Factor Analysis

Loading necessary libraries

```
library(cluster)
library(data.table)
library(magrittr)
library(stringr)
library(ggplot2)
library(knitr)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------ tidyverse 1.3.0 --

## v tibble  3.0.3     v purrr   0.3.4
## v tidyr   1.1.2     v dplyr   1.0.2
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts --------------------------------------------------------- tidyverse_conflicts() --
## x dplyr::between()   masks data.table::between()
## x tidyr::extract()   masks magrittr::extract()
## 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::set_names() masks magrittr::set_names()
## x purrr::transpose() masks data.table::transpose()
```

```
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(psych)
```

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

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

```r
library(FactoMineR)
library(nFactors)
```

```
## Loading required package: lattice

##
## Attaching package: 'nFactors'

## The following object is masked from 'package:lattice':
##
##     parallel
```

Data Loading

```r
Lending_Data <- read_csv('Lending_Data.csv')
```

```
## Parsed with column specification:
## cols(
##   member_id = col_character(),
##   loan_status = col_character(),
##   int_rate = col_character(),
##   Bin_int = col_double(),
##   dti = col_double(),
##   Bin_dti = col_double(),
##   Default_flag = col_double(),
##   No_of_Enquiry = col_double(),
##   enq_buckets = col_character(),
##   annual_inc = col_double(),
##   Income_bins = col_double(),
##   home_ownership = col_character(),
##   purpose = col_character(),
##   open_acc = col_double(),
##   emp_length = col_character(),
##   verification_status = col_character(),
##   delinq_2yrs = col_double(),
##   loan_amnt = col_double(),
##   Bins_loan_amt = col_double()
## )
```

```r
Lend = copy(Lending_Data)
Lend = setDT(Lend)
view(Lend)
str(Lend)
```

```
## Classes 'data.table' and 'data.frame':   35808 obs. of  19 variables:
##  $ member_id          : chr  "LC1" "LC10" "LC100" "LC1000" ...
##  $ loan_status        : chr  "Charged Off" "Fully Paid" "Fully Paid" "Fully Paid" ...
##  $ int_rate           : chr  "11.71%" "15.96%" "10.65%" "12.69%" ...
##  $ Bin_int            : num  10 16 8 11 22 1 23 10 5 16 ...
```

```
##  $ dti                : num  1.06 2.61 11.34 14 13.01 ...
##  $ Bin_dti            : num  2 3 11 14 13 11 5 10 24 14 ...
##  $ Default_flag       : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ No_of_Enquiry      : num  0 1 1 1 0 0 3 0 1 2 ...
##  $ enq_buckets        : chr  "0" "1-4" "1-4" "1-4" ...
##  $ annual_inc         : num  110000 135000 75000 51000 41500 ...
##  $ Income_bins        : num  9 11 6 4 3 4 12 7 6 4 ...
##  $ home_ownership     : chr  "MORTGAGE" "RENT" "MORTGAGE" "RENT" ...
##  $ purpose            : chr  "credit_card" "other" "educational" "credit_card" ...
##  $ open_acc           : num  6 3 7 5 8 5 4 7 6 9 ...
##  $ emp_length         : chr  "LT 1year" "10+ years" "2 years" "1 year" ...
##  $ verification_status: chr  "Not Verified" "Source Verified" "Source Verified" "Source Verified" ..
##  $ delinq_2yrs        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ loan_amnt          : num  7000 2000 12000 9350 6000 ...
##  $ Bins_loan_amt      : num  6 2 10 8 5 8 5 10 2 8 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   member_id = col_character(),
##   ..   loan_status = col_character(),
##   ..   int_rate = col_character(),
##   ..   Bin_int = col_double(),
##   ..   dti = col_double(),
##   ..   Bin_dti = col_double(),
##   ..   Default_flag = col_double(),
##   ..   No_of_Enquiry = col_double(),
##   ..   enq_buckets = col_character(),
##   ..   annual_inc = col_double(),
##   ..   Income_bins = col_double(),
##   ..   home_ownership = col_character(),
##   ..   purpose = col_character(),
##   ..   open_acc = col_double(),
##   ..   emp_length = col_character(),
##   ..   verification_status = col_character(),
##   ..   delinq_2yrs = col_double(),
##   ..   loan_amnt = col_double(),
##   ..   Bins_loan_amt = col_double()
##   .. )
##  - attr(*, ".internal.selfref")=<externalptr>
```

Data Cleaning

```
Lend[, member_id := factor(member_id)]
Lend[, loan_status := factor(loan_status)]
Lend[, home_ownership := factor(home_ownership)]
Lend[, purpose := factor(purpose)]
Lend[, verification_status := factor(verification_status)]

Lend[, int_rate := gsub('[%]', '', int_rate)]
Lend[, int_rate := trimws(int_rate)]
Lend[, int_rate := suppressWarnings(as.numeric(int_rate))]

Lend[open_acc %in% c(1,2,3,4,5), 'x' := 'LT5']
```

```r
Lend[open_acc %in% c(6,7,8,9,10), 'x' := '6-10']
Lend[open_acc %in% c(11,12,13,14,15), 'x' := '11-15']
Lend[open_acc >15, 'x' := '15+']
Lend = Lend %>% rename(no_of_acct = x)
str(Lend)
```

```
## Classes 'data.table' and 'data.frame':   35808 obs. of  20 variables:
##  $ member_id          : Factor w/ 35808 levels "LC1","LC10","LC100",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ loan_status        : Factor w/ 2 levels "Charged Off",..: 1 2 2 2 2 2 2 2 2 2 ...
##  $ int_rate           : num  11.7 16 10.7 12.7 19.7 ...
##  $ Bin_int            : num  10 16 8 11 22 1 23 10 5 16 ...
##  $ dti                : num  1.06 2.61 11.34 14 13.01 ...
##  $ Bin_dti            : num  2 3 11 14 13 11 5 10 24 14 ...
##  $ Default_flag       : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ No_of_Enquiry      : num  0 1 1 1 0 0 3 0 1 2 ...
##  $ enq_buckets        : chr  "0" "1-4" "1-4" "1-4" ...
##  $ annual_inc         : num  110000 135000 75000 51000 41500 ...
##  $ Income_bins        : num  9 11 6 4 3 4 12 7 6 4 ...
##  $ home_ownership     : Factor w/ 5 levels "MORTGAGE","NONE",..: 1 5 1 5 1 1 1 5 5 1 ...
##  $ purpose            : Factor w/ 14 levels "car","credit_card",..: 2 10 4 2 3 3 8 2 10 3 ...
##  $ open_acc           : num  6 3 7 5 8 5 4 7 6 9 ...
##  $ emp_length         : chr  "LT 1year" "10+ years" "2 years" "1 year" ...
##  $ verification_status: Factor w/ 3 levels "Not Verified",..: 1 2 2 2 3 3 1 1 1 2 ...
##  $ delinq_2yrs        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ loan_amnt          : num  7000 2000 12000 9350 6000 ...
##  $ Bins_loan_amt      : num  6 2 10 8 5 8 5 10 2 8 ...
##  $ no_of_acct         : chr  "6-10" "LT5" "6-10" "LT5" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   member_id = col_character(),
##   ..   loan_status = col_character(),
##   ..   int_rate = col_character(),
##   ..   Bin_int = col_double(),
##   ..   dti = col_double(),
##   ..   Bin_dti = col_double(),
##   ..   Default_flag = col_double(),
##   ..   No_of_Enquiry = col_double(),
##   ..   enq_buckets = col_character(),
##   ..   annual_inc = col_double(),
##   ..   Income_bins = col_double(),
##   ..   home_ownership = col_character(),
##   ..   purpose = col_character(),
##   ..   open_acc = col_double(),
##   ..   emp_length = col_character(),
##   ..   verification_status = col_character(),
##   ..   delinq_2yrs = col_double(),
##   ..   loan_amnt = col_double(),
##   ..   Bins_loan_amt = col_double()
##   .. )
##  - attr(*, ".internal.selfref")=<externalptr>
##  - attr(*, "index")= int
##   ..- attr(*, "__open_acc")= int  75 113 157 195 377 382 458 611 628 642 ...
```

```r
view(Lend)
```

Creating numeric data subset for Factor Analysis:

```r
Lend_fact = Lend[, c(7, 3, 5, 8, 10, 14, 18)]
Lend_fact[, -1]
```

```
##        int_rate   dti No_of_Enquiry annual_inc open_acc loan_amnt
##     1:    11.71  1.06             0     110000        6      7000
##     2:    15.96  2.61             1     135000        3      2000
##     3:    10.65 11.34             1      75000        7     12000
##     4:    12.69 14.00             1      51000        5      9350
##     5:    19.69 13.01             0      41500        8      6000
##    ---
## 35804:    13.49 19.13             0      48000        6     14000
## 35805:     9.99 11.40             1      50000        4     20000
## 35806:     9.99 21.12             2      45000        9      6400
## 35807:    15.23  7.64             1      30000        3      1500
## 35808:     8.49  7.10             0     107000       11      6000
```

```r
head(Lend_fact)
```

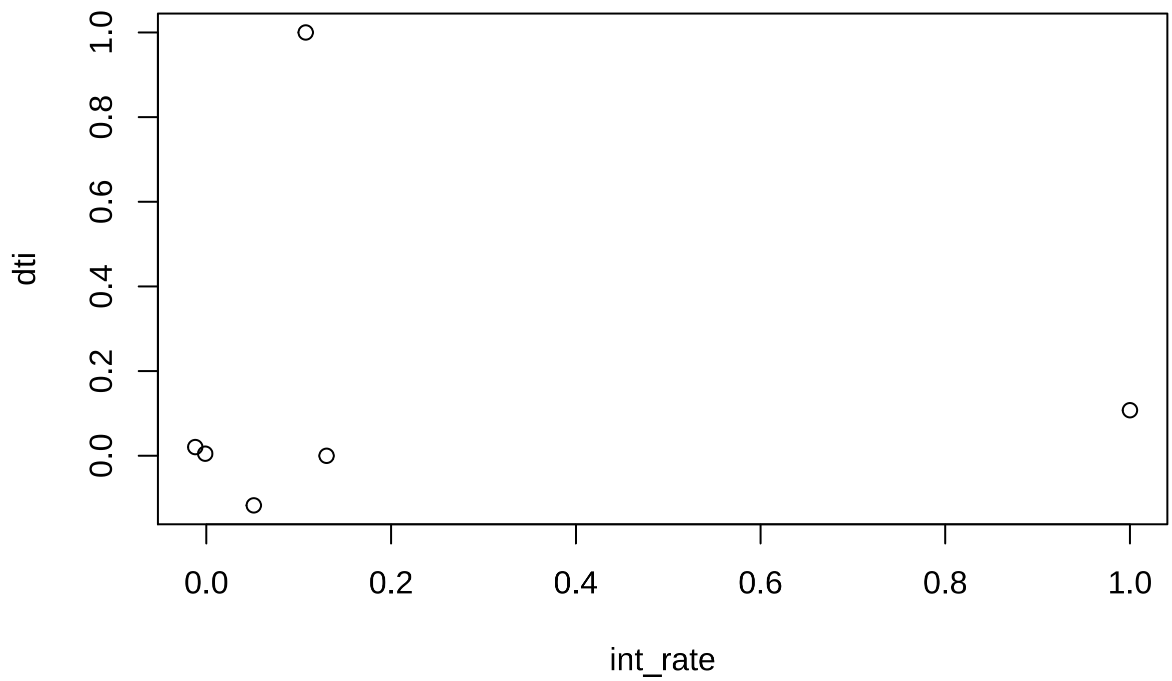
```
##    Default_flag int_rate   dti No_of_Enquiry annual_inc open_acc loan_amnt
## 1:            1    11.71  1.06             0  110000.00        6      7000
## 2:            0    15.96  2.61             1  135000.00        3      2000
## 3:            0    10.65 11.34             1   75000.00        7     12000
## 4:            0    12.69 14.00             1   51000.00        5      9350
## 5:            0    19.69 13.01             0   41500.00        8      6000
## 6:            0     5.42 11.30             0   53200.08        5     10000
```

Computing Correlation Matrix

```r
corrm_lend <- cor(Lend_fact[, -1])
corrm_lend
```

```
##                   int_rate           dti No_of_Enquiry   annual_inc
## int_rate       1.000000000  0.1075628139  0.1301617375  0.051344875
## dti            0.107562814  1.0000000000 -0.0001463515 -0.117231810
## No_of_Enquiry  0.130161738 -0.0001463515  1.0000000000  0.033140513
## annual_inc     0.051344875 -0.1172318095  0.0331405132  1.000000000
## open_acc      -0.001226094  0.0048532184  0.0019370609 -0.002605145
## loan_amnt     -0.012022853  0.0204957868 -0.0067366966 -0.085831026
##                   open_acc    loan_amnt
## int_rate      -0.001226094 -0.012022853
## dti            0.004853218  0.020495787
## No_of_Enquiry  0.001937061 -0.006736697
## annual_inc    -0.002605145 -0.085831026
## open_acc       1.000000000  0.180181997
## loan_amnt      0.180181997  1.000000000
```

```r
plot(corrm_lend)
```



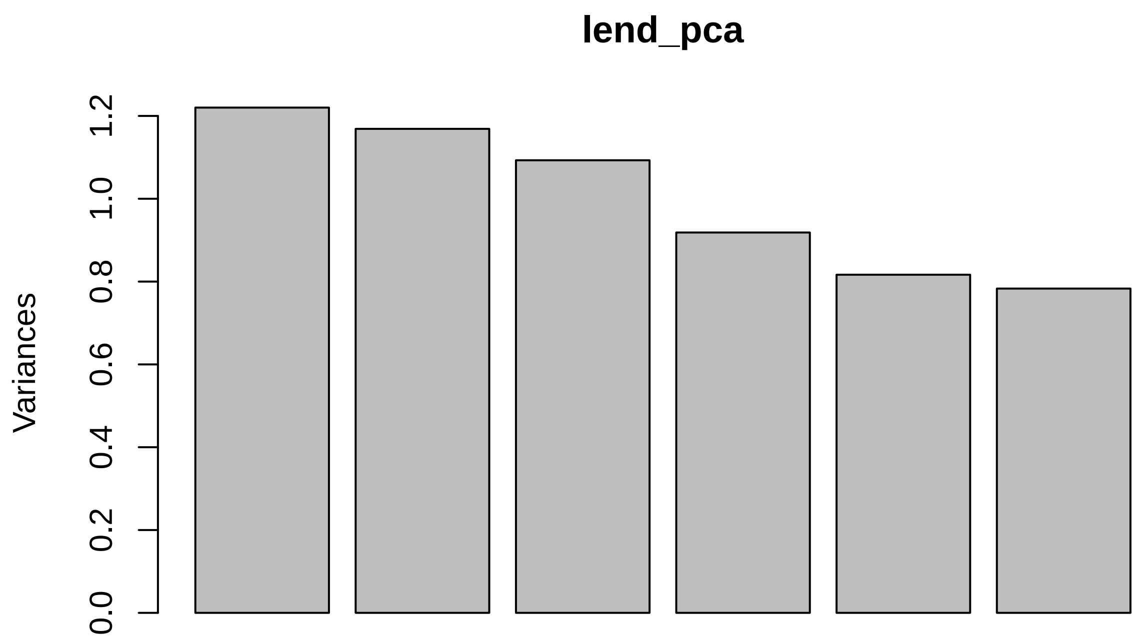

```r
lend_pca <- prcomp(Lend_fact[, -1], scale = TRUE)
lend_pca
```

```
## Standard deviations (1, .., p=6):
## [1] 1.1046047 1.0811152 1.0454206 0.9583681 0.9036363 0.8849328
##
## Rotation (n x k) = (6 x 6):
##                      PC1           PC2         PC3        PC4         PC5
## int_rate      -0.09562723 -0.7061927415  0.08031202  0.2899078  0.46291537
## dti            0.25994780 -0.4521514772 -0.53064118  0.3553719 -0.35348269
## No_of_Enquiry -0.13803378 -0.5438666419  0.30398089 -0.6820655 -0.29338659
## annual_inc    -0.44022257  0.0001169514  0.56076768  0.5061564 -0.05404644
## open_acc       0.53996076 -0.0292878372  0.48268242  0.2406027 -0.52229259
## loan_amnt      0.64720609 -0.0142175352  0.26855693 -0.1018187  0.54678432
##                     PC6
## int_rate      -0.4328510
## dti            0.4417770
## No_of_Enquiry  0.2036232
## annual_inc     0.4823157
## open_acc      -0.3793719
## loan_amnt      0.4466085
```

```r
summary(lend_pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5    PC6
## Standard deviation     1.1046 1.0811 1.0454 0.9584 0.9036 0.8849
## Proportion of Variance 0.2034 0.1948 0.1822 0.1531 0.1361 0.1305
## Cumulative Proportion  0.2034 0.3982 0.5803 0.7334 0.8695 1.0000
```

```r
plot(lend_pca)
```



A table containing eigenvalues and %'s accounted, follows. Eigenvalues are the sdev^2

```r
eigen_lend <- round(lend_pca$sdev^2, 2)
names(eigen_lend) <- paste("PC", 1:6, sep = "")
eigen_lend
```

```
##  PC1  PC2  PC3  PC4  PC5  PC6
## 1.22 1.17 1.09 0.92 0.82 0.78
```

```r
sumlambdas_lend <- sum(eigen_lend)
sumlambdas_lend
```

```
## [1] 6
```

```r
propvar_lend <- round(eigen_lend/sumlambdas_lend, 2)
propvar_lend
```

```
##  PC1  PC2  PC3  PC4  PC5  PC6
## 0.20 0.19 0.18 0.15 0.14 0.13
```

```r
cumvar_lend <- cumsum(propvar_lend)
cumvar_lend
```

```
##  PC1  PC2  PC3  PC4  PC5  PC6
## 0.20 0.39 0.57 0.72 0.86 0.99
```

```r
matlambdas_lend <- rbind(eigen_lend, propvar_lend, cumvar_lend)
matlambdas_lend
```

```
##               PC1  PC2  PC3  PC4  PC5  PC6
## eigen_lend   1.22 1.17 1.09 0.92 0.82 0.78
## propvar_lend 0.20 0.19 0.18 0.15 0.14 0.13
## cumvar_lend  0.20 0.39 0.57 0.72 0.86 0.99
```

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

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

```r
eigvec_lend <- lend_pca$rotation
print(lend_pca)
```

```
## Standard deviations (1, .., p=6):
## [1] 1.1046047 1.0811152 1.0454206 0.9583681 0.9036363 0.8849328
##
## Rotation (n x k) = (6 x 6):
##                      PC1           PC2         PC3        PC4         PC5
## int_rate      -0.09562723 -0.7061927415  0.08031202  0.2899078  0.46291537
## dti            0.25994780 -0.4521514772 -0.53064118  0.3553719 -0.35348269
## No_of_Enquiry -0.13803378 -0.5438666419  0.30398089 -0.6820655 -0.29338659
## annual_inc    -0.44022257  0.0001169514  0.56076768  0.5061564 -0.05404644
## open_acc       0.53996076 -0.0292878372  0.48268242  0.2406027 -0.52229259
## loan_amnt      0.64720609 -0.0142175352  0.26855693 -0.1018187  0.54678432
##                      PC6
## int_rate      -0.4328510
## dti            0.4417770
## No_of_Enquiry  0.2036232
## annual_inc     0.4823157
## open_acc      -0.3793719
## loan_amnt      0.4466085
```

Taking the first four PCs to generate linear combinations for all the variables with four factors:

```r
pcafactors_lend <- eigvec_lend[, 1:4]
pcafactors_lend
```

```
##                         PC1           PC2         PC3        PC4
## int_rate      -0.09562723 -0.7061927415  0.08031202  0.2899078
## dti            0.25994780 -0.4521514772 -0.53064118  0.3553719
## No_of_Enquiry -0.13803378 -0.5438666419  0.30398089 -0.6820655
## annual_inc    -0.44022257  0.0001169514  0.56076768  0.5061564
## open_acc       0.53996076 -0.0292878372  0.48268242  0.2406027
## loan_amnt      0.64720609 -0.0142175352  0.26855693 -0.1018187
```

Multiplying each column of the eigenvector's matrix by the square-root of the corresponding eigenvalue in order to get the factor loadings:

```r
unrot_fact_lend <- sweep(pcafactors_lend, MARGIN = 2, lend_pca$sdev[1:4], `*`)
unrot_fact_lend
```

```
##                        PC1          PC2         PC3         PC4
## int_rate      -0.1056303 -0.763475713  0.08395985  0.27783844
## dti            0.2871396 -0.488827838 -0.55474324  0.34057706
## No_of_Enquiry -0.1524728 -0.587982498  0.31778790 -0.65366979
## annual_inc    -0.4862719  0.000126438  0.58623811  0.48508413
## open_acc       0.5964432 -0.031663526  0.50460616  0.23058599
## loan_amnt      0.7149069 -0.015370794  0.28075496 -0.09757982
```

Computing communalities:

```r
communalities_lend <- rowSums(unrot_fact_lend^2)
communalities_lend
```

```
##      int_rate           dti No_of_Enquiry    annual_inc      open_acc
##     0.6782964     0.7451346     0.8972447     0.8154421     0.6645443
##     loan_amnt
##     0.5996733
```

Performing the varimax rotation. The default in the varimax function is norm=TRUE thus, Kaiser normalization is carried out:

```r
rot_fact_lend <- varimax(unrot_fact_lend)
#View(unrot_fact_lend)
rot_fact_lend
```

```
## $loadings
##
## Loadings:
##               PC1    PC2    PC3    PC4
## int_rate             -0.693  0.305 -0.325
## dti                  -0.786 -0.301  0.190
## No_of_Enquiry                       -0.946
## annual_inc                  0.901
```

```
## open_acc       0.796         0.164
## loan_amnt      0.738        -0.225
##
##                  PC1   PC2   PC3   PC4
## SS loadings    1.180 1.103 1.074 1.044
## Proportion Var 0.197 0.184 0.179 0.174
## Cumulative Var 0.197 0.380 0.559 0.733
##
## $rotmat
##             [,1]       [,2]        [,3]       [,4]
## [1,]  0.83434885 -0.1373138 -0.50402609  0.1759677
## [2,] -0.03334336  0.7910104 -0.05839255  0.6080962
## [3,]  0.53983613  0.3554496  0.66809937 -0.3686133
## [4,]  0.10642912 -0.4786470  0.54423455  0.6807192
```

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_lend <- rot_fact_lend$loadings[1:6, 1:4]
fact_load_lend
```

```
##                       PC1         PC2         PC3         PC4
## int_rate       0.012219002 -0.69255582  0.30512451 -0.32467297
## dti           -0.007349407 -0.78629560 -0.30145174  0.18959599
## No_of_Enquiry -0.005626273 -0.01832900 -0.03225174 -0.94648662
## annual_inc    -0.037625038  0.04306586  0.90075121  0.02861966
## open_acc       0.795643163 -0.03795330  0.16384591  0.05666000
## loan_amnt      0.738170586  0.03617550 -0.22496827 -0.05346091
```

Computing the rotated factor scores for the borrowers and displaying top six:

```
scale_lend <- scale(Lend_fact[, -1])
head(scale_lend)
```

```
##         int_rate         dti No_of_Enquiry  annual_inc   open_acc  loan_amnt
## [1,] -0.05862895 -1.81148017    -0.8077044  0.61884642 -0.7391090 -0.5483201
## [2,]  1.10158281 -1.58009441     0.1200054  0.99877218 -1.4151766 -1.2278888
## [3,] -0.34799941 -0.27687010     0.1200054  0.08695035 -0.5137532  0.1312485
## [4,]  0.20890223  0.12021771     0.1200054 -0.27777838 -0.9644649 -0.2289229
## [5,]  2.11983924 -0.02757061    -0.8077044 -0.42215017 -0.2883973 -0.6842339
## [6,] -1.77574235 -0.28284135    -0.8077044 -0.24434370 -0.9644649 -0.1405789
```

```
head(as.matrix(scale_lend)%*%fact_load_lend%*%solve(t(fact_load_lend)%*%fact_load_lend))
```

```
##             PC1        PC2        PC3        PC4
## [1,] -0.7883754  1.2918674  0.9582642  0.3902663
## [2,] -1.6498123  0.4542124  1.4657893 -0.6466192
## [3,] -0.2718044  0.4527940 -0.1046096 -0.1289179
## [4,] -0.8040622 -0.1757852 -0.3647132 -0.2089395
## [5,] -0.5700736 -1.3695085  0.4273507  0.2288565
## [6,] -0.7796288  1.2954864 -0.7565350  1.0155960
```

```r
fit_pc <- principal(Lend_fact[, -1], nfactors = 4, rotate = "varimax")
fit_pc
```

```
## Principal Components Analysis
## Call: principal(r = Lend_fact[, -1], nfactors = 4, rotate = "varimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                RC1   RC2   RC3   RC4   h2   u2 com
## int_rate       0.01  0.69  0.31  0.32 0.68 0.32 1.8
## dti           -0.01  0.79 -0.30 -0.19 0.75 0.25 1.4
## No_of_Enquiry -0.01  0.02 -0.03  0.95 0.90 0.10 1.0
## annual_inc    -0.04 -0.04  0.90 -0.03 0.82 0.18 1.0
## open_acc       0.80  0.04  0.16 -0.06 0.66 0.34 1.1
## loan_amnt      0.74 -0.04 -0.22  0.05 0.60 0.40 1.2
##
##                        RC1  RC2  RC3  RC4
## SS loadings           1.18 1.10 1.07 1.04
## Proportion Var        0.20 0.18 0.18 0.17
## Cumulative Var        0.20 0.38 0.56 0.73
## Proportion Explained  0.27 0.25 0.24 0.24
## Cumulative Proportion 0.27 0.52 0.76 1.00
##
## Mean item complexity =  1.3
## Test of the hypothesis that 4 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.16
##  with the empirical chi square  28437.32  with prob <  NA
##
## Fit based upon off diagonal values = -3.59
```

```r
round(fit_pc$values, 3)
```

```
## [1] 1.220 1.169 1.093 0.918 0.817 0.783
```

```r
fit_pc$loadings
```

```
##
## Loadings:
##               RC1    RC2    RC3    RC4
## int_rate              0.693  0.305  0.325
## dti                   0.786 -0.301 -0.190
## No_of_Enquiry                        0.946
## annual_inc                   0.901
## open_acc       0.796         0.164
## loan_amnt      0.738        -0.225
##
##                  RC1   RC2   RC3   RC4
## SS loadings    1.180 1.103 1.074 1.044
## Proportion Var 0.197 0.184 0.179 0.174
## Cumulative Var 0.197 0.380 0.559 0.733
```

Loadings with more digits:

```r
for (i in c(1, 3, 2, 4)) { print(fit_pc$loadings[[1, i]])}
```

```
## [1] 0.012219
## [1] 0.3051245
## [1] 0.6925558
## [1] 0.324673
```

Communalities:

```r
fit_pc$communality
```

```
##      int_rate          dti No_of_Enquiry    annual_inc      open_acc
##     0.6782964    0.7451346     0.8972447     0.8154421     0.6645443
##     loan_amnt
##     0.5996733
```

Rotated factor scores, Notice the columns ordering: RC1, RC3, RC2 and RC4

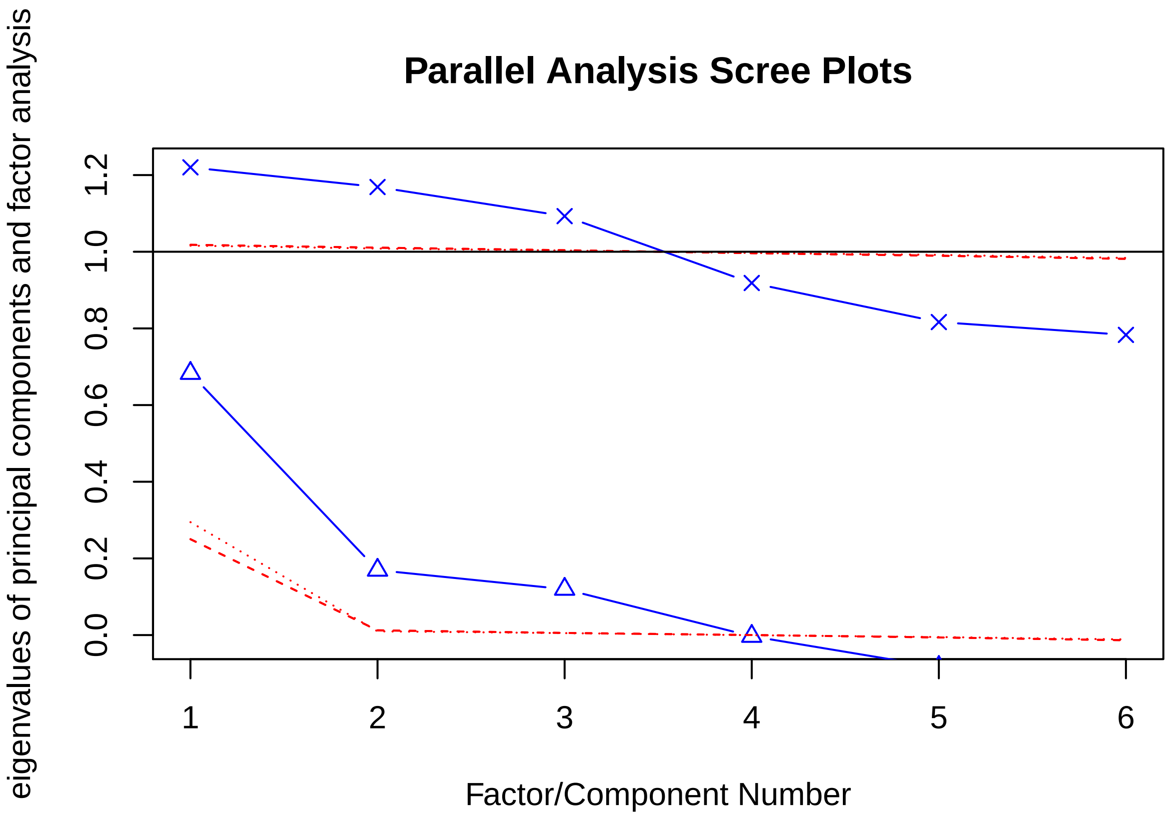
```r
head(fit_pc$scores)
```

```
##             RC1        RC2        RC3        RC4
## [1,] -0.7883754 -1.2918674  0.9582642 -0.3902663
## [2,] -1.6498123 -0.4542124  1.4657893  0.6466192
## [3,] -0.2718044 -0.4527940 -0.1046096  0.1289179
## [4,] -0.8040622  0.1757852 -0.3647132  0.2089395
## [5,] -0.5700736  1.3695085  0.4273507 -0.2288565
## [6,] -0.7796288 -1.2954864 -0.7565350 -1.0155960
```

Play with FA utilities

See factor recommendation:

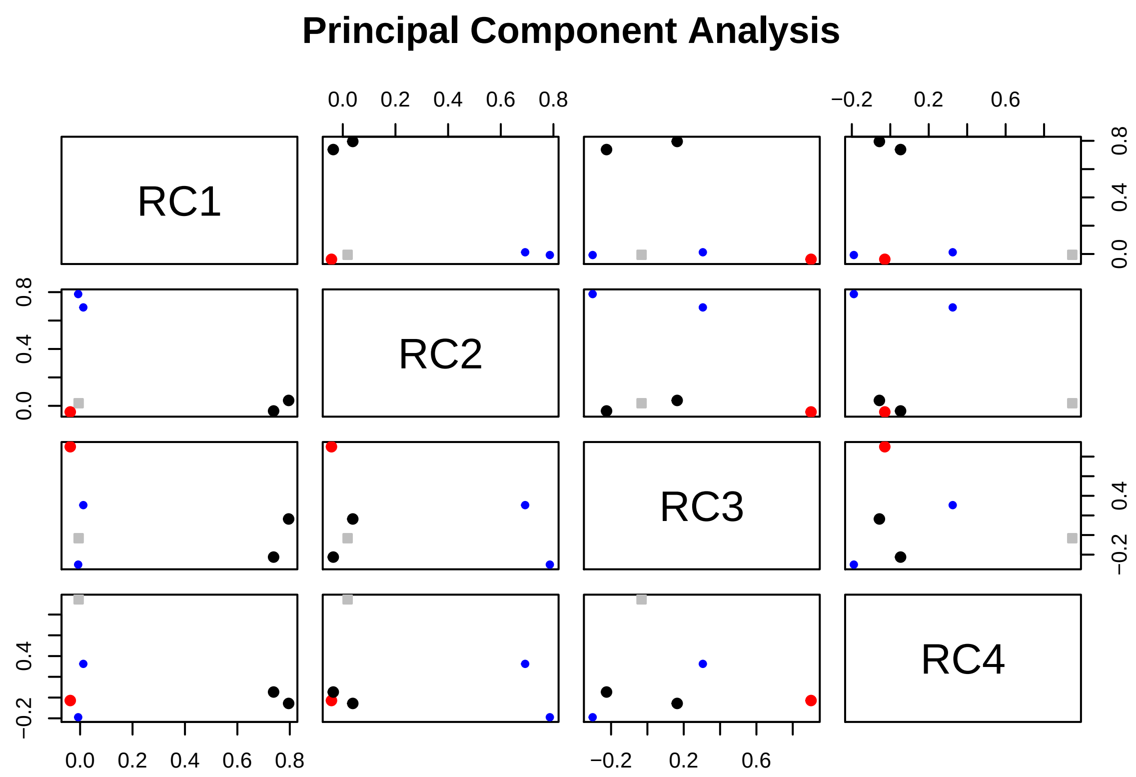
```r
fa.parallel(Lend_fact[, -1], show.legend = FALSE)
```




```
## Parallel analysis suggests that the number of factors =  0  and the number of components =  3
```

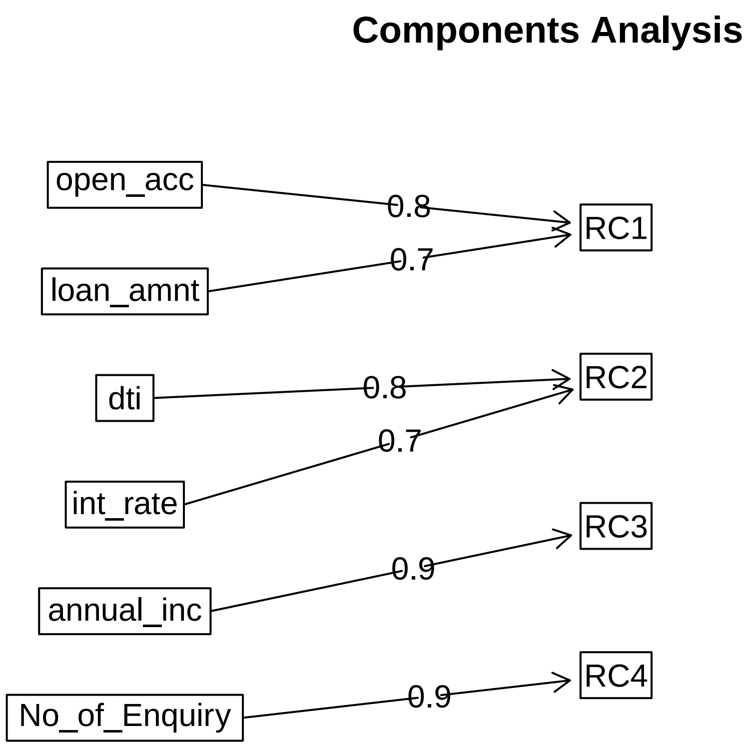
See Correlations within Factors:

```r
fa.plot(fit_pc)
```



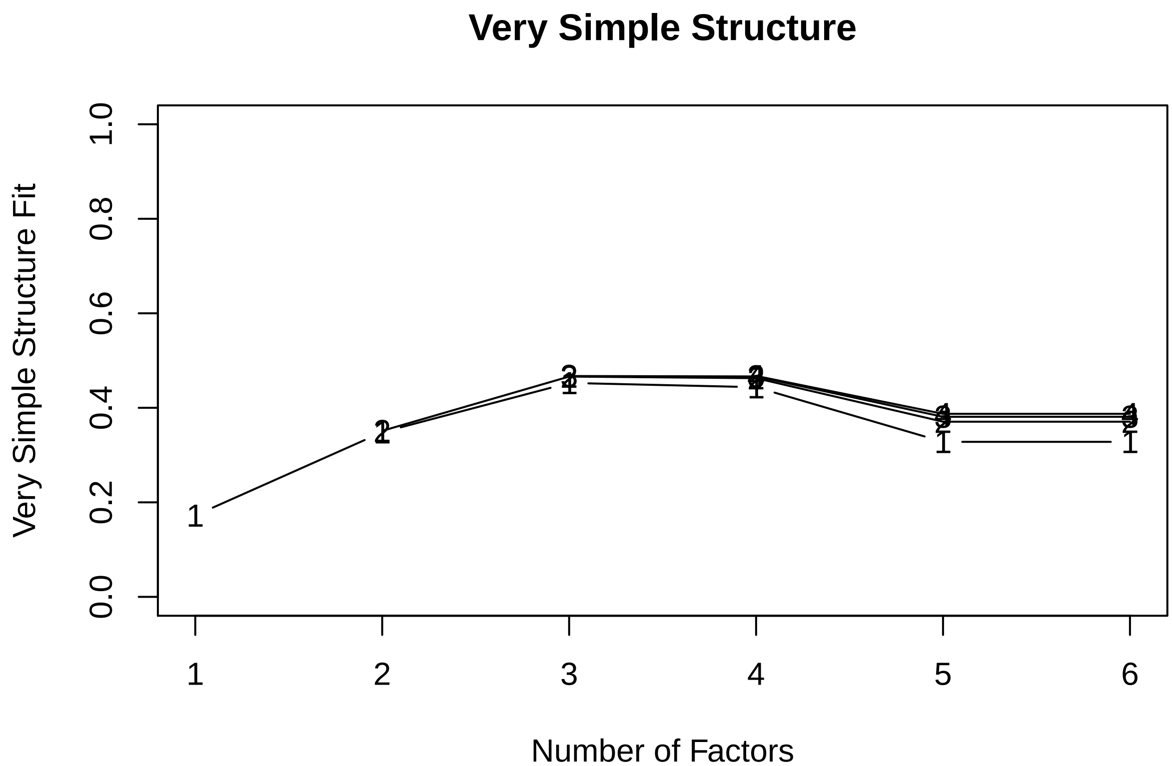
Visualize the relationship:

```
fa.diagram(fit_pc)
```



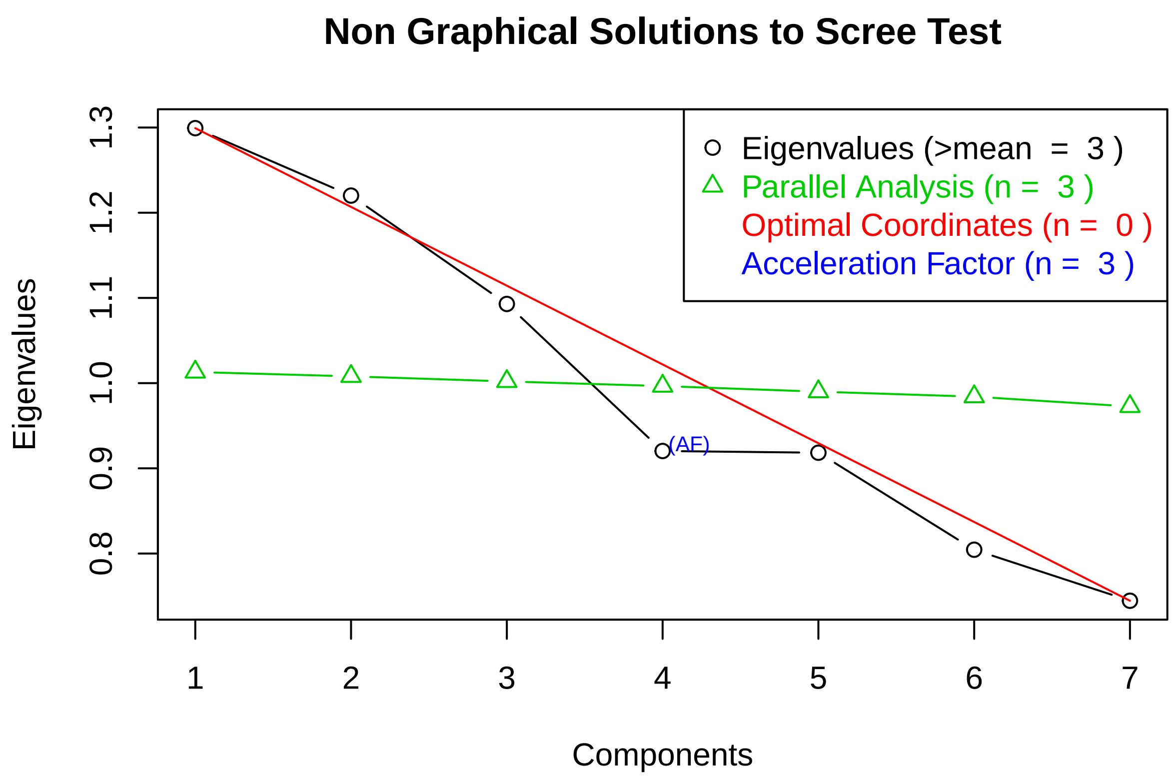
See Factor recommendations for a simple structure:

```
vss(Lend_fact[, -1])
```




```
##
## Very Simple Structure
## Call: vss(x = Lend_fact[, -1])
## VSS complexity 1 achieves a maximimum of 0.45  with  3  factors
## VSS complexity 2 achieves a maximimum of 0.47  with  3  factors
##
## The Velicer MAP achieves a minimum of NA  with  1  factors
## BIC achieves a minimum of  NA  with  2  factors
## Sample Size adjusted BIC achieves a minimum of  NA  with  2  factors
##
## Statistics by number of factors
##   vss1 vss2  map dof   chisq     prob sqresid  fit RMSEA  BIC SABIC complex
## 1 0.17 0.00 0.05   9 1699.61  0.0e+00     5.1 0.17 0.072 1605  1634     1.0
## 2 0.35 0.35 0.11   4  583.49 5.8e-125     4.0 0.35 0.064  542   554     1.1
## 3 0.45 0.47 0.20   0    2.05       NA     3.3 0.47    NA   NA    NA     1.1
## 4 0.44 0.46 0.42  -3    0.16       NA     3.3 0.47    NA   NA    NA     1.1
## 5 0.33 0.37 1.00  -5    0.00       NA     3.8 0.39    NA   NA    NA     1.3
## 6 0.33 0.37   NA  -6    0.00       NA     3.8 0.39    NA   NA    NA     1.3
##    eChisq    SRMR eCRMS eBIC
## 1 3.3e+03 5.5e-02 0.071 3161
## 2 1.1e+03 3.2e-02 0.062 1055
## 3 3.2e+00 1.7e-03    NA   NA
## 4 2.3e-01 4.6e-04    NA   NA
## 5 1.4e-14 1.2e-10    NA   NA
## 6 1.4e-14 1.2e-10    NA   NA
```

```r
eigenvals <- eigen(cor(Lend_fact)) # get eigenvalues
par <- parallel(subject = NROW(Lend_fact), var = NCOL(Lend_fact), rep = 100, cent = .05)
Scree <- nScree(x = eigenvals$values, aparallel = par$eigen$qevpea)
plotnScree(Scree)
```




```r
rot_varimax <- factanal(covmat = corrm_lend, factors = 2,
                        n.obs = NROW(Lend_fact), rotation = "varimax")
rot_varimax
```

```
##
## Call:
## factanal(factors = 2, covmat = corrm_lend, n.obs = NROW(Lend_fact),     rotation = "varimax")
##
## Uniquenesses:
##      int_rate           dti No_of_Enquiry    annual_inc      open_acc
##         0.005         0.988         0.983         0.988         0.960
##     loan_amnt
##         0.187
##
## Loadings:
##               Factor1 Factor2
## int_rate       0.993
## dti            0.110
## No_of_Enquiry  0.129
## annual_inc
```

```
## open_acc                0.199
## loan_amnt               0.899
##
##                Factor1 Factor2
## SS loadings      1.022   0.867
## Proportion Var   0.170   0.145
## Cumulative Var   0.170   0.315
##
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 574.94 on 4 degrees of freedom.
## The p-value is 4.11e-123
```