Factor_Analysis

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

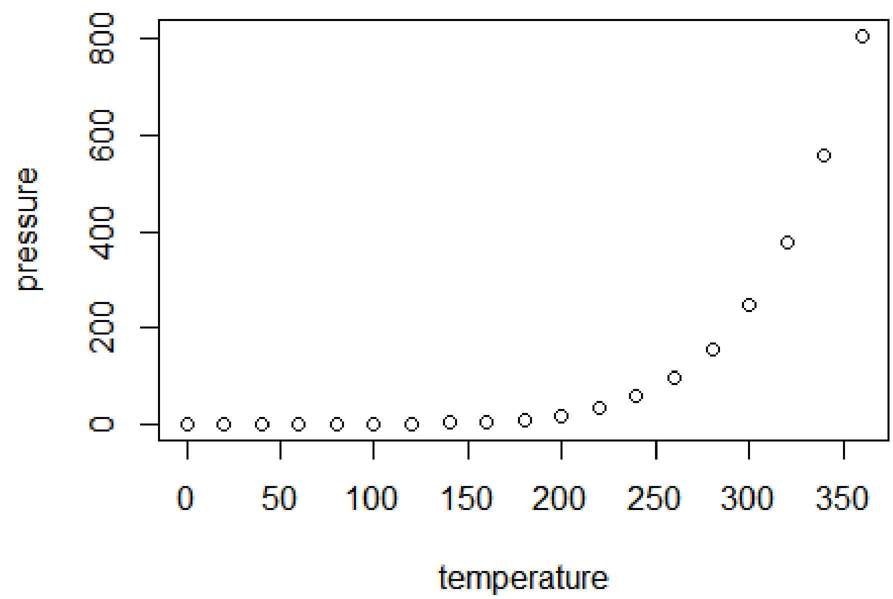
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
library(readxl)
library(data.table)
library(tidyverse)

## -- Attaching packages ------------------------------------------- tidyvers
e 1.3.0 --

## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.4
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ---------------------------------------------- tidyverse_conf
licts() --
## x dplyr::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::transpose() masks data.table::transpose()

library(gridExtra)

##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

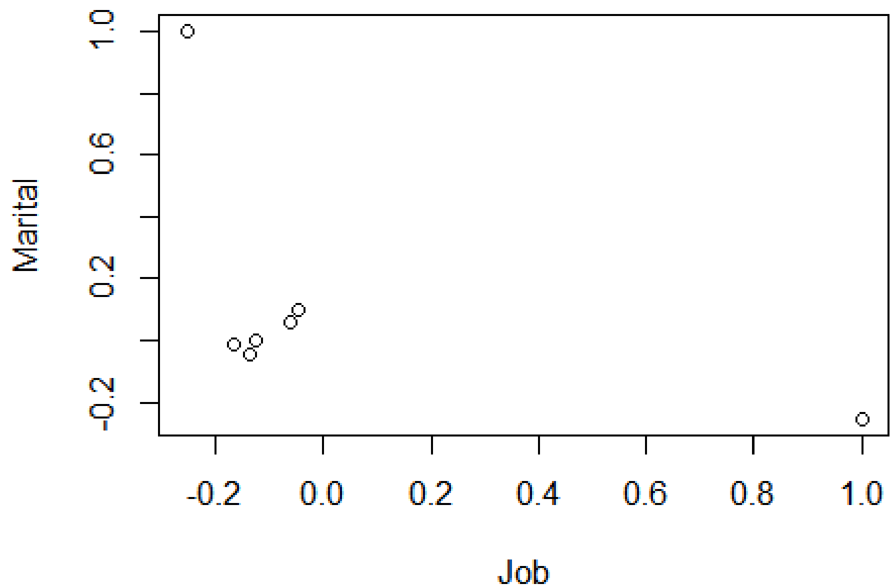
```
library(rmarkdown)
library(tinytex)
library(latexpdf)
library(latex2exp)
library(knitr)
```

```
bank=read.csv("C:/Users/Shamali/Desktop/RutgersSpring/multivariat/project/New
folder/bank.csv",row.names=1,fill=TRUE)
attach(bank)
```

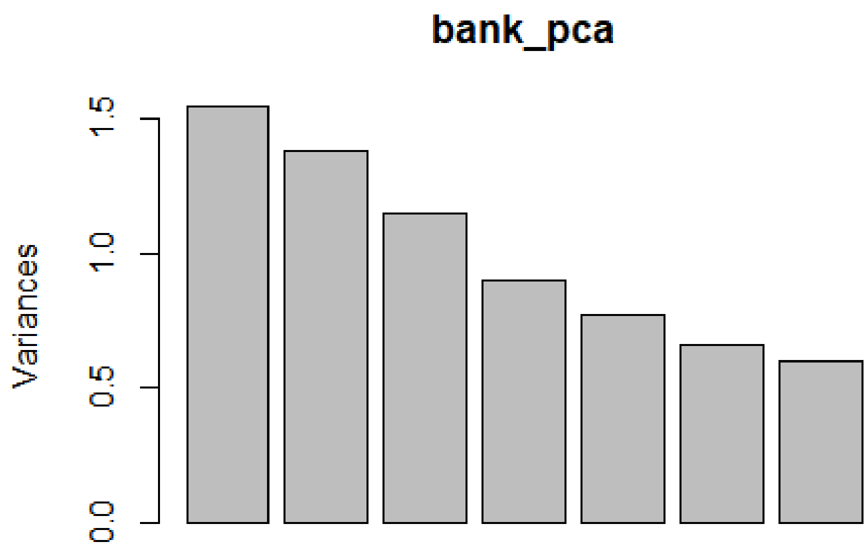
```
corrm.bank <- cor(bank)
corrm.bank
```

```
##                   Job     Marital   Education    Housing        Loan
## Job        1.00000000 -0.25351544 -0.04588504 -0.1230462 -0.16683538
## Marital   -0.25351544  1.00000000  0.09700388  0.0000000 -0.01455710
## Education -0.04588504  0.09700388  1.00000000 -0.1462388  0.14262166
## Housing   -0.12304619  0.00000000 -0.14623885  1.0000000  0.14484136
## Loan      -0.16683538 -0.01455710  0.14262166  0.1448414  1.00000000
## Default   -0.13476487 -0.04307305 -0.24651691  0.1428571 -0.07586929
## Deposit   -0.06152309  0.06030227  0.08774331 -0.2800000 -0.04828045
##               Default     Deposit
## Job       -0.13476487 -0.06152309
## Marital   -0.04307305  0.06030227
## Education -0.24651691  0.08774331
## Housing    0.14285714 -0.28000000
## Loan      -0.07586929 -0.04828045
## Default    1.00000000 -0.14285714
## Deposit   -0.14285714  1.00000000
```

```
#high negative number is good #they will get grouped together for sure
plot(corrm.bank)
```




```
bank_pca <- prcomp(bank, scale=TRUE)#pca
summary(bank_pca)#see sCREe diagram

## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5     PC6     PC7
## Standard deviation     1.2431 1.1756 1.0727 0.9475 0.8784 0.81052 0.77178
## Proportion of Variance 0.2208 0.1974 0.1644 0.1283 0.1102 0.09385 0.08509
## Cumulative Proportion  0.2208 0.4182 0.5826 0.7108 0.8211 0.91491 1.00000

plot(bank_pca)
```




```
# A table containing eigenvalues and %'s accounted, follows. Eigenvalues are
the sdev^2
(eigen_bank <- round(bank_pca$sdev^2,2))

## [1] 1.55 1.38 1.15 0.90 0.77 0.66 0.60

names(eigen_bank) <- paste("PC",1:7,sep="")
eigen_bank

##  PC1  PC2  PC3  PC4  PC5  PC6  PC7
## 1.55 1.38 1.15 0.90 0.77 0.66 0.60

sumlambdas <- sum(eigen_bank)
sumlambdas

## [1] 7.01

#cumvar_bank <- cumsum(propvar)

propvar <- round(eigen_bank/sumlambdas,2)
propvar

##  PC1  PC2  PC3  PC4  PC5  PC6  PC7
## 0.22 0.20 0.16 0.13 0.11 0.09 0.09

cumvar_bank <- cumsum(propvar)
cumvar_bank
```

```
##  PC1  PC2  PC3  PC4  PC5  PC6  PC7
## 0.22 0.42 0.58 0.71 0.82 0.91 1.00
```

```
matlambdas <- rbind(eigen_bank,propvar,cumvar_bank)
matlambdas
```

```
##              PC1  PC2  PC3  PC4  PC5  PC6  PC7
## eigen_bank  1.55 1.38 1.15 0.90 0.77 0.66 0.60
## propvar     0.22 0.20 0.16 0.13 0.11 0.09 0.09
## cumvar_bank 0.22 0.42 0.58 0.71 0.82 0.91 1.00
```

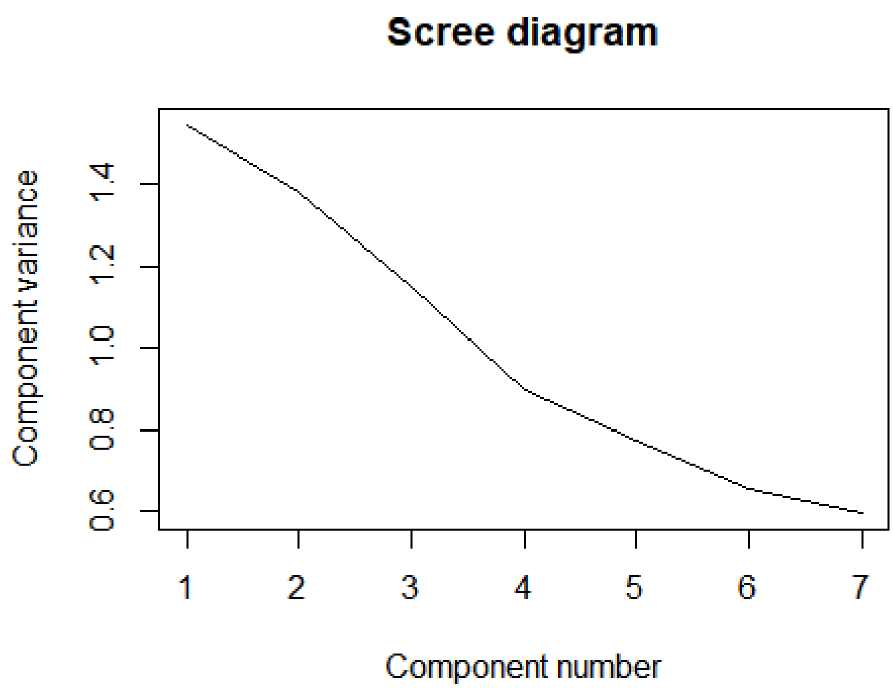
```
rownames(matlambdas) <- c("Eigenvalues","Prop. variance","Cum. prop. variance
")
rownames(matlambdas)
```

```
## [1] "Eigenvalues"         "Prop. variance"      "Cum. prop. variance"
```

```
eigvec.bank<- bank_pca$rotation
print(bank_pca)#pc1=0.17*marital
```

```
## Standard deviations (1, .., p=7):
## [1] 1.2430941 1.1755738 1.0727453 0.9474932 0.8784315 0.8105174 0.7717753
##
## Rotation (n x k) = (7 x 7):
##                  PC1         PC2        PC3        PC4         PC5        
PC6
## Job       -0.10131024  0.63291628 -0.3217743  0.2474470 -0.08269113 -0.199
60210
## Marital   -0.12733172 -0.48134638  0.4202993  0.5673798 -0.15479643 -0.343
60498
## Education -0.45327940 -0.29394405 -0.3201907  0.1792481  0.60533294  0.407
71561
## Housing    0.53163682 -0.25125215 -0.2401007  0.1569638 -0.38616782  0.580
03709
## Loan       0.03000185 -0.46638562 -0.5284382 -0.4493333 -0.08893867 -0.495
61041
## Default    0.50157425  0.02350968  0.3866175 -0.2882092  0.56791200 -0.054
20378
## Deposit   -0.48264490 -0.01111983  0.3618278 -0.5245166 -0.35101649  0.301
41640
##                 PC7
## Job       -0.6145776
## Marital   -0.3339081
## Education -0.2020463
## Housing   -0.2939048
## Loan      -0.2165578
## Default   -0.4357417
## Deposit   -0.3831825
```

```r
# Taking the first four PCs to generate linear combinations for all the varia
bles with four factors
(eigen_bank <- bank_pca$sdev^2)

## [1] 1.5452828 1.3819738 1.1507825 0.8977433 0.7716419 0.6569384 0.5956371

names(eigen_bank) <- paste("PC",1:7,sep="")
eigen_bank

##       PC1       PC2       PC3       PC4       PC5       PC6       PC7
## 1.5452828 1.3819738 1.1507825 0.8977433 0.7716419 0.6569384 0.5956371

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




```r
#FROM SCREE DIAGRAM WE UNDERSTOOD THAT WE NEED YTO MAKE 4 FACTORS
pcafactors.bank <- eigvec.bank[,1:4]#according to ske
pcafactors.bank

##                   PC1         PC2        PC3        PC4
## Job       -0.10131024  0.63291628 -0.3217743  0.2474470
## Marital   -0.12733172 -0.48134638  0.4202993  0.5673798
## Education -0.45327940 -0.29394405 -0.3201907  0.1792481
## Housing    0.53163682 -0.25125215 -0.2401007  0.1569638
## Loan       0.03000185 -0.46638562 -0.5284382 -0.4493333
## Default    0.50157425  0.02350968  0.3866175 -0.2882092
## Deposit   -0.48264490 -0.01111983  0.3618278 -0.5245166
```

```
# Multiplying each column of the eigenvector's matrix by the square-root of t
he corresponding eigenvalue in order to get the factor loadings
unrot.fact.bank <- sweep(pcafactors.bank,MARGIN=2,bank_pca$sdev[1:4],`*`)
unrot.fact.bank #factors education housing and default can come together in p
c1 as they have high correlation

##                   PC1         PC2        PC3        PC4
## Job       -0.12593815  0.74403982 -0.3451819  0.2344543
## Marital   -0.15828531 -0.56585820  0.4508741  0.5375885
## Education -0.56346893 -0.34555294 -0.3434831  0.1698364
## Housing    0.66087458 -0.29536545 -0.2575669  0.1487221
## Loan       0.03729512 -0.54827073 -0.5668796 -0.4257403
## Default    0.62350397  0.02763736  0.4147421 -0.2730763
## Deposit   -0.59997301 -0.01307218  0.3881491 -0.4969759

# Computing communalities is the common variance
communalities.bank<- rowSums(unrot.fact.bank^2)#square of that factor
communalities.bank#1-this will be its unique variance #what the common varian
ce is

##       Job   Marital Education   Housing      Loan   Default   Deposit
## 0.7435750 0.8375386 0.5837291 0.6124549 0.8045989 0.6361027 0.7577833

# Performing the varimax rotation. The default in the varimax function is nor
m=TRUE thus, Kaiser normalization is carried out
rot.fact.bank <- varimax(unrot.fact.bank)
#View(unrot.fact.bank)
rot.fact.bank

## $loadings
##
## Loadings:
##           PC1    PC2    PC3    PC4
## Job        0.106  0.256  0.493 -0.651
## Marital           0.179  0.182  0.877
## Education         0.714 -0.203  0.154
## Housing    0.692 -0.222 -0.275
## Loan       0.118  0.194 -0.865
## Default          -0.787
## Deposit   -0.862
##
##                  PC1   PC2   PC3   PC4
## SS loadings    1.267 1.314 1.151 1.244
## Proportion Var 0.181 0.188 0.164 0.178
## Cumulative Var 0.181 0.369 0.533 0.711
##
## $rotmat
##            [,1]       [,2]       [,3]        [,4]
## [1,]  0.6921501 -0.7122493 -0.1041086 -0.05282619
## [2,] -0.1237056 -0.1594531  0.6432231 -0.73860387
```

```
## [3,] -0.4537025 -0.5447203  0.4244804  0.56324976
## [4,]  0.5475210  0.4129836  0.6286826  0.36663802
```

```
# The print method of varimax omits loadings less than abs(0.1). In order to display all the loadings, it is necessary to ask explicitly the contents of the object $loadings
fact.load.bank <- rot.fact.bank$loadings[1:7,1:4]
fact.load.bank
```

```
##                   PC1         PC2         PC3         PC4
## Job        0.10576852  0.25591332  0.49256927 -0.65136162
## Marital    0.05022098  0.17938145  0.18186555  0.87736177
## Education -0.09843010  0.71367168 -0.20263431  0.15379434
## Housing    0.69225000 -0.22188880 -0.27462173  0.09269926
## Loan       0.11773095  0.19382705 -0.86482790 -0.07240262
## Default    0.09045491 -0.78719159 -0.04276357  0.08013286
## Deposit   -0.86186321  0.01273915 -0.09362438  0.07776408
```

```
# Computing the rotated factor scores for the 30 European Countries. Notice that signs are reversed for factors F2 (PC2), F3 (PC3) and F4 (PC4)
scale.bank <- scale(bank)
scale.bank
```

```
##           Job   Marital  Education    Housing       Loan    Default    Deposit
## 1   1.3500554  1.193924  1.1871064 -0.9899495 -0.5257473 -0.1414214  0.9899495
## 2   1.3500554 -0.298481 -0.2605843 -0.9899495  1.8640133 -0.1414214  0.9899495
## 3  -0.9338729 -0.298481 -1.7082750 -0.9899495 -0.5257473 -0.1414214 -0.9899495
## 4  -0.1725635 -0.298481 -3.1559657 -0.9899495 -0.5257473 -0.1414214  0.9899495
## 5   0.8425158 -0.298481 -0.2605843 -0.9899495 -0.5257473 -0.1414214  0.9899495
## 6   1.6038252 -0.298481 -0.2605843  0.9899495  1.8640133 -0.1414214 -0.9899495
## 7   1.6038252 -0.298481 -0.2605843 -0.9899495 -0.5257473 -0.1414214  0.9899495
## 8  -0.1725635  1.193924  1.1871064  0.9899495 -0.5257473 -0.1414214 -0.9899495
## 9  -1.1876427 -0.298481 -0.2605843  0.9899495 -0.5257473 -0.1414214 -0.9899495
## 10 -0.6801031 -1.790886 -0.2605843  0.9899495 -0.5257473 -0.1414214  0.9899495
## 11  0.8425158 -1.790886 -0.2605843  0.9899495 -0.5257473 -0.1414214 -0.9899495
## 12  1.3500554 -1.790886 -0.2605843  0.9899495 -0.5257473 -0.1414214 -0.9899495
## 13 -1.1876427 -0.298481 -0.2605843  0.9899495  1.8640133 -0.1414214  0.9899495
```

```
## 14  1.3500554 -0.298481 -0.2605843  0.9899495 -0.5257473 -0.1414214 -0.989
9495
## 15  1.3500554  1.193924 -0.2605843  0.9899495 -0.5257473 -0.1414214 -0.989
9495
## 16 -1.1876427 -0.298481 -0.2605843  0.9899495  1.8640133 -0.1414214 -0.989
9495
## 17 -0.9338729 -0.298481 -0.2605843  0.9899495 -0.5257473 -0.1414214 -0.989
9495
## 18 -0.1725635 -0.298481  1.1871064 -0.9899495 -0.5257473 -0.1414214  0.989
9495
## 19  1.3500554 -1.790886 -0.2605843  0.9899495 -0.5257473 -0.1414214 -0.989
9495
## 20  1.0962856  1.193924 -3.1559657 -0.9899495 -0.5257473 -0.1414214 -0.989
9495
## 21  1.3500554 -0.298481  1.1871064 -0.9899495 -0.5257473 -0.1414214  0.989
9495
## 22 -1.1876427 -0.298481  1.1871064  0.9899495  1.8640133 -0.1414214 -0.989
9495
## 23 -0.1725635  1.193924  1.1871064 -0.9899495 -0.5257473 -0.1414214 -0.989
9495
## 24 -0.9338729 -0.298481 -1.7082750  0.9899495 -0.5257473  6.9296465 -0.989
9495
## 25  0.8425158  1.193924 -0.2605843  0.9899495 -0.5257473 -0.1414214 -0.989
9495
## 26  0.8425158  1.193924 -0.2605843  0.9899495 -0.5257473 -0.1414214  0.989
9495
## 27 -0.9338729  1.193924 -0.2605843  0.9899495 -0.5257473 -0.1414214  0.989
9495
## 28  0.8425158 -1.790886 -0.2605843 -0.9899495 -0.5257473 -0.1414214  0.989
9495
## 29 -0.1725635 -0.298481  1.1871064 -0.9899495 -0.5257473 -0.1414214  0.989
9495
## 30  0.3349762 -0.298481 -0.2605843  0.9899495 -0.5257473 -0.1414214  0.989
9495
## 31 -0.1725635 -0.298481  1.1871064 -0.9899495 -0.5257473 -0.1414214  0.989
9495
## 32 -0.9338729  1.193924 -0.2605843  0.9899495  1.8640133 -0.1414214  0.989
9495
## 33 -1.1876427  1.193924  1.1871064 -0.9899495  1.8640133 -0.1414214  0.989
9495
## 34 -0.1725635  1.193924  1.1871064 -0.9899495 -0.5257473 -0.1414214 -0.989
9495
## 35 -0.6801031 -0.298481  1.1871064 -0.9899495  1.8640133 -0.1414214 -0.989
9495
## 36 -0.1725635 -0.298481  1.1871064 -0.9899495 -0.5257473 -0.1414214  0.989
9495
## 37 -0.9338729 -0.298481 -0.2605843  0.9899495 -0.5257473 -0.1414214 -0.989
9495
## 38 -0.9338729  1.193924 -0.2605843 -0.9899495 -0.5257473 -0.1414214 -0.989
9495
```

```
## 39  1.3500554 -1.790886 -0.2605843 -0.9899495  1.8640133 -0.1414214 -0.989
9495
## 40 -0.9338729 -0.298481 -0.2605843 -0.9899495 -0.5257473 -0.1414214  0.989
9495
## 41  1.3500554 -1.790886 -0.2605843 -0.9899495 -0.5257473 -0.1414214 -0.989
9495
## 42 -0.1725635 -0.298481  1.1871064 -0.9899495 -0.5257473 -0.1414214 -0.989
9495
## 43 -1.1876427  1.193924 -0.2605843  0.9899495 -0.5257473 -0.1414214 -0.989
9495
## 44 -1.1876427  1.193924 -0.2605843 -0.9899495 -0.5257473 -0.1414214  0.989
9495
## 45 -0.1725635  1.193924  1.1871064  0.9899495  1.8640133 -0.1414214 -0.989
9495
## 46 -1.1876427 -0.298481 -0.2605843  0.9899495  1.8640133 -0.1414214  0.989
9495
## 47 -1.1876427  1.193924 -0.2605843 -0.9899495 -0.5257473 -0.1414214  0.989
9495
## 48  0.3349762 -0.298481  1.1871064 -0.9899495 -0.5257473 -0.1414214  0.989
9495
## 49  0.8425158  1.193924 -0.2605843  0.9899495 -0.5257473 -0.1414214  0.989
9495
## 50 -0.9338729 -0.298481 -0.2605843  0.9899495 -0.5257473 -0.1414214  0.989
9495
## attr(,"scaled:center")
##       Job   Marital Education   Housing      Loan   Default   Deposit
##      5.68      2.20      3.18      1.50      1.22      1.02      1.50
## attr(,"scaled:scale")
##       Job   Marital Education   Housing      Loan   Default   Deposit
## 3.9405791 0.6700594 0.6907553 0.5050763 0.4184520 0.1414214 0.5050763
```

```
as.matrix(scale.bank)%*%fact.load.bank%*%solve(t(fact.load.bank)%*%fact.load.
bank)
```

```
##            PC1         PC2          PC3          PC4
## 1  -0.84741371  1.14629658  1.134009292  0.409334350
## 2  -0.99582846  0.39053058 -0.937804457 -1.240723832
## 3  -0.01564183 -0.96506567  0.522446459  0.025185806
## 4  -1.35506902 -1.82782204  0.672124472 -0.453552012
## 5  -1.14583689 -0.01633397  0.691444828 -0.622574900
## 6   1.49567746  0.55943844 -0.843045855 -1.249377787
## 7  -1.02701982  0.16900309  0.989612416 -0.977763145
## 8   1.36685239  0.88275128  0.533043522  1.229452966
## 9   0.98921785 -0.40343732 -0.108299334  0.434335879
## 10 -0.55821744 -0.80267324 -0.680033018 -0.890145832
## 11  1.11225624 -0.16609166  0.253909947 -1.629253467
## 12  1.19146762 -0.04253361  0.452688338 -1.866045630
## 13 -0.37282466 -0.38603825 -2.273934800  0.081710256
## 14  1.38527472  0.21435290  0.885592625 -0.749624937
## 15  1.57908183  0.47123942  1.318496911  0.366795757
```

```
## 16  1.06001490 -0.12013081 -1.936327011  0.052979111
## 17  1.02882353 -0.34165830 -0.008910138  0.315939797
## 18 -1.27885494  0.51873593  0.104769830  0.003290145
## 19  1.19146762 -0.04253361  0.452688338 -1.866045630
## 20  0.46960608 -0.99613297  1.939582527  0.042157128
## 21 -1.04122082  0.88941006  0.701105006 -0.707086344
## 22  1.08541959  0.66205518 -2.125445225  0.205259830
## 23  0.34779172  1.04152989  0.875281906  1.090979694
## 24  0.62682056 -5.45495943 -0.296336421  0.555292421
## 25  1.49987045  0.34768138  1.119718520  0.603587920
## 26  0.06703089  0.08177393  0.782110731  0.632319066
## 27 -0.21020892 -0.35067922  0.086386359  1.461091637
## 28 -1.33964399 -0.27322049  0.258540541 -1.738995594
## 29 -1.27885494  0.51873593  0.104769830  0.003290145
## 30 -0.20598759 -0.29867063  0.150428052 -0.247309465
## 31 -1.27885494  0.51873593  0.104769830  0.003290145
## 32 -0.13941187 -0.06737271 -1.741641317  1.079734868
## 33 -1.17267353  0.81181287 -1.687910343  1.211938397
## 34  0.34779172  1.04152989  0.875281906  1.090979694
## 35  0.14557030  0.94439184 -1.584428449 -0.170005606
## 36 -1.27885494  0.51873593  0.104769830  0.003290145
## 37  1.02882353 -0.34165830 -0.008910138  0.315939797
## 38  0.20356997  0.07400683  0.766232532  1.293887219
## 39  0.24320400  0.39955151 -1.033100954 -2.385875671
## 40 -1.42307670 -0.44878713 -0.004279544  0.206197671
## 41  0.17240694  0.11624500  0.794926722 -2.004518903
## 42  0.15398462  0.78464337  0.442377619 -0.025441000
## 43  1.18302495 -0.14655080  0.324604952  1.550756573
## 44 -1.26887528 -0.25367963  0.329235547  1.441014446
## 45  1.43764945  1.16605779 -1.294984154  0.848096197
## 46 -0.37282466 -0.38603825 -2.273934800  0.081710256
## 47 -1.26887528 -0.25367963  0.329235547  1.441014446
## 48 -1.19964357  0.64229397  0.303548222 -0.233502018
## 49  0.06703089  0.08177393  0.782110731  0.632319066
## 50 -0.40401603 -0.60756574 -0.346517927  0.344670943

#   We get new values

#simple way of doing the whole process
##Since we have 4 columns that we are considering for factor analysis , we ar
e checking
#how will the variance be distributed across 4 factors and if we really need
4 factors
#for our analysis.
library(psych)

## Warning: package 'psych' was built under R version 3.6.3

##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```r
fit.pc <- principal(bank, nfactors=4, rotate="varimax")
fit.pc
```

```
## Principal Components Analysis
## Call: principal(r = bank, nfactors = 4, rotate = "varimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##             RC2   RC1   RC4   RC3   h2   u2 com
## Job        0.26  0.11 -0.65 -0.49 0.74 0.26 2.3
## Marital    0.18  0.05  0.88 -0.18 0.84 0.16 1.2
## Education  0.71 -0.10  0.15  0.20 0.58 0.42 1.3
## Housing   -0.22  0.69  0.09  0.27 0.61 0.39 1.6
## Loan       0.19  0.12 -0.07  0.86 0.80 0.20 1.2
## Default   -0.79  0.09  0.08  0.04 0.64 0.36 1.1
## Deposit    0.01 -0.86  0.08  0.09 0.76 0.24 1.0
##
##                        RC2  RC1  RC4  RC3
## SS loadings           1.31 1.27 1.24 1.15
## Proportion Var        0.19 0.18 0.18 0.16
## Cumulative Var        0.19 0.37 0.55 0.71
## Proportion Explained  0.26 0.25 0.25 0.23
## Cumulative Proportion 0.26 0.52 0.77 1.00
##
## Mean item complexity =  1.4
## Test of the hypothesis that 4 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.13
##  with the empirical chi square  36.84  with prob <  NA
##
## Fit based upon off diagonal values = 0.09
```

```r
round(fit.pc$values, 4)
```

```
## [1] 1.5453 1.3820 1.1508 0.8977 0.7716 0.6569 0.5956
```

```r
fit.pc$loadings
```

```
##
## Loadings:
##           RC2    RC1    RC4    RC3
## Job        0.256  0.106 -0.651 -0.493
## Marital    0.179         0.877 -0.182
## Education  0.714         0.154  0.203
## Housing   -0.222  0.692         0.275
## Loan       0.194  0.118         0.865
## Default   -0.787
## Deposit          -0.862
##
```

```
##                   RC2   RC1   RC4   RC3
## SS loadings    1.314 1.267 1.244 1.151
## Proportion Var 0.188 0.181 0.178 0.164
## Cumulative Var 0.188 0.369 0.546 0.711
```

```
# Loadings with more digits
for (i in c(1,3,2,4)) { print(fit.pc$loadings[[1,i]])}
```

```
## [1] 0.2559133
## [1] -0.6513616
## [1] 0.1057685
## [1] -0.4925693
```

```
# Communalities
fit.pc$communality
```

```
##       Job   Marital Education   Housing      Loan   Default   Deposit
## 0.7435750 0.8375386 0.5837291 0.6124549 0.8045989 0.6361027 0.7577833
```

```
# Rotated factor scores, Notice the columns ordering: RC1, RC3, RC2 and RC4
fit.pc$scores
```

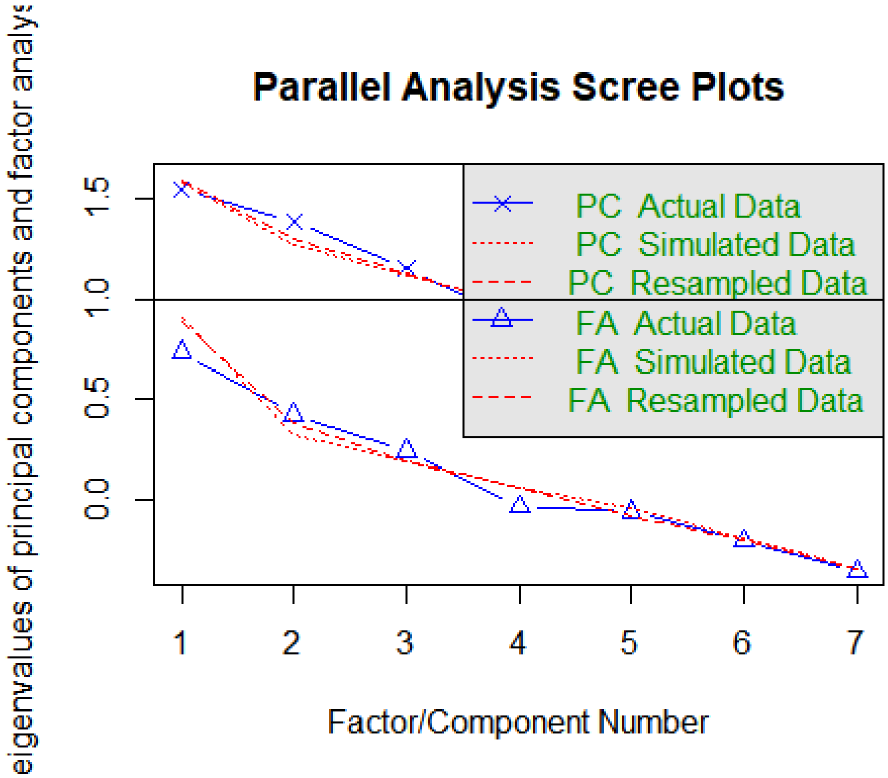
```
##            RC2         RC1          RC4          RC3
## 1   1.14629658 -0.84741371  0.409334350 -1.134009292
## 2   0.39053058 -0.99582846 -1.240723832  0.937804457
## 3  -0.96506567 -0.01564183  0.025185806 -0.522446459
## 4  -1.82782204 -1.35506902 -0.453552012 -0.672124472
## 5  -0.01633397 -1.14583689 -0.622574900 -0.691444828
## 6   0.55943844  1.49567746 -1.249377787  0.843045855
## 7   0.16900309 -1.02701982 -0.977763145 -0.989612416
## 8   0.88275128  1.36685239  1.229452966 -0.533043522
## 9  -0.40343732  0.98921785  0.434335879  0.108299334
## 10 -0.80267324 -0.55821744 -0.890145832  0.680033018
## 11 -0.16609166  1.11225624 -1.629253467 -0.253909947
## 12 -0.04253361  1.19146762 -1.866045630 -0.452688338
## 13 -0.38603825 -0.37282466  0.081710256  2.273934800
## 14  0.21435290  1.38527472 -0.749624937 -0.885592625
## 15  0.47123942  1.57908183  0.366795757 -1.318496911
## 16 -0.12013081  1.06001490  0.052979111  1.936327011
## 17 -0.34165830  1.02882353  0.315939797  0.008910138
## 18  0.51873593 -1.27885494  0.003290145 -0.104769830
## 19 -0.04253361  1.19146762 -1.866045630 -0.452688338
## 20 -0.99613297  0.46960608  0.042157128 -1.939582527
## 21  0.88941006 -1.04122082 -0.707086344 -0.701105006
## 22  0.66205518  1.08541959  0.205259830  2.125445225
## 23  1.04152989  0.34779172  1.090979694 -0.875281906
## 24 -5.45495943  0.62682056  0.555292421  0.296336421
## 25  0.34768138  1.49987045  0.603587920 -1.119718520
## 26  0.08177393  0.06703089  0.632319066 -0.782110731
## 27 -0.35067922 -0.21020892  1.461091637 -0.086386359
## 28 -0.27322049 -1.33964399 -1.738995594 -0.258540541
```

```
## 29  0.51873593 -1.27885494  0.003290145 -0.104769830
## 30 -0.29867063 -0.20598759 -0.247309465 -0.150428052
## 31  0.51873593 -1.27885494  0.003290145 -0.104769830
## 32 -0.06737271 -0.13941187  1.079734868  1.741641317
## 33  0.81181287 -1.17267353  1.211938397  1.687910343
## 34  1.04152989  0.34779172  1.090979694 -0.875281906
## 35  0.94439184  0.14557030 -0.170005606  1.584428449
## 36  0.51873593 -1.27885494  0.003290145 -0.104769830
## 37 -0.34165830  1.02882353  0.315939797  0.008910138
## 38  0.07400683  0.20356997  1.293887219 -0.766232532
## 39  0.39955151  0.24320400 -2.385875671  1.033100954
## 40 -0.44878713 -1.42307670  0.206197671  0.004279544
## 41  0.11624500  0.17240694 -2.004518903 -0.794926722
## 42  0.78464337  0.15398462 -0.025441000 -0.442377619
## 43 -0.14655080  1.18302495  1.550756573 -0.324604952
## 44 -0.25367963 -1.26887528  1.441014446 -0.329235547
## 45  1.16605779  1.43764945  0.848096197  1.294984154
## 46 -0.38603825 -0.37282466  0.081710256  2.273934800
## 47 -0.25367963 -1.26887528  1.441014446 -0.329235547
## 48  0.64229397 -1.19964357 -0.233502018 -0.303548222
## 49  0.08177393  0.06703089  0.632319066 -0.782110731
## 50 -0.60756574 -0.40401603  0.344670943  0.346517927
```

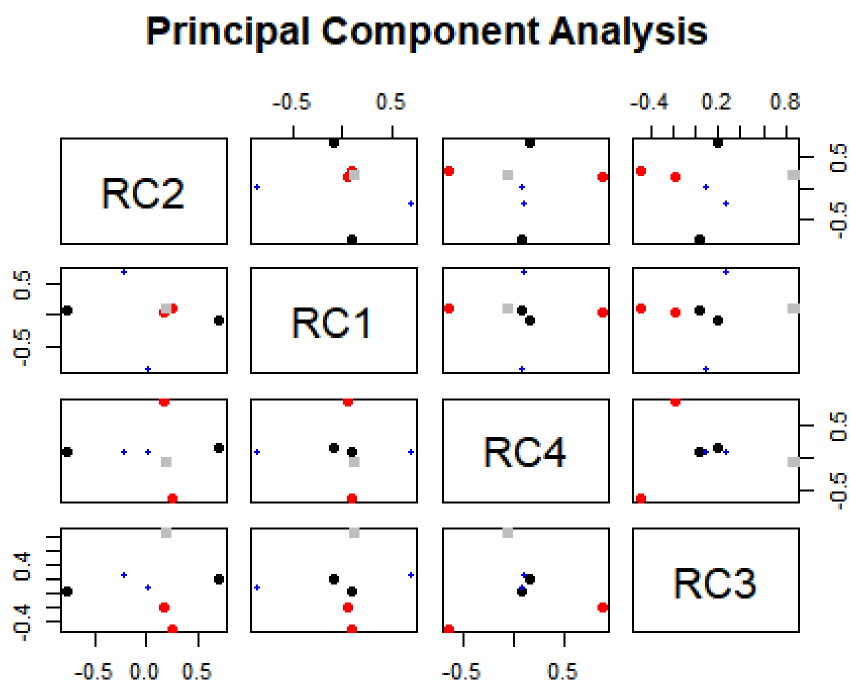
```r
# Play with FA utilities

#factor rotation only in 4 lines
fa.parallel(bank) # See factor recommendation
```

```
## Warning in cor(sampledata, use = use): the standard deviation is zero

## Warning in cor(sampledata, use = use): the standard deviation is zero

## Warning in cor(sampledata, use = use): the standard deviation is zero

## Warning in cor(sampledata, use = use): the standard deviation is zero

## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.ob
s, :
## The estimated weights for the factor scores are probably incorrect. Try a
## different factor score estimation method.

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate,
: An
## ultra-Heywood case was detected. Examine the results carefully

## Warning in cor(sampledata, use = use): the standard deviation is zero

## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.ob
s, :
## The estimated weights for the factor scores are probably incorrect. Try a
## different factor score estimation method.
```

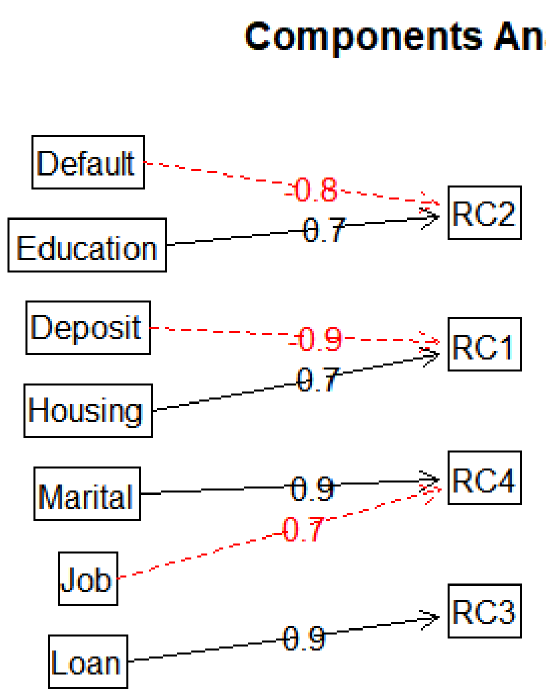
```
## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate,
: An
## ultra-Heywood case was detected. Examine the results carefully

## Warning in cor(sampledata, use = use): the standard deviation is zero

## Warning in cor(sampledata, use = use): the standard deviation is zero

## Warning in cor(sampledata, use = use): the standard deviation is zero

## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.ob
s, :
## The estimated weights for the factor scores are probably incorrect. Try a
## different factor score estimation method.

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate,
: An
## ultra-Heywood case was detected. Examine the results carefully

## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.ob
s, :
## The estimated weights for the factor scores are probably incorrect. Try a
## different factor score estimation method.

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate,
: An
## ultra-Heywood case was detected. Examine the results carefully

## Warning in cor(sampledata, use = use): the standard deviation is zero

## Warning in cor(sampledata, use = use): the standard deviation is zero

## Warning in cor(sampledata, use = use): the standard deviation is zero

## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.ob
s, :
## The estimated weights for the factor scores are probably incorrect. Try a
## different factor score estimation method.

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate,
: An
## ultra-Heywood case was detected. Examine the results carefully
```



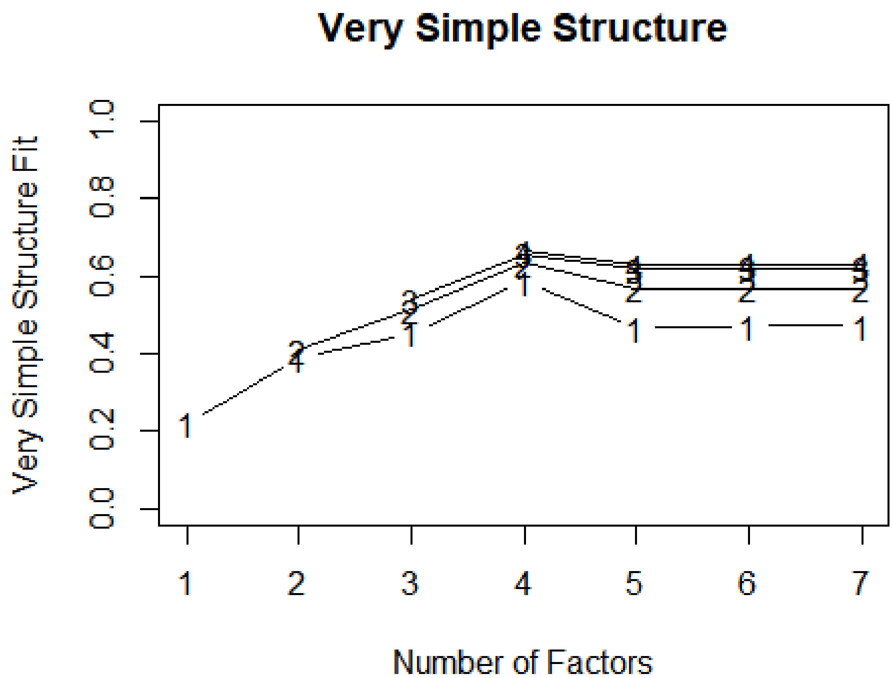

```
## Parallel analysis suggests that the number of factors =  0  and the number
of components =  0

fa.plot(fit.pc) # See Correlations within Factors
```




```
fa.diagram(fit.pc) # Visualize the relationship #to decide which rc to keep l
ook for communalities #so adding it it should be high communalities
```



```r
vss(bank) # See Factor recommendations for a simple structure
```




```
##
## Very Simple Structure
## Call: vss(x = bank)
## VSS complexity 1 achieves a maximimum of 0.59  with  4  factors
## VSS complexity 2 achieves a maximimum of 0.63  with  4  factors
##
## The Velicer MAP achieves a minimum of NA  with  1  factors
## BIC achieves a minimum of  NA  with  1  factors
## Sample Size adjusted BIC achieves a minimum of  NA  with  3  factors
##
## Statistics by number of factors
##   vss1 vss2   map dof   chisq prob sqresid  fit RMSEA BIC  SABIC complex
## 1 0.22 0.00 0.046  14 1.1e+01 0.70     6.1 0.22     0 -44 -0.001     1.0
## 2 0.39 0.41 0.079   8 4.7e+00 0.79     4.6 0.41     0 -27 -1.530     1.1
## 3 0.45 0.51 0.131   3 6.6e-01 0.88     3.6 0.54     0 -11 -1.664     1.4
## 4 0.59 0.63 0.237  -1 8.0e-02   NA     2.6 0.67    NA  NA     NA     1.3
## 5 0.47 0.57 0.417  -4 1.4e-11   NA     2.9 0.63    NA  NA     NA     1.6
## 6 0.47 0.57 1.000  -6 0.0e+00   NA     2.8 0.64    NA  NA     NA     1.6
## 7 0.47 0.57    NA  -7 0.0e+00   NA     2.8 0.64    NA  NA     NA     1.6
##    eChisq    SRMR eCRMS eBIC
## 1 2.1e+01 9.9e-02 0.122  -34
## 2 7.9e+00 6.1e-02 0.099  -23
## 3 8.7e-01 2.0e-02 0.054  -11
## 4 1.1e-01 7.1e-03    NA   NA
## 5 1.4e-11 8.3e-08    NA   NA
```

```
## 6 2.5e-15 1.1e-09    NA   NA
## 7 2.5e-15 1.1e-09    NA   NA
```

#From Compnent analysis we come to that there are 3 Factors which combine multiple columns.

#Loan which tends to RC3 we won't convert it into RC3 as it makes no sense to convert it into RC3. vss(bank) # See Factor recommendations for a simple structure