"
>
```

To get more insight we also made a metric of the performance

```
> CrossTable(prediction1, testData$y, dnn=c('predicted', 'actual'), prop.chisq=FALSE, prop.t=TRUE
 prop.r=FALSE, prop.c=FALSE)


   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  9043


             | actual
   predicted |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
           0 |      7918 |       946 |      8864 |
             |     0.876 |     0.105 |           |
-------------|-----------|-----------|-----------|
           1 |        59 |       120 |       179 |
             |     0.007 |     0.013 |           |
-------------|-----------|-----------|-----------|
Column Total |      7977 |      1066 |      9043 |
-------------|-----------|-----------|-----------|


>
```

j) I used the neuralnet package in r to generate a neural network model with 1 hidden layer.



k)

The accuracy of the model was 0.89

```
> error <- mean(pred != testData$y)
> print(paste('Accuracy of model',1-error))
[1] "Accuracy of model 0.891850049762247"
```

Here is my cross table of the predictions with the Testdata.

```
> Predict <- compute(nn,testData)
> prob <- Predict$net.result
> pred <- ifelse(prob>0.5, 1, 0)
> CrossTable(pred, testData$y, dnn=c('predicted', 'actual'), prop.chisq=FALSE, prop.t=TRUE,
 prop.r=FALSE, prop.c=FALSE)


   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  9043


             | actual
   predicted |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
           0 |      7779 |       780 |      8559 |
             |     0.860 |     0.086 |           |
-------------|-----------|-----------|-----------|
           1 |       198 |       286 |       484 |
             |     0.022 |     0.032 |           |
-------------|-----------|-----------|-----------|
Column Total |      7977 |      1066 |      9043 |
-------------|-----------|-----------|-----------|
```

l)

The conclusion of this analysis Is that both the models have similar accuracy on predicting y values in this dataset. The neural network however requires a lot more processing power and memory. Because of this i think the Linear regression model is the best choice for predicting the success of bank-telemarketing in this company.