Bank_Data_Variance_Analysis.R

Deepti Khatri

Thu Feb 21 21:10:29 2019

```
bank <- read.csv("~/Spring 19 Sem/Multi Analysis/bank-additional/bank-
additional-full.csv", sep=";")
head(bank)

##   age       job marital   education default housing loan   contact month
## 1  56 housemaid married    basic.4y      no      no   no telephone   may
## 2  57  services married high.school unknown      no   no telephone   may
## 3  37  services married high.school      no     yes   no telephone   may
## 4  40    admin. married    basic.6y      no      no   no telephone   may
## 5  56  services married high.school      no      no  yes telephone   may
## 6  45  services married    basic.9y unknown      no   no telephone   may
##   day_of_week duration campaign pdays previous    poutcome emp.var.rate
## 1         mon      261        1   999        0 nonexistent          1.1
## 2         mon      149        1   999        0 nonexistent          1.1
## 3         mon      226        1   999        0 nonexistent          1.1
## 4         mon      151        1   999        0 nonexistent          1.1
## 5         mon      307        1   999        0 nonexistent          1.1
## 6         mon      198        1   999        0 nonexistent          1.1
##   cons.price.idx cons.conf.idx euribor3m nr.employed  y
## 1         93.994         -36.4     4.857        5191 no
## 2         93.994         -36.4     4.857        5191 no
## 3         93.994         -36.4     4.857        5191 no
## 4         93.994         -36.4     4.857        5191 no
## 5         93.994         -36.4     4.857        5191 no
## 6         93.994         -36.4     4.857        5191 no

str(bank)

## 'data.frame':    41188 obs. of  21 variables:
##  $ age           : int  56 57 37 40 56 45 59 41 24 25 ...
##  $ job           : Factor w/ 12 levels "admin.","blue-collar",..: 4 8 8 1
8 8 1 2 10 8 ...
##  $ marital       : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 2 2
2 2 3 3 ...
##  $ education     : Factor w/ 8 levels "basic.4y","basic.6y",..: 1 4 4 2 4
3 6 8 6 4 ...
##  $ default       : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1
1 ...
##  $ housing       : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3
3 ...
##  $ loan          : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 3 1 1 1 1
1 ...
```

```
##  $ contact       : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 
2 2 2 2 ...
##  $ month         : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7 
7 7 7 ...
##  $ day_of_week   : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2 
2 2 2 ...
##  $ duration      : int  261 149 226 151 307 198 139 217 380 50 ...
##  $ campaign      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ pdays         : int  999 999 999 999 999 999 999 999 999 999 ...
##  $ previous      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ poutcome      : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 
2 2 2 2 2 2 ...
##  $ emp.var.rate  : num  1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...
##  $ cons.price.idx: num  94 94 94 94 94 ...
##  $ cons.conf.idx : num  -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -
36.4 -36.4 ...
##  $ euribor3m     : num  4.86 4.86 4.86 4.86 4.86 ...
##  $ nr.employed   : num  5191 5191 5191 5191 5191 ...
##  $ y             : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...

bank_modified=bank
unique(bank$job)

##  [1] housemaid     services      admin.        blue-collar   technician   
##  [6] retired       management    unemployed    self-employed unknown      
## [11] entrepreneur  student      
## 12 Levels: admin. blue-collar entrepreneur housemaid ... unknown

summary(bank$job)

##        admin.   blue-collar  entrepreneur     housemaid    management 
##         10422          9254          1456          1060          2924 
##       retired self-employed      services       student    technician 
##          1720          1421          3969           875          6743 
##    unemployed       unknown 
##          1014           330

#unknown = 0 ,student=1, unemployed=2, housemaid=3
#self-employed 4, entrepreneur 5, retired 6, management 7,services 8 ,
technician 9
# admin 11   blue-collar    10
bank_job= ifelse(bank$job== 'admin.', 11,
                 ifelse(bank$job=='blue-collar', 10,
                    ifelse(bank$job=='technician',9,
                          ifelse(bank$job=='services',8,
                              ifelse(bank$job=='management',7,
                                 ifelse(bank$job=='retired',6,
                                     ifelse(bank$job=='entrepreneur',5,
                                            ifelse(bank$job=='self-
employed',4,
```

```r
ifelse(bank$job=='housemaid',3,

ifelse(bank$job=='unemployed',2,

ifelse(bank$job=='student',1,0)))))))))))

#added column in new dataframe bank_modified
bank_modified=cbind(bank_modified,bank_job)
#head(bank_modified[,c('education','bank_education')],30)

#month from factor to numeric
unique(bank$month)

##  [1] may jun jul aug oct nov dec mar apr sep
## Levels: apr aug dec jul jun mar may nov oct sep

#may jun jul aug oct nov dec mar apr sep
bank_month=ifelse(bank$month=='mar',3,
                  ifelse(bank$month=='apr',4,
                         ifelse(bank$month=='may',5,
                                ifelse(bank$month=='jun',6,
                                       ifelse(bank$month=='jul',7,
                                              ifelse(bank$month=='aug',8,

ifelse(bank$month=='sep',9,

ifelse(bank$month=='oct',10,

ifelse(bank$month=='nov',11,

ifelse(bank$month=='dec',12,0))))))))))


#adding it to data frame bank_modified
bank_modified=cbind(bank_modified,bank_month)


#changing day of the week to numeric
#mon tue wed thu fri
bank_days=ifelse(bank$day_of_week=='mon',1,
                 ifelse(bank$day_of_week=='tue',2,
                        ifelse(bank$day_of_week=='wed',3,
                               ifelse(bank$day_of_week=='thu',4,
                                      ifelse(bank$day_of_week=='fri',5,0)))))


bank_modified=cbind(bank_modified, bank_days)

#loan from factor to numeric
```

```r
bank_loan= ifelse(bank$loan=='yes',1,0)
bank_modified=cbind(bank_modified,bank_loan)

#default from factor to numeric
bank_default= ifelse(bank$default=='yes',1,0)
bank_modified=cbind(bank_modified,bank_default)

#education from factor to numeric in the order of highest count: higher count
get the highest number
bank_education=ifelse(bank$education=='illiterate',1,
                      ifelse(bank$education=='basic.6y',2,
                             ifelse(bank$education=='basic.4y',3,

ifelse(bank$education=='professional.course',4,

ifelse(bank$education=='basic.9y',5,

ifelse(bank$education=='high.school',6,

ifelse(bank$education=='university.degree',7,0) ))))))

bank_modified=cbind(bank_modified,bank_education)


bank_contact=ifelse(bank$contact=='cellular',2,1)
bank_modified=cbind(bank_modified,bank_contact)

#changing marital from factor to numeric
#married 3, single 2, divorced 1 and unknown 0
bank_marital=ifelse(bank$marital=='married',3,
                    ifelse(bank$marital=='single',2,
                           ifelse(bank$marital=='divorced',1,0)))

bank_modified=cbind(bank_modified,bank_marital)


#Housing from factor to numeric
bank_housing= ifelse(bank$housing=='yes',1,0)

bank_modified=cbind(bank_modified,bank_housing)
head(bank_modified)

##   age       job marital   education default housing loan   contact month
## 1  56 housemaid married    basic.4y      no      no   no telephone   may
## 2  57  services married high.school unknown      no   no telephone   may
## 3  37  services married high.school      no     yes   no telephone   may
## 4  40    admin. married    basic.6y      no      no   no telephone   may
## 5  56  services married high.school      no      no  yes telephone   may
```

```
## 6  45   services married    basic.9y unknown      no    no telephone   may
##   day_of_week duration campaign pdays previous    poutcome emp.var.rate
## 1         mon      261        1   999        0 nonexistent          1.1
## 2         mon      149        1   999        0 nonexistent          1.1
## 3         mon      226        1   999        0 nonexistent          1.1
## 4         mon      151        1   999        0 nonexistent          1.1
## 5         mon      307        1   999        0 nonexistent          1.1
## 6         mon      198        1   999        0 nonexistent          1.1
##   cons.price.idx cons.conf.idx euribor3m nr.employed  y bank_job
## 1         93.994         -36.4     4.857        5191 no        3
## 2         93.994         -36.4     4.857        5191 no        8
## 3         93.994         -36.4     4.857        5191 no        8
## 4         93.994         -36.4     4.857        5191 no       11
## 5         93.994         -36.4     4.857        5191 no        8
## 6         93.994         -36.4     4.857        5191 no        8
##   bank_month bank_days bank_loan bank_default bank_education bank_contact
## 1          5         1         0            0              3            1
## 2          5         1         0            0              6            1
## 3          5         1         0            0              6            1
## 4          5         1         0            0              2            1
## 5          5         1         1            0              6            1
## 6          5         1         0            0              5            1
##   bank_marital bank_housing
## 1            3            0
## 2            3            0
## 3            3            1
## 4            3            0
## 5            3            0
## 6            3            0
```

```
bank_y=ifelse(bank$y=='yes',1,0)
bank_modified=cbind(bank_modified,bank_y)
str(bank_modified)
```

```
## 'data.frame':    41188 obs. of  31 variables:
##  $ age           : int  56 57 37 40 56 45 59 41 24 25 ...
##  $ job           : Factor w/ 12 levels "admin.","blue-collar",..: 4 8 8 1
8 8 1 2 10 8 ...
##  $ marital       : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 2 2
2 2 3 3 ...
##  $ education     : Factor w/ 8 levels "basic.4y","basic.6y",..: 1 4 4 2 4
3 6 8 6 4 ...
##  $ default       : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1
1 ...
##  $ housing       : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3
3 ...
##  $ loan          : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 3 1 1 1 1
1 ...
##  $ contact       : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2
2 2 2 2 ...
```

```
##  $ month          : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7
7 7 7 ...
##  $ day_of_week    : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2
2 2 2 ...
##  $ duration       : int  261 149 226 151 307 198 139 217 380 50 ...
##  $ campaign       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ pdays          : int  999 999 999 999 999 999 999 999 999 999 ...
##  $ previous       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ poutcome       : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2
2 2 2 2 2 2 ...
##  $ emp.var.rate   : num  1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...
##  $ cons.price.idx: num  94 94 94 94 94 ...
##  $ cons.conf.idx : num  -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -
36.4 -36.4 ...
##  $ euribor3m      : num  4.86 4.86 4.86 4.86 4.86 ...
##  $ nr.employed    : num  5191 5191 5191 5191 5191 ...
##  $ y              : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ bank_job       : num  3 8 8 11 8 8 11 10 9 8 ...
##  $ bank_month     : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ bank_days      : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ bank_loan      : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ bank_default   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ bank_education: num  3 6 6 2 6 5 4 0 4 6 ...
##  $ bank_contact   : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ bank_marital   : num  3 3 3 3 3 3 3 3 2 2 ...
##  $ bank_housing   : num  0 0 1 0 0 0 0 0 1 1 ...
##  $ bank_y         : num  0 0 0 0 0 0 0 0 0 0 ...

bank_int =
bank_modified[,c('age','duration','campaign','pdays','previous','emp.var.rate
','cons.price.idx','cons.conf.idx','euribor3m',

'nr.employed','bank_housing','bank_loan','bank_job','bank_education','bank_mo
nth',

'bank_days','bank_contact','bank_marital','bank_y')]

#x <- dist(scale(bank_int, center = FALSE))
#x
#as.dist(round(as.matrix(x), 2)[1:12, 1:12])

cm <- colMeans(bank_int)
S <- cov(bank_int) #diagonals are variances
d <- apply(bank_int, MARGIN = 1, function(bank_int)t(bank_int - cm) %*%
solve(S) %*% (bank_int - cm))
d

##     [1]  13.483570  10.965225   8.465088  11.200340  16.403112   7.753476
##     [7]  11.963118  14.971289  11.415000  10.980666  15.700590  10.306592
##    [13]  15.187492  19.965754  11.485333  15.530186  11.344293  16.562457
```

```
## [41113]  81.033098  85.361935  88.177385  75.392997  81.964863  84.208448
## [41119]  85.141638  80.272217  86.662477  88.238195  89.367902  99.472986
## [41125] 102.761993  77.067039 145.321422  87.138631  76.448562  92.293726
## [41131]  78.955814  78.715215  78.677503  87.524382  85.685552  76.106175
## [41137] 123.862141  71.644332  84.645921  76.892583  80.549722  81.009612
## [41143]  87.927478  88.317288 141.653435  80.691857  82.076280  86.412189
## [41149]  79.520392  77.112528  70.023167  84.653831  77.547485 152.020299
## [41155]  75.347386  89.714883  77.400000  79.860380  81.576760 106.642264
## [41161]  80.558208  72.149816  94.636587  78.714148 121.887284  81.597929
## [41167]  86.176878  76.333764  76.053576  79.277298 140.953475  97.895356
## [41173]  78.012911  87.086140 197.290508  89.405360  85.291229  78.584659
## [41179]  86.998770  84.778475  78.979693  81.191204  96.162788  87.684716
## [41185]  78.492152  79.813539  78.866403  79.565061
```

```
S
```

```
##                          age      duration      campaign         pdays
## age              108.602451165 -2.339147e+00   0.132602905 -6.694540e+01
## duration          -2.339146942  6.722573e+04 -51.494888397 -2.305683e+03
## campaign           0.132602905 -5.149489e+01   7.672975028  2.722492e+01
## pdays            -66.945399639 -2.305683e+03  27.224921359  3.493569e+04
## previous           0.125660856  2.648520e+00  -0.108493675 -5.434645e+01
## emp.var.rate      -0.006068627 -1.139180e+01   0.656017208  7.957482e+01
## cons.price.idx     0.005167908  7.972716e-01   0.204971433  8.535132e+00
## cons.conf.idx      6.239800832 -9.807412e+00  -0.176060671 -7.901668e+01
## euribor3m          0.194622473 -1.479383e+01   0.649236419  9.625087e+01
## nr.employed      -13.346159899 -8.374399e+02  28.838822297  5.031877e+03
## bank_housing      -0.007359781 -8.810154e-01  -0.014941021 -1.000641e+00
## bank_loan         -0.026909690  1.127191e-02   0.005260933 -5.769701e-03
## bank_job          -2.942361746 -4.332399e+00   0.099531269  2.971107e+01
## bank_education    -3.499404013 -4.902691e+00  -0.014834970 -5.131918e+00
## bank_month         1.643396595 -1.021420e+01  -0.173194966 -3.034951e+01
## bank_days         -0.271924763  3.860555e+00   0.059031862  1.784832e+00
## bank_contact      -0.035230419  3.327960e+00  -0.103191259 -1.061719e+01
## bank_marital       0.834924575 -2.204498e-01  -0.004379889  2.296641e+00
## bank_y             0.100161695  3.322321e+01  -0.058116134 -1.920123e+01
##                       previous  emp.var.rate cons.price.idx cons.conf.idx
## age               1.256609e-01  -0.006068627    0.005167908   6.239800832
## duration          2.648520e+00 -11.391802119    0.797271590  -9.807411936
## campaign         -1.084937e-01   0.656017208    0.204971433  -0.176060671
## pdays            -5.434645e+01  79.574823033    8.535132485 -79.016677080
## previous          2.449271e-01  -0.326917530   -0.058190350  -0.116669718
## emp.var.rate     -3.269175e-01   2.467914506    0.705038043   1.425359699
## cons.price.idx   -5.819035e-02   0.705038043    0.335055802   0.158023171
## cons.conf.idx    -1.166697e-01   1.425359699    0.158023171  21.420215396
## euribor3m        -3.901282e-01   2.649120795    0.690960743   2.229088852
## nr.employed      -1.792634e+01 102.944953096   21.832545607  33.611125259
## bank_housing      5.053787e-03  -0.046967770   -0.023682580  -0.079736240
## bank_loan        -3.805283e-04   0.000672042   -0.001234151  -0.022700209
## bank_job         -7.011819e-02   0.305567788    0.033233387  -0.543150008
```

```
## bank_education  1.717054e-02  -0.134824416   -0.091864771   0.080741473
## bank_month      6.439781e-02   0.188769535   -0.177624990   2.495921179
## bank_days       2.803357e-03  -0.009758766   -0.003747063  -0.000647943
## bank_contact    5.072123e-02  -0.297718079   -0.164852751  -0.560723493
## bank_marital   -1.042179e-02   0.053560974    0.010080996   0.164508488
## bank_y          3.601747e-02  -0.148181434   -0.024928537   0.080303622
##                    euribor3m   nr.employed bank_housing     bank_loan
## age             1.946225e-01  -13.34615990 -0.007359781 -0.0269096903
## duration       -1.479383e+01 -837.43986594 -0.881015369  0.0112719132
## campaign        6.492364e-01   28.83882230 -0.014941021  0.0052609329
## pdays           9.625087e+01 5031.87747726 -1.000640878 -0.0057697014
## previous       -3.901282e-01  -17.92634155  0.005053787 -0.0003805283
## emp.var.rate    2.649121e+00  102.94495310 -0.046967770  0.0006720420
## cons.price.idx  6.909607e-01   21.83254561 -0.023682580 -0.0012341509
## cons.conf.idx   2.229089e+00   33.61112526 -0.079736240 -0.0227002088
## euribor3m       3.008308e+00  118.44342135 -0.051032986 -0.0003029111
## nr.employed     1.184434e+02 5220.28325040 -1.625501118  0.1089889629
## bank_housing   -5.103299e-02   -1.62550112  0.249437620  0.0101497034
## bank_loan      -3.029111e-04    0.10898896  0.010149703  0.1286865203
## bank_job        3.009990e-01   16.87042768 -0.000599546  0.0095512656
## bank_education -1.215829e-01   -3.46369470  0.013488811  0.0083561490
## bank_month      5.784763e-01   19.56820263  0.033895322 -0.0012414380
## bank_days      -1.359129e-02   -0.07489496 -0.006542088  0.0009365872
## bank_contact   -3.338696e-01   -9.36380789  0.020314116  0.0023088497
## bank_marital    6.601380e-02    2.52009321 -0.001858772  0.0003992101
## bank_y         -1.687776e-01   -8.10227564  0.001854313 -0.0005065497
##                    bank_job bank_education    bank_month     bank_days
## age            -2.942361746   -3.499404013   1.643396595 -0.2719247626
## duration       -4.332399142   -4.902691443 -10.214195870  3.8605553591
## campaign        0.099531269   -0.014834970  -0.173194966  0.0590318619
## pdays          29.711067606   -5.131917536 -30.349513624  1.7848315446
## previous       -0.070118187    0.017170543   0.064397806  0.0028033574
## emp.var.rate    0.305567788   -0.134824416   0.188769535 -0.0097587660
## cons.price.idx  0.033233387   -0.091864771  -0.177624990 -0.0037470626
## cons.conf.idx  -0.543150008    0.080741473   2.495921179 -0.0006479430
## euribor3m       0.300998973   -0.121582906   0.578476282 -0.0135912892
## nr.employed    16.870427679   -3.463694700  19.568202631 -0.0748949643
## bank_housing   -0.000599546    0.013488811   0.033895322 -0.0065420877
## bank_loan       0.009551266    0.008356149  -0.001241438  0.0009365872
## bank_job        7.296841069    0.458591949  -0.241356037  0.0399007279
## bank_education  0.458591949    3.491377171   0.324628213 -0.0185007988
## bank_month     -0.241356037    0.324628213   4.165671508 -0.0200490417
## bank_days       0.039900728   -0.018500799  -0.020049042  1.9923720588
## bank_contact    0.001663345    0.078289694   0.318721440 -0.0133094652
## bank_marital   -0.008857822   -0.128851803   0.005007848  0.0041410316
## bank_y         -0.047701363    0.013616436   0.023997010  0.0044854483
##                bank_contact  bank_marital        bank_y
## age             -0.035230419  0.8349245747  1.001617e-01
## duration         3.327960119 -0.2204497908  3.322321e+01
## campaign        -0.103191259 -0.0043798889 -5.811613e-02
```

```
## pdays          -10.617191584  2.2966409027 -1.920123e+01
## previous         0.050721231 -0.0104217886  3.601747e-02
## emp.var.rate    -0.297718079  0.0535609743 -1.481814e-01
## cons.price.idx  -0.164852751  0.0100809963 -2.492854e-02
## cons.conf.idx   -0.560723493  0.1645084877  8.030362e-02
## euribor3m       -0.333869619  0.0660137999 -1.687776e-01
## nr.employed     -9.363807894  2.5200932089 -8.102276e+00
## bank_housing     0.020314116 -0.0018587722  1.854313e-03
## bank_loan        0.002308850  0.0003992101 -5.065497e-04
## bank_job         0.001663345 -0.0088578225 -4.770136e-02
## bank_education   0.078289694 -0.1288518035  1.361644e-02
## bank_month       0.318721440  0.0050078479  2.399701e-02
## bank_days       -0.013309465  0.0041410316  4.485448e-03
## bank_contact     0.231848610 -0.0146157861  2.204019e-02
## bank_marital    -0.014615786  0.4855013047 -5.794485e-03
## bank_y           0.022040191 -0.0057944855  9.996564e-02
```

```
cm
```

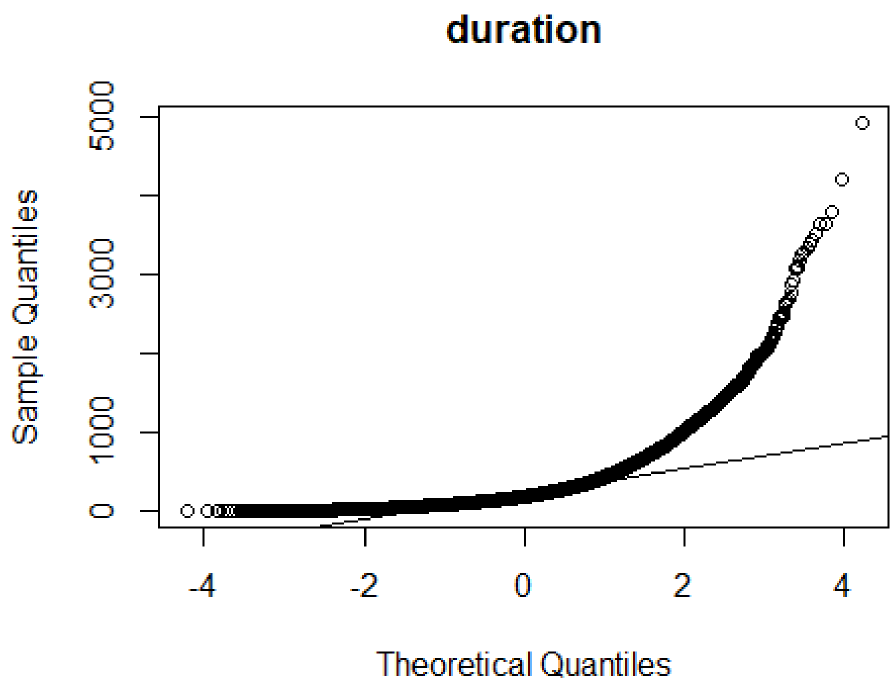
```
##            age       duration       campaign          pdays       previous
##     40.0240604    258.2850102      2.5675925    962.4754540      0.1729630
##   emp.var.rate cons.price.idx  cons.conf.idx      euribor3m    nr.employed
##      0.0818855     93.5756644    -40.5026003      3.6212908   5167.0359109
##   bank_housing      bank_loan       bank_job bank_education     bank_month
##      0.5238419      0.1516947      8.4844129      5.1129698      6.6078955
##      bank_days   bank_contact   bank_marital         bank_y
##      2.9795814      1.6347480      2.4893658      0.1126542
```

```
str(bank_int)
```

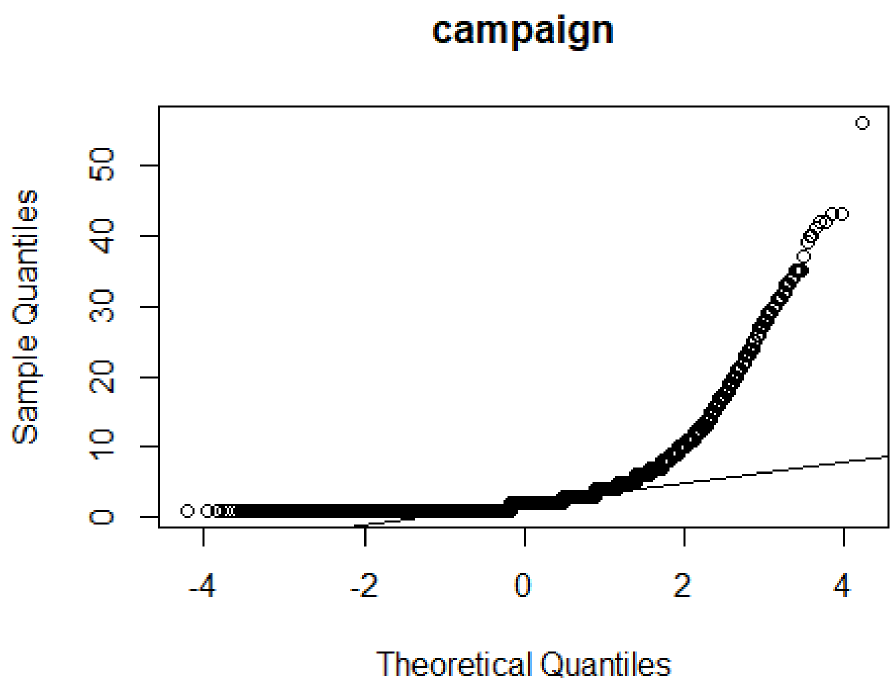
```
## 'data.frame':    41188 obs. of  19 variables:
##  $ age           : int  56 57 37 40 56 45 59 41 24 25 ...
##  $ duration      : int  261 149 226 151 307 198 139 217 380 50 ...
##  $ campaign      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ pdays         : int  999 999 999 999 999 999 999 999 999 999 ...
##  $ previous      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ emp.var.rate  : num  1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...
##  $ cons.price.idx: num  94 94 94 94 94 ...
##  $ cons.conf.idx : num  -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -
36.4 -36.4 ...
##  $ euribor3m     : num  4.86 4.86 4.86 4.86 4.86 ...
##  $ nr.employed   : num  5191 5191 5191 5191 5191 ...
##  $ bank_housing  : num  0 0 1 0 0 0 0 0 1 1 ...
##  $ bank_loan     : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ bank_job      : num  3 8 8 11 8 8 11 10 9 8 ...
##  $ bank_education: num  3 6 6 2 6 5 4 0 4 6 ...
##  $ bank_month    : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ bank_days     : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ bank_contact  : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ bank_marital  : num  3 3 3 3 3 3 3 3 2 2 ...
##  $ bank_y        : num  0 0 0 0 0 0 0 0 0 0 ...
```

```r
qqnorm(bank_int[,"age"], main = "age")
#how nomal looks like - univariate normalization
qqline(bank_int[,"age"])
```

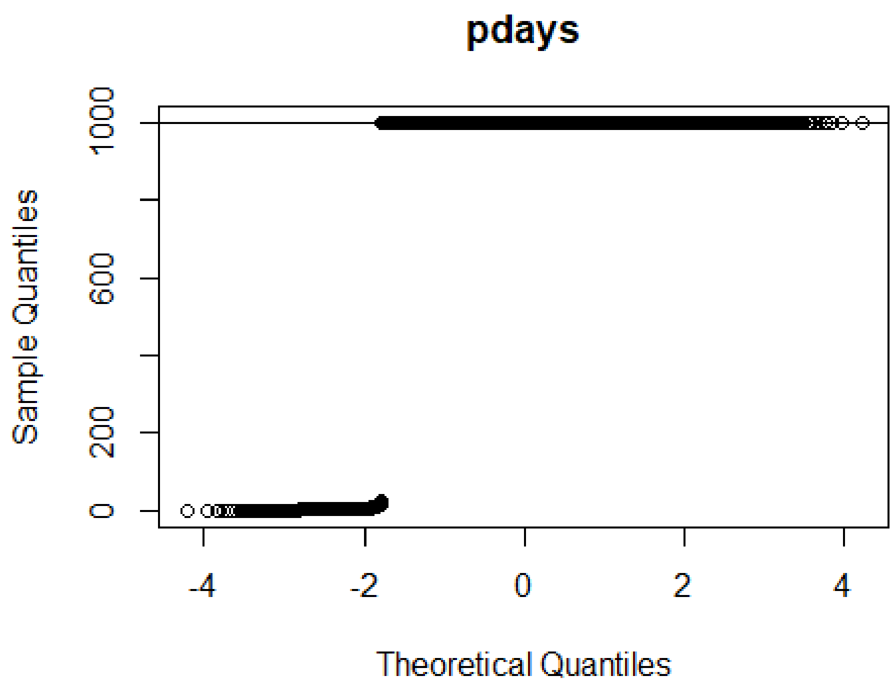
```r
qqnorm(bank_int[,"duration"], main = "duration")
#how nomal looks like - univariate normalization
qqline(bank_int[,"duration"])
```



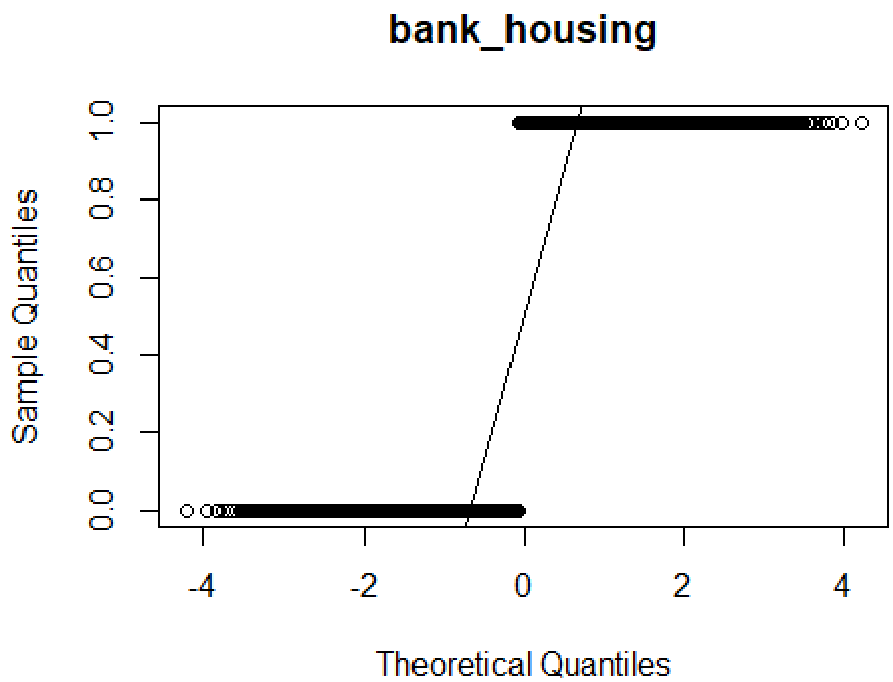
```r
qqnorm(bank_int[,"campaign"], main = "campaign")
#how nomal looks like - univariate normalization
qqline(bank_int[,"campaign"])
```



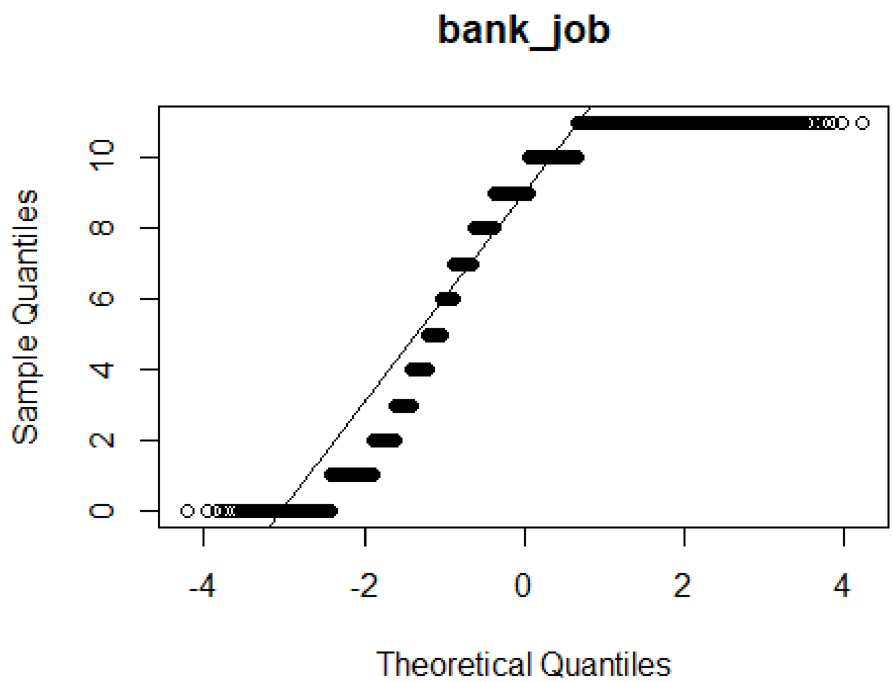
```r
qqnorm(bank_int[,"pdays"], main = "pdays")
#how nomal looks like - univariate normalization
qqline(bank_int[,"pdays"])
```



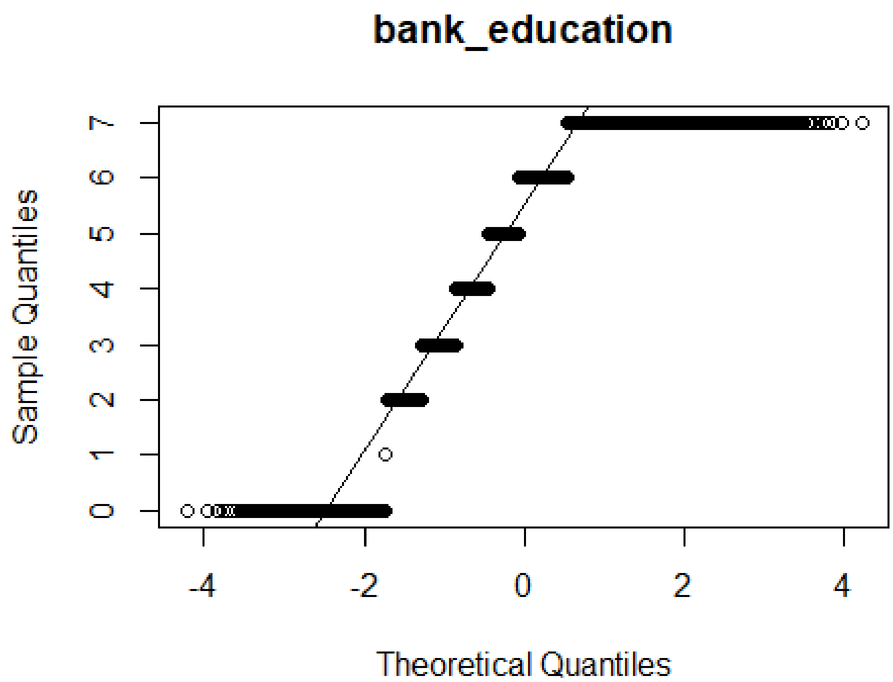
```r
qqnorm(bank_int[,"bank_housing"], main = "bank_housing")
#how nomal looks like - univariate normalization
qqline(bank_int[,"bank_housing"])
```



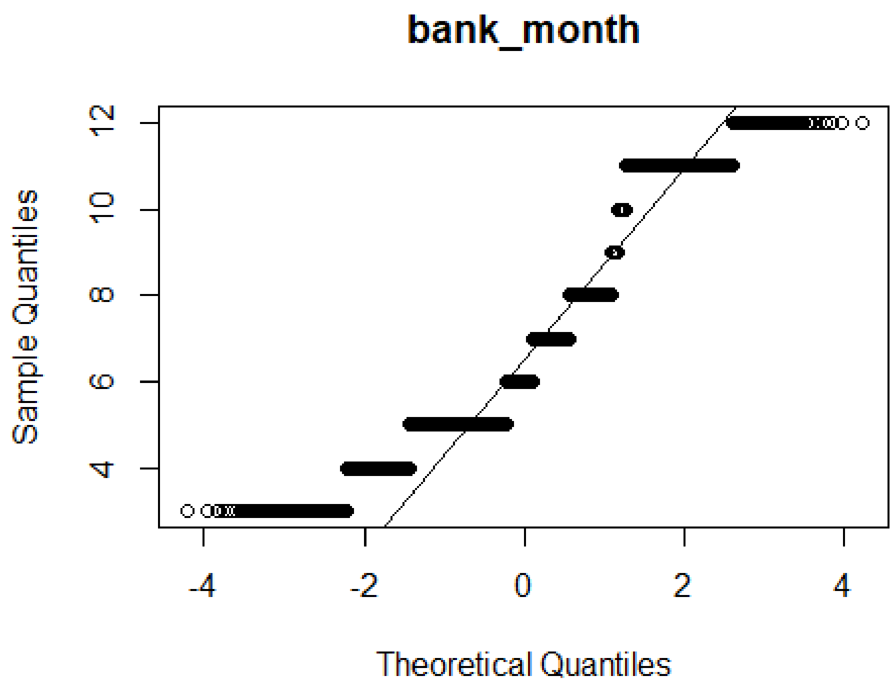
```r
qqnorm(bank_int[,"bank_job"], main = "bank_job")
#how nomal looks like - univariate normalization
qqline(bank_int[,"bank_job"])
```



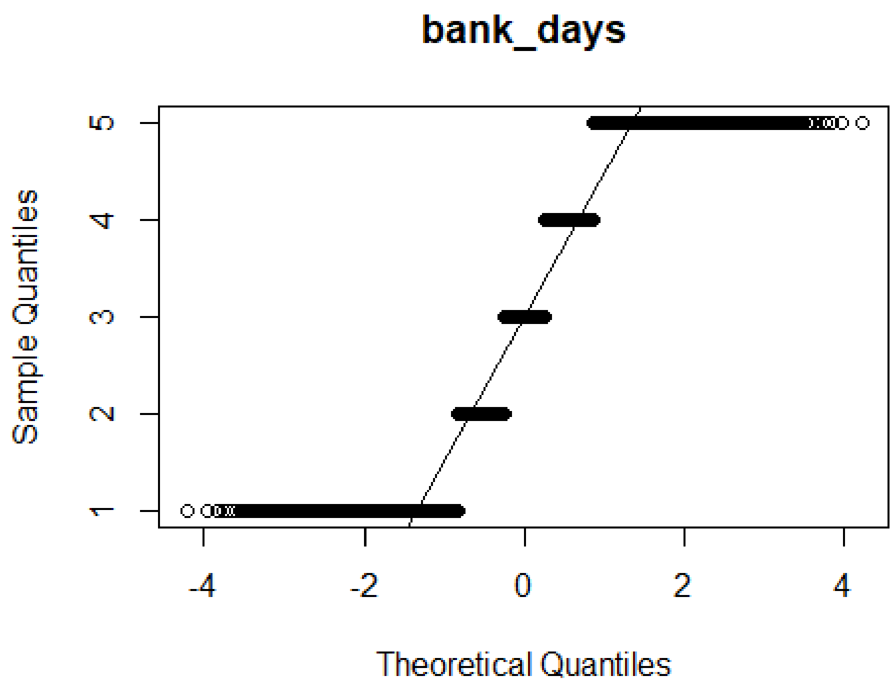
```r
qqnorm(bank_int[,"bank_education"], main = "bank_education")
#how nomal looks like - univariate normalization
qqline(bank_int[,"bank_education"])
```



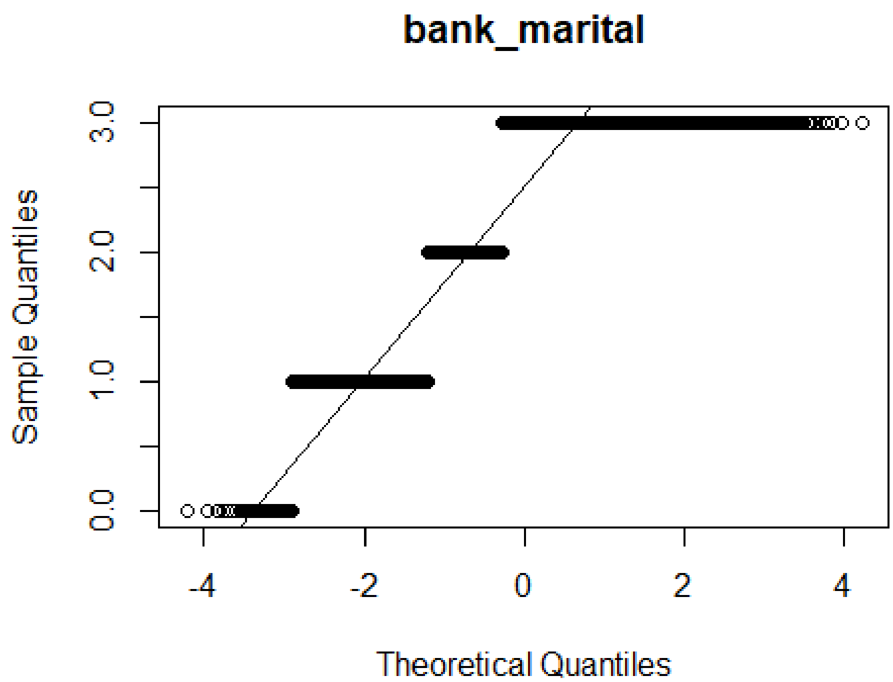
```r
qqnorm(bank_int[,"bank_month"], main = "bank_month")
#how nomal looks like - univariate normalization
qqline(bank_int[,"bank_month"])
```



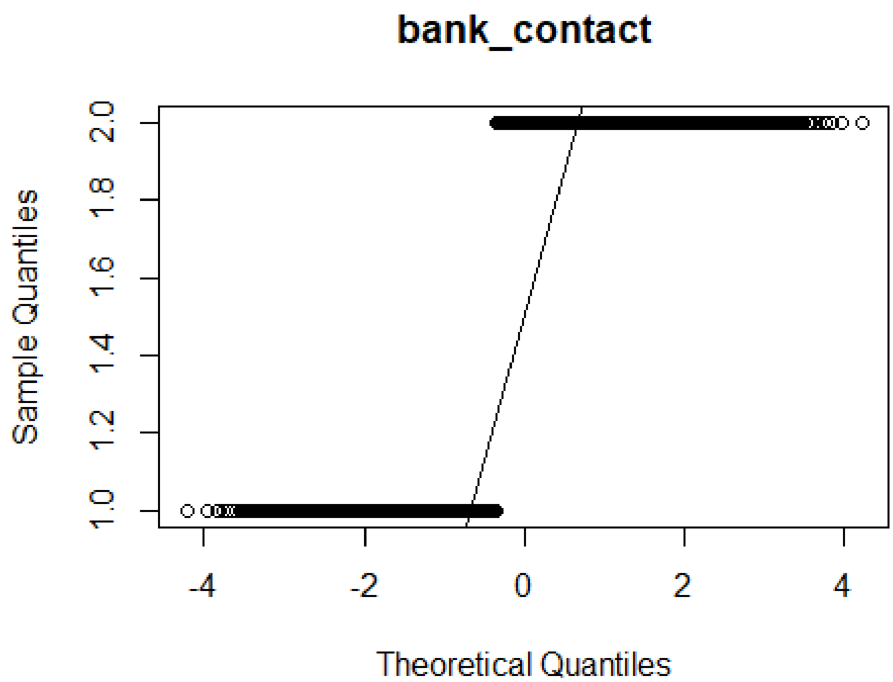
```r
qqnorm(bank_int[,"bank_days"], main = "bank_days")
#how nomal looks like - univariate normalization
qqline(bank_int[,"bank_days"])
```



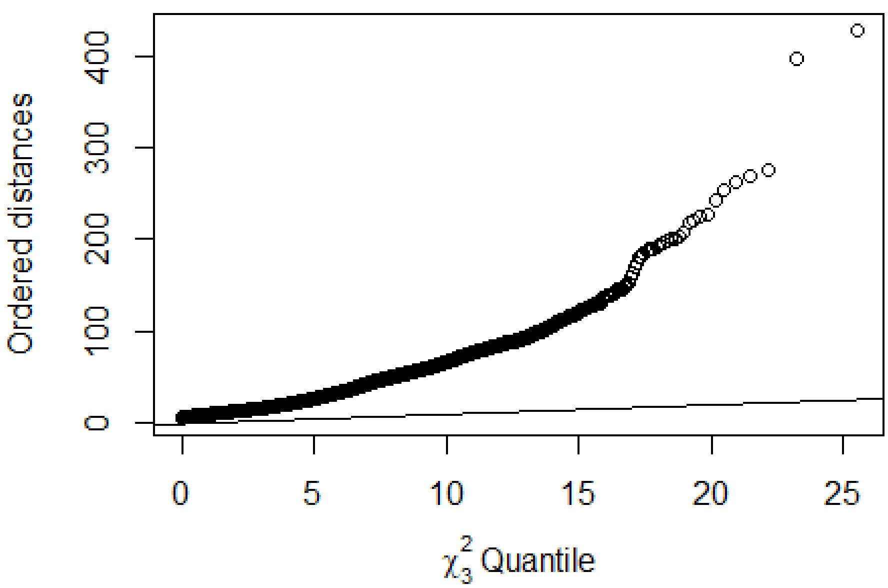
```r
qqnorm(bank_int[,"bank_marital"], main = "bank_marital")
#how nomal looks like - univariate normalization
qqline(bank_int[,"bank_marital"])
```



```r
qqnorm(bank_int[,"bank_contact"], main = "bank_contact")
#how nomal looks like - univariate normalization
qqline(bank_int[,"bank_contact"])
```



```r
#individually they had outliers
#all of them together or how they interact with each other
#they look they are normally multivariate
plot(qchisq((1:nrow(bank_int) - 1/2) / nrow(bank_int), df = 3), sort(d),
     xlab = expression(paste(chi[3]^2, " Quantile")),
     ylab = "Ordered distances")
abline(a = 0, b = 1)
```




```r
t.test(bank$age[bank$y=="yes"],bank$age[bank$y=="no"],var.equal=TRUE)

##
##  Two Sample t-test
##
## data:  bank$age[bank$y == "yes"] and bank$age[bank$y == "no"]
## t = 6.1721, df = 41186, p-value = 6.802e-10
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.6837762 1.3201463
## sample estimates:
## mean of x mean of y
##  40.91315  39.91119



#Age is not significant
```

```r
t.test(bank$duration[bank$y=="no"],bank$duration[bank$y=="yes"],var.equal=TRU
E)
```

```
##
##  Two Sample t-test
##
## data:  bank$duration[bank$y == "no"] and bank$duration[bank$y == "yes"]
## t = -89.967, df = 41186, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -339.5868 -325.1059
## sample estimates:
## mean of x mean of y
##  220.8448  553.1912
```

```r
#Duration is Significant

t.test(bank_modified$bank_job[bank_modified$y=='yes'],bank_modified$bank_job[
bank_modified$y=='no'],var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  bank_modified$bank_job[bank_modified$y == "yes"] and
bank_modified$bank_job[bank_modified$y == "no"]
## t = -11.352, df = 41186, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.5595629 -0.3947923
## sample estimates:
## mean of x mean of y
##  8.060991  8.538169
```

```r
#Job is significant

t.test(bank_modified$bank_housing[bank_modified$y=='yes'],bank_modified$bank_
housing[bank_modified$y=='no'],var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  bank_modified$bank_housing[bank_modified$y == "yes"] and
bank_modified$bank_housing[bank_modified$y == "no"]
## t = 2.3833, df = 41186, p-value = 0.01716
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.003294525 0.033804490
## sample estimates:
## mean of x mean of y
## 0.5403017 0.5217522
```

```r
#housing is significant

t.test(bank_modified$bank_month[bank_modified$y=='yes'],bank_modified$bank_mo
nth[bank_modified$y=='no'],var.equal=TRUE)

##
##  Two Sample t-test
##
## data:  bank_modified$bank_month[bank_modified$y == "yes"] and
bank_modified$bank_month[bank_modified$y == "no"]
## t = 7.552, df = 41186, p-value = 4.373e-14
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1777505 0.3023547
## sample estimates:
## mean of x mean of y
##  6.820905  6.580853

#month is signficant

t.test(bank_modified$bank_loan[bank_modified$y=='yes'],bank_modified$bank_lao
n[bank_modified$y=='no'],var.equal=TRUE)

##
##  One Sample t-test
##
## data:  bank_modified$bank_loan[bank_modified$y == "yes"]
## t = 28.297, df = 4639, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.1370000 0.1573965
## sample estimates:
## mean of x
## 0.1471983

#laon is significant

t.test(bank_modified$bank_days[bank_modified$y=='yes'],bank_modified$bank_day
s[bank_modified$y=='no'],var.equal=TRUE)

##
##  Two Sample t-test
##
## data:  bank_modified$bank_days[bank_modified$y == "yes"] and
bank_modified$bank_days[bank_modified$y == "no"]
## t = 2.0398, df = 41186, p-value = 0.04137
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.001755341 0.087984462
## sample estimates:
```

```
## mean of x mean of y
##  3.019397  2.974527

#Days of week is significant

t.test(bank_modified$bank_default[bank_modified$y=='yes'],bank_modified$bank_
default[bank_modified$y=='no'],var.equal=TRUE)

##
##  Two Sample t-test
##
## data:  bank_modified$bank_default[bank_modified$y == "yes"] and
bank_modified$bank_default[bank_modified$y == "no"]
## t = -0.61716, df = 41186, p-value = 0.5371
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.0003427733  0.0001786056
## sample estimates:
##    mean of x    mean of y
## 0.000000e+00 8.208383e-05

#Default is not significant

t.test(bank_modified$bank_month[bank_modified$y=='yes'],bank_modified$bank_mo
nth[bank_modified$y=='no'],var.equal=TRUE)

##
##  Two Sample t-test
##
## data:  bank_modified$bank_month[bank_modified$y == "yes"] and
bank_modified$bank_month[bank_modified$y == "no"]
## t = 7.552, df = 41186, p-value = 4.373e-14
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1777505 0.3023547
## sample estimates:
## mean of x mean of y
##  6.820905  6.580853

#month is signficant

t.test(bank_modified$bank_education[bank_modified$y=='yes'],bank_modified$ban
k_education[bank_modified$y=='no'],var.equal=TRUE)

##
##  Two Sample t-test
##
## data:  bank_modified$bank_education[bank_modified$y == "yes"] and
bank_modified$bank_education[bank_modified$y == "no"]
## t = 4.6788, df = 41186, p-value = 2.895e-06
## alternative hypothesis: true difference in means is not equal to 0
```

```
## 95 percent confidence interval:
##  0.0791496 0.1932727
## sample estimates:
## mean of x mean of y
##  5.233836  5.097625
```

```r
#Significant

t.test(bank_modified$bank_contact[bank_modified$y=='yes'],bank_modified$bank_contact[bank_modified$y=='no'],var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  bank_modified$bank_contact[bank_modified$y == "yes"] and
bank_modified$bank_contact[bank_modified$y == "no"]
## t = 29.694, df = 41186, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.2059243 0.2350310
## sample estimates:
## mean of x mean of y
##  1.830388  1.609910
```

```r
#significant

t.test(bank_modified$bank_marital[bank_modified$y=='yes'],bank_modified$bank_marital[bank_modified$y=='no'],var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  bank_modified$bank_marital[bank_modified$y == "yes"] and
bank_modified$bank_marital[bank_modified$y == "no"]
## t = -5.3397, df = 41186, p-value = 9.358e-08
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.07924154 -0.03668801
## sample estimates:
## mean of x mean of y
##  2.437931  2.495896
```

```r
#significant

t.test(bank$emp.var.rate[bank$y=="no"],bank$emp.var.rate[bank$y=="yes"],var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  bank$emp.var.rate[bank$y == "no"] and bank$emp.var.rate[bank$y ==
```

```
"yes"]
## t = 63.434, df = 41186, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.436522 1.528126
## sample estimates:
##  mean of x  mean of y
##  0.2488755 -1.2334483

#Significant

t.test(bank$cons.price.idx[bank$y=="no"],bank$cons.price.idx[bank$y=="yes"],v
ar.equal=TRUE)

##
##  Two Sample t-test
##
## data:  bank$cons.price.idx[bank$y == "no"] and bank$cons.price.idx[bank$y
== "yes"]
## t = 27.903, df = 41186, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.2318544 0.2668878
## sample estimates:
## mean of x mean of y
##  93.60376  93.35439

#Significant

t.test(bank$cons.conf.idx[bank$y=="no"],bank$cons.conf.idx[bank$y=="yes"],var
.equal=TRUE)

##
##  Two Sample t-test
##
## data:  bank$cons.conf.idx[bank$y == "no"] and bank$cons.conf.idx[bank$y ==
"yes"]
## t = -11.154, df = 41186, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.9444742 -0.6621503
## sample estimates:
## mean of x mean of y
## -40.59310 -39.78978

#Significant
```

```r
t.test(bank$euribor3m[bank$y=="no"],bank$euribor3m[bank$y=="yes"],var.equal=T
RUE)

##
##  Two Sample t-test
##
## data:  bank$euribor3m[bank$y == "no"] and bank$euribor3m[bank$y == "yes"]
## t = 65.647, df = 41186, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.637947 1.738766
## sample estimates:
## mean of x mean of y
##  3.811491  2.123135

#Significant

t.test(bank$nr.employed[bank$y=="no"],bank$nr.employed[bank$y=="yes"],var.equ
al=TRUE)

##
##  Two Sample t-test
##
## data:  bank$nr.employed[bank$y == "no"] and bank$nr.employed[bank$y ==
"yes"]
## t = 76.984, df = 41186, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  78.98706 83.11415
## sample estimates:
## mean of x mean of y
##  5176.167  5095.116

#Significant

t.test(bank$campaign[bank$y=="no"],bank$campaign[bank$y=="yes"],var.equal=TRU
E)

##
##  Two Sample t-test
##
## data:  bank$campaign[bank$y == "no"] and bank$campaign[bank$y == "yes"]
## t = 13.497, df = 41186, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.4969335 0.6657887
## sample estimates:
## mean of x mean of y
##  2.633085  2.051724
```

```r
#significant

library(Hotelling)

## Warning: package 'Hotelling' was built under R version 3.5.2

## Loading required package: corpcor

## Warning: package 'corpcor' was built under R version 3.5.2

t2testbank_int <- hotelling.test( .  ~ bank_y, data=bank_int)
cat("T2 statistic =",t2testbank_int$stat[[1]],"\n")

## T2 statistic = 20882.55

print(t2testbank_int)

## Test stat:  1159.7
## Numerator df:  18
## Denominator df:  41169
## P-value:  0

View(t2testbank_int)
```

Name	Type	Value
t2testbank_int	list [2] (S3: hotelling.test)	List of length 2
stats	list [6]	List of length 6
statistic	double [1]	16.99393
m	double [1]	0.05553262
df	double [2]	18 41169
nx	integer [1]	41185
ny	integer [1]	3
p	integer [1]	18
pval	double [1]	0.5240185

```r
#if we include bank_default it becomes singular
#and it gives error Lapack routine dgesv: system is exactly singular:
U[13,13] = 0
```

```r
#
#no much info p should be less than .05
# testing Variation
# F-test for Total length (not recommended)


#close to 1 F=.7 is not helping
var.test(bank_int$age[bank_int$bank_y=="1"],bank_int$age[bank_int$bank_y=="0"
])
```

```
##
##  F test to compare two variances
##
## data:  bank_int$age[bank_int$bank_y == "1"] and
bank_int$age[bank_int$bank_y == "0"]
## F = 1.9544, num df = 4639, denom df = 36547, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.872416 2.041412
## sample estimates:
## ratio of variances
##           1.954372
```

Significant

```r
var.test(bank_int$duration[bank_int$bank_y=="1"],bank_int$duration[bank_int$b
ank_y=="0"])
```

```
##
##  F test to compare two variances
##
## data:  bank_int$duration[bank_int$bank_y == "1"] and
bank_int$duration[bank_int$bank_y == "0"]
## F = 3.7525, num df = 4639, denom df = 36547, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  3.595103 3.919580
## sample estimates:
## ratio of variances
##           3.752462
```

```r
var.test(bank_int$campaign[bank_int$bank_y=="1"],bank_int$campaign[bank_int$b
ank_y=="0"])
```

```
##
##  F test to compare two variances
##
```

```
## data:  bank_int$campaign[bank_int$bank_y == "1"] and
bank_int$campaign[bank_int$bank_y == "0"]
## F = 0.33626, num df = 4639, denom df = 36547, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.3221580 0.3512344
## sample estimates:
## ratio of variances
##          0.3362589
```

```
var.test(bank_int$pdays[bank_int$bank_y=="1"],bank_int$pdays[bank_int$bank_y=
="0"])
```

```
##
##  F test to compare two variances
##
## data:  bank_int$pdays[bank_int$bank_y == "1"] and
bank_int$pdays[bank_int$bank_y == "0"]
## F = 11.178, num df = 4639, denom df = 36547, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  10.70972 11.67633
## sample estimates:
## ratio of variances
##           11.17849
```

```
var.test(bank_int$previous[bank_int$bank_y=="1"],bank_int$previous[bank_int$b
ank_y=="0"])
```

```
##
##  F test to compare two variances
##
## data:  bank_int$previous[bank_int$bank_y == "1"] and
bank_int$previous[bank_int$bank_y == "0"]
## F = 4.4205, num df = 4639, denom df = 36547, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  4.235161 4.617407
## sample estimates:
## ratio of variances
##           4.420535
```

```
var.test(bank_int$emp.var.rate[bank_int$bank_y=="1"],bank_int$emp.var.rate[ba
nk_int$bank_y=="0"])
```

```
##
##  F test to compare two variances
##
## data:  bank_int$emp.var.rate[bank_int$bank_y == "1"] and
bank_int$emp.var.rate[bank_int$bank_y == "0"]
## F = 1.1988, num df = 4639, denom df = 36547, p-value < 2.2e-16
```

```
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.148483 1.252140
## sample estimates:
## ratio of variances
##           1.198753
```

```
var.test(bank_int$cons.price.idx[bank_int$bank_y=="1"],bank_int$cons.price.id
x[bank_int$bank_y=="0"])
```

```
##
##  F test to compare two variances
##
## data:  bank_int$cons.price.idx[bank_int$bank_y == "1"] and
bank_int$cons.price.idx[bank_int$bank_y == "0"]
## F = 1.4652, num df = 4639, denom df = 36547, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.403792 1.530491
## sample estimates:
## ratio of variances
##           1.465236
```

```
var.test(bank_int$cons.conf.idx[bank_int$bank_y=="1"],bank_int$cons.conf.idx[
bank_int$bank_y=="0"])
```

```
##
##  F test to compare two variances
##
## data:  bank_int$cons.conf.idx[bank_int$bank_y == "1"] and
bank_int$cons.conf.idx[bank_int$bank_y == "0"]
## F = 1.9549, num df = 4639, denom df = 36547, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.872955 2.041999
## sample estimates:
## ratio of variances
##           1.954935
```

```
var.test(bank_int$euribor3m[bank_int$bank_y=="1"],bank_int$euribor3m[bank_int
$bank_y=="0"])
```

```
##
##  F test to compare two variances
##
## data:  bank_int$euribor3m[bank_int$bank_y == "1"] and
bank_int$euribor3m[bank_int$bank_y == "0"]
## F = 1.1315, num df = 4639, denom df = 36547, p-value = 1.184e-08
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.084082 1.181926
```

```
## sample estimates:
## ratio of variances
##           1.131533
```

```r
var.test(bank_int$nr.employed[bank_int$bank_y=="1"],bank_int$nr.employed[bank
_int$bank_y=="0"])
```

```
##
##  F test to compare two variances
##
## data:  bank_int$nr.employed[bank_int$bank_y == "1"] and
bank_int$nr.employed[bank_int$bank_y == "0"]
## F = 1.8393, num df = 4639, denom df = 36547, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.762154 1.921197
## sample estimates:
## ratio of variances
##           1.839284
```

```r
var.test(bank_int$bank_housing[bank_int$bank_y=="1"],bank_int$bank_housing[ba
nk_int$bank_y=="0"])
```

```
##
##  F test to compare two variances
##
## data:  bank_int$bank_housing[bank_int$bank_y == "1"] and
bank_int$bank_housing[bank_int$bank_y == "0"]
## F = 0.99557, num df = 4639, denom df = 36547, p-value = 0.8451
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.9538252 1.0399128
## sample estimates:
## ratio of variances
##          0.9955743
```

```r
var.test(bank_int$bank_loan[bank_int$bank_y=="1"],bank_int$bank_loan[bank_int
$bank_y=="0"])
```

```
##
##  F test to compare two variances
##
## data:  bank_int$bank_loan[bank_int$bank_y == "1"] and
bank_int$bank_loan[bank_int$bank_y == "0"]
## F = 0.97268, num df = 4639, denom df = 36547, p-value = 0.2126
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.9318933 1.0160015
## sample estimates:
## ratio of variances
##          0.9726825
```

```r
var.test(bank_int$bank_job[bank_int$bank_y=="1"],bank_int$bank_job[bank_int$b
ank_y=="0"])

##
##  F test to compare two variances
##
## data:  bank_int$bank_job[bank_int$bank_y == "1"] and
bank_int$bank_job[bank_int$bank_y == "0"]
## F = 1.3806, num df = 4639, denom df = 36547, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.322709 1.442091
## sample estimates:
## ratio of variances
##           1.380605

var.test(bank_int$bank_education[bank_int$bank_y=="1"],bank_int$bank_educatio
n[bank_int$bank_y=="0"])

##
##  F test to compare two variances
##
## data:  bank_int$bank_education[bank_int$bank_y == "1"] and
bank_int$bank_education[bank_int$bank_y == "0"]
## F = 1.1133, num df = 4639, denom df = 36547, p-value = 7.705e-07
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.066612 1.162879
## sample estimates:
## ratio of variances
##           1.113297

var.test(bank_int$bank_month[bank_int$bank_y=="1"],bank_int$bank_month[bank_i
nt$bank_y=="0"])

##
##  F test to compare two variances
##
## data:  bank_int$bank_month[bank_int$bank_y == "1"] and
bank_int$bank_month[bank_int$bank_y == "0"]
## F = 1.4489, num df = 4639, denom df = 36547, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.388178 1.513469
## sample estimates:
## ratio of variances
##           1.448939

var.test(bank_int$bank_days[bank_int$bank_y=="1"],bank_int$bank_days[bank_int
$bank_y=="0"])
```

```
##
##  F test to compare two variances
##
## data:  bank_int$bank_days[bank_int$bank_y == "1"] and
bank_int$bank_days[bank_int$bank_y == "0"]
## F = 0.94262, num df = 4639, denom df = 36547, p-value = 0.007934
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.9030892 0.9845977
## sample estimates:
## ratio of variances
##          0.9426177
```

```
var.test(bank_int$bank_contact[bank_int$bank_y=="1"],bank_int$bank_contact[ba
nk_int$bank_y=="0"])
```

```
##
##  F test to compare two variances
##
## data:  bank_int$bank_contact[bank_int$bank_y == "1"] and
bank_int$bank_contact[bank_int$bank_y == "0"]
## F = 0.59209, num df = 4639, denom df = 36547, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.5672626 0.6184610
## sample estimates:
## ratio of variances
##          0.5920918
```

```
var.test(bank_int$bank_marital[bank_int$bank_y=="1"],bank_int$bank_marital[ba
nk_int$bank_y=="0"])
```

```
##
##  F test to compare two variances
##
## data:  bank_int$bank_marital[bank_int$bank_y == "1"] and
bank_int$bank_marital[bank_int$bank_y == "0"]
## F = 0.95784, num df = 4639, denom df = 36547, p-value = 0.05265
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.9176693 1.0004937
## sample estimates:
## ratio of variances
##          0.9578359
```

```
# Leverne test is used to verify Homoscedasticity. It tests if the variance
of two samples are # #equal. Levene's test is an inferential statistic used
to assess the equality of variances for a #variable calculated for two or
more groups.[1] Some common statistical procedures assume that #variances of
the populations from which different samples are drawn are equal. Levene's
test #assesses this assumption.
```

```
bank_int$y=bank$y
library(car)

## Warning: package 'car' was built under R version 3.5.2

## Loading required package: carData

## Warning: package 'carData' was built under R version 3.5.2

leveneTest(age ~ y, data=bank_int)

## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  689.32 < 2.2e-16 ***
##       41186
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#leveneTest() produces a two-sided test
leveneTest(duration ~ y, data=bank_int)

## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1    3130 < 2.2e-16 ***
##       41186
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

leveneTest(campaign ~ y, data=bank_int)

## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  127.71 < 2.2e-16 ***
##       41186
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

leveneTest(pdays ~ y, data=bank_int)

## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  4861.2 < 2.2e-16 ***
##       41186
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

leveneTest(previous ~ y, data=bank_int)

## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  2304.3 < 2.2e-16 ***
##       41186
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

leveneTest(emp.var.rate ~ y, data=bank_int)

## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  26.155 3.165e-07 ***
##       41186
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

leveneTest(cons.conf.idx ~ y, data=bank_int)

## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  1048.5 < 2.2e-16 ***
##       41186
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

leveneTest(cons.price.idx ~ y, data=bank_int)

## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  237.42 < 2.2e-16 ***
##       41186
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

leveneTest(euribor3m ~ y, data=bank_int)

## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  23.775 1.087e-06 ***
##       41186
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

leveneTest(nr.employed ~ y, data=bank_int)

## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1    1038 < 2.2e-16 ***
##       41186
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

leveneTest(bank_housing ~ y, data=bank_int)

## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value  Pr(>F)
```

```
## group     1  5.6802 0.01716 *
##       41186
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
leveneTest(bank_loan ~ y, data=bank_int)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value Pr(>F)
## group     1  0.8215 0.3647
##       41186
```

```r
leveneTest(bank_job ~ y, data=bank_int)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1   233.1 < 2.2e-16 ***
##       41186
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
leveneTest(bank_education ~ y, data=bank_int)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value Pr(>F)
## group     1  0.1642 0.6853
##       41186
```

```r
leveneTest(bank_month ~ y, data=bank_int)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  428.11 < 2.2e-16 ***
##       41186
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
leveneTest(bank_days ~ y, data=bank_int)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  14.283 0.0001575 ***
##       41186
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
leveneTest(bank_contact ~ y, data=bank_int)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  881.71 < 2.2e-16 ***
##       41186
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
leveneTest(bank_marital ~ y, data=bank_int)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)    
## group     1  28.513 9.358e-08 ***
##       41186                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For this Data we don’t need to standardize, therefore we have not applied scale function