Bank_Data_Principal_Component_Analysis.R

deepti

Thu Mar 07 15:41:36 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
```

```r
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 ...
```

```r
bank_pca_data=bank[,c(1,11:14,16:20)]
str(bank_pca_data)
```

```
## 'data.frame':    41188 obs. of  10 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 ...
```

```r
summary(bank_pca_data)
```

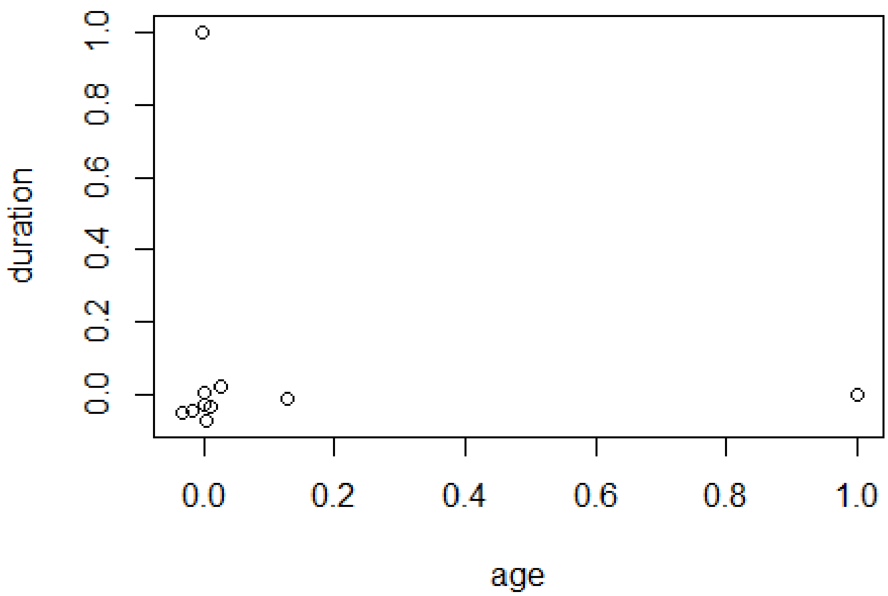
```
##       age           duration         campaign          pdays      
##  Min.   :17.00   Min.   :   0.0   Min.   : 1.000   Min.   :  0.0  
##  1st Qu.:32.00   1st Qu.: 102.0   1st Qu.: 1.000   1st Qu.:999.0  
##  Median :38.00   Median : 180.0   Median : 2.000   Median :999.0  
##  Mean   :40.02   Mean   : 258.3   Mean   : 2.568   Mean   :962.5  
##  3rd Qu.:47.00   3rd Qu.: 319.0   3rd Qu.: 3.000   3rd Qu.:999.0  
##  Max.   :98.00   Max.   :4918.0   Max.   :56.000   Max.   :999.0  
##     previous      emp.var.rate      cons.price.idx  cons.conf.idx 
##  Min.   :0.000   Min.   :-3.40000   Min.   :92.20   Min.   :-50.8  
##  1st Qu.:0.000   1st Qu.:-1.80000   1st Qu.:93.08   1st Qu.:-42.7  
##  Median :0.000   Median : 1.10000   Median :93.75   Median :-41.8  
##  Mean   :0.173   Mean   : 0.08189   Mean   :93.58   Mean   :-40.5  
##  3rd Qu.:0.000   3rd Qu.: 1.40000   3rd Qu.:93.99   3rd Qu.:-36.4  
##  Max.   :7.000   Max.   : 1.40000   Max.   :94.77   Max.   :-26.9  
##    euribor3m      nr.employed  
##  Min.   :0.634   Min.   :4964  
##  1st Qu.:1.344   1st Qu.:5099  
##  Median :4.857   Median :5191  
##  Mean   :3.621   Mean   :5167  
##  3rd Qu.:4.961   3rd Qu.:5228  
##  Max.   :5.045   Max.   :5228
```

```r
cor(bank_pca_data)
```

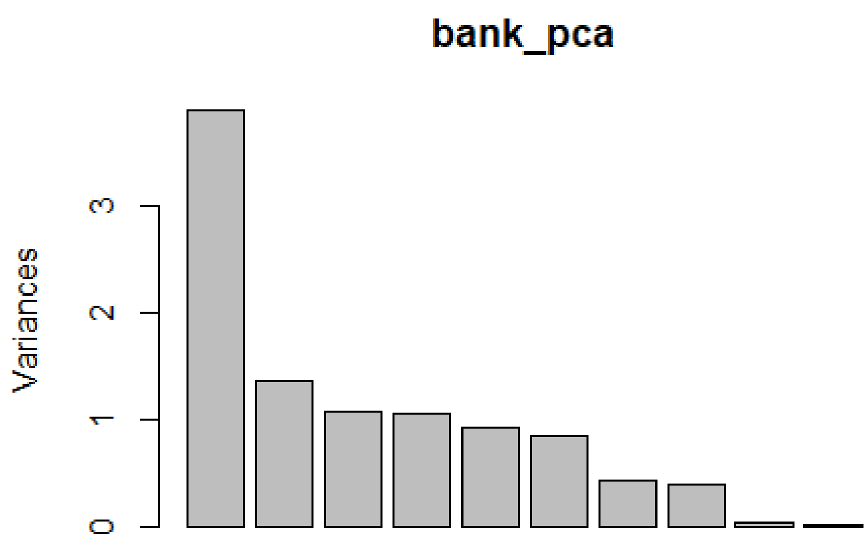
```
##                          age     duration    campaign       pdays
## age             1.0000000000 -0.000865705  0.00459358 -0.03436895
## duration       -0.0008657050  1.000000000 -0.07169923 -0.04757702
## campaign        0.0045935805 -0.071699226  1.00000000  0.05258357
## pdays          -0.0343689512 -0.047577015  0.05258357  1.00000000
## previous        0.0243647409  0.020640351 -0.07914147 -0.58751386
## emp.var.rate   -0.0003706855 -0.027967884  0.15075381  0.27100417
## cons.price.idx  0.0008567150  0.005312268  0.12783591  0.07888911
## cons.conf.idx   0.1293716142 -0.008172873 -0.01373310 -0.09134235
## euribor3m       0.0107674295 -0.032896656  0.13513251  0.29689911
## nr.employed    -0.0177251319 -0.044703223  0.14409489  0.37260474
##                   previous  emp.var.rate cons.price.idx cons.conf.idx
## age             0.02436474 -0.0003706855    0.000856715   0.129371614
## duration        0.02064035 -0.0279678845    0.005312268  -0.008172873
## campaign       -0.07914147  0.1507538056    0.127835912  -0.013733099
## pdays          -0.58751386  0.2710041743    0.078889109  -0.091342354
## previous        1.00000000 -0.4204891094   -0.203129967  -0.050936351
## emp.var.rate   -0.42048911  1.0000000000    0.775334171   0.196041268
## cons.price.idx -0.20312997  0.7753341708    1.000000000   0.058986182
## cons.conf.idx  -0.05093635  0.1960412681    0.058986182   1.000000000
## euribor3m      -0.45449365  0.9722446712    0.688230107   0.277686220
## nr.employed    -0.50133293  0.9069701013    0.522033977   0.100513432
##                  euribor3m nr.employed
## age             0.01076743 -0.01772513
## duration       -0.03289666 -0.04470322
## campaign        0.13513251  0.14409489
## pdays           0.29689911  0.37260474
## previous       -0.45449365 -0.50133293
## emp.var.rate    0.97224467  0.90697010
## cons.price.idx  0.68823011  0.52203398
```

```
## cons.conf.idx   0.27768622  0.10051343
## euribor3m       1.00000000  0.94515443
## nr.employed     0.94515443  1.00000000
```

```
plot(cor(bank_pca_data))
```



```r
# Using prcomp to compute the principal components (eigenvalues and eigenvectors). With s
cale=TRUE, variable means are set to zero, and variances set to one

bank_pca <- prcomp(bank_pca_data,scale=TRUE)
plot(bank_pca)
```




```r
summary(bank_pca)

## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     1.9737 1.1657 1.0381 1.0249 0.96408 0.91751 0.65201
## Proportion of Variance 0.3896 0.1359 0.1078 0.1050 0.09295 0.08418 0.04251
## Cumulative Proportion  0.3896 0.5254 0.6332 0.7382 0.83118 0.91537 0.95788
##                            PC8     PC9    PC10
## Standard deviation     0.62106 0.15776 0.10298
## Proportion of Variance 0.03857 0.00249 0.00106
## Cumulative Proportion  0.99645 0.99894 1.00000
```

```r
View(bank_pca)
```

Name	Type	Value
bank_pca	list [5] (S3: prcomp)	List of length 5
sdev	double [10]	1.974 1.166 1.038 1.025 0.964 0.918 ...
rotation	double [10 x 10]	-1.58e-03 -2.56e-02 1.00e-01 2.28e-01 -3.06e-01 4.88e-01 2.52e-01 8.14e-02 ...
center	double [10]	40.0241 258.2850 2.5676 962.4755 0.1730 0.0819 ...
age	double [1]	40.02406
duration	double [1]	258.285
campaign	double [1]	2.567593
pdays	double [1]	962.4755
previous	double [1]	0.172963
emp.var.rate	double [1]	0.0818855
cons.price.idx	double [1]	93.57566
cons.conf.idx	double [1]	-40.5026
euribor3m	double [1]	3.621291
nr.employed	double [1]	5167.036
scale	double [10]	10.421 259.279 2.770 186.911 0.495 1.571 ...
age	double [1]	10.42125
duration	double [1]	259.2792
campaign	double [1]	2.770014
pdays	double [1]	186.9109
previous	double [1]	0.4949011
emp.var.rate	double [1]	1.57096
cons.price.idx	double [1]	0.57884
cons.conf.idx	double [1]	4.628198
euribor3m	double [1]	1.734447
nr.employed	double [1]	72.25153

```r
# x has new values of data, after
(eigen_bank <- bank_pca$sdev^2)

##  [1] 3.89549575 1.35888318 1.07764506 1.05036054 0.92945344 0.84183259
##  [7] 0.42511495 0.38572154 0.02488887 0.01060409

names(eigen_bank) <- paste("PC",1:10,sep="")
eigen_bank

##        PC1        PC2        PC3        PC4        PC5        PC6
## 3.89549575 1.35888318 1.07764506 1.05036054 0.92945344 0.84183259
##        PC7        PC8        PC9       PC10
## 0.42511495 0.38572154 0.02488887 0.01060409
```

```
sumlambdas <- sum(eigen_bank)
sumlambdas

## [1] 10

propvar <- eigen_bank/sumlambdas
propvar

##         PC1         PC2         PC3         PC4         PC5         PC6
## 0.389549575 0.135888318 0.107764506 0.105036054 0.092945344 0.084183259
##         PC7         PC8         PC9        PC10
## 0.042511495 0.038572154 0.002488887 0.001060409

cumvar_bank <- cumsum(propvar)
cumvar_bank

##       PC1       PC2       PC3       PC4       PC5       PC6       PC7
## 0.3895496 0.5254379 0.6332024 0.7382385 0.8311838 0.9153671 0.9578786
##       PC8       PC9      PC10
## 0.9964507 0.9989396 1.0000000

matlambdas <- rbind(eigen_bank,propvar,cumvar_bank)
rownames(matlambdas) <- c("Eigenvalues","Prop. variance","Cum. prop. variance")
round(matlambdas,4)

##                        PC1    PC2    PC3    PC4    PC5    PC6    PC7
## Eigenvalues         3.8955 1.3589 1.0776 1.0504 0.9295 0.8418 0.4251
## Prop. variance      0.3895 0.1359 0.1078 0.1050 0.0929 0.0842 0.0425
## Cum. prop. variance 0.3895 0.5254 0.6332 0.7382 0.8312 0.9154 0.9579
##                        PC8    PC9   PC10
## Eigenvalues         0.3857 0.0249 0.0106
## Prop. variance      0.0386 0.0025 0.0011
## Cum. prop. variance 0.9965 0.9989 1.0000

summary(bank_pca)

## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     1.9737 1.1657 1.0381 1.0249 0.96408 0.91751 0.65201
## Proportion of Variance 0.3896 0.1359 0.1078 0.1050 0.09295 0.08418 0.04251
## Cumulative Proportion  0.3896 0.5254 0.6332 0.7382 0.83118 0.91537 0.95788
##                            PC8     PC9    PC10
## Standard deviation     0.62106 0.15776 0.10298
## Proportion of Variance 0.03857 0.00249 0.00106
## Cumulative Proportion  0.99645 0.99894 1.00000
```

```r
bank_pca$rotation
```

```
##                        PC1          PC2          PC3          PC4
## age           -0.001577131  0.251900655  0.635282811 -0.253370761
## duration      -0.025564414  0.081409042  0.040921453  0.767885976
## campaign       0.100490892 -0.007934948 -0.324028903 -0.575546022
## pdays          0.227536614 -0.628711981  0.252674001 -0.006719777
## previous      -0.305815059  0.474453454 -0.281754824 -0.021267326
## emp.var.rate   0.488002497  0.163001272 -0.091015114  0.044439336
## cons.price.idx  0.366097505  0.279060437 -0.276172371  0.073400593
## cons.conf.idx  0.101572714  0.427668539  0.510937020 -0.070490333
## euribor3m      0.490377105  0.148132110 -0.002732445  0.036446035
## nr.employed    0.470094939 -0.013534619 -0.029958110  0.027276466
##                       PC5          PC6         PC7          PC8
## age            0.44501766 -0.519053816  0.03130209 -0.017883104
## duration       0.59089472  0.222453386  0.03759640  0.036557378
## campaign       0.61975410  0.411271670  0.00457756  0.015618439
## pdays          0.05446567  0.017271037 -0.22220217  0.660616627
## previous      -0.03416913 -0.146373752  0.19252528  0.735882396
## emp.var.rate  -0.03068934 -0.075546254  0.07039768  0.047342403
## cons.price.idx  0.04059256 -0.249851072 -0.73172135  0.003923396
## cons.conf.idx -0.23898992  0.647573823 -0.17299928  0.120583312
## euribor3m     -0.06312538  0.004006082  0.21662283  0.052635324
## nr.employed   -0.02971432 -0.052239619  0.54214369  0.024394248
##                        PC9          PC10
## age             1.877379e-03  0.0013597361
## duration       -1.291328e-03  0.0013824218
## campaign        1.144203e-05 -0.0092865724
## pdays           2.347902e-03  0.0007657793
## previous       -1.826083e-02  0.0042240094
## emp.var.rate    7.938804e-01  0.2844876703
## cons.price.idx -3.114171e-01  0.0997677428
## cons.conf.idx  -7.079644e-02  0.1216873967
## euribor3m      -6.342829e-02 -0.8237302095
## nr.employed    -5.132191e-01  0.4643979470
```

```r
print(bank_pca)
```

```
## Standard deviations (1, .., p=10):
##  [1] 1.9737010 1.1657114 1.0380968 1.0248710 0.9640817 0.9175144 0.6520084
##  [8] 0.6210648 0.1577621 0.1029762
##
## Rotation (n x k) = (10 x 10):
##                        PC1          PC2          PC3          PC4
## age           -0.001577131  0.251900655  0.635282811 -0.253370761
## duration      -0.025564414  0.081409042  0.040921453  0.767885976
## campaign       0.100490892 -0.007934948 -0.324028903 -0.575546022
## pdays          0.227536614 -0.628711981  0.252674001 -0.006719777
## previous      -0.305815059  0.474453454 -0.281754824 -0.021267326
## emp.var.rate   0.488002497  0.163001272 -0.091015114  0.044439336
## cons.price.idx  0.366097505  0.279060437 -0.276172371  0.073400593
## cons.conf.idx  0.101572714  0.427668539  0.510937020 -0.070490333
## euribor3m      0.490377105  0.148132110 -0.002732445  0.036446035
## nr.employed    0.470094939 -0.013534619 -0.029958110  0.027276466
##                       PC5          PC6         PC7          PC8
## age            0.44501766 -0.519053816  0.03130209 -0.017883104
```

```
## duration        0.59089472  0.222453386  0.03759640  0.036557378
## campaign        0.61975410  0.411271670  0.00457756  0.015618439
## pdays           0.05446567  0.017271037 -0.22220217  0.660616627
## previous       -0.03416913 -0.146373752  0.19252528  0.735882396
## emp.var.rate   -0.03068934 -0.075546254  0.07039768  0.047342403
## cons.price.idx  0.04059256 -0.249851072 -0.73172135  0.003923396
## cons.conf.idx  -0.23898992  0.647573823 -0.17299928  0.120583312
## euribor3m      -0.06312538  0.004006082  0.21662283  0.052635324
## nr.employed    -0.02971432 -0.052239619  0.54214369  0.024394248
##                          PC9          PC10
## age             1.877379e-03  0.0013597361
## duration       -1.291328e-03  0.0013824218
## campaign        1.144203e-05 -0.0092865724
## pdays           2.347902e-03  0.0007657793
## previous       -1.826083e-02  0.0042240094
## emp.var.rate    7.938804e-01  0.2844876703
## cons.price.idx -3.114171e-01  0.0997677428
## cons.conf.idx  -7.079644e-02  0.1216873967
## euribor3m      -6.342829e-02 -0.8237302095
## nr.employed    -5.132191e-01  0.4643979470
```

```
# Sample scores stored in bank_pca$x
head(bank_pca$x)
```

```
##           PC1       PC2       PC3          PC4        PC5         PC6
## [1,] 1.267965 0.8903042 1.4879965  0.005825065  0.1030323 -0.64157299
## [2,] 1.278857 0.8793100 1.5312801 -0.350188998 -0.1095116 -0.78747269
## [3,] 1.274291 0.4200501 0.3242263  0.364113463 -0.7880876  0.27473589
## [4,] 1.281232 0.4690169 0.4952702  0.069053460 -0.8309031  0.06096652
## [5,] 1.263429 0.9047474 1.4952566  0.142059472  0.2078658 -0.60210644
## [6,] 1.275841 0.6046332 0.8074897  0.086685008 -0.5102760 -0.14774523
##             PC7         PC8        PC9        PC10
## [1,] -0.3673217 0.022035364 0.02096648 -0.06246663
## [2,] -0.3805585 0.004527771 0.02170444 -0.06293331
## [3,] -0.4294668 0.049704934 0.01771796 -0.06513231
## [4,] -0.4313310 0.033982154 0.01863195 -0.06514076
## [5,] -0.3606516 0.028521187 0.02073738 -0.06222136
## [6,] -0.4094974 0.032028858 0.01929861 -0.06423778
```

```
# Identifying the scores by their conversion status
banktyp_pca <- cbind(data.frame(bank$y),bank_pca$x)
head(banktyp_pca)
```

```
##   bank.y      PC1       PC2       PC3          PC4        PC5         PC6
## 1     no 1.267965 0.8903042 1.4879965  0.005825065  0.1030323 -0.64157299
## 2     no 1.278857 0.8793100 1.5312801 -0.350188998 -0.1095116 -0.78747269
## 3     no 1.274291 0.4200501 0.3242263  0.364113463 -0.7880876  0.27473589
## 4     no 1.281232 0.4690169 0.4952702  0.069053460 -0.8309031  0.06096652
## 5     no 1.263429 0.9047474 1.4952566  0.142059472  0.2078658 -0.60210644
## 6     no 1.275841 0.6046332 0.8074897  0.086685008 -0.5102760 -0.14774523
##          PC7         PC8        PC9        PC10
## 1 -0.3673217 0.022035364 0.02096648 -0.06246663
## 2 -0.3805585 0.004527771 0.02170444 -0.06293331
## 3 -0.4294668 0.049704934 0.01771796 -0.06513231
## 4 -0.4313310 0.033982154 0.01863195 -0.06514076
```

```
## 5 -0.3606516 0.028521187 0.02073738 -0.06222136
## 6 -0.4094974 0.032028858 0.01929861 -0.06423778

# Means of scores for all the PC's classified by Cumstomer's response towards fixed depos
it
tabmeansPC <- aggregate(banktyp_pca[,2:11],by=list(y=bank$y),mean)
tabmeansPC

##     y        PC1         PC2          PC3        PC4         PC5
## 1  no  0.2383454 -0.08932705 -0.005245612 -0.1036727 -0.07368158
## 2 yes -1.8773812  0.70360454  0.041318235  0.8166011  0.58036947
##           PC6         PC7         PC8           PC9          PC10
## 1 -0.04175336  0.02044733  0.02312943  0.0007812426  0.0003684071
## 2  0.32887966 -0.16105800 -0.18218418 -0.0061536325 -0.0029018412

tabmeansPC <- tabmeansPC[rev(order(tabmeansPC$y)),]
tabmeansPC

##     y        PC1         PC2          PC3        PC4         PC5
## 2 yes -1.8773812  0.70360454  0.041318235  0.8166011  0.58036947
## 1  no  0.2383454 -0.08932705 -0.005245612 -0.1036727 -0.07368158
##           PC6         PC7         PC8           PC9          PC10
## 2  0.32887966 -0.16105800 -0.18218418 -0.0061536325 -0.0029018412
## 1 -0.04175336  0.02044733  0.02312943  0.0007812426  0.0003684071

tabfmeans <- t(tabmeansPC[,-1])
tabfmeans

##                 2             1
## PC1  -1.877381240  0.2383454348
## PC2   0.703604537 -0.0893270508
## PC3   0.041318235 -0.0052456115
## PC4   0.816601100 -0.1036726798
## PC5   0.580369471 -0.0736815789
## PC6   0.328879662 -0.0417533554
## PC7  -0.161057997  0.0204473325
## PC8  -0.182184175  0.0231294346
## PC9  -0.006153632  0.0007812426
## PC10 -0.002901841  0.0003684071

colnames(tabfmeans) <- t(as.vector(tabmeansPC[1]))
tabfmeans

##               yes            no
## PC1  -1.877381240  0.2383454348
## PC2   0.703604537 -0.0893270508
## PC3   0.041318235 -0.0052456115
## PC4   0.816601100 -0.1036726798
## PC5   0.580369471 -0.0736815789
## PC6   0.328879662 -0.0417533554
## PC7  -0.161057997  0.0204473325
## PC8  -0.182184175  0.0231294346
## PC9  -0.006153632  0.0007812426
## PC10 -0.002901841  0.0003684071
```

```r
# Standard deviations of scores for all the PC's classified by Bank$y
tabsdsPC <- aggregate(banktyp_pca[,2:11], by=list(y=bank$y),sd)
tabfsds <- t(tabsdsPC[,-1])
colnames(tabfsds) <- t(as.vector(tabsdsPC[1]))
tabfsds
```

```
##               no       yes
## PC1  1.80836703 2.2022105
## PC2  0.92144630 2.1949059
## PC3  0.96048176 1.5159219
## PC4  0.92523896 1.3526315
## PC5  0.89843006 1.2301639
## PC6  0.86386338 1.2136866
## PC7  0.59178344 0.9930270
## PC8  0.54586330 1.0196774
## PC9  0.14816401 0.2190531
## PC10 0.09892813 0.1305217
```

```r
#t test on all the Principal Components
t.test(PC1~bank$y,data=banktyp_pca)
```

```
## 
##  Welch Two Sample t-test
## 
## data:  PC1 by bank$y
## t = 62.809, df = 5462.2, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  2.049691 2.181763
## sample estimates:
##  mean in group no mean in group yes 
##         0.2383454        -1.8773812
```

#Significant

```r
t.test(PC2~bank$y,data=banktyp_pca)
```

```
## 
##  Welch Two Sample t-test
## 
## data:  PC2 by bank$y
## t = -24.337, df = 4848.6, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.8568048 -0.7290584
## sample estimates:
##  mean in group no mean in group yes 
##       -0.08932705        0.70360454
```

#Significant

```r
t.test(PC3~bank$y,data=banktyp_pca)
```

```
##
##  Welch Two Sample t-test
##
## data:  PC3 by bank$y
## t = -2.041, df = 5122.2, p-value = 0.0413
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.091290135 -0.001837558
## sample estimates:
##  mean in group no mean in group yes
##      -0.005245612       0.041318235
```

#Significant

```r
t.test(PC4~bank$y,data=banktyp_pca)
```

```
##
##  Welch Two Sample t-test
##
## data:  PC4 by bank$y
## t = -45.026, df = 5204.2, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.9603420 -0.8802056
## sample estimates:
##  mean in group no mean in group yes
##        -0.1036727         0.8166011
```

#Significant

```r
t.test(PC5~bank$y,data=banktyp_pca)
```

```
##
##  Welch Two Sample t-test
##
## data:  PC5 by bank$y
## t = -35.049, df = 5285.5, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.6906341 -0.6174680
## sample estimates:
##  mean in group no mean in group yes
##       -0.07368158        0.58036947
```

#Significant

```r
t.test(PC6~bank$y,data=banktyp_pca)
```

```
##
##  Welch Two Sample t-test
##
## data:  PC6 by bank$y
## t = -20.163, df = 5252.2, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.4066686 -0.3345974
## sample estimates:
##  mean in group no mean in group yes
##       -0.04175336        0.32887966
```

#Significant

```r
t.test(PC7~bank$y,data=banktyp_pca)
```

```
##
##  Welch Two Sample t-test
##
## data:  PC7 by bank$y
## t = 12.179, df = 5065.4, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1522887 0.2107220
## sample estimates:
##  mean in group no mean in group yes
##        0.02044733       -0.16105800
```

#Significant

```r
t.test(PC8~bank$y,data=banktyp_pca)
```

```
##
##  Welch Two Sample t-test
##
## data:  PC8 by bank$y
## t = 13.473, df = 4981.9, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1754379 0.2351893
## sample estimates:
##  mean in group no mean in group yes
##        0.02312943       -0.18218418
```

#Significant

```r
t.test(PC9~bank$y,data=banktyp_pca)

##
##  Welch Two Sample t-test
##
## data:  PC9 by bank$y
## t = 2.0965, df = 5191.3, p-value = 0.03609
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.0004500319 0.0134197183
## sample estimates:
##  mean in group no mean in group yes
##      0.0007812426     -0.0061536325
```

#Significant

```r
t.test(PC10~bank$y,data=banktyp_pca)

##
##  Welch Two Sample t-test
##
## data:  PC10 by bank$y
## t = 1.6477, df = 5336.8, p-value = 0.09948
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.0006207109  0.0071612075
## sample estimates:
##  mean in group no mean in group yes
##      0.0003684071     -0.0029018412
```

#Not Significant

```r
# F ratio tests
var.test(PC1~bank$y,data=banktyp_pca)

##
##  F test to compare two variances
##
## data:  PC1 by bank$y
## F = 0.6743, num df = 36547, denom df = 4639, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.6455535 0.7038181
## sample estimates:
## ratio of variances
##          0.6743036

var.test(PC2~bank$y,data=banktyp_pca)

##
##  F test to compare two variances
##
## data:  PC2 by bank$y
## F = 0.17624, num df = 36547, denom df = 4639, p-value < 2.2e-16
```

```
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.1687272 0.1839557
## sample estimates:
## ratio of variances
##          0.1762415
```

```r
var.test(PC3~bank$y,data=banktyp_pca)
```

```
##
##  F test to compare two variances
##
## data:  PC3 by bank$y
## F = 0.40144, num df = 36547, denom df = 4639, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.3843274 0.4190149
## sample estimates:
## ratio of variances
##          0.4014436
```

```r
var.test(PC4~bank$y,data=banktyp_pca)
```

```
##
##  F test to compare two variances
##
## data:  PC4 by bank$y
## F = 0.4679, num df = 36547, denom df = 4639, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.4479459 0.4883753
## sample estimates:
## ratio of variances
##          0.4678954
```

```r
var.test(PC5~bank$y,data=banktyp_pca)
```

```
##
##  F test to compare two variances
##
## data:  PC5 by bank$y
## F = 0.53339, num df = 36547, denom df = 4639, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.5106453 0.5567337
## sample estimates:
## ratio of variances
##          0.5333872
```

```r
var.test(PC6~bank$y,data=banktyp_pca)
```

```
##
##  F test to compare two variances
##
## data:  PC6 by bank$y
## F = 0.50661, num df = 36547, denom df = 4639, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
```

```
## 95 percent confidence interval:
##  0.4850134 0.5287884
## sample estimates:
## ratio of variances
##          0.5066138
```

```r
var.test(PC7~bank$y,data=banktyp_pca)
```

```
##
##  F test to compare two variances
##
## data:  PC7 by bank$y
## F = 0.35514, num df = 36547, denom df = 4639, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.3400010 0.3706879
## sample estimates:
## ratio of variances
##          0.3551432
```

```r
var.test(PC8~bank$y,data=banktyp_pca)
```

```
##
##  F test to compare two variances
##
## data:  PC8 by bank$y
## F = 0.28658, num df = 36547, denom df = 4639, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.2743588 0.2991211
## sample estimates:
## ratio of variances
##          0.2865776
```

```r
var.test(PC9~bank$y,data=banktyp_pca)
```

```
##
##  F test to compare two variances
##
## data:  PC9 by bank$y
## F = 0.4575, num df = 36547, denom df = 4639, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.4379890 0.4775198
## sample estimates:
## ratio of variances
##          0.4574952
```

```r
var.test(PC10~bank$y,data=banktyp_pca)
```

```
##
##  F test to compare two variances
##
## data:  PC10 by bank$y
## F = 0.57448, num df = 36547, denom df = 4639, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
```

```
##  0.5499854 0.5996244
## sample estimates:
## ratio of variances
##          0.5744793
```

#All are Significant

```
# Levene's tests (one-sided)
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

(LTPC1 <- leveneTest(PC1~bank$y,data=banktyp_pca))

## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  248.99 < 2.2e-16 ***
##       41186
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(LTPC1 <- leveneTest(PC1~bank$y,data=banktyp_pca))

## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  248.99 < 2.2e-16 ***
##       41186
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(p_PC1_1sided <- LTPC1[[3]][1]/2)

## [1] 3.145583e-56

(LTPC2 <- leveneTest(PC2~bank$y,data=banktyp_pca))

## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  4799.7 < 2.2e-16 ***
##       41186
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(p_PC2_1sided=LTPC2[[3]][1]/2)

## [1] 0

(LTPC3 <- leveneTest(PC3~bank$y,data=banktyp_pca))

## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  1515.2 < 2.2e-16 ***
##       41186
```

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

```r
(p_PC3_1sided=LTPC3[[3]][1]/2)
```

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

```r
(LTPC4 <- leveneTest(PC4~bank$y,data=banktyp_pca))
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)    
## group     1  1363.3 < 2.2e-16 ***
##       41186                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
(p_PC4_1sided=LTPC4[[3]][1]/2)
```

```
## [1] 6.313288e-294
```

```r
(LTPC5 <- leveneTest(PC5~bank$y,data=banktyp_pca))
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)    
## group     1  917.34 < 2.2e-16 ***
##       41186                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
(p_PC5_1sided=LTPC5[[3]][1]/2)
```

```
## [1] 1.29368e-199
```

```r
(LTPC6 <- leveneTest(PC6~bank$y,data=banktyp_pca))
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)    
## group     1   844.1 < 2.2e-16 ***
##       41186                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
(p_PC6_1sided=LTPC6[[3]][1]/2)
```

```
## [1] 5.025655e-184
```

```r
(LTPC7 <- leveneTest(PC7~bank$y,data=banktyp_pca))
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)    
## group     1  1507.7 < 2.2e-16 ***
##       41186                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
(p_PC7_1sided=LTPC7[[3]][1]/2)
```

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

```r
(LTPC8 <- leveneTest(PC8~bank$y,data=banktyp_pca))
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)    
## group     1  2031.5 < 2.2e-16 ***
##       41186                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(p_PC8_1sided=LTPC8[[3]][1]/2)
```

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

```
(LTPC9 <- leveneTest(PC9~bank$y,data=banktyp_pca))
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)    
## group     1  1123.2 < 2.2e-16 ***
##       41186                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(p_PC9_1sided=LTPC9[[3]][1]/2)
```

```
## [1] 2.814464e-243
```

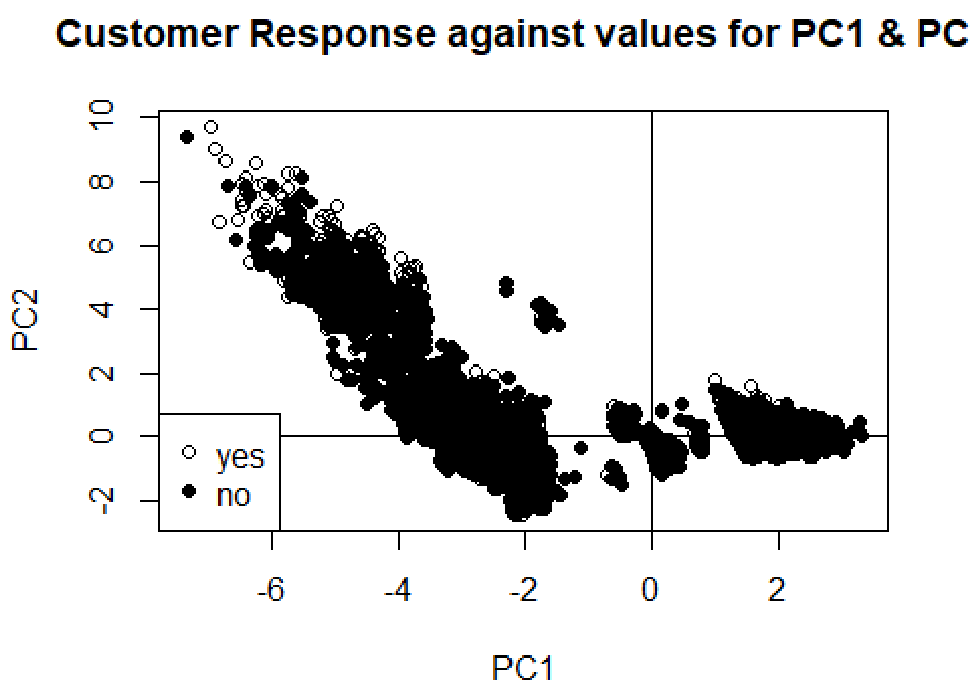
```
(LTPC10 <- leveneTest(PC10~bank$y,data=banktyp_pca))
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)    
## group     1   343.1 < 2.2e-16 ***
##       41186                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

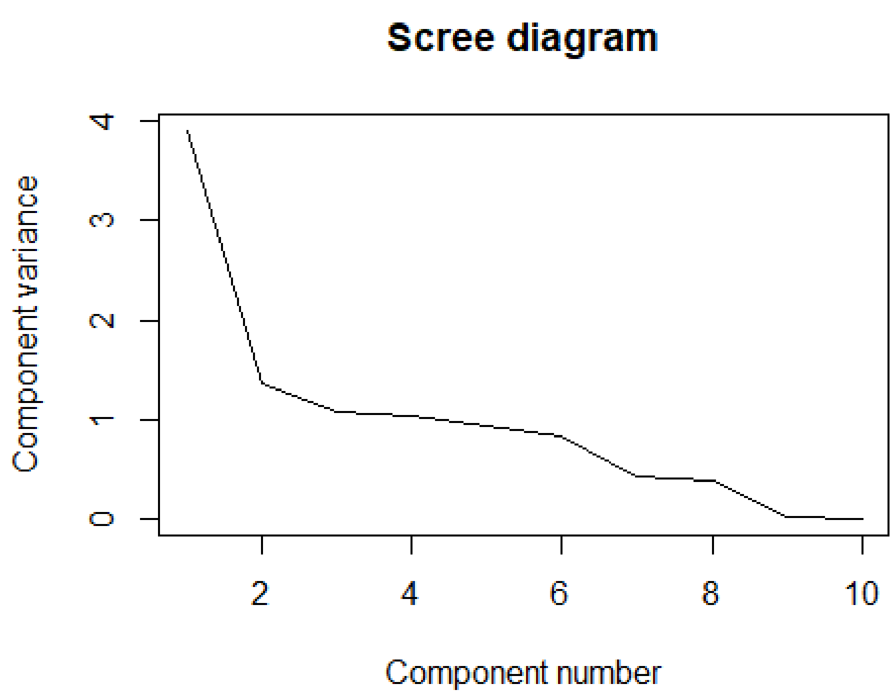
```
(p_PC10_1sided=LTPC10[[3]][1]/2)
```

```
## [1] 1.378121e-76
```

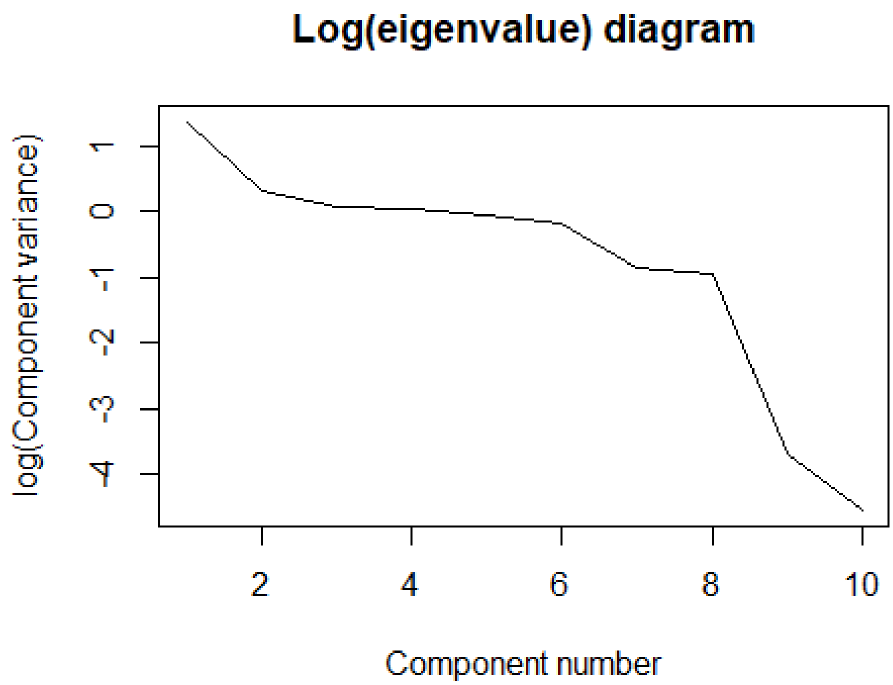
```
# Plotting the scores for the first and second components
plot(banktyp_pca$PC1, banktyp_pca$PC2,pch=ifelse(banktyp_pca$bank.y == "yes",1,16),xlab="
PC1", ylab="PC2", main="Customer Response against values for PC1 & PC2")
abline(h=0)
abline(v=0)
legend("bottomleft", legend=c("yes","no"), pch=c(1,16))
```




```
plot(eigen_bank, xlab = "Component number", ylab = "Component variance", type = "l", main
= "Scree diagram")
```



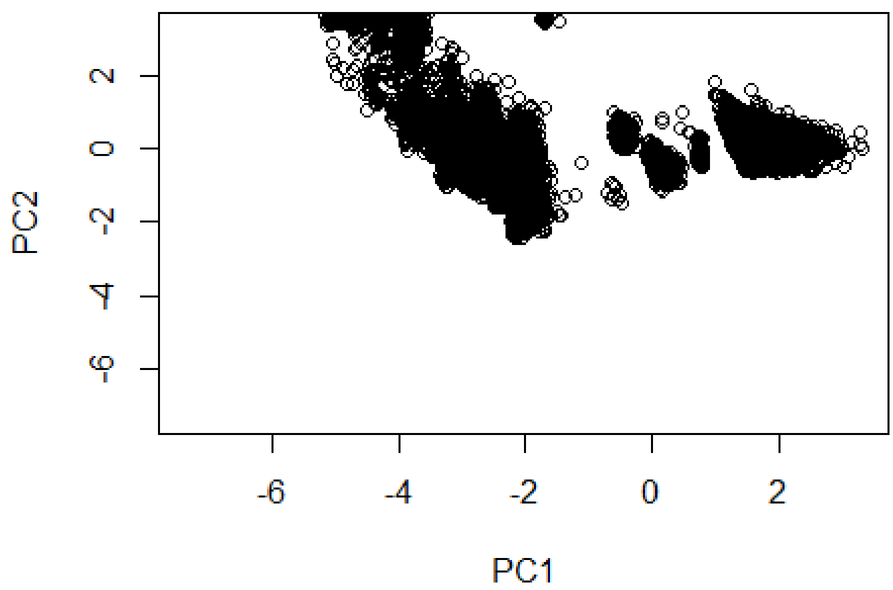

```
#where bending - chossing pC componeet or >.7
#6
plot(log(eigen_bank), xlab = "Component number",ylab = "log(Component variance)", type="l
",main = "Log(eigenvalue) diagram")
```




```
#9 are good
print(summary(bank_pca))

## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     1.9737 1.1657 1.0381 1.0249 0.96408 0.91751 0.65201
## Proportion of Variance 0.3896 0.1359 0.1078 0.1050 0.09295 0.08418 0.04251
## Cumulative Proportion  0.3896 0.5254 0.6332 0.7382 0.83118 0.91537 0.95788
##                            PC8     PC9    PC10
## Standard deviation     0.62106 0.15776 0.10298
## Proportion of Variance 0.03857 0.00249 0.00106
## Cumulative Proportion  0.99645 0.99894 1.00000

#View(bank_pca)
diag(cov(bank_pca$x))

##        PC1        PC2        PC3        PC4        PC5        PC6
## 3.89549575 1.35888318 1.07764506 1.05036054 0.92945344 0.84183259
##        PC7        PC8        PC9       PC10
## 0.42511495 0.38572154 0.02488887 0.01060409

xlim <- range(bank_pca$x[,1])
head(bank_pca$x[,1])

## [1] 1.267965 1.278857 1.274291 1.281232 1.263429 1.275841

head(bank_pca$x)

##           PC1       PC2       PC3          PC4        PC5         PC6
## [1,] 1.267965 0.8903042 1.4879965  0.005825065  0.1030323 -0.64157299
## [2,] 1.278857 0.8793100 1.5312801 -0.350188998 -0.1095116 -0.78747269
## [3,] 1.274291 0.4200501 0.3242263  0.364113463 -0.7880876  0.27473589
## [4,] 1.281232 0.4690169 0.4952702  0.069053460 -0.8309031  0.06096652
## [5,] 1.263429 0.9047474 1.4952566  0.142059472  0.2078658 -0.60210644
```

```
## [6,] 1.275841 0.6046332 0.8074897  0.086685008 -0.5102760 -0.14774523
##             PC7         PC8        PC9        PC10
## [1,] -0.3673217 0.022035364 0.02096648 -0.06246663
## [2,] -0.3805585 0.004527771 0.02170444 -0.06293331
## [3,] -0.4294668 0.049704934 0.01771796 -0.06513231
## [4,] -0.4313310 0.033982154 0.01863195 -0.06514076
## [5,] -0.3606516 0.028521187 0.02073738 -0.06222136
## [6,] -0.4094974 0.032028858 0.01929861 -0.06423778
```

```r
plot(bank_pca$x,xlim=xlim,ylim=xlim)
```




```r
bank_pca$rotation[,1]
```

```
##          age      duration      campaign         pdays      previous
##  -0.001577131  -0.025564414   0.100490892   0.227536614  -0.305815059
##  emp.var.rate cons.price.idx cons.conf.idx     euribor3m   nr.employed
##   0.488002497   0.366097505   0.101572714   0.490377105   0.470094939
```

First Component (PC1)could be named as SocioEconomic as the Social and Economic attributes are relev ant here

Second Component (PC2) could be names as Past_Contact (Contact while previous campaign)

PC3 could be named as Age

PC4 could be named as Current_Contact

```
bank_pca$rotation
```

```
##                          PC1          PC2          PC3          PC4
## age             -0.001577131  0.251900655  0.635282811 -0.253370761
## duration        -0.025564414  0.081409042  0.040921453  0.767885976
## campaign         0.100490892 -0.007934948 -0.324028903 -0.575546022
## pdays            0.227536614 -0.628711981  0.252674001 -0.006719777
## previous        -0.305815059  0.474453454 -0.281754824 -0.021267326
## emp.var.rate     0.488002497  0.163001272 -0.091015114  0.044439336
## cons.price.idx   0.366097505  0.279060437 -0.276172371  0.073400593
## cons.conf.idx    0.101572714  0.427668539  0.510937020 -0.070490333
## euribor3m        0.490377105  0.148132110 -0.002732445  0.036446035
## nr.employed      0.470094939 -0.013534619 -0.029958110  0.027276466
##                         PC5          PC6         PC7          PC8
## age              0.44501766 -0.519053816  0.03130209 -0.017883104
## duration         0.59089472  0.222453386  0.03759640  0.036557378
## campaign         0.61975410  0.411271670  0.00457756  0.015618439
## pdays            0.05446567  0.017271037 -0.22220217  0.660616627
## previous        -0.03416913 -0.146373752  0.19252528  0.735882396
## emp.var.rate    -0.03068934 -0.075546254  0.07039768  0.047342403
## cons.price.idx   0.04059256 -0.249851072 -0.73172135  0.003923396
## cons.conf.idx   -0.23898992  0.647573823 -0.17299928  0.120583312
## euribor3m       -0.06312538  0.004006082  0.21662283  0.052635324
## nr.employed     -0.02971432 -0.052239619  0.54214369  0.024394248
##                           PC9          PC10
## age             1.877379e-03  0.0013597361
## duration       -1.291328e-03  0.0013824218
## campaign        1.144203e-05 -0.0092865724
## pdays           2.347902e-03  0.0007657793
## previous       -1.826083e-02  0.0042240094
## emp.var.rate    7.938804e-01  0.2844876703
## cons.price.idx -3.114171e-01  0.0997677428
## cons.conf.idx  -7.079644e-02  0.1216873967
## euribor3m      -6.342829e-02 -0.8237302095
## nr.employed    -5.132191e-01  0.4643979470
```

```
#get the original value of the data based on PCA
#center <- bank_pca$center
#scale <- bank_pca$scale
#new_bank <- as.matrix(bank_pca_data)
#head(new_bank)
#drop(scale(new_bank,center=center, scale=scale)%*%bank_pca$rotation[,1])
#drop(new_bank%*%bank_pca$rotation[,1])
#predict(bank_pca)[,1]
#scale it back up
#The aboved two gives us the same thing. predict is a good function to know.
```