Social Media

Yuefei Chen

2024-03-29

Loading the Dataset

In the dataset of "Social_media_cleaned.csv", the data is cleaned and every time stamp data has been transformed into numeric value data. These values are in the new columns "XXX_value". XXX means the Apps we use. Mean value of the data is also a kind of numeric data, which are shown in the last line. Additionally, N/A value has been replaced by 0.00. The point of my mean value of social media is in the line 23.

```r
library(readr)
APP_data <- read_csv("Social Media_cleaned.csv")
```

```
## New names:
## Rows: 23 Columns: 33
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (15): ID, Instagram, Linkedin, Snapchat, Twitter, Whatsapp/ Wechat, You... dbl
## (12): Instagram_value, Linkedin_value, Snapchat_value, Twitter_value, W... time
## (6): Hours spent...3, Hours spent...6, Hours spent...9, Hours spent......
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `Hours spent` -> `Hours spent...3`
## * `Hours spent` -> `Hours spent...6`
## * `Hours spent` -> `Hours spent...9`
## * `Hours spent` -> `Hours spent...12`
## * `Hours spent` -> `Hours spent...15`
## * `Hours spent` -> `Hours spent...18`
## * `Hours spent` -> `Hours spent...21`
## * `Hours spent` -> `Hours spent...24`
```

```r
APP_data <- APP_data[c(1:22), c(1:2, 4:5, 7:8, 10:11, 13:14, 16:17, 19:20, 22:23, 25:33)]
str(APP_data)
```

```
## tibble [22 x 25] (S3: tbl_df/tbl/data.frame)
##  $ ID                                                    : chr [1:22] "masinl" "peace" "Patty" "Bunny"
##  $ Instagram                                             : chr [1:22] "Yes" "Yes" "Yes" "Yes" ...
##  $ Instagram_value                                       : num [1:22] 3.5 7.73 3.77 5.38 0 2.33 5.37 7
##  $ Linkedin                                              : chr [1:22] "Yes" "Yes" "Yes" "Yes" ...
##  $ Linkedin_value                                        : num [1:22] 4 5.2 7 5.32 0.58 7 4 4 10 0 ...
##  $ Snapchat                                              : chr [1:22] "Yes" "Yes" "Yes" "Yes" ...
##  $ Snapchat_value                                        : num [1:22] 1 3.68 0.53 1.3 0 0.47 0 3 3.83
```

```
##  $ Twitter                                              : chr [1:22] "Yes" "No" "No" "No" ...
##  $ Twitter_value                                        : num [1:22] 5 0 0 0 0.67 0 0 0 0 0 ...
##  $ Whatsapp/ Wechat                                     : chr [1:22] "Yes" "Yes" "Yes" "Yes" ...
##  $ Whatsapp/ Wechat_value                               : num [1:22] 1 4.18 9.83 5.3 3 12 6 10 6.15 1
##  $ Youtube                                              : chr [1:22] "Yes" "Yes" "Yes" "Yes" ...
##  $ Youtube_value                                        : num [1:22] 2.5 4.25 1.85 2 3.5 7 3 2 4 3 ..
##  $ OTT (Netflix, Hulu, Prime video)                     : chr [1:22] "Yes" "No" "Yes" "Yes" ...
##  $ OTT (Netflix, Hulu, Prime video)_value               : num [1:22] 14.5 0 2 2 2 3 0 3 3 0 ...
##  $ Reddit                                               : chr [1:22] "Yes" "No" "No" "No" ...
##  $ Reddit_value                                         : num [1:22] 2.5 0 0 0 1 0 0 0 0 0 ...
##  $ Application type(Social media, OTT, Learning)        : chr [1:22] "OTT" "Social Media" "Social Medi
##  $ How many job interview calls received in this week.?: num [1:22] 0 0 0 2 0 0 0 0 1 0 ...
##  $ How much networking done with coffee chats?          : num [1:22] 0 1 0 0 2 0 2 0 0 0 ...
##  $ How many learning done in terms of items created?   : num [1:22] 3 3 4 4 4 4 3 2 6 2 ...
##  $ Mood Productivity                                    : chr [1:22] "Yes" "Yes" "Yes" "Yes" ...
##  $ Tired waking up in morning                           : chr [1:22] "No" "No" "No" "No" ...
##  $ Trouble falling asleep                               : chr [1:22] "No" "Yes" "No" "No" ...
##  $ How you felt the entire week?                        : num [1:22] 3 3 4 4 3 5 4 4 3 2 ...
```

Part I: Calculate the MVA distance of your social media usage and the class average

```r
MVA_data <- APP_data[c(1:22), c(3,5,7,9,11,13,15,17,19,20,21,25)]
cov_matrix <- cov(MVA_data)
mean_vector <- colMeans(MVA_data)
mahalanobis_distances <- mahalanobis(MVA_data, center = mean_vector, cov = cov_matrix)
mahalanobis_distances[22]
```

```
## [1] 7.126339
```

The MVA distance of my social media usage and the class average is 7.126339.

Part II: Social Media Data - Midterm Prep

Summary and Takeaway In the PCA model tells us these social media usage variables can be transformed into three components. Since after component 3 point, the curve decreasing becomes slow, and additionally, only first 3 components' variance are larger than 1, 3 principal components will be selected as PCA analysis model. When we test whether PCs will affect the "Tired waking up in morning". The p-value shows the hypothesis is not significant, so we cannot conclude these usage will affect classmates feeling when waking up in morning. In this cluster analysis, since the dataset is not large and we do not know how many cluster we need. Hierarchical cluster analysis will be used in this model. These points will be clustered into two clusters. The cluster one is {9, 15, 20}. The cluster 2 is {1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 16, 17, 18, 19, 21, 22} In this factor analysis model, four factors are ideal for the dataset. That is because from the scree plot there are significant decrease of the line before the factor is 4. After factor = 4, the change of the line is not significant. And after factor = 4, the data point is under the eigenvalue line. Additionally, from the chart of Very Simple Structure, factor = 4 line has good performance in fit. In component analysis, the factor loading between PC1 and Instagram, Snapchat, WhatsApp/Wechat are 0.9, 0.8, 0.6. The factor loading between PC2 and Twitter, OTT are 0.9, 0.7. The factor loading between PC3 and Linkedin, Youtube are 0.8, 0.8. The factor loading between PC4 and Reddit is 1.

PCA Analysis

```
APP_pca <- prcomp(MVA_data[,-c(9:12)],scale=TRUE)
APP_pca
```

```
## Standard deviations (1, .., p=8):
## [1] 1.6689580 1.3514365 1.0162846 0.9242447 0.8374943 0.6433195 0.5412065
## [8] 0.3049175
##
## Rotation (n x k) = (8 x 8):
##                                              PC1         PC2         PC3
## Instagram_value                       0.49725527  0.02316484 -0.33976112
## Linkedin_value                        0.34780303  0.11301260  0.44613172
## Snapchat_value                        0.47020393  0.21122319 -0.27891701
## Twitter_value                        -0.22616734  0.60007954 -0.12431651
## Whatsapp/ Wechat_value                0.43020230 -0.25189162 -0.05736183
## Youtube_value                         0.35828706 -0.02738113  0.56626815
## OTT (Netflix, Hulu, Prime video)_value  0.20262681  0.63223685 -0.12732923
## Reddit_value                         -0.07089142  0.34359615  0.50211237
##                                              PC4         PC5         PC6
## Instagram_value                      -0.13120229 -0.05305024  0.31084478
## Linkedin_value                        0.48390734 -0.45913621 -0.41859226
## Snapchat_value                       -0.07052415 -0.39487038  0.25247196
## Twitter_value                         0.26237627  0.28036296  0.02516454
## Whatsapp/ Wechat_value               -0.29690816  0.44478593 -0.57838795
## Youtube_value                         0.18797492  0.48942132  0.50833975
## OTT (Netflix, Hulu, Prime video)_value -0.01385949  0.26355504 -0.26523569
## Reddit_value                         -0.74237697 -0.21218998  0.02210878
##                                              PC7         PC8
## Instagram_value                      -0.65599734  0.29962912
## Linkedin_value                       -0.21662083  0.01054774
## Snapchat_value                        0.45936501 -0.46994016
## Twitter_value                        -0.38834971 -0.52383666
## Whatsapp/ Wechat_value               -0.06427857 -0.35147243
## Youtube_value                         0.12585533 -0.03323202
## OTT (Netflix, Hulu, Prime video)_value  0.34194385  0.53486194
## Reddit_value                         -0.15496549 -0.06440494
```

```
summary(APP_pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     1.6690 1.3514 1.0163 0.9242 0.83749 0.64332 0.54121
## Proportion of Variance 0.3482 0.2283 0.1291 0.1068 0.08767 0.05173 0.03661
## Cumulative Proportion  0.3482 0.5765 0.7056 0.8124 0.90003 0.95177 0.98838
##                            PC8
## Standard deviation     0.30492
## Proportion of Variance 0.01162
## Cumulative Proportion  1.00000
```

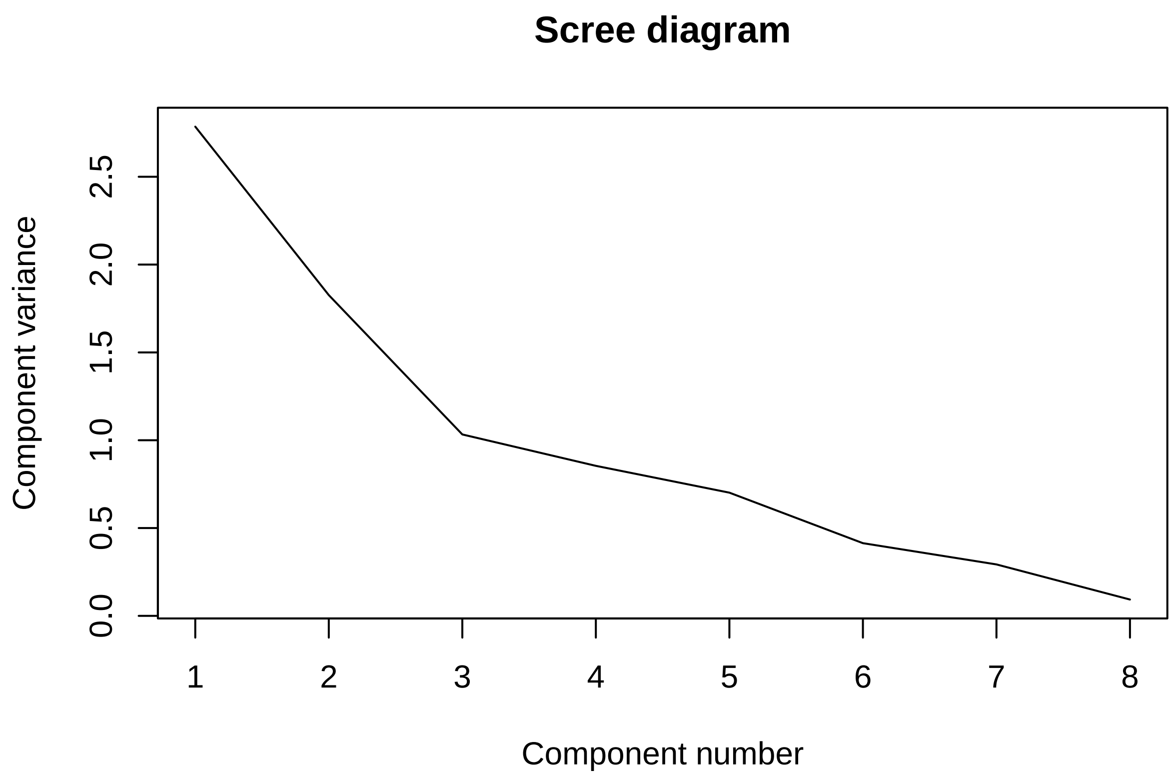
```r
(eigen_rent <- APP_pca$sdev^2)
```

```
## [1] 2.7854209 1.8263807 1.0328343 0.8542282 0.7013967 0.4138600 0.2929045
## [8] 0.0929747
```

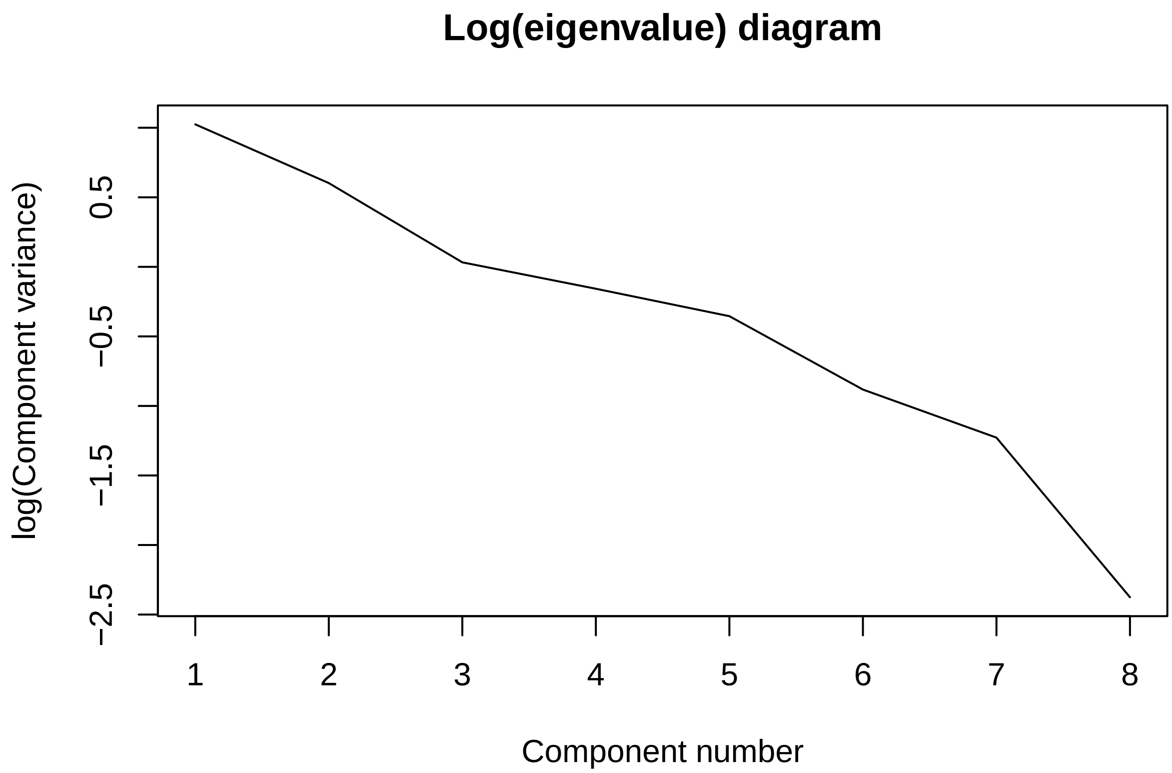
```r
names(eigen_rent) <- paste("PC",1:8,sep="")
eigen_rent
```

```
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7       PC8
## 2.7854209 1.8263807 1.0328343 0.8542282 0.7013967 0.4138600 0.2929045 0.0929747
```

```r
plot(eigen_rent, xlab = "Component number", ylab = "Component variance", type = "l", main = "Scree diag
```




```r
plot(log(eigen_rent), xlab = "Component number",ylab = "log(Component variance)", type="l",main = "Log(
```




```
APP_pca_id <- cbind(APP_data[1:22,23],APP_pca$x)
APP_pca_id
```

```
##    Tired waking up in morning          PC1         PC2         PC3         PC4
## 1                          No -1.160796114  5.10850988 -0.01765301  0.44987868
## 2                          No  1.189991321 -0.27032404  0.11437308  0.59539705
## 3                          No  0.251872718 -0.52941816  0.38051959  0.54798468
## 4                          No  0.005125007 -0.23024082 -0.09320668  0.46694531
## 5                         Yes -1.885435804  0.01644305  0.56189276 -0.22205566
## 6                          No  1.389623882 -0.58010836  2.16883299  1.00996887
## 7                         Yes -0.358636386 -0.86638173  0.26247798  0.32384476
## 8                         Yes  1.048381721 -0.18713328 -0.86167999 -0.26742605
## 9                          No  2.375553276  0.40360635  0.65694128  1.32382762
## 10                         No -2.208571481 -0.78008578  0.11808967  0.09202957
## 11                         No -2.004650434  0.57727275 -0.89513570  0.61072236
## 12                         No -1.549782025  0.85434716 -0.32579181  0.78618795
## 13                        Yes -1.270901522 -0.63449430 -1.06865108 -0.37322236
## 14                         No -0.643032069  0.03120748 -0.40227259  0.39718339
## 15                         No -0.254556930  1.14479617  2.29097492 -3.12095953
## 16                        Yes  0.479930110 -0.43837591  1.01792290  0.49244984
## 17                         No  0.760126075 -1.01076000 -0.77790683 -0.90439450
## 18                        Yes -0.174319520 -0.30765384 -1.38117347 -0.53674132
## 19                         No  0.429273173 -1.08153502  0.88812578  0.32133139
## 20                         No  4.969803179  1.26514408 -1.62136884 -0.82150402
## 21                        Yes -1.965687543 -1.15964857 -0.89777737 -0.67388653
```

```
## 22                         No  0.576689366 -1.32516709 -0.11753358 -0.49756150
##            PC5          PC6         PC7         PC8
## 1   0.87444225 -0.459138482 -0.03114110  0.34531812
## 2  -1.02613144  1.137556191  0.17446702 -0.34044347
## 3  -0.53304520 -1.634446415 -0.09967347  0.01485383
## 4  -0.85917404 -0.409206191  0.01939590  0.33198182
## 5   0.59330119  0.528304805  0.98076480  0.02928655
## 6   1.28490890 -0.627021790  0.60394544 -0.22769378
## 7  -0.11469227 -0.016001352 -0.32798526  0.27909785
## 8  -0.42513265 -0.534718425  0.28521062 -0.22530780
## 9  -1.60713181 -0.160907067 -0.14419934  0.01870842
## 10  0.17671427  0.901309783  1.10290662  0.23405182
## 11 -0.42006534  0.450581482 -0.71436962 -0.44308529
## 12  0.47173599  0.223902115 -0.44047366 -0.72441318
## 13 -0.68190030 -0.004391436 -0.05727361  0.40165547
## 14 -0.28139428  0.512023692 -0.30934502  0.03847819
## 15 -1.08609776  0.082366759 -0.31083274 -0.19665184
## 16  0.05035790  0.685179351 -0.41095939  0.49604849
## 17  1.55290671 -0.065061918 -0.88799155  0.16042926
## 18 -0.49574406 -0.241362634  0.12659035  0.24682089
## 19  0.82636382  0.251869222 -0.26615937  0.01340769
## 20  0.52503202  0.464564697  0.69258845 -0.09744000
## 21  0.01847379 -1.103176307  0.69009580 -0.32310888
## 22  1.15627232  0.017773920 -0.67556088 -0.03199417
```

```
var.test(PC1~APP_data$`Tired waking up in morning`,data=APP_pca_id)
```

```
##
##  F test to compare two variances
##
## data:  PC1 by APP_data$`Tired waking up in morning`
## F = 2.4813, num df = 14, denom df = 6, p-value = 0.2702
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.4684457 8.6878274
## sample estimates:
## ratio of variances
##           2.481269
```

```
var.test(PC2~APP_data$`Tired waking up in morning`,data=APP_pca_id)
```

```
##
##  F test to compare two variances
##
## data:  PC2 by APP_data$`Tired waking up in morning`
## F = 14.983, num df = 14, denom df = 6, p-value = 0.003185
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##   2.828689 52.461062
## sample estimates:
## ratio of variances
##           14.98303
```

```r
var.test(PC3~APP_data$`Tired waking up in morning`,data=APP_pca_id)
```

```
##
##  F test to compare two variances
##
## data:  PC3 by APP_data$`Tired waking up in morning`
## F = 1.2563, num df = 14, denom df = 6, p-value = 0.8231
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.2371736 4.3986382
## sample estimates:
## ratio of variances
##           1.256264
```

```r
var.test(PC4~APP_data$`Tired waking up in morning`,data=APP_pca_id)
```

```
##
##  F test to compare two variances
##
## data:  PC4 by APP_data$`Tired waking up in morning`
## F = 6.2852, num df = 14, denom df = 6, p-value = 0.03264
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##   1.186601 22.006791
## sample estimates:
## ratio of variances
##           6.285203
```

```r
var.test(PC5~APP_data$`Tired waking up in morning`,data=APP_pca_id)
```

```
##
##  F test to compare two variances
##
## data:  PC5 by APP_data$`Tired waking up in morning`
## F = 5.2171, num df = 14, denom df = 6, p-value = 0.05189
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##   0.984942 18.266803
## sample estimates:
## ratio of variances
##           5.217052
```

```r
var.test(PC6~APP_data$`Tired waking up in morning`,data=APP_pca_id)
```

```
##
##  F test to compare two variances
##
## data:  PC6 by APP_data$`Tired waking up in morning`
## F = 1.2134, num df = 14, denom df = 6, p-value = 0.8604
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
```

```
##  0.2290727 4.2483981
## sample estimates:
## ratio of variances
##           1.213355
```

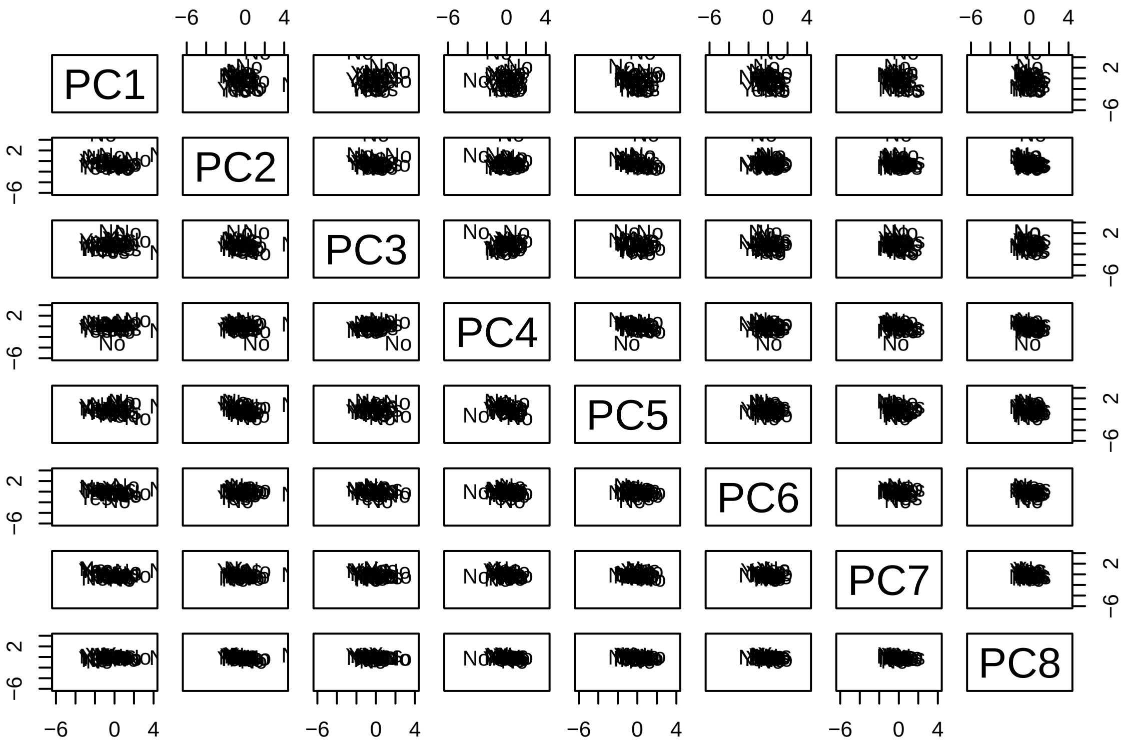
```r
var.test(PC7~APP_data$`Tired waking up in morning`,data=APP_pca_id)
```

```
##
##  F test to compare two variances
##
## data:  PC7 by APP_data$`Tired waking up in morning`
## F = 1.147, num df = 14, denom df = 6, p-value = 0.9216
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.2165433 4.0160267
## sample estimates:
## ratio of variances
##           1.146989
```

```r
var.test(PC8~APP_data$`Tired waking up in morning`,data=APP_pca_id)
```

```
##
##  F test to compare two variances
##
## data:  PC8 by APP_data$`Tired waking up in morning`
## F = 0.87544, num df = 14, denom df = 6, p-value = 0.7775
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.1652765 3.0652302
## sample estimates:
## ratio of variances
##          0.8754387
```

```r
pairs(APP_pca$x[,1:8], ylim = c(-6,4),xlim = c(-6,4),panel=function(x,y,...){text(x,y,APP_pca_id$`Tired
```



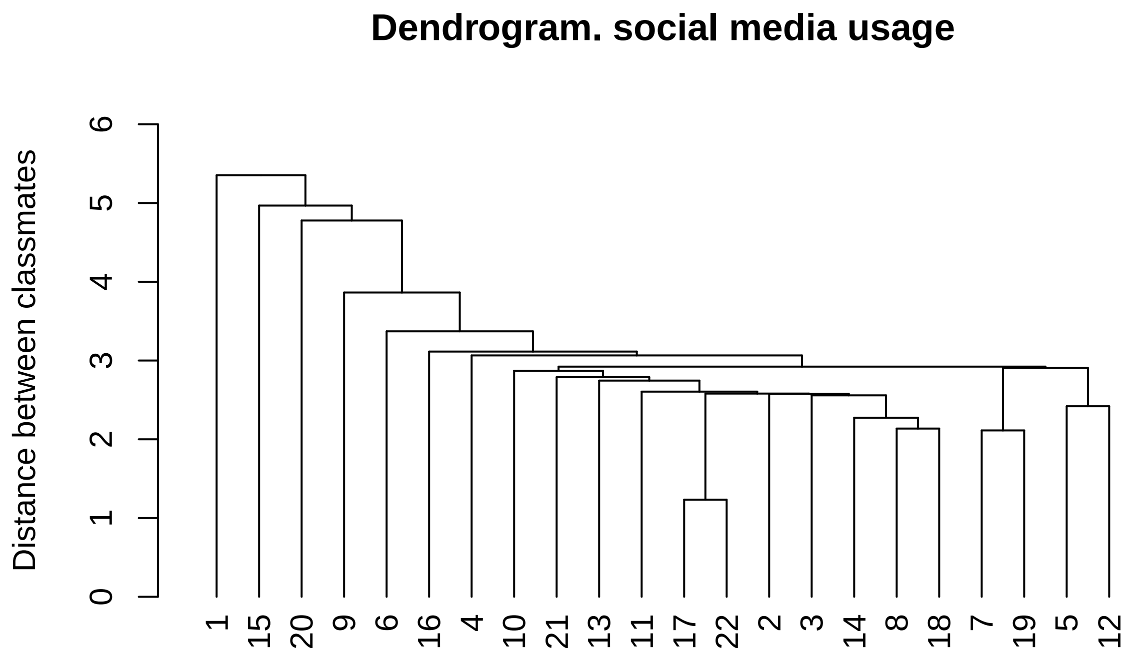
Cluster Analysis

```r
library(cluster)
library(readr)
library(factoextra)
```

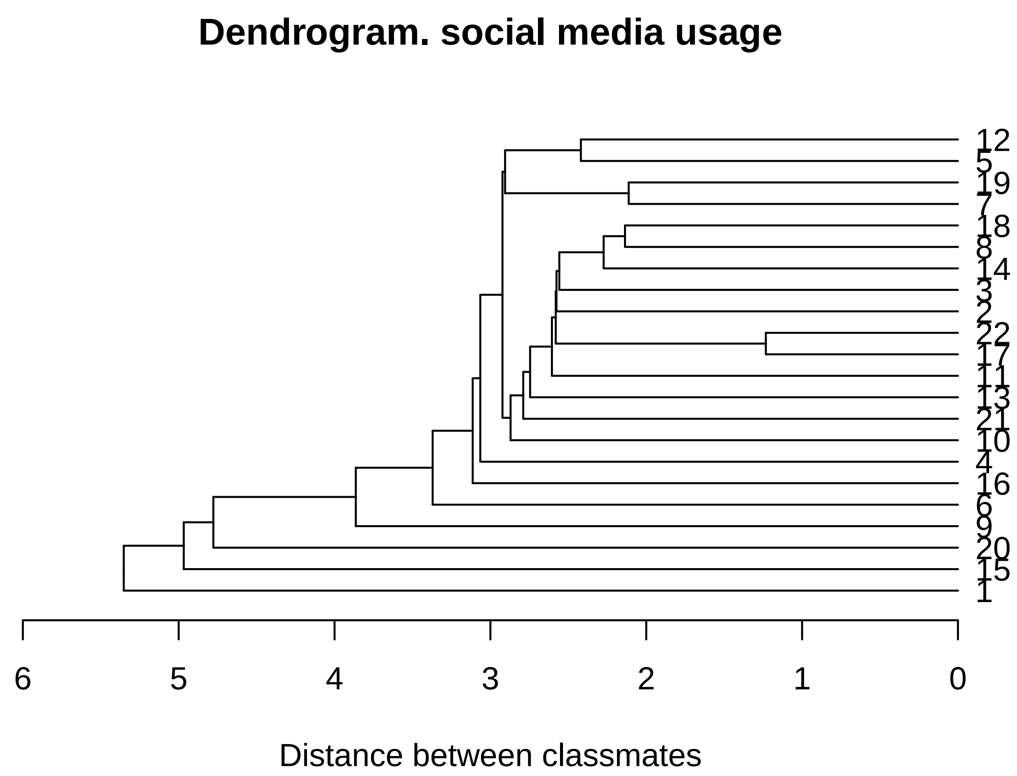
```
## Loading required package: ggplot2

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

```r
library(magrittr)
library(NbClust)
matstd.APP <- scale(MVA_data)
dist.APP <- dist(matstd.APP, method="euclidean")
clusAPP.nn <- hclust(dist.APP, method = "single")
plot(as.dendrogram(clusAPP.nn),ylab="Distance between classmates",ylim=c(0,6),main="Dendrogram. social r
```




```
plot(as.dendrogram(clusAPP.nn), xlab= "Distance between classmates", xlim=c(6,0), horiz = TRUE,main="Den
```




```
(agn.APP <- agnes(MVA_data, metric="euclidean", stand=TRUE, method = "single"))
```

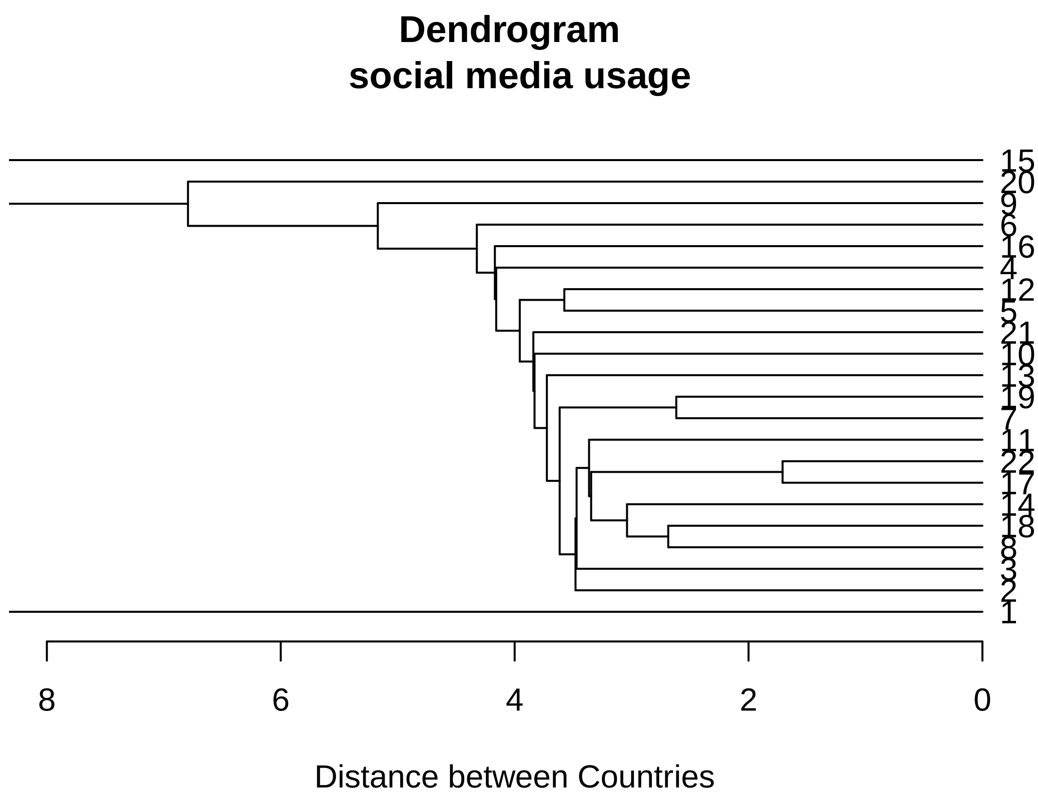
```
## Call:     agnes(x = MVA_data, metric = "euclidean", stand = TRUE, method = "single")
## Agglomerative coefficient:  0.5367198
## Order of objects:
##  [1]  1  2  3  8 18 14 17 22 11  7 19 13 10 21  5 12  4 16  6  9 20 15
## Height (summary):
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.709   3.364   3.724   4.183   4.169   8.587
##
## Available components:
## [1] "order"  "height" "ac"     "merge"  "diss"   "call"   "method" "data"
```

```
agn.APP$merge
```

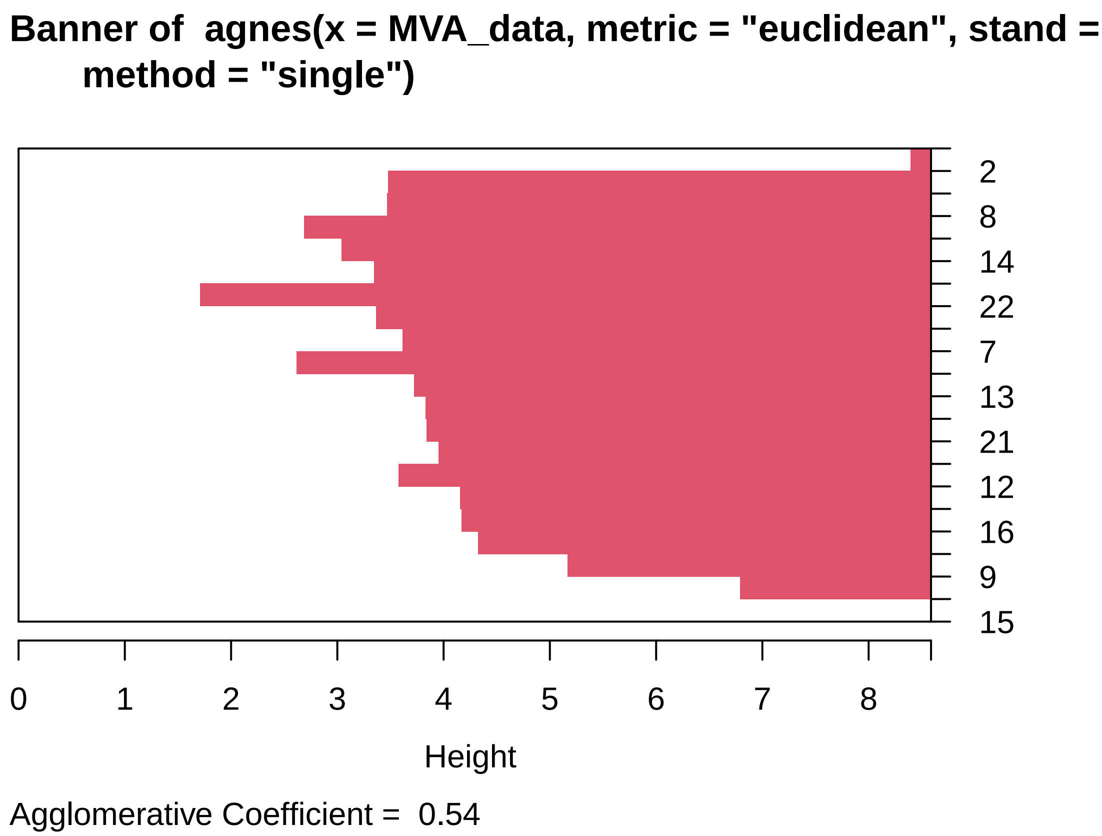
```
##       [,1] [,2]
##  [1,]  -17  -22
##  [2,]   -7  -19
##  [3,]   -8  -18
##  [4,]    3  -14
##  [5,]    4    1
##  [6,]    5  -11
##  [7,]   -3    6
##  [8,]   -2    7
##  [9,]   -5  -12
```

```
## [10,]    8    2
## [11,]   10  -13
## [12,]   11  -10
## [13,]   12  -21
## [14,]   13    9
## [15,]   14   -4
## [16,]   15  -16
## [17,]   16   -6
## [18,]   17   -9
## [19,]   18  -20
## [20,]   -1   19
## [21,]   20  -15
```

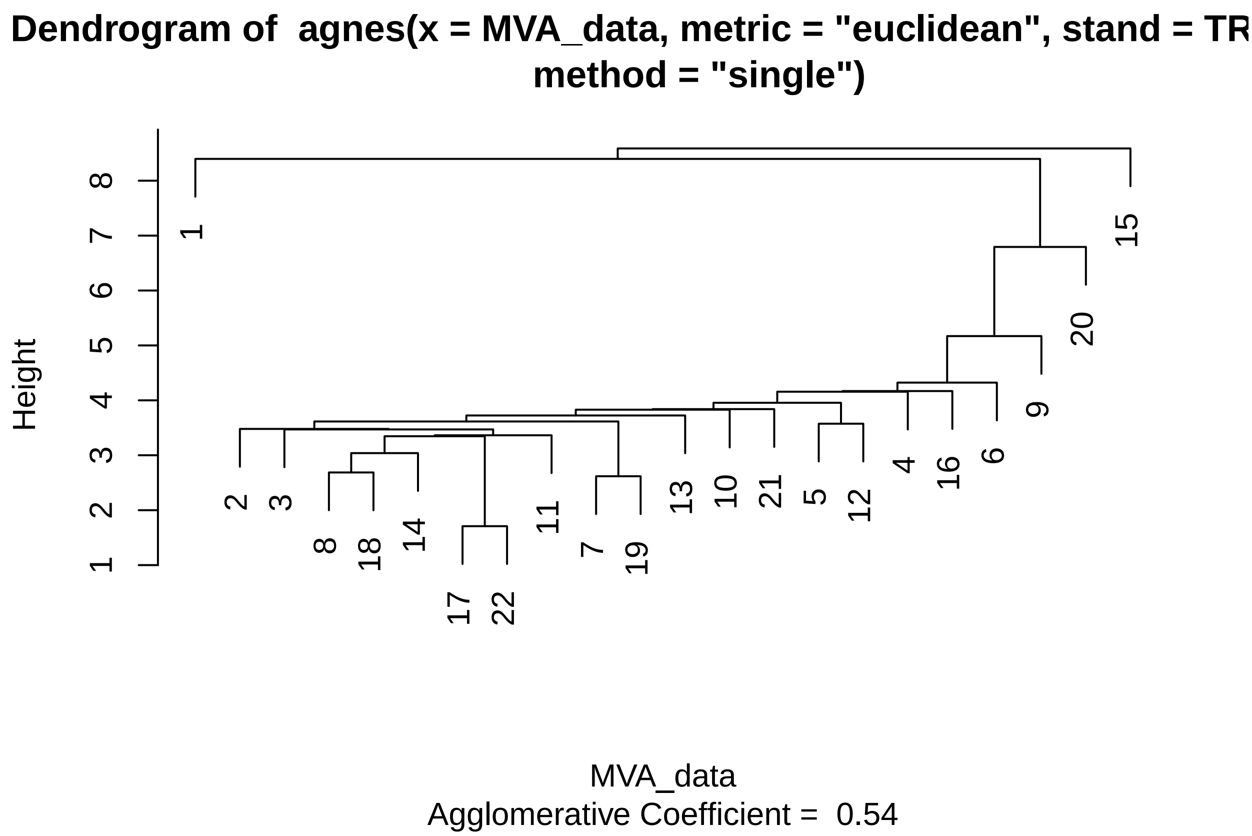
```
plot(as.dendrogram(agn.APP), xlab= "Distance between Countries",xlim=c(8,0), horiz = TRUE,main="Dendrog
```




```
plot(agn.APP, which.plots=1)
```




```
plot(agn.APP, which.plots=2)
```




```
plot(agn.APP, which.plots=3)
```

#factor analysis

```
library(psych)
```

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

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

```
fit.pc <- principal(MVA_data[,-c(9:12)], nfactors=4, rotate="varimax")
fit.pc
```

```
## Principal Components Analysis
## Call: principal(r = MVA_data[, -c(9:12)], nfactors = 4, rotate = "varimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                                          RC1   RC2   RC3   RC4   h2    u2 com
## Instagram_value                         0.89 -0.09  0.14 -0.10 0.82 0.176 1.1
## Linkedin_value                          0.17  0.10  0.85 -0.09 0.77 0.234 1.1
## Snapchat_value                          0.85  0.15  0.21 -0.02 0.78 0.218 1.2
## Twitter_value                          -0.19  0.91 -0.11  0.06 0.87 0.125 1.1
```

```
## Whatsapp/ Wechat_value                    0.64 -0.51  0.19  0.03 0.71 0.290 2.1
## Youtube_value                             0.17 -0.19  0.80  0.12 0.72 0.280 1.3
## OTT (Netflix, Hulu, Prime video)_value    0.48  0.74  0.15  0.24 0.86 0.139 2.1
## Reddit_value                             -0.07  0.13  0.01  0.97 0.96 0.039 1.0
##
##                          RC1  RC2  RC3  RC4
## SS loadings             2.24 1.73 1.49 1.03
## Proportion Var          0.28 0.22 0.19 0.13
## Cumulative Var          0.28 0.50 0.68 0.81
## Proportion Explained    0.34 0.27 0.23 0.16
## Cumulative Proportion 0.34 0.61 0.84 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  0.09
##  with the empirical chi square  9.36  with prob <  0.0093
##
## Fit based upon off diagonal values = 0.92
```

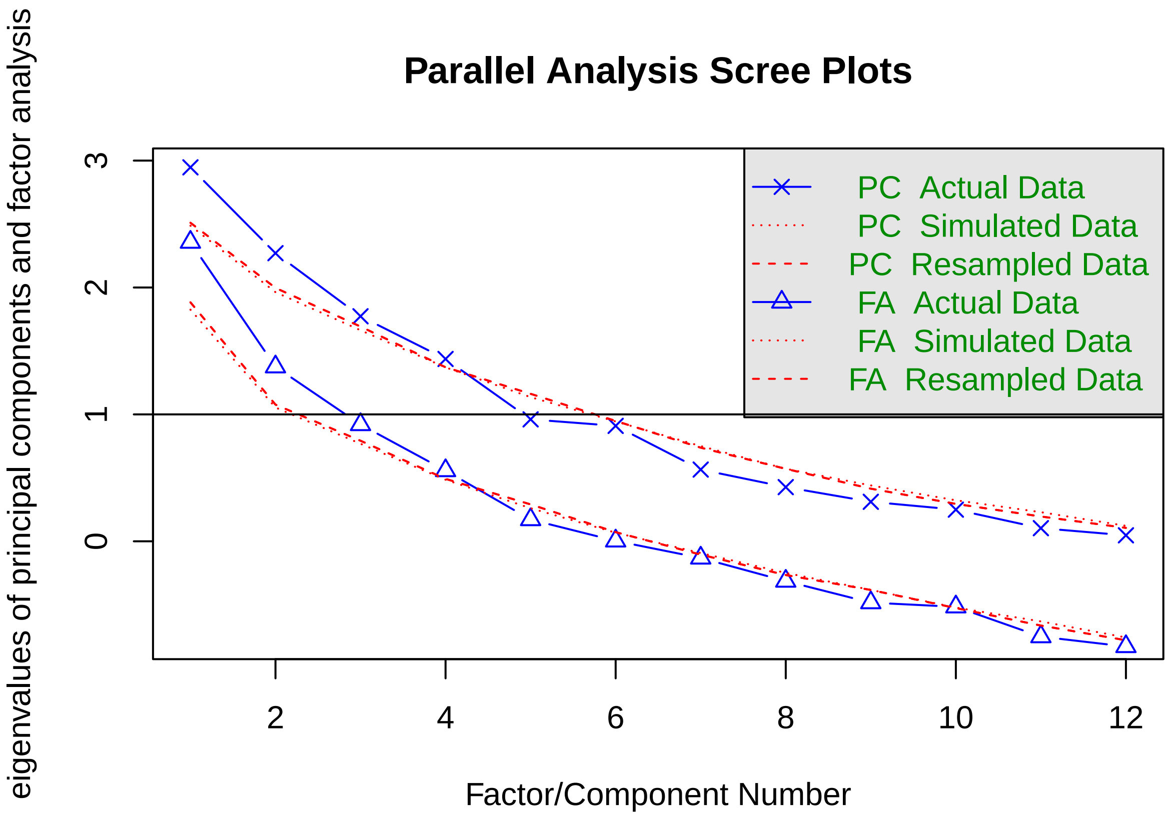
```r
round(fit.pc$values, 3)
```

```
## [1] 2.785 1.826 1.033 0.854 0.701 0.414 0.293 0.093
```

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

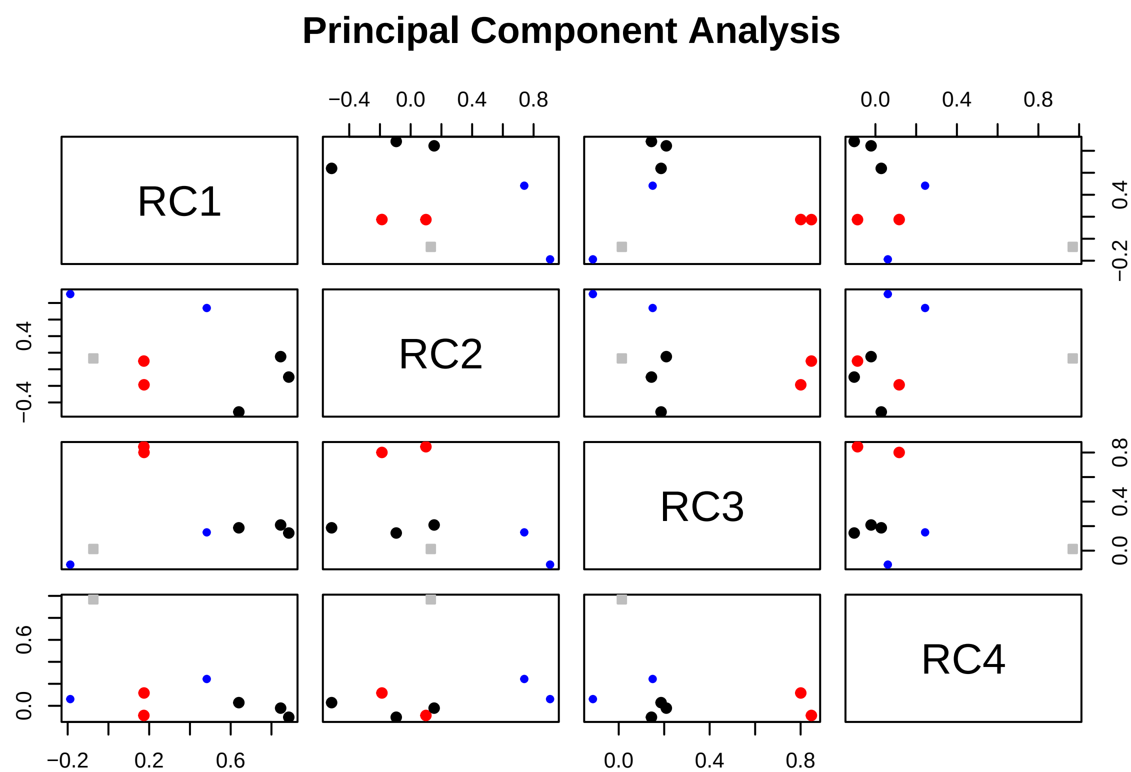
```
##
## Loadings:
##                                           RC1    RC2    RC3    RC4
## Instagram_value                            0.885         0.144 -0.104
## Linkedin_value                             0.174         0.847
## Snapchat_value                             0.845  0.153  0.209
## Twitter_value                             -0.187  0.907 -0.113
## Whatsapp/ Wechat_value                     0.640 -0.515  0.186
## Youtube_value                              0.175 -0.187  0.801  0.117
## OTT (Netflix, Hulu, Prime video)_value     0.482  0.739  0.149  0.244
## Reddit_value                                      0.131         0.968
##
##                  RC1   RC2   RC3   RC4
## SS loadings    2.241 1.729 1.494 1.035
## Proportion Var 0.280 0.216 0.187 0.129
## Cumulative Var 0.280 0.496 0.683 0.812
```

```r
fa.parallel(MVA_data)
```

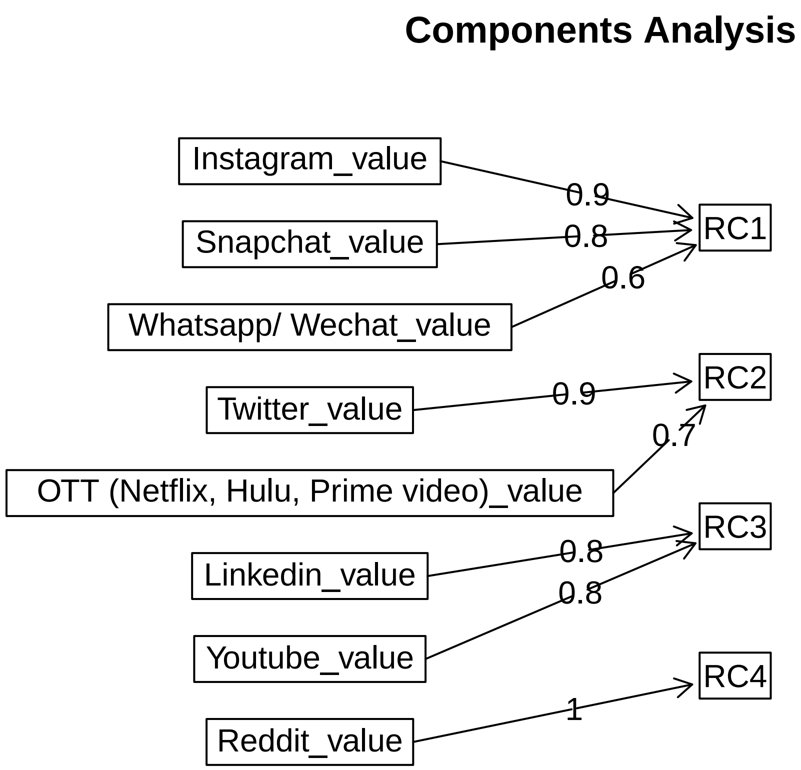



```
## Parallel analysis suggests that the number of factors =  0  and the number of components =  0
```

```
fa.plot(fit.pc)
```




```
fa.diagram(fit.pc)
```




```r
vss(MVA_data)
```

```
## 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
```

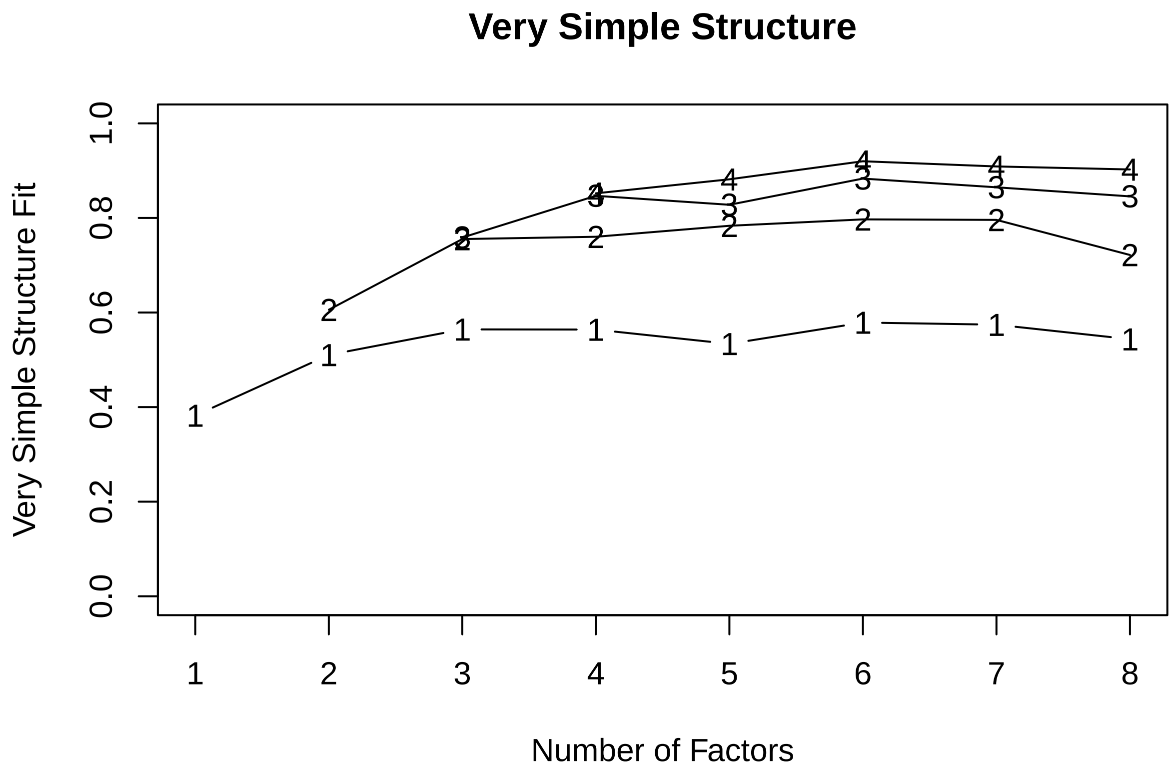
```
## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## The estimated weights for the factor scores are probably incorrect.  Try a
## different factor score estimation method.
```

```
## Warning in 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
```

```
## 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
```




```
##
## Very Simple Structure
## Call: vss(x = MVA_data)
## Although the VSS complexity 1 shows  6  factors, it is probably more reasonable to think about  3  fa
## VSS complexity 2 achieves a maximimum of 0.8  with  6  factors
##
## The Velicer MAP achieves a minimum of 0.08  with  1  factors
## BIC achieves a minimum of  -87.78  with  1  factors
## Sample Size adjusted BIC achieves a minimum of  7.73  with  7  factors
##
## Statistics by number of factors
##   vss1 vss2   map dof chisq  prob sqresid  fit RMSEA   BIC SABIC complex eChisq
## 1 0.38 0.00 0.075  54  79.1 0.015   13.26 0.38 0.138 -87.8  79.1     1.0 108.82
## 2 0.51 0.61 0.084  43  58.7 0.056    8.46 0.61 0.120 -74.2  58.7     1.3  58.29
## 3 0.56 0.76 0.085  33  37.4 0.275    5.18 0.76 0.062 -64.6  37.4     1.4  24.66
## 4 0.56 0.76 0.099  24  27.9 0.264    3.18 0.85 0.072 -46.3  27.9     1.7  10.75
## 5 0.53 0.78 0.139  16  17.6 0.349    2.39 0.89 0.048 -31.9  17.6     1.9   4.78
## 6 0.58 0.80 0.141   9  12.1 0.207    1.37 0.94 0.116 -15.7  12.1     1.7   1.76
## 7 0.57 0.80 0.189   3   7.7 0.052    1.11 0.95 0.264  -1.5   7.7     1.9   0.76
## 8 0.54 0.72 0.263  -2   1.7    NA    0.88 0.96    NA    NA    NA     2.2   0.16
##     SRMR eCRMS  eBIC
## 1 0.1936 0.214 -58.1
```

```
## 2 0.1417 0.176 -74.6
## 3 0.0921 0.130 -77.3
## 4 0.0608 0.101 -63.4
## 5 0.0406 0.082 -44.7
## 6 0.0246 0.067 -26.1
## 7 0.0162 0.076  -8.5
## 8 0.0074    NA    NA
```

```
(eigen_APP_vars <- round(APP_pca$sdev^2,3))
```

```
## [1] 2.785 1.826 1.033 0.854 0.701 0.414 0.293 0.093
```

```
names(eigen_APP_vars) <- paste("PC",1:8,sep="")
sumlambdas <- sum(eigen_APP_vars)
propvar <- round(eigen_APP_vars/sumlambdas,2)
cumvar_APP_vars <- cumsum(propvar)
matlambdas <- rbind(eigen_APP_vars,propvar,cumvar_APP_vars)
rownames(matlambdas) <- c("Eigenvalues","Prop. variance","Cum. prop. variance")
eigvec.emp <- APP_pca$rotation
pcafactors.emp <- eigvec.emp[,1:2]
unrot.fact.emp <- sweep(pcafactors.emp,MARGIN=2,APP_pca$sdev[1:2],`*`)
communalities.emp <- rowSums(unrot.fact.emp^2)
communalities.emp
```

```
##                        Instagram_value                         Linkedin_value
##                              0.6897110                              0.3602701
##                         Snapchat_value                          Twitter_value
##                              0.6973179                              0.8001503
##                 Whatsapp/ Wechat_value                          Youtube_value
##                              0.6313918                              0.3589327
## OTT (Netflix, Hulu, Prime video)_value                           Reddit_value
##                              0.8444099                              0.2296178
```

```
rot.fact.emp <- varimax(unrot.fact.emp)
rot.fact.emp
```

```
## $loadings
##
## Loadings:
##                                         PC1    PC2
## Instagram_value                          0.825
## Linkedin_value                           0.597
## Snapchat_value                           0.820  0.160
## Twitter_value                           -0.247  0.860
## Whatsapp/ Wechat_value                   0.656 -0.448
## Youtube_value                            0.585 -0.129
## OTT (Netflix, Hulu, Prime video)_value  0.467  0.792
## Reddit_value                                    0.477
##
##                  PC1   PC2
## SS loadings    2.762 1.850
## Proportion Var 0.345 0.231
```

```
## Cumulative Var 0.345 0.576
##
## $rotmat
##           [,1]       [,2]
## [1,] 0.9878642 -0.1553198
## [2,] 0.1553198  0.9878642
```

```
fact.load.emp <- rot.fact.emp$loadings[1:6,1:2]
fact.load.emp
```

```
##                              PC1         PC2
## Instagram_value        0.8246891 -0.09797374
## Linkedin_value         0.5971461  0.06071758
## Snapchat_value         0.8195639  0.16010320
## Twitter_value         -0.2469234  0.85975529
## Whatsapp/ Wechat_value  0.6564029 -0.44780236
## Youtube_value          0.5849619 -0.12943078
```

```
scale.emp <- scale(MVA_data[,-c(9:12)])
scale.emp
```

```
##       Instagram_value Linkedin_value Snapchat_value Twitter_value
##  [1,]     -0.57711580     0.18451345    -0.13137365    3.51308901
##  [2,]      0.66007371     0.67605281     1.36107147   -0.43783412
##  [3,]     -0.49814626     1.41336184    -0.39310842   -0.43783412
##  [4,]     -0.02725380     0.72520674     0.03569111   -0.43783412
##  [5,]     -1.60079506    -1.21637372    -0.68825615    0.09158958
##  [6,]     -0.91931715     1.41336184    -0.42652138   -0.43783412
##  [7,]     -0.03017860     0.18451345    -0.68825615   -0.43783412
##  [8,]      0.44656346     0.18451345     0.98239137   -0.43783412
##  [9,]      0.92915511     2.64221024     1.44460385   -0.43783412
## [10,]     -1.55107350    -1.45395108    -0.68825615   -0.43783412
## [11,]     -0.26123763    -0.58146872    -0.45436550    1.47441268
## [12,]     -0.62683737    -0.42991075    -0.50448493    1.79838837
## [13,]     -0.04187779    -0.83952688    -0.35412665   -0.43783412
## [14,]      0.15408367    -0.22510268    -0.13137365    0.35235051
## [15,]     -0.24076404     0.08210942     0.10251701   -0.43783412
## [16,]      0.44656346     0.59412958    -0.45436550   -0.43783412
## [17,]      1.26550687    -1.12625817    -0.68825615   -0.22448427
## [18,]      0.38806750    -0.66748810     0.35311414   -0.43783412
## [19,]      0.05756534     0.15174416    -0.68825615   -0.43783412
## [20,]      2.79225137     0.16403264     3.38812380   -0.43783412
## [21,]     -1.50427673    -1.24095069    -0.68825615   -0.43783412
## [22,]      0.73904325    -0.63471881    -0.68825615   -0.43783412
##       Whatsapp/ Wechat_value Youtube_value
##  [1,]            -1.38359636   -0.30202800
##  [2,]            -0.60922515    0.71977147
##  [3,]             0.76662307   -0.68155352
##  [4,]            -0.33649063   -0.59397070
##  [5,]            -0.89657044    0.28185741
##  [6,]             1.29504619    2.32545635
##  [7,]            -0.16603156   -0.01008529
##  [8,]             0.80802027   -0.59397070
```

```
##  [9,]           -0.12950462   0.57380012
## [10,]           -1.38359636  -0.01008529
## [11,]           -1.38359636  -1.17785611
## [12,]           -0.73341676  -0.30202800
## [13,]           -0.65305748  -1.46979882
## [14,]           -0.79429500  -0.30202800
## [15,]           -0.04427508   0.06581981
## [16,]           -0.40954452   1.15768553
## [17,]            1.62378869   0.29937397
## [18,]            0.06530575  -1.29463320
## [19,]            0.54502628   1.24526834
## [20,]            2.11081460   1.25694605
## [21,]            0.41352928  -1.76174152
## [22,]            1.29504619   0.57380012
##       OTT (Netflix, Hulu, Prime video)_value Reddit_value
##  [1,]                             3.51688501  1.281598165
##  [2,]                            -0.64719456 -0.324048739
##  [3,]                            -0.07283876 -0.324048739
##  [4,]                            -0.07283876 -0.324048739
##  [5,]                            -0.07283876  0.318210023
##  [6,]                             0.21433914 -0.324048739
##  [7,]                            -0.64719456 -0.324048739
##  [8,]                             0.21433914 -0.324048739
##  [9,]                             0.21433914 -0.324048739
## [10,]                            -0.64719456 -0.324048739
## [11,]                            -0.64719456 -0.324048739
## [12,]                            -0.21642771 -0.324048739
## [13,]                            -0.50073383 -0.259822863
## [14,]                            -0.36001666 -0.324048739
## [15,]                            -0.09581299  4.171762593
## [16,]                            -0.36001666 -0.002919358
## [17,]                            -0.16473569 -0.324048739
## [18,]                             0.06213485 -0.324048739
## [19,]                            -0.64719456 -0.324048739
## [20,]                             2.22458445 -0.324048739
## [21,]                            -0.64719456 -0.324048739
## [22,]                            -0.64719456 -0.324048739
## attr(,"scaled:center")
##                      Instagram_value                         Linkedin_value
##                            5.4731818                              3.5495455
##                       Snapchat_value                          Twitter_value
##                            1.2359091                              0.5540909
##               Whatsapp/ Wechat_value                          Youtube_value
##                            6.6818182                              3.0172727
## OTT (Netflix, Hulu, Prime video)_value                         Reddit_value
##                            2.2536364                              0.5045455
## attr(,"scaled:scale")
##                      Instagram_value                         Linkedin_value
##                             3.419040                               2.441310
##                       Snapchat_value                          Twitter_value
##                             1.795711                               1.265527
##               Whatsapp/ Wechat_value                          Youtube_value
##                             4.106558                               1.712665
## OTT (Netflix, Hulu, Prime video)_value                         Reddit_value
```

```
##                              3.482162                              1.557005
```