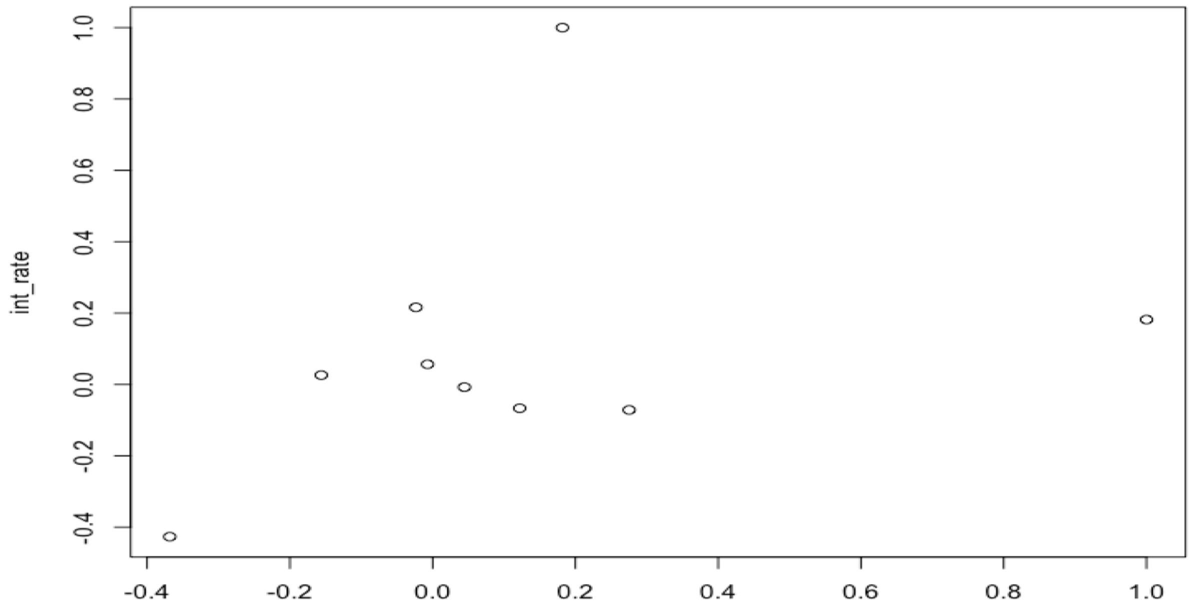
Factor Analysis

ABC<-euroemp
# Computing Correlation Matrix
corrm.emp <- cor(euroemp[-1])
corrm.emp

```
> ABC<-euroemp
>             # Computing Correlation Matrix
>             corrm.emp <- cor(euroemp[-1])
>             corrm.emp
                  loan_amnt    int_rate      issue_d      purpose          dti   emp_length home_ownership
loan_amnt       1.000000000  0.18182824 -0.007165254 -0.156003584 -0.023428386  0.044541782    0.121961884
int_rate        0.181828244  1.00000000  0.056567247  0.026051583  0.216011067 -0.007511820   -0.066756409
issue_d        -0.007165254  0.05656725  1.000000000  0.046423193 -0.067248173  0.009880283    0.023826468
purpose        -0.156003584  0.02605158  0.046423193  1.000000000 -0.100760314  0.005532303    0.048543854
dti            -0.023428386  0.21601107 -0.067248173 -0.100760314  1.000000000  0.048241195    0.004595834
emp_length      0.044541782 -0.00751182  0.009880283  0.005532303  0.048241195  1.000000000    0.181587513
home_ownership  0.121961884 -0.06675641  0.023826468  0.048543854  0.004595834  0.181587513    1.000000000
annual_inc      0.275302638 -0.07140994  0.024494141  0.026711033 -0.213059622  0.030683145    0.101969604
term           -0.368270502 -0.42675736 -0.023469236  0.063222698 -0.026617172 -0.029370193   -0.057841985
                 annual_inc        term
loan_amnt       0.27530264 -0.36827050
int_rate       -0.07140994 -0.42675736
issue_d         0.02449414 -0.02346924
purpose         0.02671103  0.06322270
dti            -0.21305962 -0.02661717
emp_length      0.03068314 -0.02937019
home_ownership  0.10196960 -0.05784198
annual_inc      1.00000000 -0.04585242
term           -0.04585242  1.00000000
```

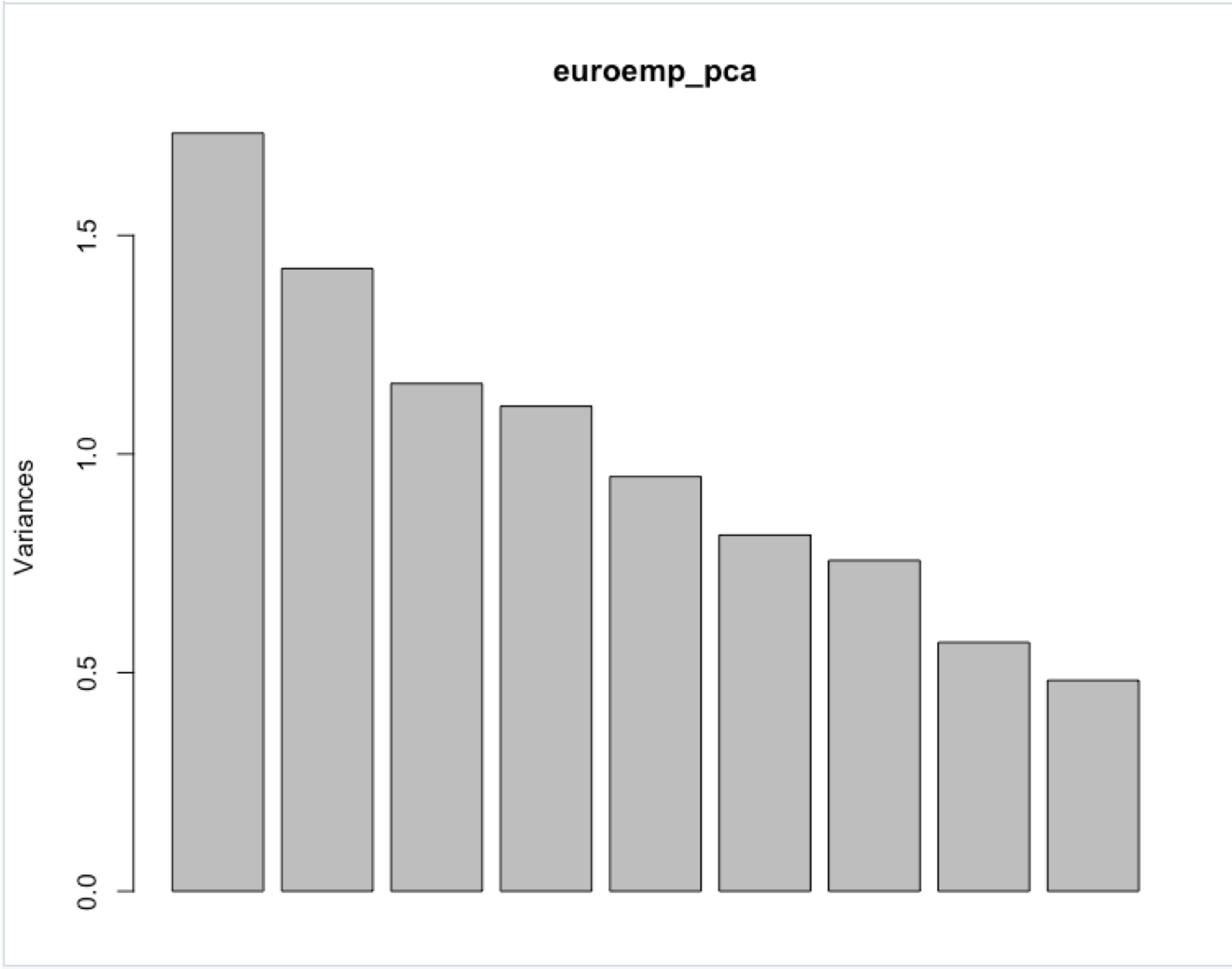
plot(corrm.emp)



euroemp_pca <- prcomp(euroemp[-1], scale=TRUE)
summary(euroemp_pca)

```
>             summary(euroemp_pca)
Importance of components:
                          PC1    PC2   PC3    PC4    PC5     PC6     PC7     PC8     PC9
Standard deviation     1.3168 1.1935 1.078 1.0533 0.9738 0.90255 0.86975 0.75436 0.69463
Proportion of Variance 0.1927 0.1583 0.129 0.1233 0.1054 0.09051 0.08405 0.06323 0.05361
Cumulative Proportion  0.1927 0.3509 0.480 0.6032 0.7086 0.79911 0.88316 0.94639 1.00000
```

plot(euroemp_pca)



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

```
>                 eigen_euroemp
 PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8  PC9
1.73 1.42 1.16 1.11 0.95 0.81 0.76 0.57 0.48
```

names(eigen_euroemp) <- paste("PC",1:9,sep="")
eigen_euroemp

```
>                eigen_euroemp
 PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8  PC9
1.73 1.42 1.16 1.11 0.95 0.81 0.76 0.57 0.48
```

sumlambdas <- sum(eigen_euroemp)
sumlambdas

```
> sumlambdas <- sum(eigen_euroemp)
>              sumlambdas
[1] 8.99
```

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

```
> propvar <- round(eigen_euroemp/sumlambdas,2)
>              propvar
 PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8  PC9
0.19 0.16 0.13 0.12 0.11 0.09 0.08 0.06 0.05
```

cumvar_euroemp <- cumsum(propvar)
cumvar_euroemp
matlambdas <- rbind(eigen_euroemp,propvar,cumvar_euroemp)
matlambdas

```
> matlambdas <- rbind(eigen_euroemp,propvar,cumvar_euroemp)
> matlambdas
                PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8  PC9
eigen_euroemp  1.73 1.42 1.16 1.11 0.95 0.81 0.76 0.57 0.48
propvar        0.19 0.16 0.13 0.12 0.11 0.09 0.08 0.06 0.05
cumvar_euroemp 0.19 0.35 0.48 0.60 0.71 0.80 0.88 0.94 0.99
```

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

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

eigvec.emp <- euroemp_pca$rotation
print(euroemp_pca)

```
Standard deviations (1, .., p=9):
[1] 1.3168091 1.1935147 1.0775910 1.0532695 0.9738201 0.9025509 0.8697546 0.7543563 0.6946274

Rotation (n x k) = (9 x 9):
                      PC1         PC2         PC3         PC4         PC5         PC6         PC7         PC8
loan_amnt       0.54679645 -0.25377540  0.09149861 -0.22712566 -0.01696929 -0.05789206 -0.08130664 -0.69309332
int_rate        0.48836689  0.37652571  0.08137268  0.29848452  0.12493862  0.07590500 -0.08065786  0.40444721
issue_d         0.04298565 -0.09319355  0.10066581  0.56855528 -0.78969194 -0.08592919 -0.11876835 -0.08836233
purpose        -0.14824069 -0.10892940 -0.00988357  0.69900027  0.57703256 -0.02369502 -0.16678074 -0.33217563
dti             0.11988763  0.52496964 -0.38629545 -0.11896389 -0.03384585 -0.27134035 -0.61394005 -0.13197568
emp_length      0.10084538 -0.16121910 -0.67528219  0.06893525 -0.08496720  0.70346947 -0.01633217 -0.02054461
home_ownership  0.13618563 -0.34362679 -0.56937238  0.08547384  0.03673534 -0.63975078  0.25910590  0.20145387
annual_inc      0.17960841 -0.58914485  0.18550507 -0.08953343  0.09049345  0.03858871 -0.61194104  0.41477169
term           -0.60277676 -0.08074672 -0.10463248 -0.11505629 -0.09807615 -0.06063530 -0.35519709 -0.08623621
                      PC9
loan_amnt       0.29330030
int_rate        0.57667771
issue_d        -0.05628034
purpose        -0.07599686
dti            -0.27847112
emp_length      0.01740615
home_ownership  0.11633292
annual_inc     -0.14844275
term            0.67753746
```

```
# Taking the first four PCs to generate linear combinations for all the variables with four factors
pcafactors.emp <- eigvec.emp[,1:4]
pcafactors.emp
```

```
                      PC1         PC2         PC3         PC4
loan_amnt       0.54679645 -0.25377540  0.09149861 -0.22712566
int_rate        0.48836689  0.37652571  0.08137268  0.29848452
issue_d         0.04298565 -0.09319355  0.10066581  0.56855528
purpose        -0.14824069 -0.10892940 -0.00988357  0.69900027
dti             0.11988763  0.52496964 -0.38629545 -0.11896389
emp_length      0.10084538 -0.16121910 -0.67528219  0.06893525
home_ownership  0.13618563 -0.34362679 -0.56937238  0.08547384
annual_inc      0.17960841 -0.58914485  0.18550507 -0.08953343
term           -0.60277676 -0.08074672 -0.10463248 -0.11505629
```

```
# Multiplying each column of the eigenvector’s matrix by the square-root of the corresponding eigenvalue in order to get the factor loadings
unrot.fact.emp <- sweep(pcafactors.emp,MARGIN=2,euroemp_pca$sdev[1:4],`*`)
unrot.fact.emp
```

```
> unrot.fact.emp <- sweep(pcafactors.emp,MARGIN=2,euroemp_pca$sdev[1:4],`*`)
> unrot.fact.emp
                      PC1        PC2         PC3         PC4
loan_amnt      0.72002654 -0.3028847  0.09859808 -0.23922453
int_rate       0.64308597  0.4493890  0.08768647  0.31438464
issue_d        0.05660389 -0.1112279  0.10847657  0.59884194
purpose       -0.19520469 -0.1300088 -0.01065045  0.73623567
dti            0.15786912  0.6265590 -0.41626851 -0.12530104
emp_length     0.13279411 -0.1924174 -0.72767803  0.07260740
home_ownership 0.17933048 -0.4101236 -0.61355057  0.09002699
annual_inc     0.23650999 -0.7031530  0.19989860 -0.09430283
term          -0.79374192 -0.0963724 -0.11275102 -0.12118529
```

# Computing communalities
communalities.emp <- rowSums(unrot.fact.emp^2)
communalities.emp

```
> communalities.emp <- rowSums(unrot.fact.emp^2)
> communalities.emp
     loan_amnt       int_rate        issue_d        purpose            dti     emp_length home_ownership
     0.6771273      0.7220366      0.3859545      0.5971636      0.6064786      0.5894459      0.5849100
    annual_inc           term
     0.5992136      0.6667125
```

# Performing the varimax rotation. The default in the varimax function is norm=TRUE thus, Kaiser normalization is carried out
rot.fact.emp <- varimax(unrot.fact.emp)
View(unrot.fact.emp)
rot.fact.emp

```
> rot.fact.emp
$loadings

Loadings:
               PC1    PC2    PC3    PC4
loan_amnt       0.609 -0.445 -0.152 -0.292
int_rate        0.764  0.310         0.180
issue_d         0.133                0.601
purpose        -0.109                0.762
dti             0.201  0.691 -0.163 -0.250
emp_length             0.105 -0.760
home_ownership        -0.140 -0.750
annual_inc      0.100 -0.757 -0.125
term           -0.814

                 PC1   PC2   PC3   PC4
SS loadings    1.697 1.385 1.221 1.125
Proportion Var 0.189 0.154 0.136 0.125
Cumulative Var 0.189 0.343 0.478 0.603

$rotmat
          [,1]       [,2]        [,3]        [,4]
[1,] 0.9589642 -0.1472361 -0.19966157 -0.13727519
[2,] 0.2008183  0.8982626  0.37587983 -0.10728748
[3,] 0.1397422 -0.3979056  0.90216168  0.09081609
[4,] 0.1432893  0.1145290 -0.07038405  0.98050872
```

# 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 $loadings
fact.load.emp <- rot.fact.emp$loadings[1:9,1:4]
fact.load.emp

```
> fact.load.emp <- rot.fact.emp$loadings[1:9,1:4]
> fact.load.emp
                      PC1           PC2          PC3          PC4
loan_amnt       0.609154886 -0.4447147483 -0.151820862 -0.291953493
int_rate        0.764243383  0.3100990994  0.097496405  0.179726666
issue_d         0.132910931 -0.0828246304  0.002604553  0.601184213
purpose        -0.109296060  0.0005173104 -0.071320500  0.761663343
dti             0.201090741  0.6908554528 -0.162731823 -0.249556084
emp_length     -0.002579579  0.1054693899 -0.760433323  0.007521951
home_ownership  0.016772204 -0.1403567379 -0.749820881  0.051935493
annual_inc      0.100020324 -0.7567800921 -0.124544719 -0.031338172
term           -0.813644046  0.0612847473  0.029065172 -0.009762211
```

```
# 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.emp <- scale(euroemp[-1])
scale.emp
as.matrix(scale.emp)%*%fact.load.emp%*%solve(t(fact.load.emp)%*%fact.load.emp)
```

```
> as.matrix(scale.emp)%*%fact.load.emp%*%solve(t(fact.load.emp)%*%fact.load.emp)
                PC1           PC2           PC3          PC4
      [1,] -7.604773e-01 -2.796123e-01  5.885826e-01 1.571487e+00
      [2,] -5.160093e-01  3.854755e-01  1.153475e+00 3.040187e+00
      [3,] -2.364363e-02  5.470126e-01 -1.417640e-01 3.168854e+00
      [4,]  5.330852e-01  1.018750e+00  1.730391e+00 1.716740e+00
      [5,] -9.232816e-01 -5.434223e-01 -1.208899e-01 1.214668e+00
      [6,]  8.446483e-01 -5.408219e-01  5.989810e-01 3.076286e+00
      [7,]  3.134453e-01 -5.971436e-01  1.225615e-01 2.059460e+00
      [8,] -4.159506e-01  8.683714e-01  1.684780e+00 2.842129e+00
      [9,] -3.102745e-01 -8.243968e-01  1.979055e+00 1.239698e+00
```

```
library(psych)
install.packages("psych",
lib="/Library/Frameworks/R.framework/Versions/3.5/Resources/library")
library(psych)
fit.pc <- principal(euroemp[-1], nfactors=4, rotate="varimax")
fit.pc
```

```
Principal Components Analysis
Call: principal(r = euroemp[-1], nfactors = 4, rotate = "varimax")
Standardized loadings (pattern matrix) based upon correlation matrix
                RC1   RC2   RC3   RC4   h2   u2 com
loan_amnt      0.61  0.44  0.15 -0.29 0.68 0.32 2.5
int_rate       0.76 -0.31 -0.10  0.18 0.72 0.28 1.5
issue_d        0.13  0.08  0.00  0.60 0.39 0.61 1.1
purpose       -0.11  0.00  0.07  0.76 0.60 0.40 1.1
dti            0.20 -0.69  0.16 -0.25 0.61 0.39 1.6
emp_length     0.00 -0.11  0.76  0.01 0.59 0.41 1.0
home_ownership 0.02  0.14  0.75  0.05 0.58 0.42 1.1
annual_inc     0.10  0.76  0.12 -0.03 0.60 0.40 1.1
term          -0.81 -0.06 -0.03 -0.01 0.67 0.33 1.0

                       RC1  RC2  RC3  RC4
SS loadings           1.70 1.39 1.22 1.13
Proportion Var        0.19 0.15 0.14 0.13
Cumulative Var        0.19 0.34 0.48 0.60
Proportion Explained  0.31 0.26 0.22 0.21
Cumulative Proportion 0.31 0.57 0.79 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  0.13
 with the empirical chi square  157321  with prob <  0

Fit based upon off diagonal values = 0.08>
```

round(fit.pc$values, 3)
fit.pc$loadings

```
> fit.pc$loadings

Loadings:
               RC1    RC2    RC3    RC4   
loan_amnt       0.609  0.445  0.152 -0.292
int_rate        0.764 -0.310         0.180
issue_d         0.133                0.601
purpose        -0.109                0.762
dti             0.201 -0.691  0.163 -0.250
emp_length            -0.105  0.760       
home_ownership         0.140  0.750       
annual_inc      0.100  0.757  0.125       
term           -0.814                     

                 RC1   RC2   RC3   RC4
SS loadings    1.697 1.385 1.221 1.125
Proportion Var 0.189 0.154 0.136 0.125
Cumulative Var 0.189 0.343 0.478 0.603
```

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

```
> for (i in c(1,3,2,4)) { print(fit.pc$loadings[[1,i]])}
[1] 0.6091549
[1] 0.1518209
[1] 0.4447147
[1] -0.2919535
```

Communalities
fit.pc$communality

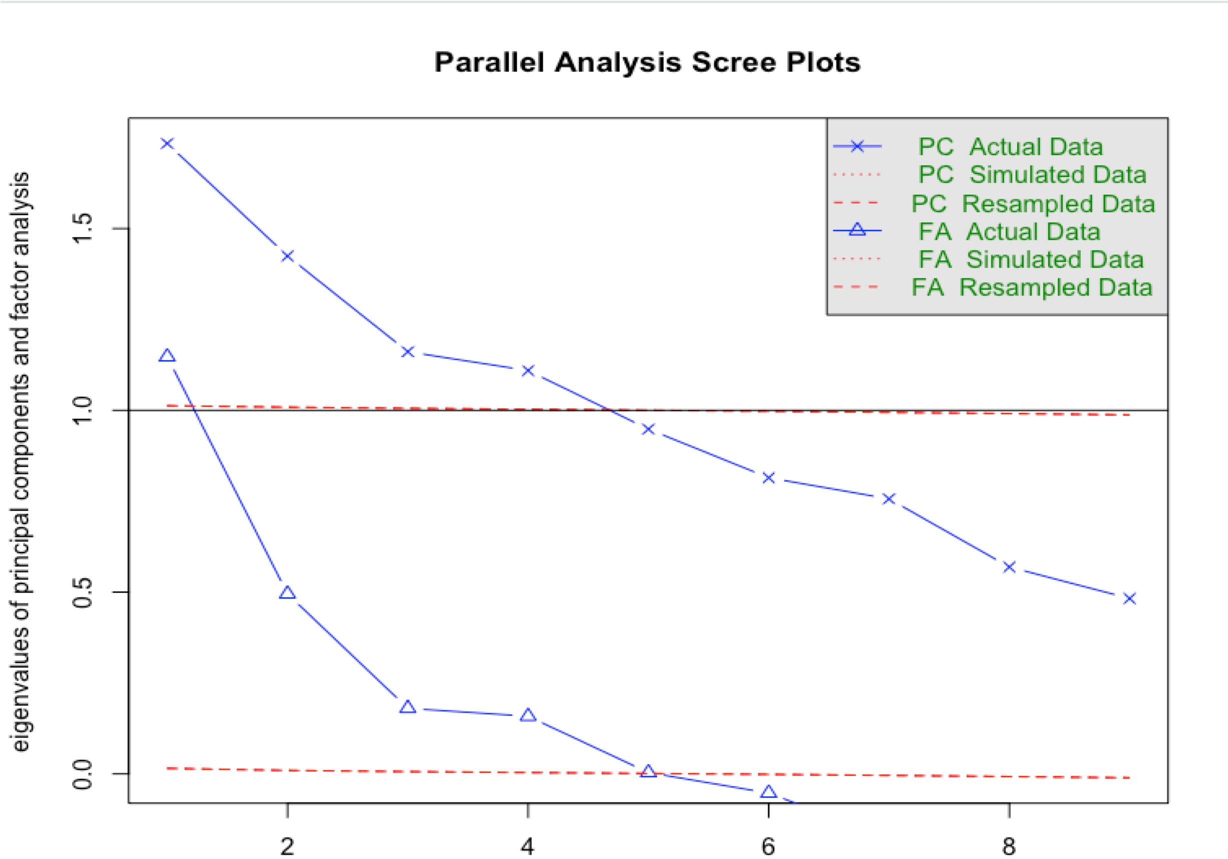
```
> fit.pc$communality
     loan_amnt       int_rate        issue_d        purpose            dti     emp_length home_ownership 
     0.6771273      0.7220366      0.3859545      0.5971636      0.6064786      0.5894459      0.5849100 
    annual_inc           term 
     0.5992136      0.6667125 
```

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

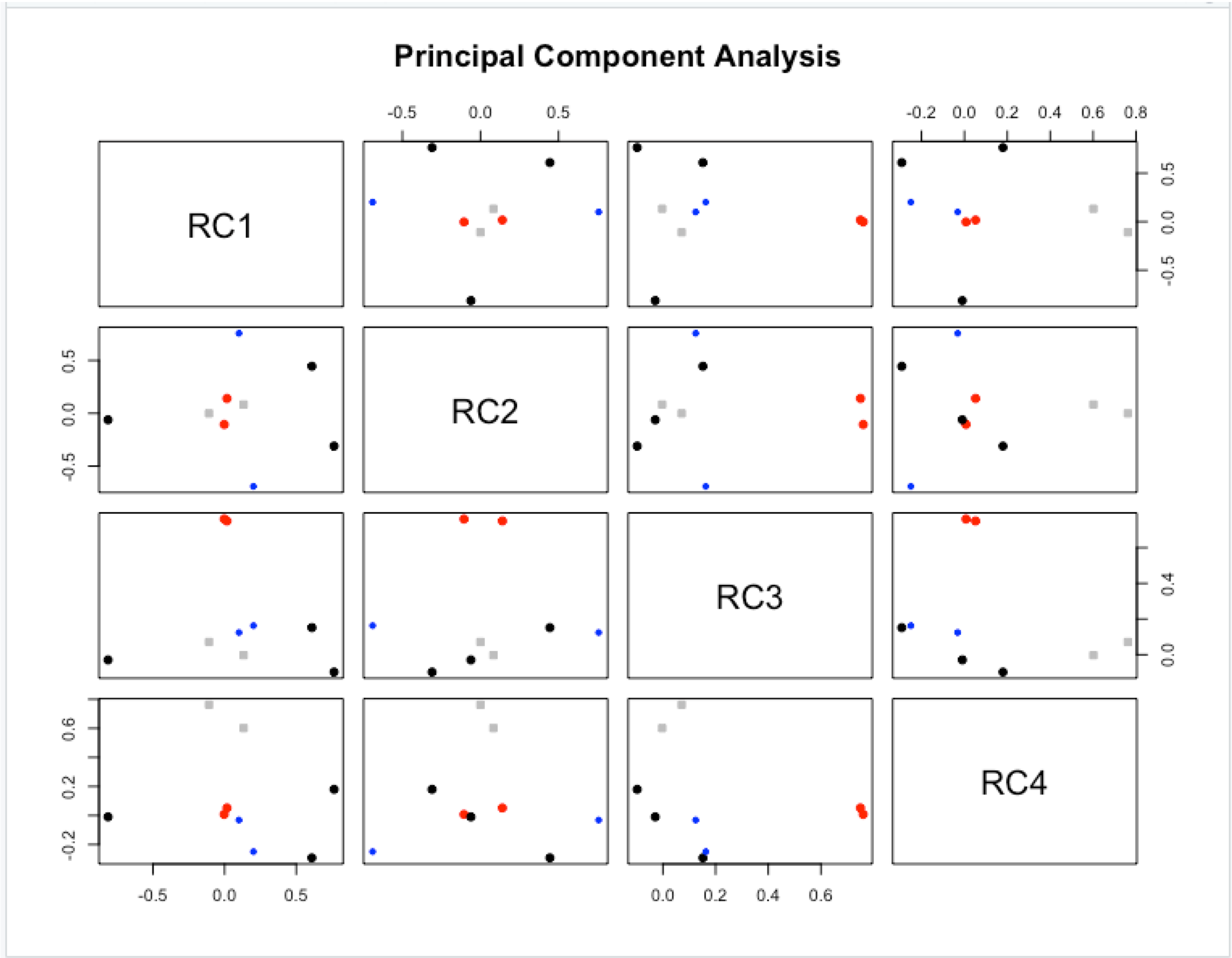
```
> fit.pc$scores
               RC1           RC2           RC3          RC4
 [1,] -7.604773e-01  2.796123e-01 -5.885826e-01  1.571487e+00
 [2,] -5.160093e-01 -3.854755e-01 -1.153475e+00  3.040187e+00
 [3,] -2.364363e-02 -5.470126e-01  1.417640e-01  3.168854e+00
 [4,]  5.330852e-01 -1.018750e+00 -1.730391e+00  1.716740e+00
 [5,] -9.232816e-01  5.434223e-01  1.208899e-01  1.214668e+00
 [6,]  8.446483e-01  5.408219e-01 -5.989810e-01  3.076286e+00
 [7,]  3.134453e-01  5.971436e-01 -1.225615e-01  2.059460e+00
 [8,] -4.159506e-01 -8.683714e-01 -1.684780e+00  2.842129e+00
 [9,] -3.102745e-01  8.243968e-01 -1.979055e+00  1.239698e+00
```

fa.parallel(euroemp[-1])

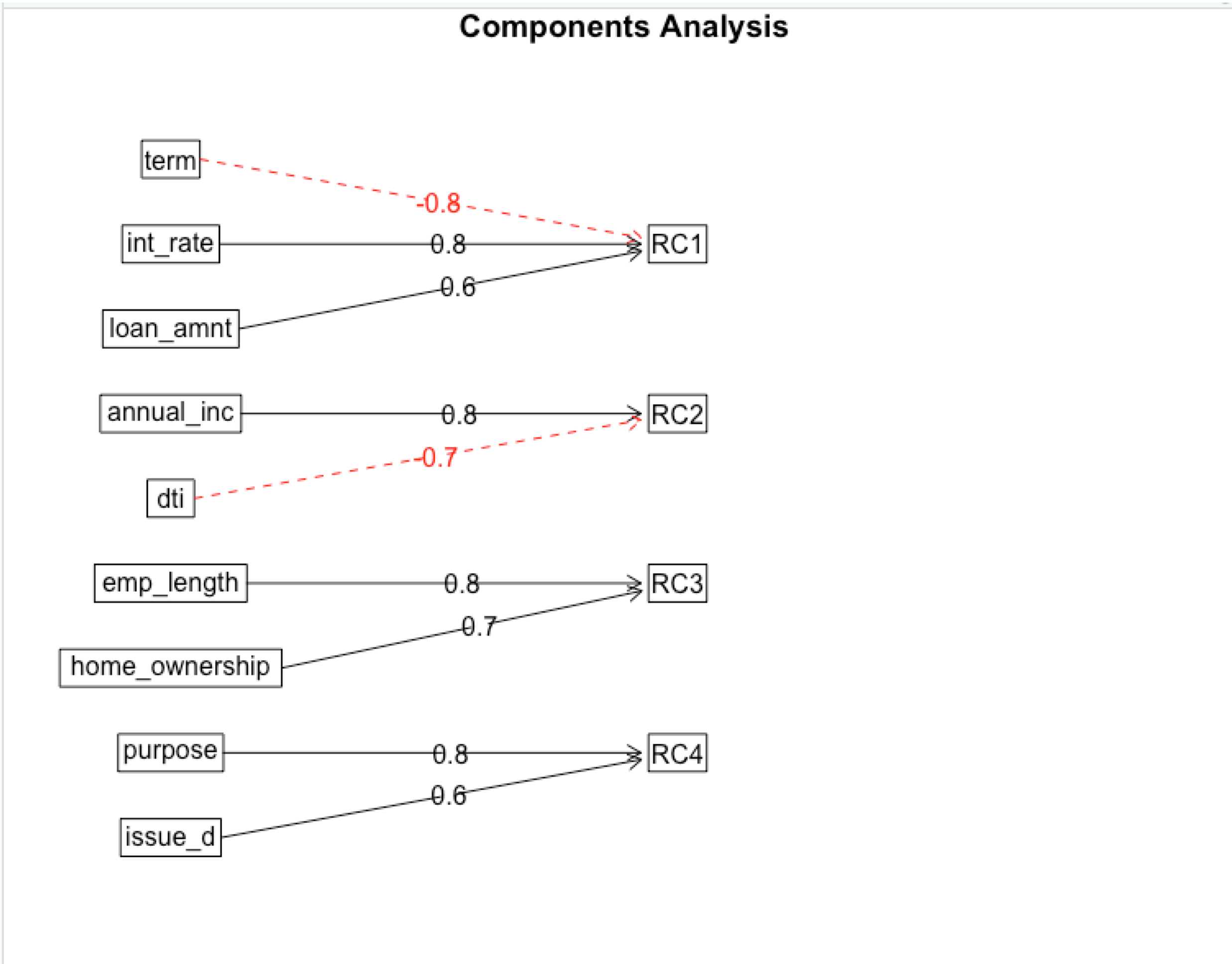
```
> fa.parallel(euroemp[-1]) # See factor recommendation
Parallel analysis suggests that the number of factors =  5  and the number of components =  4
```



fa.plot(fit.pc)



fa.diagram(fit.pc)



vss(euroemp[-1]) # See Factor recommendations for a simple structure

```
Very Simple Structure
Call: vss(x = euroemp[-1])
VSS complexity 1 achieves a maximimum of 0.53  with  6  factors
VSS complexity 2 achieves a maximimum of 0.61  with  6  factors

The Velicer MAP achieves a minimum of 0.03  with  1  factors
BIC achieves a minimum of  NA  with  5  factors
Sample Size adjusted BIC achieves a minimum of  NA  with  5  factors

Statistics by number of factors
  vss1 vss2  map dof   chisq     prob sqresid  fit  RMSEA     BIC   SABIC complex  eChisq    SRMR  eCRMS    eBIC
1 0.25 0.00 0.031  27 4.1e+04  0.0e+00     7.8 0.25 0.1075 40462.1 40547.9     1.0 6.8e+04 8.5e-02 0.0983 67827.1
2 0.31 0.40 0.047  19 1.7e+04  0.0e+00     6.2 0.40 0.0818 16408.9 16469.3     1.4 2.3e+04 5.0e-02 0.0684 23005.1
3 0.40 0.47 0.077  12 1.1e+04  0.0e+00     5.4 0.48 0.0840 10943.1 10981.2     1.3 1.3e+04 3.7e-02 0.0643 12847.8
4 0.42 0.50 0.103   6 8.6e+02 1.4e-182     4.5 0.57 0.0330   789.7   808.7     1.5 7.5e+02 8.9e-03 0.0219   680.2
5 0.52 0.57 0.184   1 1.2e+01  5.1e-04     4.0 0.61 0.0092     0.3     3.5     1.3 1.5e+01 1.3e-03 0.0077     3.6
6 0.53 0.61 0.332  -3 1.3e-03       NA     3.5 0.66     NA      NA      NA     1.4 1.3e-03 1.2e-05     NA      NA
7 0.47 0.57 0.477  -6 5.8e-05       NA     3.9 0.62     NA      NA      NA     1.5 5.1e-05 2.3e-06     NA      NA
8 0.48 0.58 1.000  -8 1.2e-08       NA     3.9 0.63     NA      NA      NA     1.5 8.1e-09 2.9e-08     NA      NA
```

Conclusion – As we can see above our proportion of variance for RC1 and RC2 is 19% and 15% respectively which is significantly low for our data. Therefore Factor analysis is not the best analysis method for our dataset.