Homework - 2

Group 4

5/19/2020

Problem 1

Perform principal component analysis on NHL.xlsx, which contains statistics of 30 teams in the National Hockey League. The description of the variables is provided in the 'Description' sheet of the file. Focus only on the variables 12 through 25, and create a new data frame.

- Input the new data frame to fa.parallel() function to determine the number of components to extract
- Input the new data frame to principal() function to extract the components. If raw data is input, the correlation matrix is automatically calculated by principal() function.
- Rotate the components
- Compute component scores
- Graph an orthogonal solution using factor.plot()
- Interpret the results

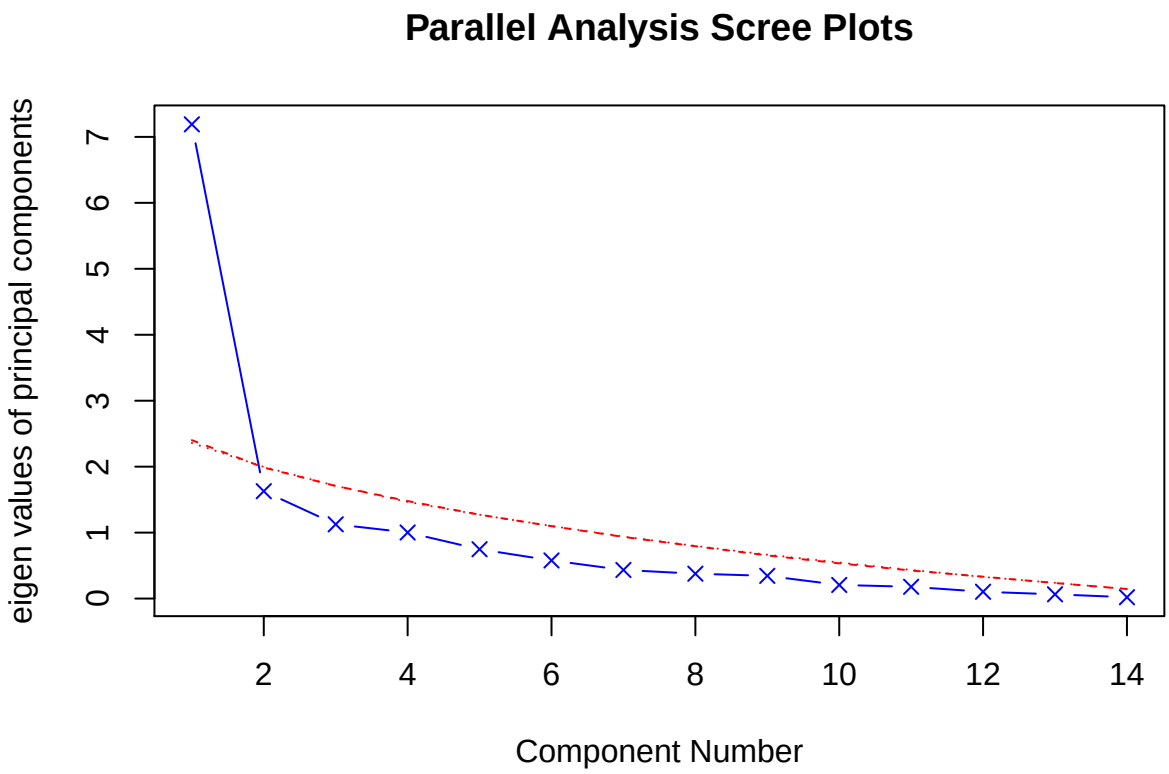
First, import all the required libraries

```
library(dplyr)
library(readxl)
library(psych)
```

```
# Import the NHL excel file as a dataframe
NHL <- data.frame(read_xlsx("./data/NHL.xlsx", sheet = "Data"))

# Select the columns 13-26, the 1st column is the index column from excel
df <- NHL[, 13:26]

# Use Parallel Analysis Scree Plots to figure out the number of factors to extract
fa.parallel(df, fa = "pc", n.iter = 100, show.legend = FALSE)
```




```
## Parallel analysis suggests that the number of factors =  NA  and the number of components =  1
```

From the above plot it will be appropriate to use 1 factor

```
# Perform PCA with varimax orthogonal rotation
pc <- principal(df, nfactors = 1, rotate = "none", scores = TRUE)

pc
```

```
## Principal Components Analysis
## Call: principal(r = df, nfactors = 1, rotate = "none", scores = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##         PC1    h2   u2 com
## gg     0.83 0.682 0.32   1
## gag   -0.82 0.671 0.33   1
## five   0.90 0.818 0.18   1
## PPP    0.16 0.026 0.97   1
## PKP    0.70 0.490 0.51   1
## shots  0.61 0.367 0.63   1
## sag   -0.63 0.400 0.60   1
## sc1    0.82 0.666 0.33   1
## tr1    0.74 0.552 0.45   1
## lead1  0.82 0.667 0.33   1
## lead2  0.75 0.564 0.44   1
## wop    0.71 0.500 0.50   1
## wosp   0.84 0.710 0.29   1
## face   0.28 0.078 0.92   1
```

```
##
##                 PC1
## SS loadings    7.19
## Proportion Var 0.51
##
## Mean item complexity =  1
## Test of the hypothesis that 1 component is sufficient.
##
## The root mean square of the residuals (RMSR) is  0.11
##  with the empirical chi square  70.28  with prob <  0.69
##
## Fit based upon off diagonal values = 0.95
```

Now rotate the components

```
# PCA with varimax rotation
pc <- principal(df, nfactors = 1, rotate = "varimax", scores = TRUE)

pc
```

```
## Principal Components Analysis
## Call: principal(r = df, nfactors = 1, rotate = "varimax", scores = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##         PC1    h2   u2 com
## gg     0.83 0.682 0.32   1
## gag   -0.82 0.671 0.33   1
## five   0.90 0.818 0.18   1
## PPP    0.16 0.026 0.97   1
## PKP    0.70 0.490 0.51   1
## shots  0.61 0.367 0.63   1
## sag   -0.63 0.400 0.60   1
## sc1    0.82 0.666 0.33   1
## tr1    0.74 0.552 0.45   1
## lead1  0.82 0.667 0.33   1
## lead2  0.75 0.564 0.44   1
## wop    0.71 0.500 0.50   1
## wosp   0.84 0.710 0.29   1
## face   0.28 0.078 0.92   1
##
##                 PC1
## SS loadings    7.19
## Proportion Var 0.51
##
## Mean item complexity =  1
## Test of the hypothesis that 1 component is sufficient.
##
## The root mean square of the residuals (RMSR) is  0.11
##  with the empirical chi square  70.28  with prob <  0.69
##
## Fit based upon off diagonal values = 0.95
```

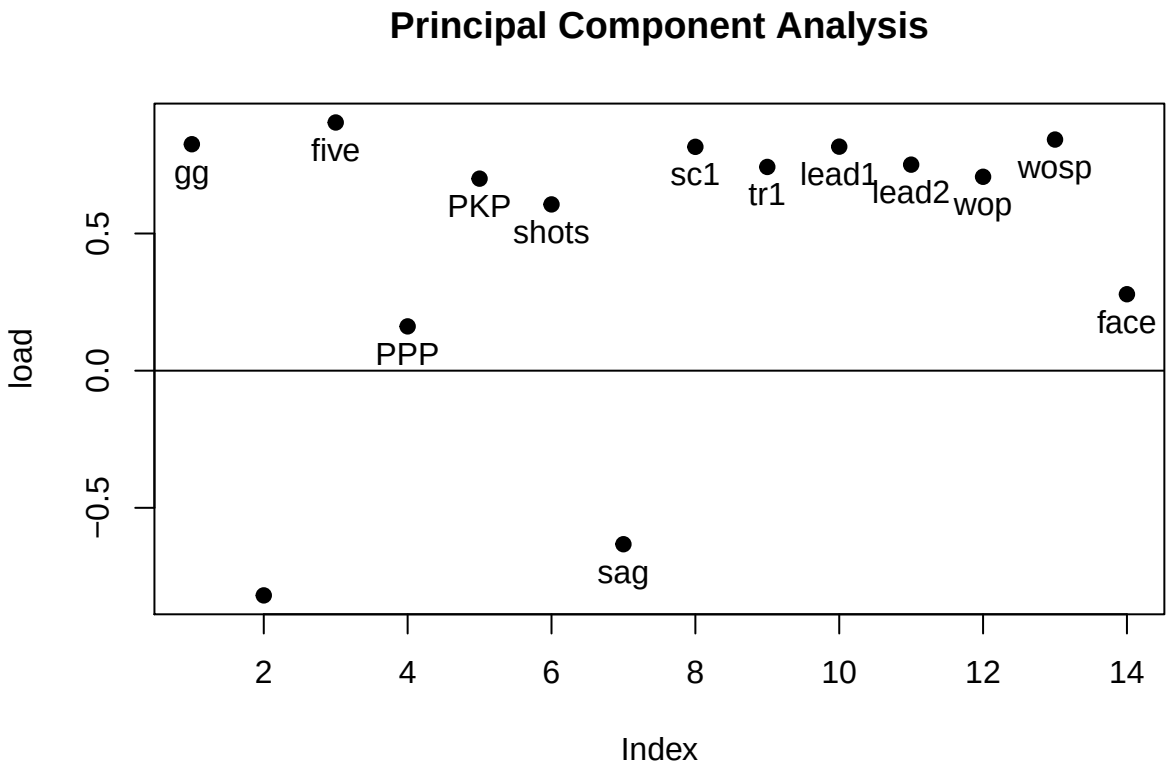
Let's see the component scores

```
head(pc$scores)
```

```
##            PC1
```

```
## [1,] 1.3503592
## [2,] 1.1106603
## [3,] 0.7062801
## [4,] 0.7251884
## [5,] 1.1956402
## [6,] 1.1813631
```

```r
# Plot the components and analyze them
factor.plot(pc, labels = colnames(df))
```




```r
rm(list = ls())
```

Observations made from the plots - - The Principal Component loads all the components except PPP and face.

- There isn't any clear indication as to what the component signifies. Further analysis may be required.

Problem 2

Perform principal component analysis on Glass Identification Data.xlsx

- Input the raw data matrix to fa.parallel() function to determine the number of components to extract
- Input the raw data matrix to principal() function to extract the components. If raw data is input, the correlation matrix is automatically calculated by principal() function.
- Rotate the components

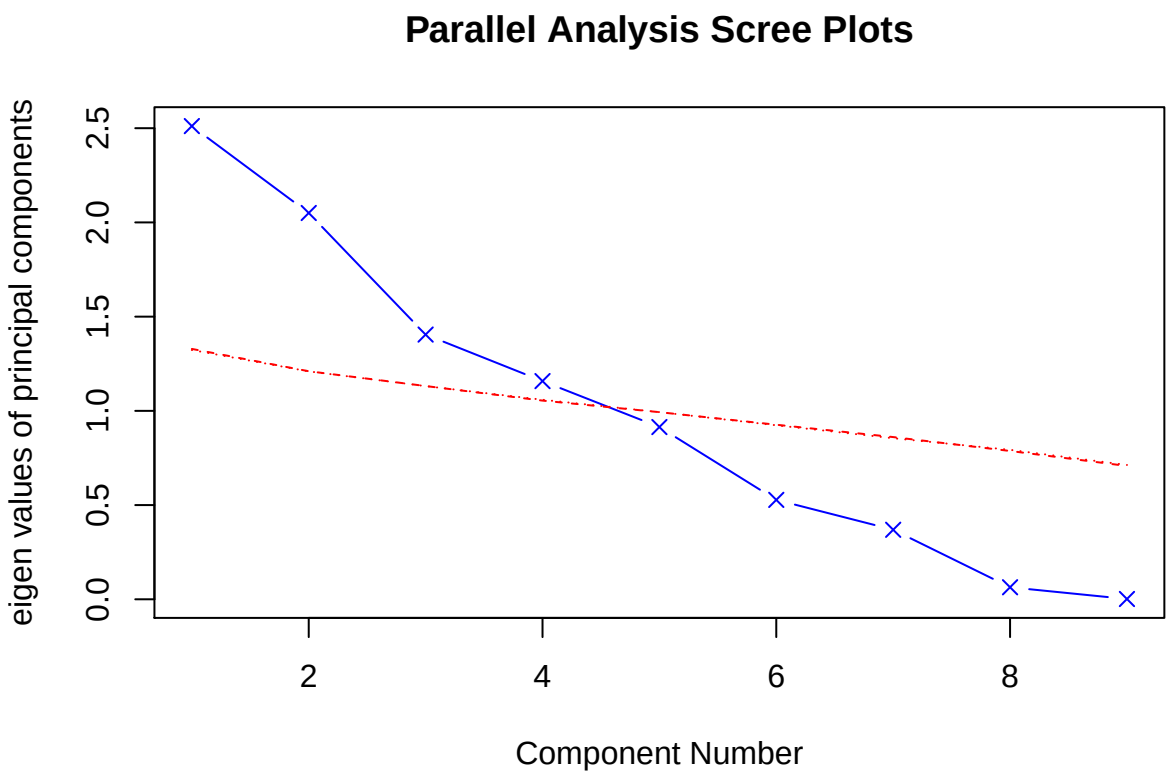
- Compute component scores
- Graph an orthogonal solution using factor.plot()
- Interpret the results

```
glass_identification_data <- data.frame(read_xlsx("./data/Glass Identification Data.xlsx",
  sheet = "Glass Data"
))

fa.parallel(glass_identification_data[, 2:10], fa = "pc", n.iter = 100, show.legend = FALSE)
```

```
## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## 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 =  NA  and the number of components =  4
```

The Skree Plot and the fa.parallel() function suggests nfactors = 4.

```
# Perform PCA without rotation
pc <- principal(glass_identification_data[, 2:10], nfactors = 4, rotate = "none", scores = TRUE)

pc
```

```
## Principal Components Analysis
## Call: principal(r = glass_identification_data[, 2:10], nfactors = 4,
```

```
##     rotate = "none", scores = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##      PC1   PC2   PC3   PC4   h2    u2 com
## RI -0.86  0.41  0.10 -0.16 0.95 0.051 1.5
## Na  0.41  0.39 -0.46 -0.53 0.80 0.195 3.8
## Mg -0.18 -0.85  0.01 -0.41 0.92 0.081 1.5
## Al  0.68  0.42  0.39  0.15 0.81 0.186 2.5
## Si  0.36 -0.22 -0.54  0.70 0.97 0.031 2.7
## K   0.35 -0.22  0.79  0.04 0.79 0.212 1.6
## CA -0.78  0.49  0.00  0.30 0.94 0.058 2.0
## Ba  0.40  0.69  0.09 -0.14 0.67 0.333 1.7
## Fe -0.29 -0.09  0.34  0.25 0.27 0.730 3.0
##
##                        PC1  PC2  PC3  PC4
## SS loadings           2.51 2.05 1.40 1.16
## Proportion Var        0.28 0.23 0.16 0.13
## Cumulative Var        0.28 0.51 0.66 0.79
## Proportion Explained  0.35 0.29 0.20 0.16
## Cumulative Proportion 0.35 0.64 0.84 1.00
##
## Mean item complexity =  2.3
## Test of the hypothesis that 4 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.08
##  with the empirical chi square  102.53  with prob <  7.4e-20
##
## Fit based upon off diagonal values = 0.92
```

Rotate the components

```
# Perform PCA with rotation
pc <- principal(glass_identification_data[, 2:10], nfactors = 4, rotate = "varimax", scores = TRUE)

pc
```

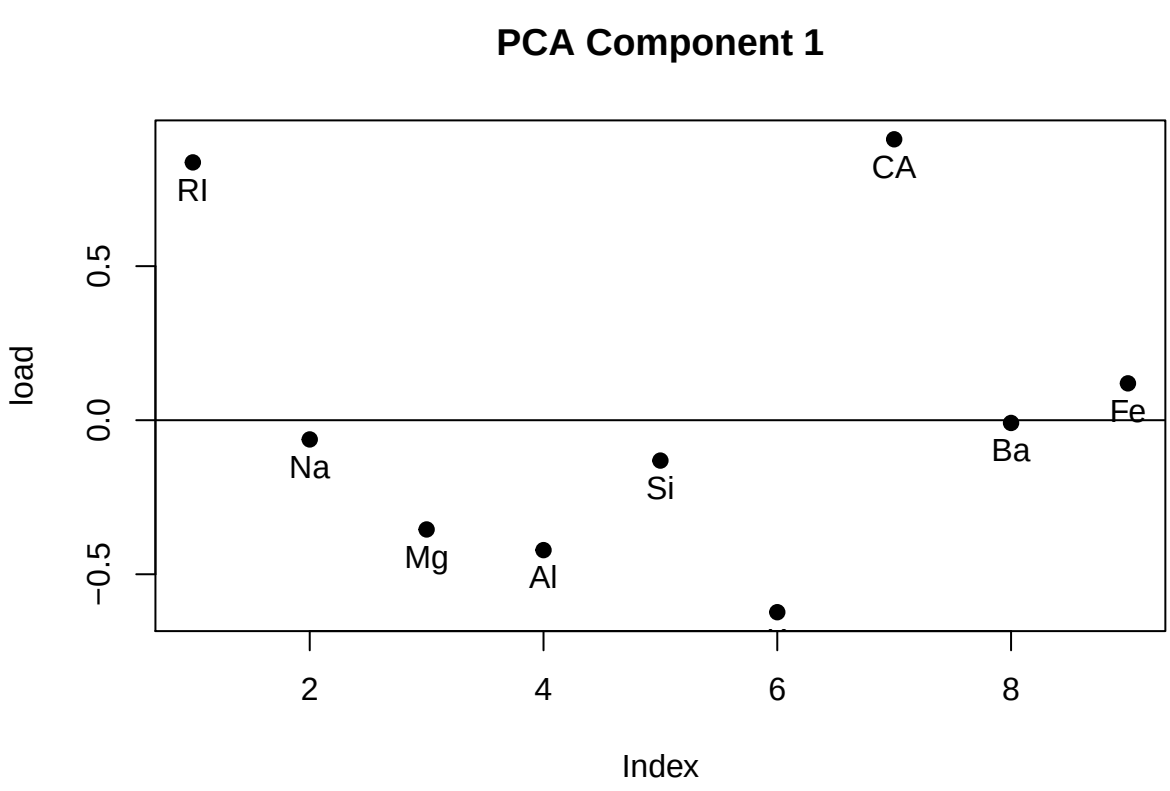
```
## Principal Components Analysis
## Call: principal(r = glass_identification_data[, 2:10], nfactors = 4,
##     rotate = "varimax", scores = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##      RC1   RC2   RC3   RC4   h2    u2 com
## RI  0.84 -0.07  0.15  0.47 0.95 0.051 1.7
## Na -0.06  0.22 -0.86  0.09 0.80 0.195 1.2
## Mg -0.35 -0.86  0.04  0.21 0.92 0.081 1.5
## Al -0.42  0.80  0.03  0.01 0.81 0.186 1.5
## Si -0.13  0.00 -0.02 -0.98 0.97 0.031 1.0
## K  -0.62  0.22  0.51  0.30 0.79 0.212 2.7
## CA  0.91  0.12  0.30  0.06 0.94 0.058 1.3
## Ba -0.01  0.72 -0.33  0.17 0.67 0.333 1.5
## Fe  0.12 -0.04  0.50  0.07 0.27 0.730 1.2
##
##                        RC1  RC2  RC3  RC4
## SS loadings           2.26 2.03 1.48 1.36
## Proportion Var        0.25 0.23 0.16 0.15
## Cumulative Var        0.25 0.48 0.64 0.79
## Proportion Explained  0.32 0.28 0.21 0.19
```

```
## Cumulative Proportion 0.32 0.60 0.81 1.00
##
## Mean item complexity =  1.5
## Test of the hypothesis that 4 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.08
##  with the empirical chi square  102.53  with prob <  7.4e-20
##
## Fit based upon off diagonal values = 0.92
```

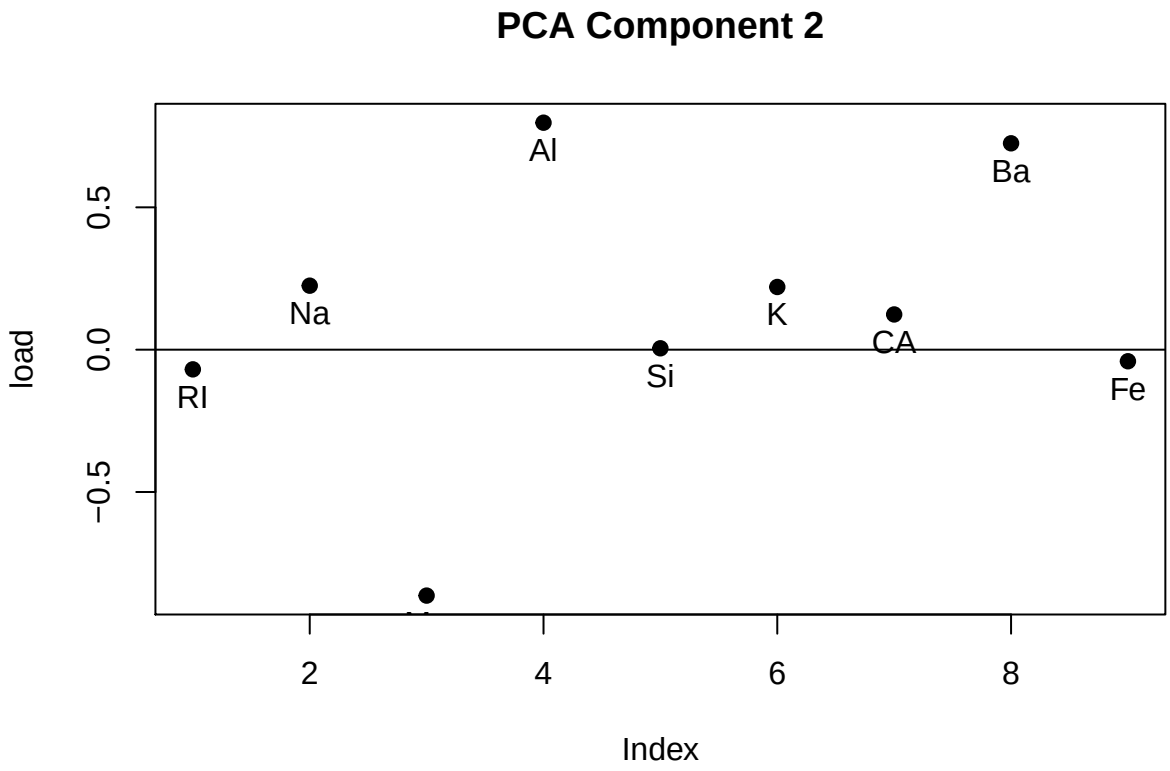
```
head(pc$scores)
```

```
##             RC1        RC2        RC3         RC4
## [1,]  0.2516834 -1.1257154 -0.8331376  1.14203433
## [2,] -0.5120556 -0.5823124 -0.7217195  0.07184681
## [3,] -0.6811108 -0.4417522 -0.4610237 -0.39146231
## [4,] -0.4363986 -0.6266048 -0.1520952  0.09532063
## [5,] -0.4446499 -0.6485935 -0.1947898 -0.37616223
## [6,] -0.7149524 -0.2237372  1.1926990 -0.41874608
```

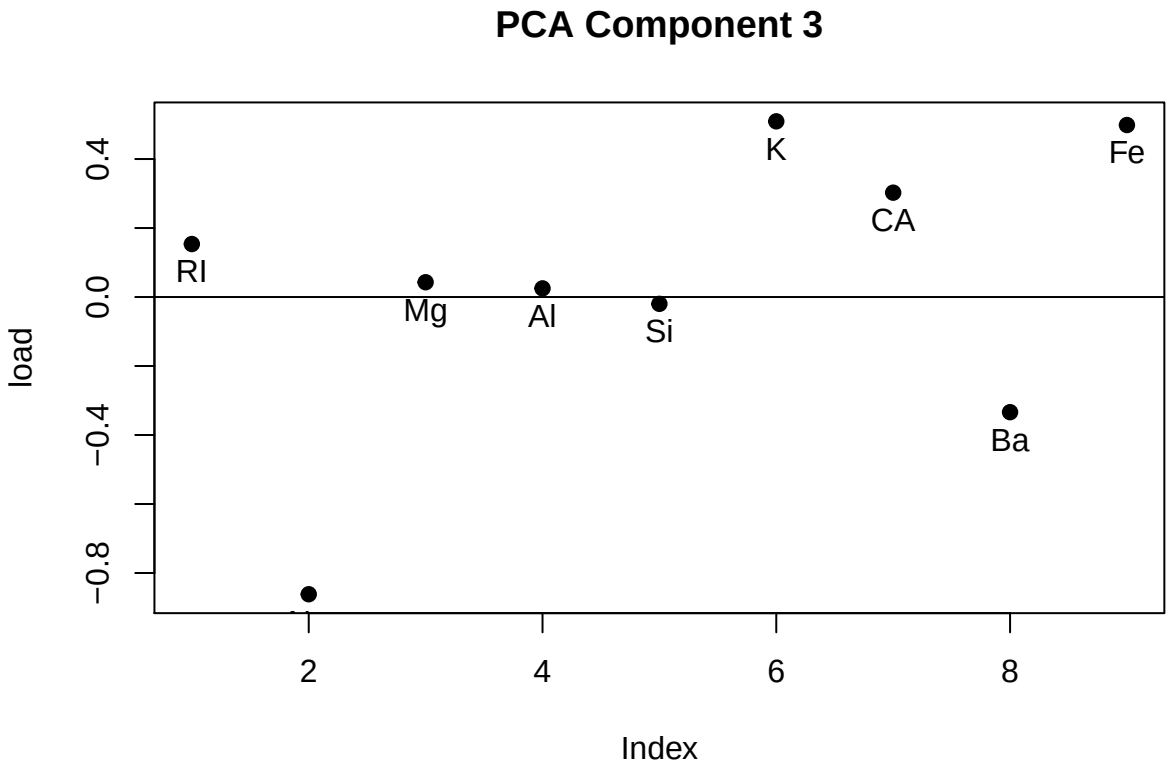
```
# Plot the components and analyze them
factor.plot(pc,
  choose = c(1),
  labels = colnames(glass_identification_data[, 2:10]),
  title = "PCA Component 1"
)
```



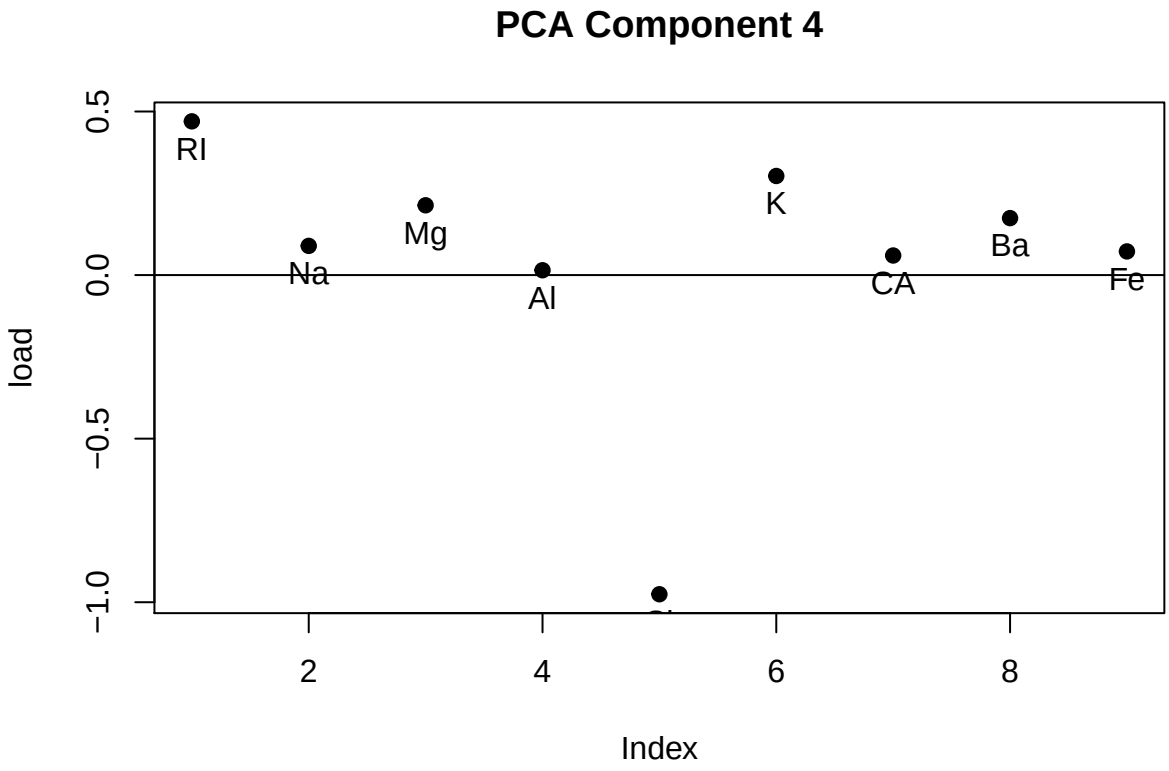
```r
factor.plot(pc,
  choose = c(2),
  labels = colnames(glass_identification_data[, 2:10]),
  title = "PCA Component 2"
)
```



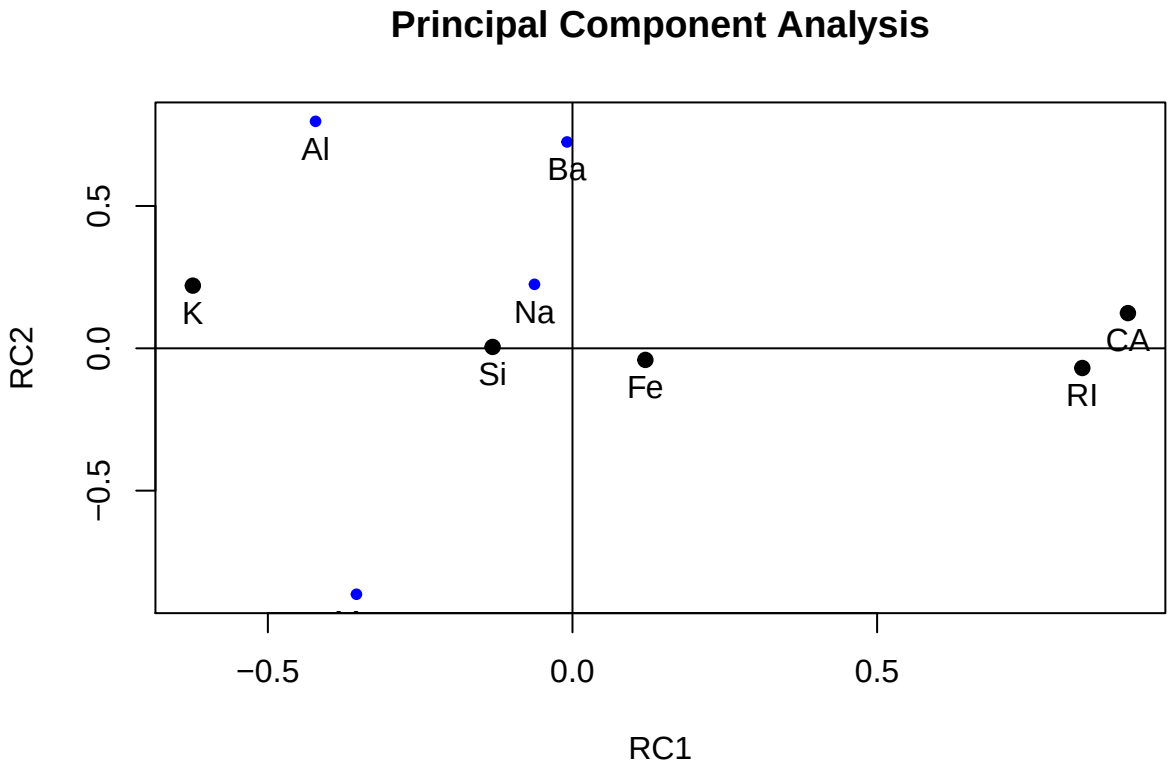

```r
factor.plot(pc,
  choose = c(3),
  labels = colnames(glass_identification_data[, 2:10]),
  title = "PCA Component 3"
)
```



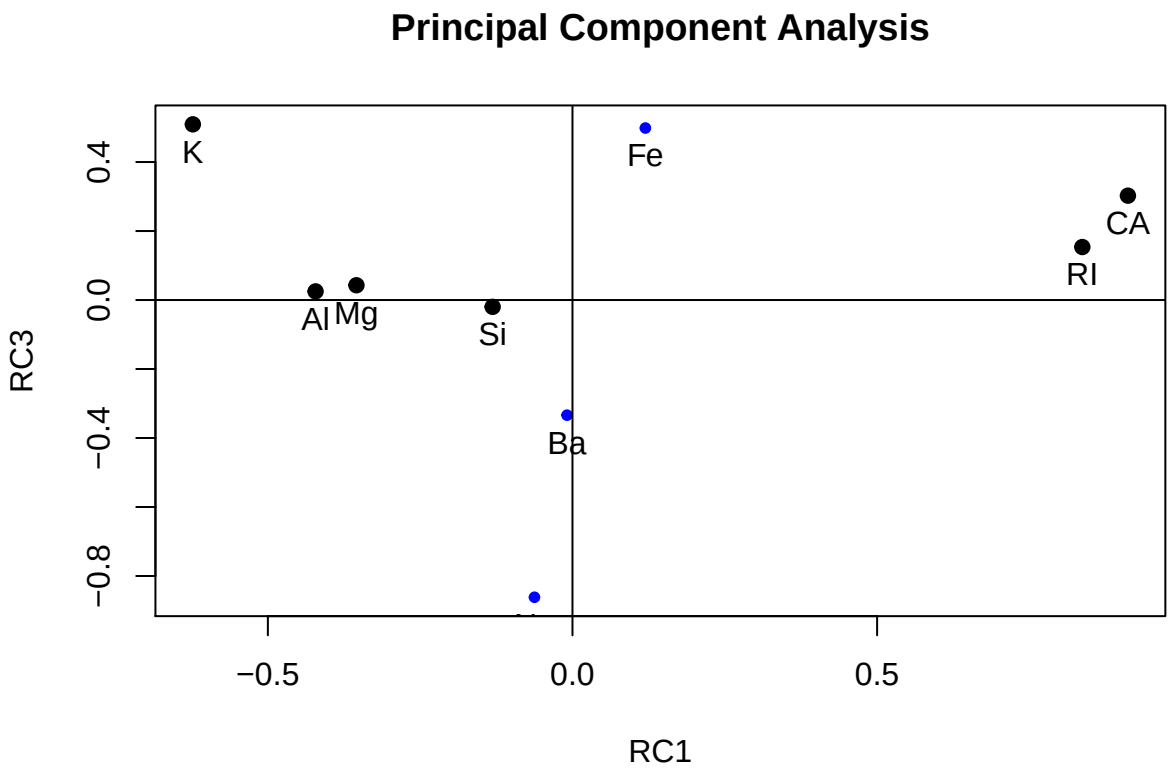

```
factor.plot(pc,
  choose = c(4),
  labels = colnames(glass_identification_data[, 2:10]),
  title = "PCA Component 4"
)
```



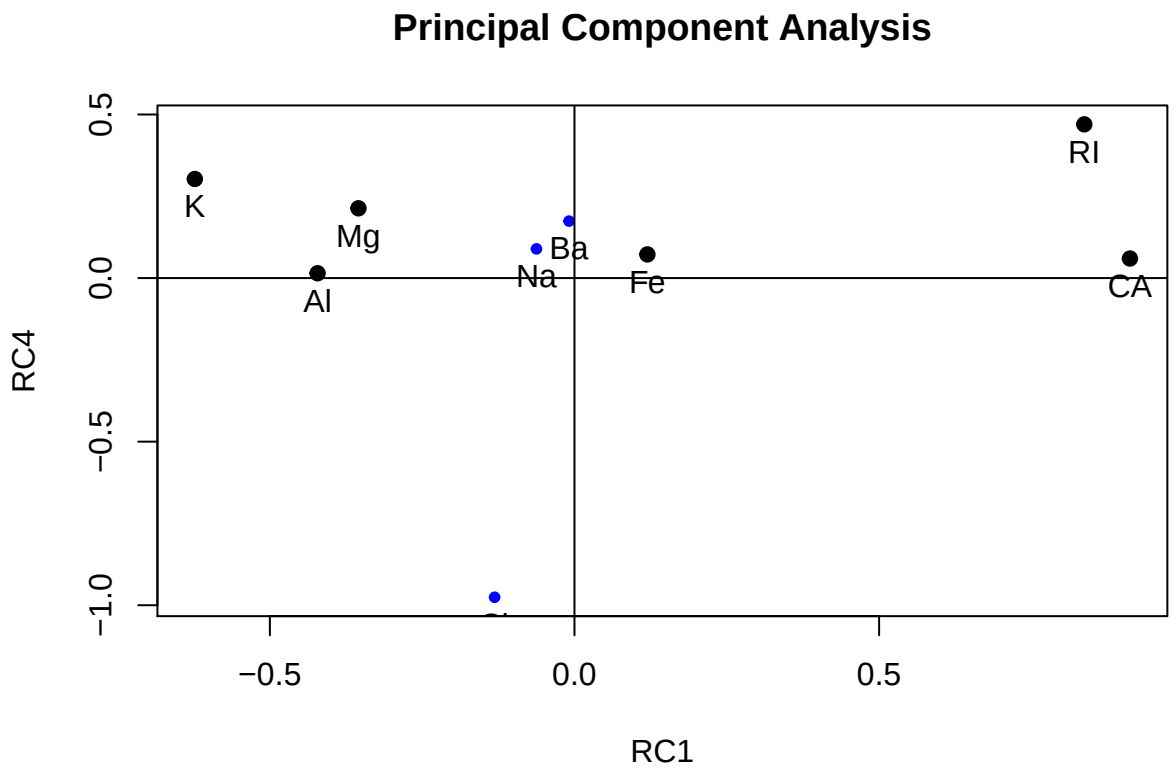

```
factor.plot(pc, choose = c(1, 2), labels = colnames(glass_identification_data[, 2:10]))
```



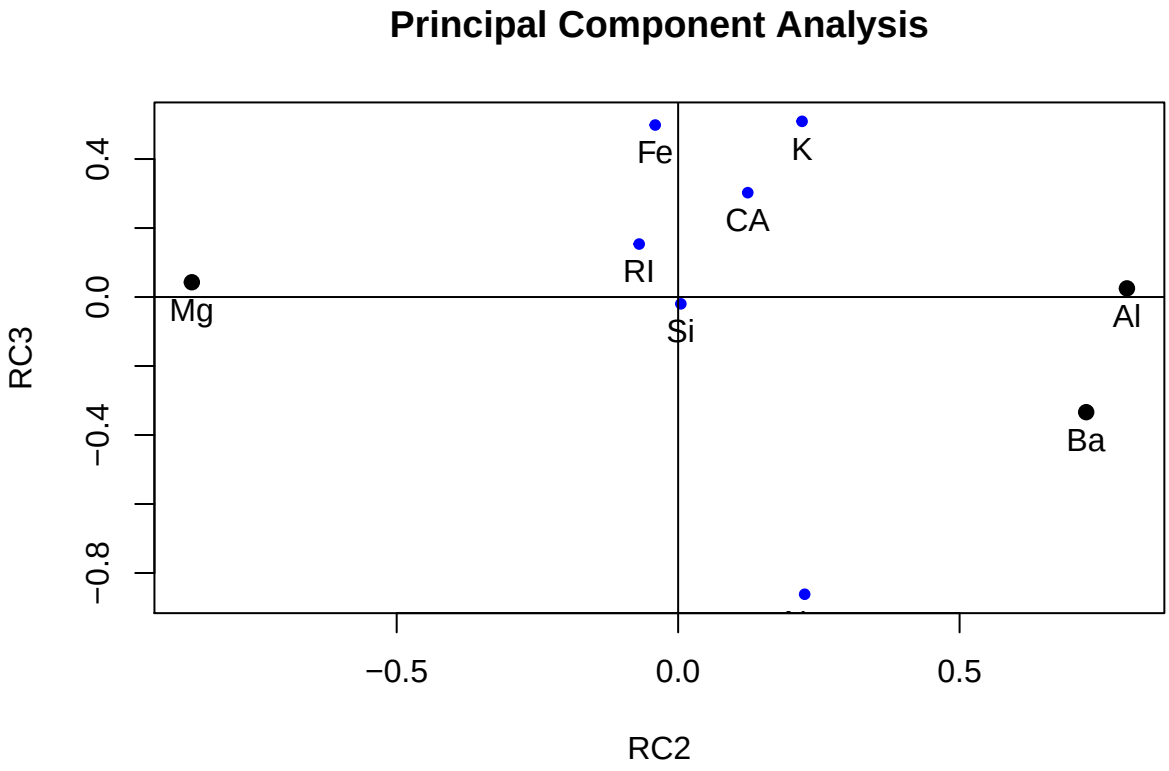

```
factor.plot(pc, choose = c(1, 3), labels = colnames(glass_identification_data[, 2:10]))
```



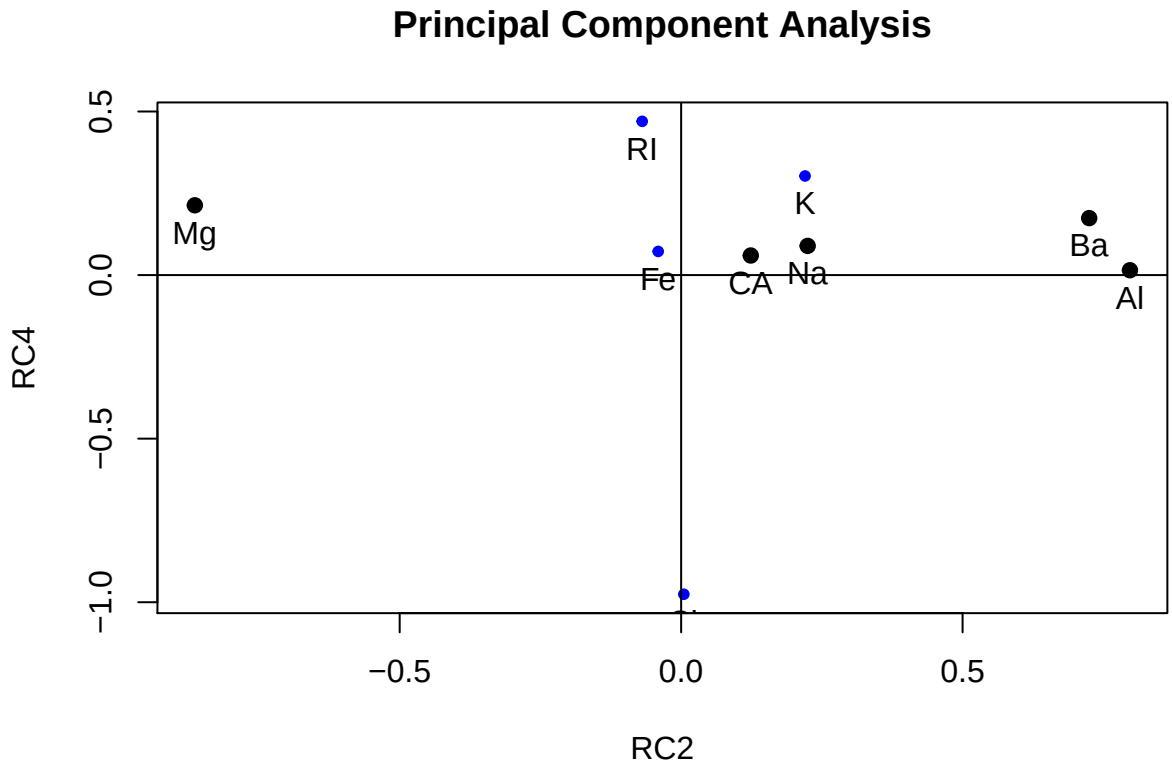

```
factor.plot(pc, choose = c(1, 4), labels = colnames(glass_identification_data[, 2:10]))
```



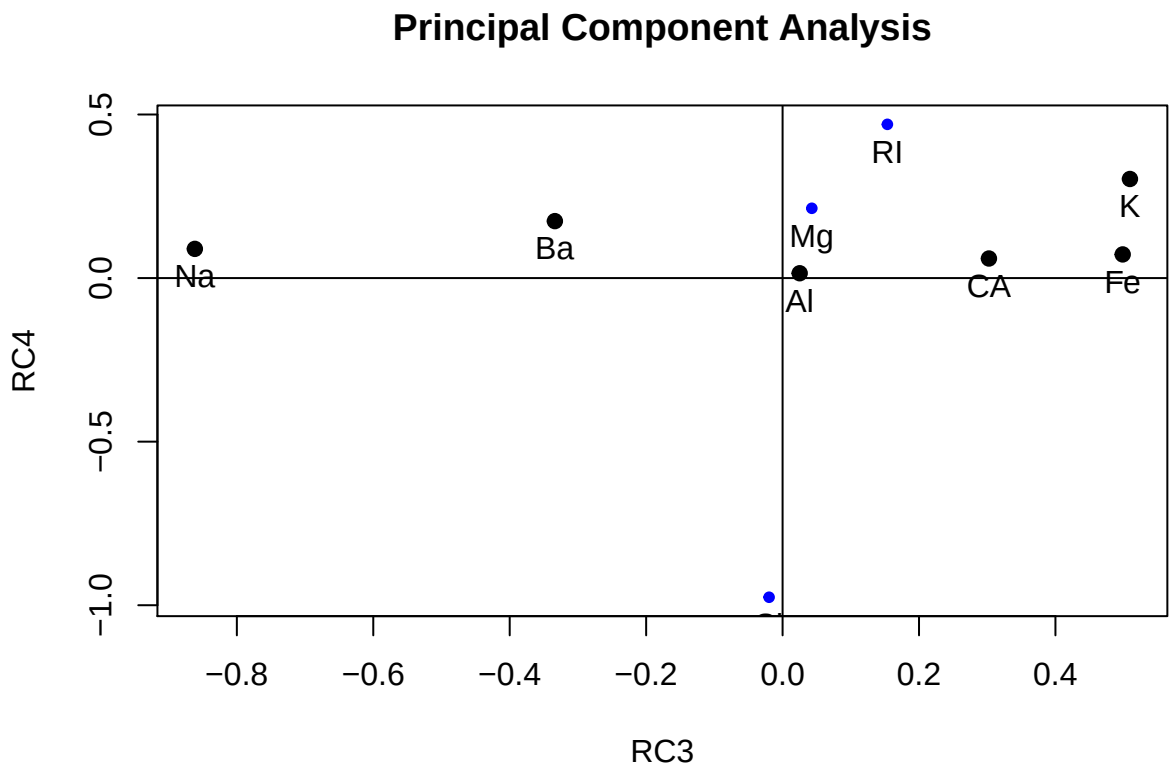

```
factor.plot(pc, choose = c(2, 3), labels = colnames(glass_identification_data[, 2:10]))
```



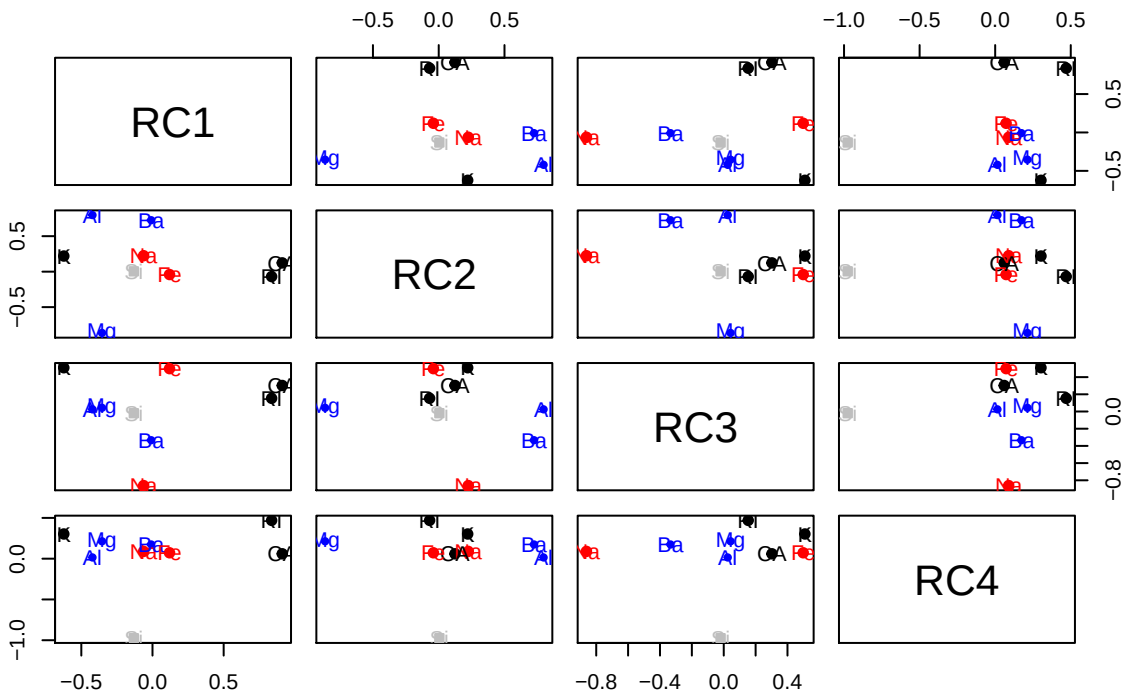

```
factor.plot(pc, choose = c(2, 4), labels = colnames(glass_identification_data[, 2:10]))
```




```
factor.plot(pc, choose = c(3, 4), labels = colnames(glass_identification_data[, 2:10]))
```




```
factor.plot(pc, labels = colnames(glass_identification_data[, 2:10]))
```




```
rm(list = ls())
```

Observations made - - PC1: loads RI, CA, Mg, Al and K.

- PC2: loads Mg, Al and Ba.
- PC3: loads Na, K, Ba and Fe.
- PC4: loads RI and Si.

Hence - 1. PC1 signifies Calcium, Potassium, Magnesium and Aluminum heavy glass. It also signifies glass with high refractive index.

2. PC2 signifies glass with high Magnesium, Aluminum and Barium concentration.
3. PC3 signifies glass with high Sodium, Potassium, Barium and Iron concentration.
4. PC4 signifies glass with high Refractive Index and Silicon concentration.

Problem 3

Perform factor analysis on Herman74.cor, which is a data structure available in the base installation (A correlation matrix of 24 psychological tests given to 145 seventh and eight-grade children in a Chicago suburb by Holzinger and Swineford).

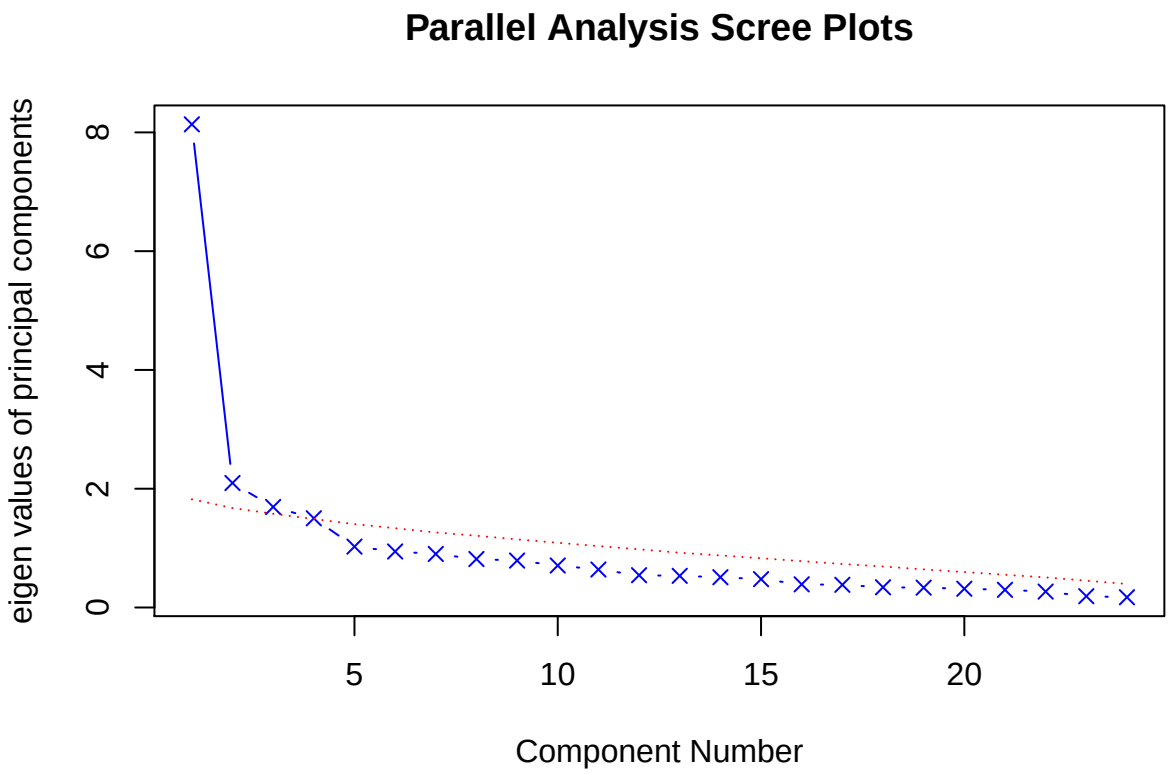
- Input the correlation matrix to fa.parallel() function to determine the number of components to extract
- Input the correlation matrix to fa() function to extract the components. If raw data is input, the correlation matrix is automatically calculated by fa() function.

- Rotate the factors
- Compute factor scores
- Graph an orthogonal solution using factor.plot()
- Graph an oblique solutions using fa.diagram()
- Interpret the results

```
"Harman75.cor" <-
  structure(list(cov = structure(c(
    1, 0.318, 0.403, 0.468, 0.321,
    0.335, 0.304, 0.332, 0.326, 0.116, 0.308, 0.314, 0.489, 0.125,
    0.238, 0.414, 0.176, 0.368, 0.27, 0.365, 0.369, 0.413, 0.474,
    0.282, 0.318, 1, 0.317, 0.23, 0.285, 0.234, 0.157, 0.157, 0.195,
    0.057, 0.15, 0.145, 0.239, 0.103, 0.131, 0.272, 0.005, 0.255,
    0.112, 0.292, 0.306, 0.232, 0.348, 0.211, 0.403, 0.317, 1, 0.305,
    0.247, 0.268, 0.223, 0.382, 0.184, -0.075, 0.091, 0.14, 0.321,
    0.177, 0.065, 0.263, 0.177, 0.211, 0.312, 0.297, 0.165, 0.25,
    0.383, 0.203, 0.468, 0.23, 0.305, 1, 0.227, 0.327, 0.335, 0.391,
    0.325, 0.099, 0.11, 0.16, 0.327, 0.066, 0.127, 0.322, 0.187,
    0.251, 0.137, 0.339, 0.349, 0.38, 0.335, 0.248, 0.321, 0.285,
    0.247, 0.227, 1, 0.622, 0.656, 0.578, 0.723, 0.311, 0.344, 0.215,
    0.344, 0.28, 0.229, 0.187, 0.208, 0.263, 0.19, 0.398, 0.318,
    0.441, 0.435, 0.42, 0.335, 0.234, 0.268, 0.327, 0.622, 1, 0.722,
    0.527, 0.714, 0.203, 0.353, 0.095, 0.309, 0.292, 0.251, 0.291,
    0.273, 0.167, 0.251, 0.435, 0.263, 0.386, 0.431, 0.433, 0.304,
    0.157, 0.223, 0.335, 0.656, 0.722, 1, 0.619, 0.685, 0.246, 0.232,
    0.181, 0.345, 0.236, 0.172, 0.18, 0.228, 0.159, 0.226, 0.451,
    0.314, 0.396, 0.405, 0.437, 0.332, 0.157, 0.382, 0.391, 0.578,
    0.527, 0.619, 1, 0.532, 0.285, 0.3, 0.271, 0.395, 0.252, 0.175,
    0.296, 0.255, 0.25, 0.274, 0.427, 0.362, 0.357, 0.501, 0.388,
    0.326, 0.195, 0.184, 0.325, 0.723, 0.714, 0.685, 0.532, 1, 0.17,
    0.28, 0.113, 0.28, 0.26, 0.248, 0.242, 0.274, 0.208, 0.274, 0.446,
    0.266, 0.483, 0.504, 0.424, 0.116, 0.057, -0.075, 0.099, 0.311,
    0.203, 0.246, 0.285, 0.17, 1, 0.484, 0.585, 0.408, 0.172, 0.154,
    0.124, 0.289, 0.317, 0.19, 0.173, 0.405, 0.16, 0.262, 0.531,
    0.308, 0.15, 0.091, 0.11, 0.344, 0.353, 0.232, 0.3, 0.28, 0.484,
    1, 0.428, 0.535, 0.35, 0.24, 0.314, 0.362, 0.35, 0.29, 0.202,
    0.399, 0.304, 0.251, 0.412, 0.314, 0.145, 0.14, 0.16, 0.215,
    0.095, 0.181, 0.271, 0.113, 0.585, 0.428, 1, 0.512, 0.131, 0.173,
    0.119, 0.278, 0.349, 0.11, 0.246, 0.355, 0.193, 0.35, 0.414,
    0.489, 0.239, 0.321, 0.327, 0.344, 0.309, 0.345, 0.395, 0.28,
    0.408, 0.535, 0.512, 1, 0.195, 0.139, 0.281, 0.194, 0.323, 0.263,
    0.241, 0.425, 0.279, 0.382, 0.358, 0.125, 0.103, 0.177, 0.066,
    0.28, 0.292, 0.236, 0.252, 0.26, 0.172, 0.35, 0.131, 0.195, 1,
    0.37, 0.412, 0.341, 0.201, 0.206, 0.302, 0.183, 0.243, 0.242,
    0.304, 0.238, 0.131, 0.065, 0.127, 0.229, 0.251, 0.172, 0.175,
    0.248, 0.154, 0.24, 0.173, 0.139, 0.37, 1, 0.325, 0.345, 0.334,
    0.192, 0.272, 0.232, 0.246, 0.256, 0.165, 0.414, 0.272, 0.263,
    0.322, 0.187, 0.291, 0.18, 0.296, 0.242, 0.124, 0.314, 0.119,
    0.281, 0.412, 0.325, 1, 0.324, 0.344, 0.258, 0.388, 0.348, 0.283,
    0.36, 0.262, 0.176, 0.005, 0.177, 0.187, 0.208, 0.273, 0.228,
    0.255, 0.274, 0.289, 0.362, 0.278, 0.194, 0.341, 0.345, 0.324,
    1, 0.448, 0.324, 0.262, 0.173, 0.273, 0.287, 0.326, 0.368, 0.255,
```

```
    0.211, 0.251, 0.263, 0.167, 0.159, 0.25, 0.208, 0.317, 0.35,
    0.349, 0.323, 0.201, 0.334, 0.344, 0.448, 1, 0.358, 0.301, 0.357,
    0.317, 0.272, 0.405, 0.27, 0.112, 0.312, 0.137, 0.19, 0.251,
    0.226, 0.274, 0.274, 0.19, 0.29, 0.11, 0.263, 0.206, 0.192, 0.258,
    0.324, 0.358, 1, 0.167, 0.331, 0.342, 0.303, 0.374, 0.365, 0.292,
    0.297, 0.339, 0.398, 0.435, 0.451, 0.427, 0.446, 0.173, 0.202,
    0.246, 0.241, 0.302, 0.272, 0.388, 0.262, 0.301, 0.167, 1, 0.413,
    0.463, 0.509, 0.366, 0.369, 0.306, 0.165, 0.349, 0.318, 0.263,
    0.314, 0.362, 0.266, 0.405, 0.399, 0.355, 0.425, 0.183, 0.232,
    0.348, 0.173, 0.357, 0.331, 0.413, 1, 0.374, 0.451, 0.448, 0.413,
    0.232, 0.25, 0.38, 0.441, 0.386, 0.396, 0.357, 0.483, 0.16, 0.304,
    0.193, 0.279, 0.243, 0.246, 0.283, 0.273, 0.317, 0.342, 0.463,
    0.374, 1, 0.503, 0.375, 0.474, 0.348, 0.383, 0.335, 0.435, 0.431,
    0.405, 0.501, 0.504, 0.262, 0.251, 0.35, 0.382, 0.242, 0.256,
    0.36, 0.287, 0.272, 0.303, 0.509, 0.451, 0.503, 1, 0.434, 0.282,
    0.211, 0.203, 0.248, 0.42, 0.433, 0.437, 0.388, 0.424, 0.531,
    0.412, 0.414, 0.358, 0.304, 0.165, 0.262, 0.326, 0.405, 0.374,
    0.366, 0.448, 0.375, 0.434, 1
  ), .Dim = c(24, 24), .Dimnames = list(
    c(
      "VisualPerception", "Cubes", "PaperFormBoard", "Flags",
      "GeneralInformation", "PargraphComprehension", "SentenceCompletion",
      "WordClassification", "WordMeaning", "Addition", "Code",
      "CountingDots", "StraightCurvedCapitals", "WordRecognition",
      "NumberRecognition", "FigureRecognition", "ObjectNumber",
      "NumberFigure", "FigureWord", "Deduction", "NumericalPuzzles",
      "ProblemReasoning", "SeriesCompletion", "ArithmeticProblems"
    ), c(
      "VisualPerception", "Cubes", "PaperFormBoard", "Flags",
      "GeneralInformation", "PargraphComprehension", "SentenceCompletion",
      "WordClassification", "WordMeaning", "Addition", "Code",
      "CountingDots", "StraightCurvedCapitals", "WordRecognition",
      "NumberRecognition", "FigureRecognition", "ObjectNumber",
      "NumberFigure", "FigureWord", "Deduction", "NumericalPuzzles",
      "ProblemReasoning", "SeriesCompletion", "ArithmeticProblems"
    )
  )), center = c(
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0
  ), n.obs = 145), .Names = c(
    "cov",
    "center", "n.obs"
  ))


fa.parallel(Harman74.cor$cov,
  n.obs = Harman74.cor$n.obs,
  fa = "pc", n.iter = 100, show.legend = FALSE
)
```




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

The Screeplot suggests 3 components.

```
# Perform Factor Analysis
correlations <- cov2cor(Harman74.cor$cov)
fa <- fa(correlations, nfactors = 3, rotate = "none", scores = TRUE, fm = "pa")

fa
```

```
## Factor Analysis using method =  pa
## Call: fa(r = correlations, nfactors = 3, rotate = "none", scores = TRUE,
##     fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                          PA1   PA2   PA3   h2   u2 com
## VisualPerception        0.59  0.03  0.37 0.49 0.51 1.7
## Cubes                   0.37 -0.03  0.26 0.21 0.79 1.8
## PaperFormBoard          0.42 -0.12  0.37 0.33 0.67 2.1
## Flags                   0.48 -0.10  0.26 0.31 0.69 1.6
## GeneralInformation      0.69 -0.30 -0.27 0.64 0.36 1.7
## PargraphComprehension   0.69 -0.40 -0.20 0.67 0.33 1.8
## SentenceCompletion      0.68 -0.41 -0.30 0.72 0.28 2.1
## WordClassification      0.68 -0.19 -0.09 0.50 0.50 1.2
## WordMeaning             0.70 -0.45 -0.22 0.74 0.26 1.9
## Addition                0.48  0.53 -0.48 0.74 0.26 3.0
## Code                    0.56  0.36 -0.17 0.47 0.53 1.9
## CountingDots            0.46  0.48 -0.13 0.47 0.53 2.1
## StraightCurvedCapitals 0.59  0.25  0.02 0.42 0.58 1.4
```

```
## WordRecognition        0.42  0.05  0.00 0.17 0.83 1.0
## NumberRecognition      0.38  0.09  0.08 0.16 0.84 1.2
## FigureRecognition      0.51  0.09  0.33 0.37 0.63 1.8
## ObjectNumber           0.46  0.19 -0.01 0.24 0.76 1.3
## NumberFigure           0.52  0.32  0.16 0.39 0.61 1.9
## FigureWord             0.44  0.10  0.10 0.22 0.78 1.2
## Deduction              0.62 -0.13  0.14 0.42 0.58 1.2
## NumericalPuzzles       0.59  0.22  0.08 0.41 0.59 1.3
## ProblemReasoning       0.61 -0.10  0.13 0.40 0.60 1.1
## SeriesCompletion       0.69 -0.06  0.15 0.50 0.50 1.1
## ArithmeticProblems     0.65  0.18 -0.19 0.49 0.51 1.3
##
##                        PA1  PA2  PA3
## SS loadings           7.61 1.66 1.19
## Proportion Var        0.32 0.07 0.05
## Cumulative Var        0.32 0.39 0.44
## Proportion Explained  0.73 0.16 0.11
## Cumulative Proportion 0.73 0.89 1.00
##
## Mean item complexity =  1.6
## Test of the hypothesis that 3 factors are sufficient.
##
## The degrees of freedom for the null model are  276  and the objective function was  11.44
## The degrees of freedom for the model are 207  and the objective function was  2.24
##
## The root mean square of the residuals (RMSR) is  0.05
## The df corrected root mean square of the residuals is  0.06
##
## Fit based upon off diagonal values = 0.97
## Measures of factor score adequacy
##                                                 PA1  PA2  PA3
## Correlation of (regression) scores with factors   0.97 0.90 0.86
## Multiple R square of scores with factors          0.94 0.82 0.74
## Minimum correlation of possible factor scores     0.88 0.63 0.47
```

Rotate the factors

```
fa_orthogonal <- fa(correlations, nfactors = 3, rotate = "varimax", scores = TRUE, fm = "pa")

fa_orthogonal
```

```
## Factor Analysis using method =  pa
## Call: fa(r = correlations, nfactors = 3, rotate = "varimax", scores = TRUE,
##     fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                         PA1   PA3  PA2   h2   u2 com
## VisualPerception       0.17  0.65 0.22 0.49 0.51 1.4
## Cubes                  0.12  0.43 0.09 0.21 0.79 1.2
## PaperFormBoard         0.16  0.55 0.01 0.33 0.67 1.2
## Flags                  0.24  0.49 0.09 0.31 0.69 1.5
## GeneralInformation     0.74  0.19 0.24 0.64 0.36 1.4
## PargraphComprehension  0.77  0.25 0.14 0.67 0.33 1.3
## SentenceCompletion     0.82  0.17 0.16 0.72 0.28 1.2
## WordClassification     0.57  0.33 0.25 0.50 0.50 2.0
## WordMeaning            0.82  0.24 0.12 0.74 0.26 1.2
```

```
## Addition               0.17 -0.13 0.83 0.74 0.26 1.1
## Code                   0.18  0.18 0.63 0.47 0.53 1.3
## CountingDots           0.03  0.15 0.67 0.47 0.53 1.1
## StraightCurvedCapitals 0.19  0.35 0.51 0.42 0.58 2.1
## WordRecognition        0.22  0.24 0.26 0.17 0.83 2.9
## NumberRecognition      0.14  0.29 0.25 0.16 0.84 2.4
## FigureRecognition      0.10  0.56 0.22 0.37 0.63 1.4
## ObjectNumber           0.16  0.25 0.39 0.24 0.76 2.1
## NumberFigure           0.04  0.42 0.47 0.39 0.61 2.0
## FigureWord             0.16  0.34 0.28 0.22 0.78 2.4
## Deduction              0.39  0.48 0.18 0.42 0.58 2.2
## NumericalPuzzles       0.19  0.40 0.46 0.41 0.59 2.3
## ProblemReasoning       0.38  0.46 0.21 0.40 0.60 2.4
## SeriesCompletion       0.39  0.53 0.27 0.50 0.50 2.4
## ArithmeticProblems     0.37  0.22 0.55 0.49 0.51 2.1
##
##                        PA1  PA3  PA2
## SS loadings           3.76 3.37 3.34
## Proportion Var        0.16 0.14 0.14
## Cumulative Var        0.16 0.30 0.44
## Proportion Explained  0.36 0.32 0.32
## Cumulative Proportion 0.36 0.68 1.00
##
## Mean item complexity =  1.8
## Test of the hypothesis that 3 factors are sufficient.
##
## The degrees of freedom for the null model are  276  and the objective function was  11.44
## The degrees of freedom for the model are 207  and the objective function was  2.24
##
## The root mean square of the residuals (RMSR) is  0.05
## The df corrected root mean square of the residuals is  0.06
##
## Fit based upon off diagonal values = 0.97
## Measures of factor score adequacy
##                                                 PA1  PA3  PA2
## Correlation of (regression) scores with factors   0.93 0.88 0.92
## Multiple R square of scores with factors          0.87 0.78 0.84
## Minimum correlation of possible factor scores     0.74 0.56 0.68
```

Let's see the factor scores

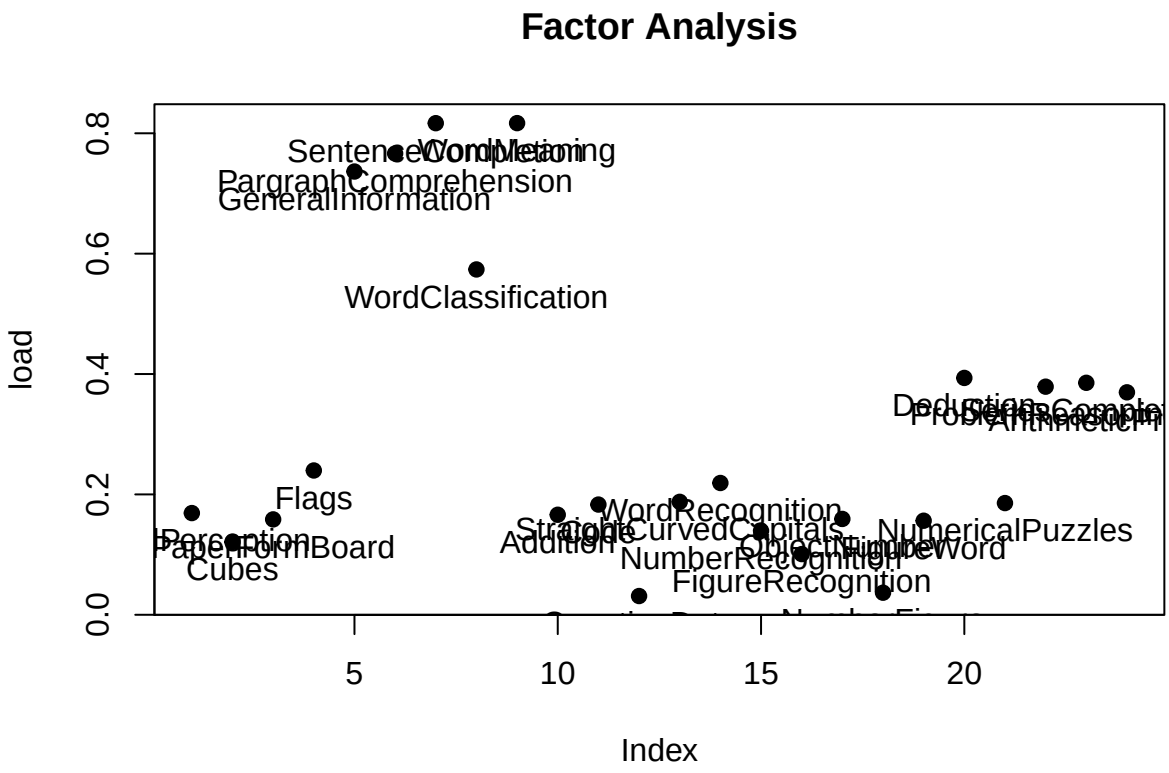
```
fa_orthogonal$weights
```

```
##                                PA1          PA3          PA2
## VisualPerception       -0.0846470536  0.240223872 -0.004751909
## Cubes                  -0.0278956779  0.100696352 -0.023482813
## PaperFormBoard         -0.0017273482  0.134714744 -0.022115974
## Flags                  -0.0175241036  0.125968576 -0.041332014
## GeneralInformation      0.1842500394 -0.049058198 -0.020538486
## PargraphComprehension   0.2134216935  0.003031936 -0.073641718
## SentenceCompletion      0.3489145257 -0.143376378 -0.035869541
## WordClassification      0.0705638354  0.053735991 -0.013437915
## WordMeaning             0.3531881175 -0.095006703 -0.058110558
## Addition                0.0200972184 -0.342238593  0.522526392
## Code                   -0.0212974198 -0.034091534  0.192656220
```

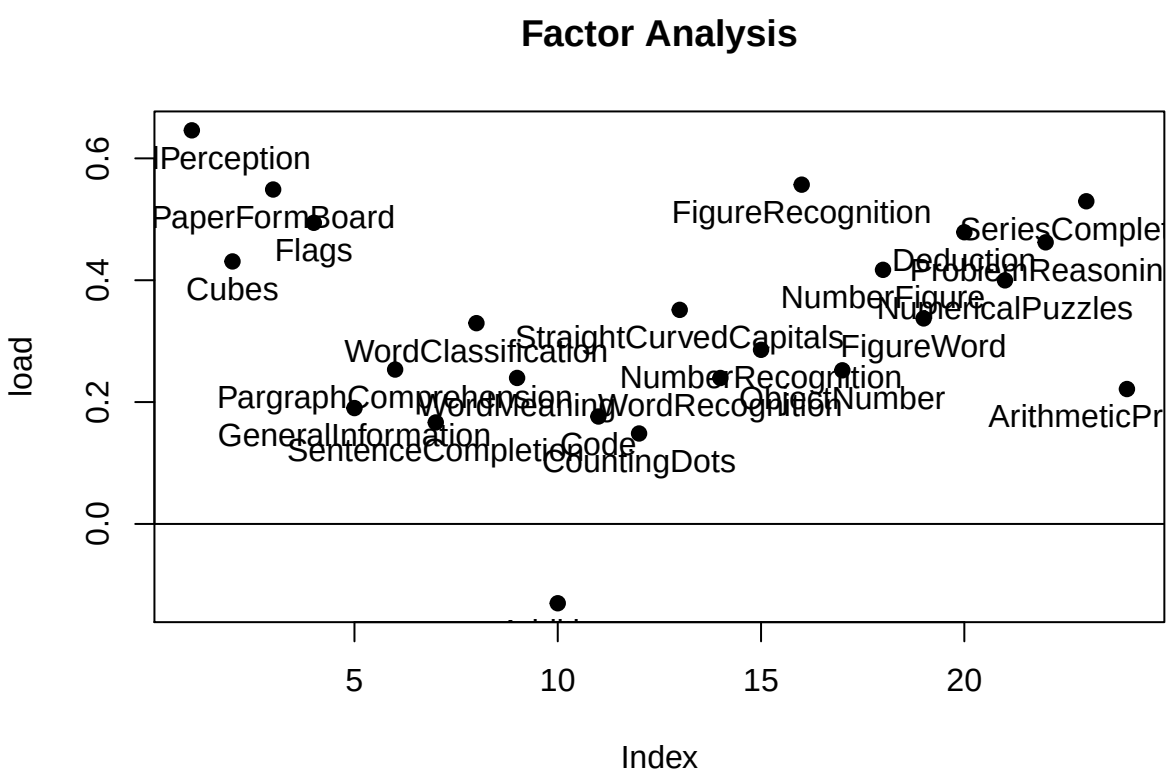
```
## CountingDots           -0.0467066495 -0.003803166  0.155392202
## StraightCurvedCapitals -0.0558212207  0.074210513  0.084854174
## WordRecognition         0.0007735856  0.029129971  0.035486962
## NumberRecognition      -0.0208541449  0.040004770  0.039872882
## FigureRecognition      -0.0453682462  0.165543229  0.007761110
## ObjectNumber           -0.0306113738  0.040899278  0.058430068
## NumberFigure           -0.0723295126  0.116287610  0.099064848
## FigureWord             -0.0276547073  0.060358621  0.028981877
## Deduction               0.0081733280  0.119615562 -0.019425087
## NumericalPuzzles       -0.0262836134  0.098265777  0.057557882
## ProblemReasoning        0.0160245946  0.098599087  0.001749052
## SeriesCompletion        0.0080520487  0.159385045 -0.011183038
## ArithmeticProblems      0.0060559812  0.010231354  0.104895879
```

Plotting an orthogonal solution

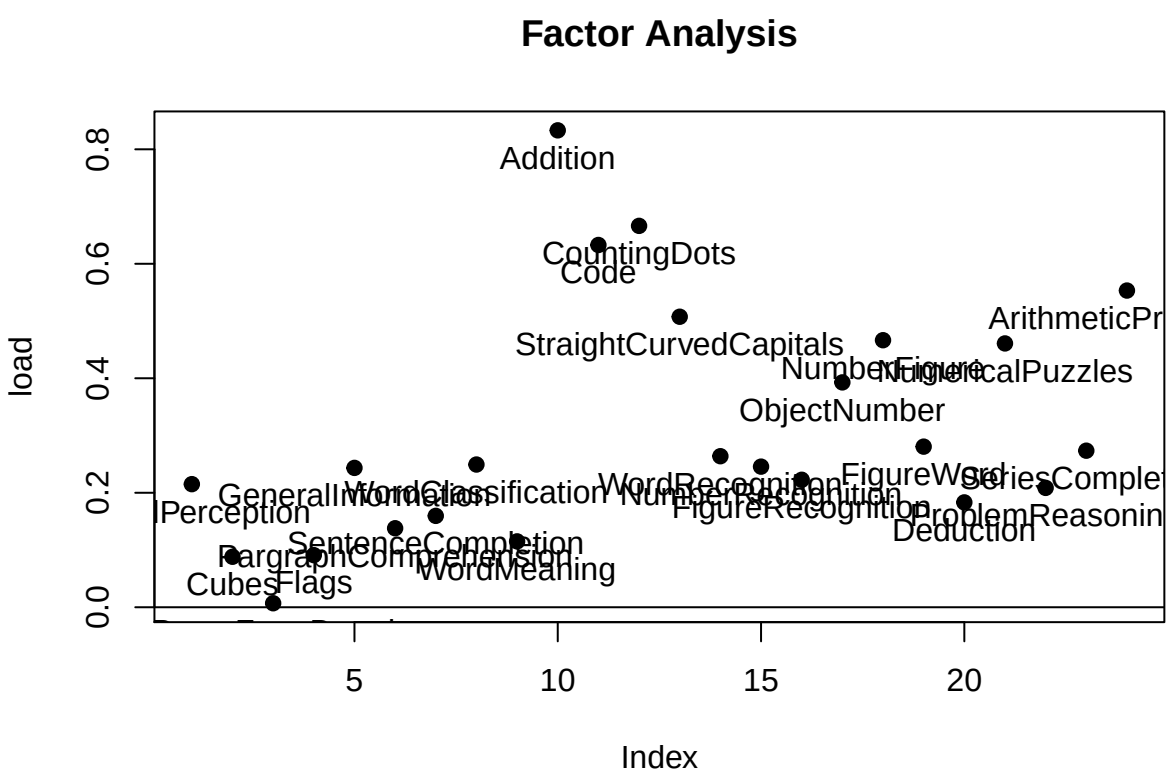
```
factor.plot(fa_orthogonal, choose = c(1), labels = rownames(fa$loadings))
```



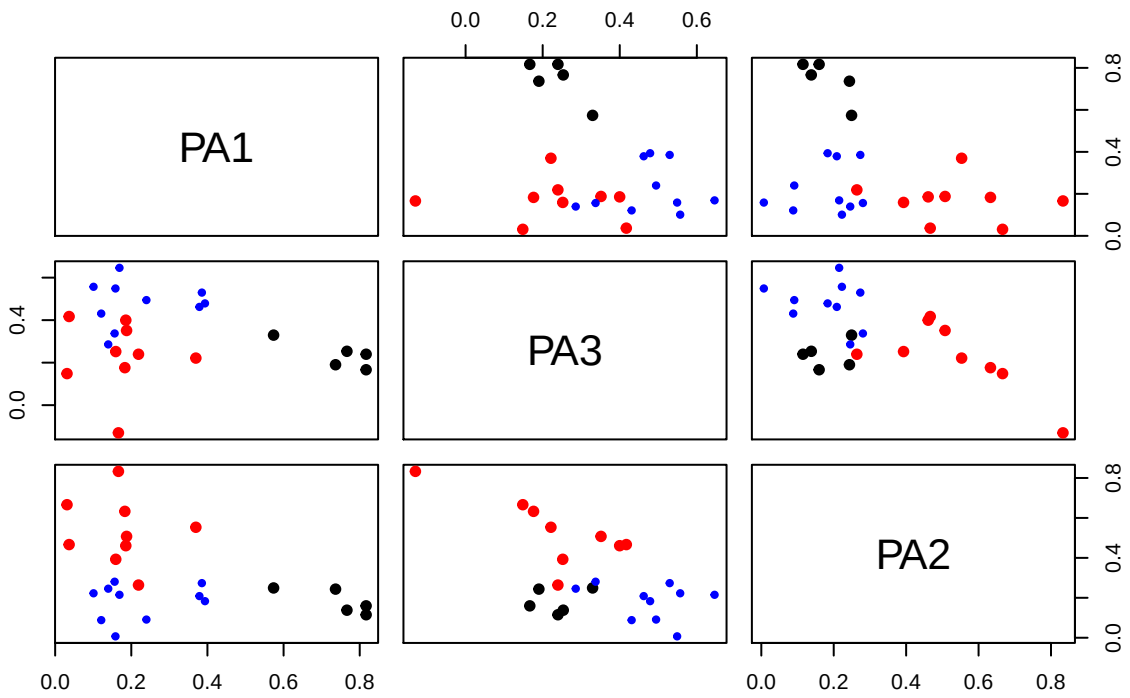

```
factor.plot(fa_orthogonal, choose = c(2), labels = rownames(fa$loadings))
```




```
factor.plot(fa_orthogonal, choose = c(3), labels = rownames(fa$loadings))
```




```
factor.plot(fa_orthogonal)
```

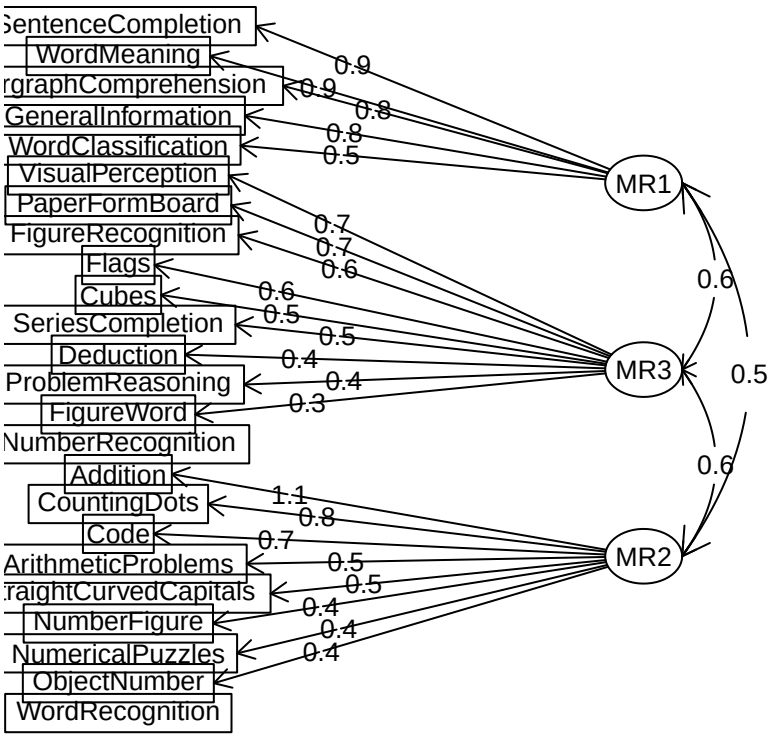


Now let's plot an oblique solution

```
fa_oblique <- fa(Harman74.cor$cov, nfactors = 3, rotate = "promax", scores = TRUE)
```

```
## Loading required namespace: GPArotation
```

```
fa.diagram(fa_oblique)
```




```
rm(list = ls())
```

We can make the following observations -

- Component 1 seems to be related to language related attributes like Sentence Completion, WordMeaning, Word Classification etc.
- Component 2 seems to be related to more general problem solving attributes like Deduction, Problem Reasoning, Cubes, Flage etc.
- Component 3 seems to be related to mathematical attributes like Code, Counting Dota, Addition, Number Recognition etc.
