final project social media Yuefei Chen

Yuefei Chen

2024-04-29

final project on Social Media

Question 1. Explain the data collection process. (10 points)

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 Additionally, N/A value has been replaced by 0.00.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.

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

```
## New names:
## Rows: 23 Columns: 33
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (15): ID, Instagram, Linkedin, Snapchat, Twitter, Whatsapp_Wechat, Yout... dbl
## (12): Instagram_value, Linkedin_value, Snapchat_value, Twitter_value, W... time
## (6): Hours_spent...3, Hours_spent...6, Hours_spent...9, Hours spent, H...
## 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...15`
## * `Hours_spent` -> `Hours_spent...18`
## * `Hours_spent` -> `Hours_spent...21`
## * `Hours_spent` -> `Hours_spent...24`
```

```
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 8.65 0.17
##  $ 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 0 ...
##  $ 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 Media" "Social
##  $ job_interview_calls                      : num [1:22] 0 0 0 2 0 0 0 0 1 0 ...
##  $ networking_done_with_coffee_chats        : num [1:22] 0 1 0 0 2 0 2 0 0 0 ...
##  $ 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" ...
##  $ felt_the_entire_week                     : num [1:22] 3 3 4 4 3 5 4 4 3 2 ...
```

```
summary(APP_data)
```

```
##       ID             Instagram         Instagram_value    Linkedin
##  Length:22          Length:22          Min.   : 0.000   Length:22
##  Class :character   Class :character   1st Qu.: 3.567   Class :character
##  Mode  :character   Mode  :character   Median : 5.375   Mode  :character
##                                        Mean   : 5.473
##                                        3rd Qu.: 7.000
##                                        Max.   :15.020
##  Linkedin_value     Snapchat          Snapchat_value    Twitter
##  Min.   : 0.000   Length:22          Min.   :0.000   Length:22
##  1st Qu.: 1.940   Class :character   1st Qu.:0.000   Class :character
##  Median : 3.835   Mode  :character   Median :0.500   Mode  :character
##  Mean   : 3.550                      Mean   :1.236
##  3rd Qu.: 4.750                      3rd Qu.:1.390
##  Max.   :10.000                      Max.   :7.320
##  Twitter_value    Whatsapp_Wechat    Whatsapp_Wechat_value   Youtube
##  Min.   :0.0000   Length:22          Min.   : 1.000        Length:22
##  1st Qu.:0.0000   Class :character   1st Qu.: 3.752        Class :character
##  Median :0.0000   Mode  :character   Median : 6.075        Mode  :character
##  Mean   :0.5541                      Mean   : 6.682
##  3rd Qu.:0.2025                      3rd Qu.: 9.602
##  Max.   :5.0000                      Max.   :15.350
##  Youtube_value   OTT_Netflix_Hulu_Prime video
##  Min.   :0.000   Length:22
##  1st Qu.:2.000   Class :character
##  Median :3.000   Mode  :character
##  Mean   :3.017
##  3rd Qu.:4.000
##  Max.   :7.000
##  OTT_Netflix_Hulu_Prime_video_value    Reddit           Reddit_value
##  Min.   : 0.000                     Length:22          Min.   :0.0000
##  1st Qu.: 0.000                     Class :character   1st Qu.:0.0000
```

```
##  Median : 1.590                     Mode  :character   Median :0.0000
##  Mean   : 2.254                                       Mean   :0.5045
##  3rd Qu.: 2.353                                       3rd Qu.:0.0000
##  Max.   :14.500                                       Max.   :7.0000
##  Application_type_Social media_OTT_Learning job_interview_calls
##  Length:22                                  Min.   :0.0000
##  Class :character                           1st Qu.:0.0000
##  Mode  :character                           Median :0.0000
##                                             Mean   :0.2273
##                                             3rd Qu.:0.0000
##                                             Max.   :2.0000
##  networking_done_with_coffee_chats learning_done_in_terms_of_items_created
##  Min.   :0.0                       Min.   :1.000
##  1st Qu.:0.0                       1st Qu.:2.000
##  Median :0.0                       Median :3.000
##  Mean   :0.5                       Mean   :3.045
##  3rd Qu.:1.0                       3rd Qu.:4.000
##  Max.   :2.0                       Max.   :6.000
##  Mood_Productivity  Tired_waking_up_in_morning Trouble_falling_asleep
##  Length:22          Length:22                  Length:22
##  Class :character   Class :character           Class :character
##  Mode  :character   Mode  :character           Mode  :character
##
##
##
##  felt_the_entire_week
##  Min.   :2.000
##  1st Qu.:3.000
##  Median :3.000
##  Mean   :3.455
##  3rd Qu.:4.000
##  Max.   :5.000
```

Question 2. Exploratory Data Analysis and Visualizations (50 points)

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

PCA Analysis

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.

```r
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
```

```r
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
```

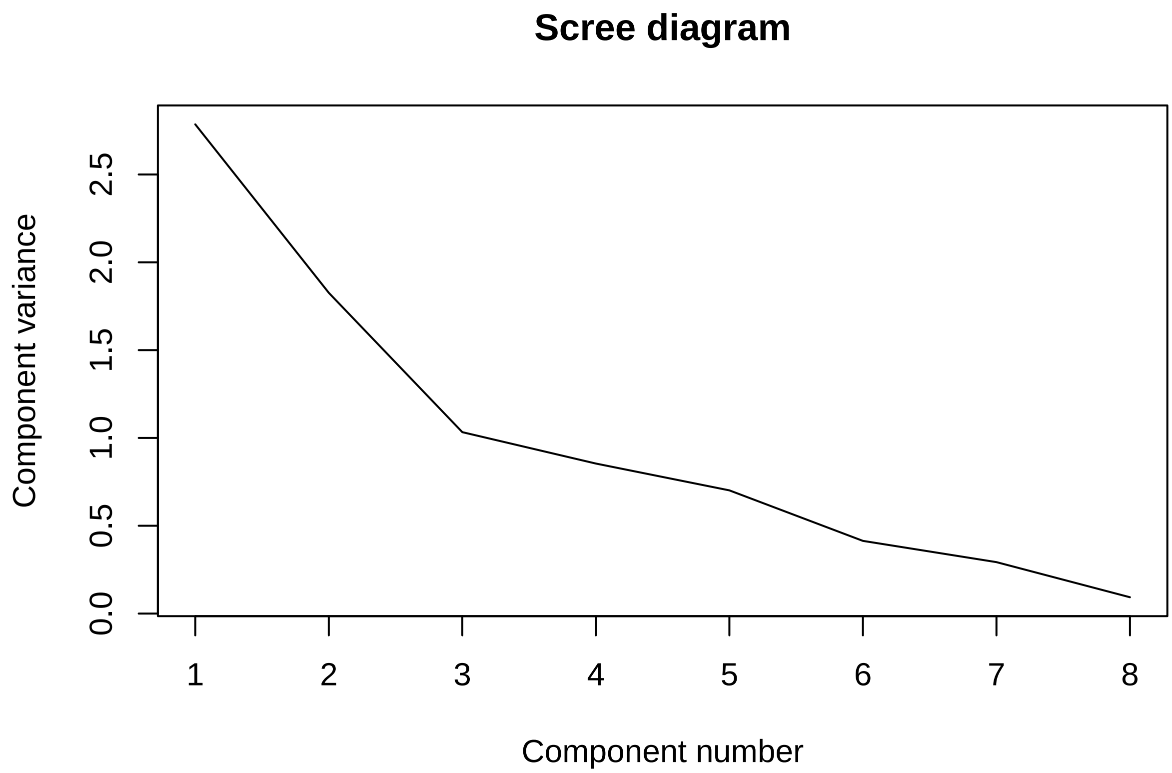
```r
(eigen_rent <- APP_pca$sdev^2)
```

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

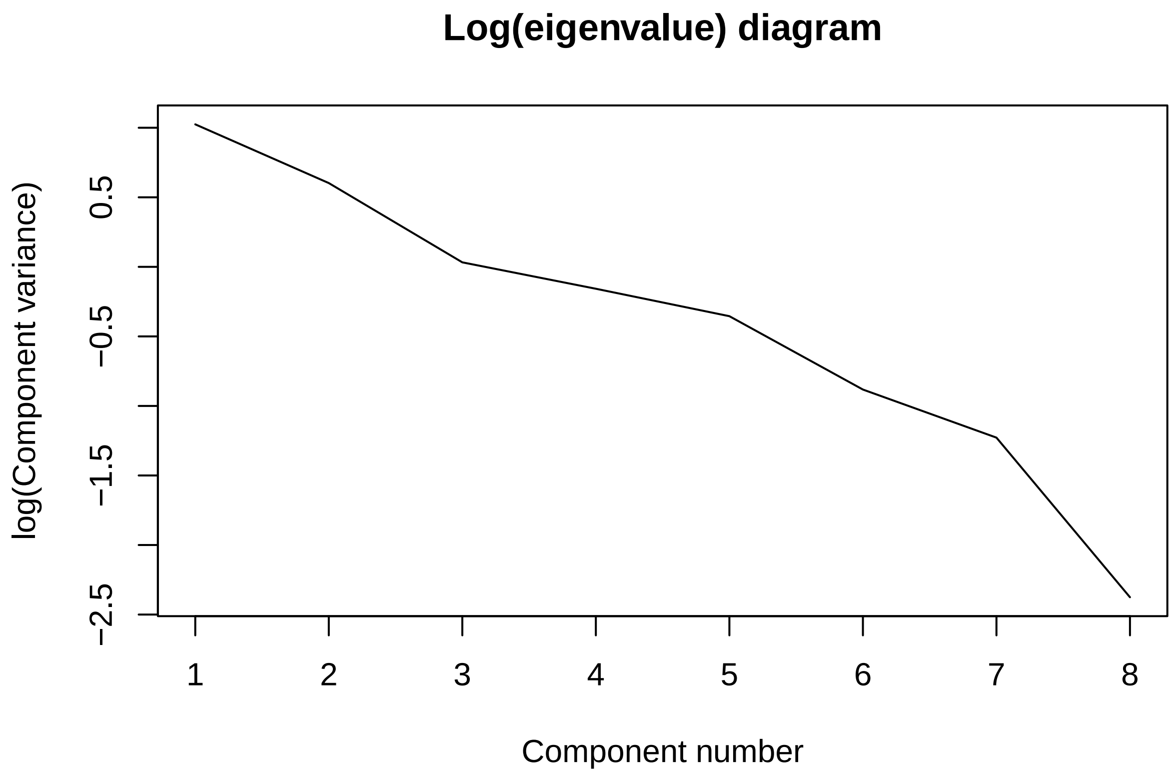
```r
names(eigen_rent) <- paste("PC",1:8,sep="")
eigen_rent
```

```
##       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
```




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




```r
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
```

```r
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
```

```r
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
```

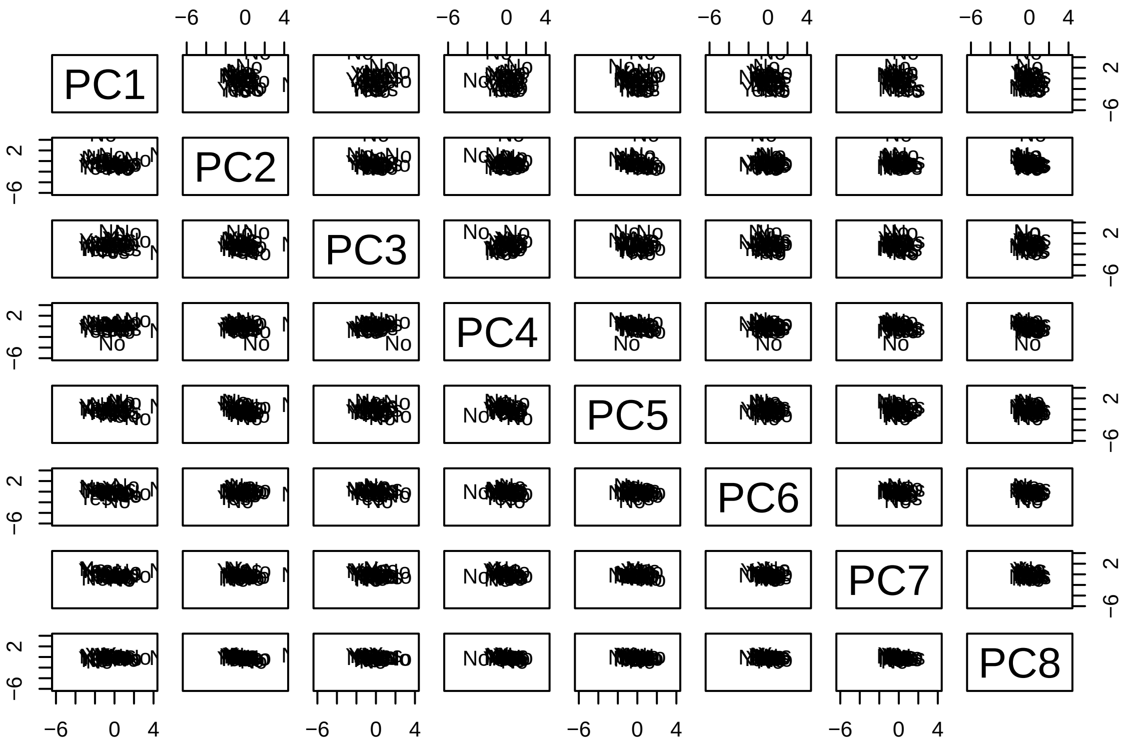
```
var.test(PC7~APP_data$`Tired_waking_up_in_morning`,data=APP_pca_id)
```

```
##
##  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
```

```
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
```

```
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_
```



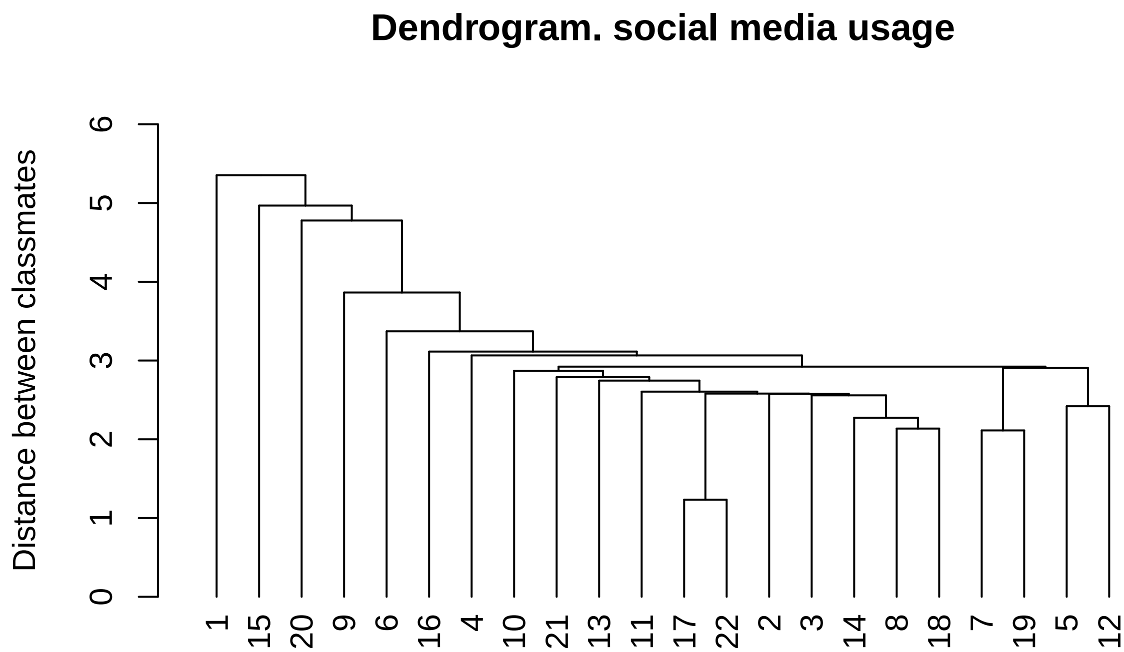
Cluster Analysis

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

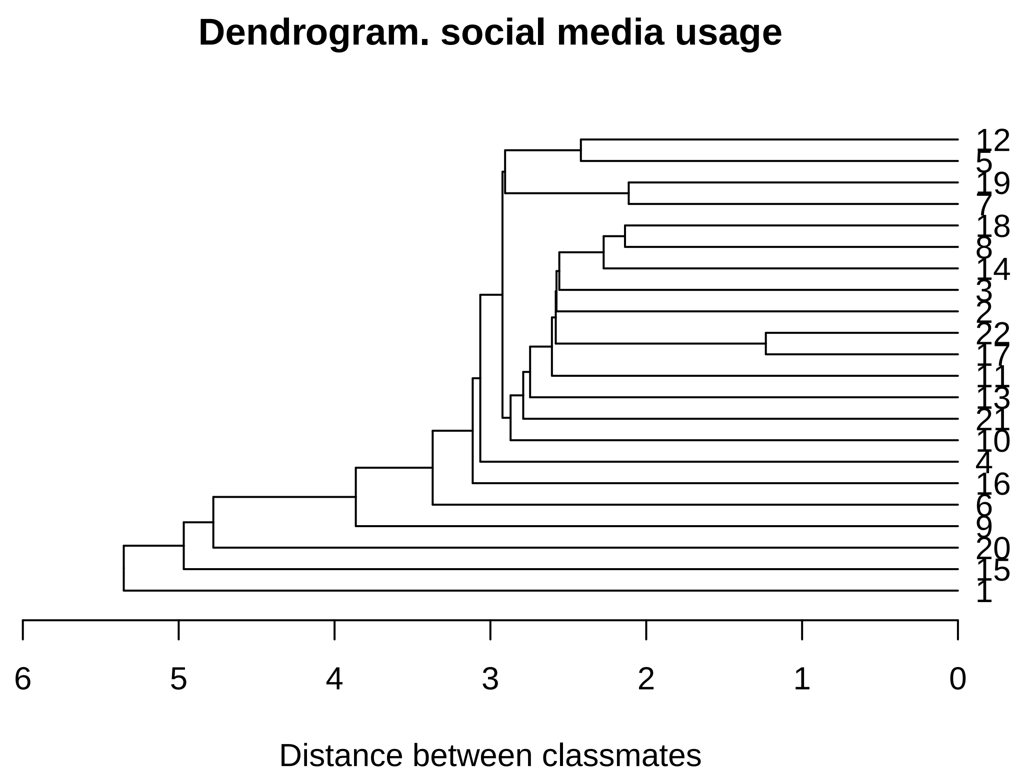
```
## Loading required package: ggplot2
```

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

```
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"))
```

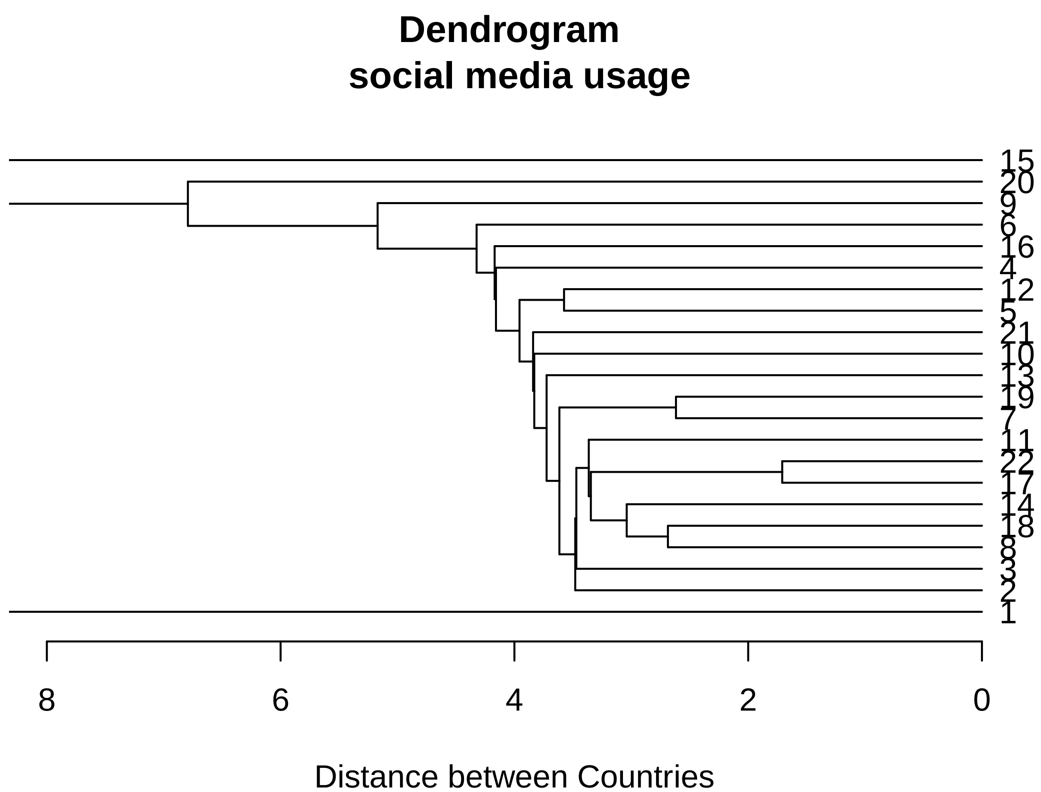
```
## Call:     agnes(x = MVA_data, metric = "euclidean", stand = TRUE, method = "single")
## Agglomerative coefficient:  0.5367198
## Order of objects:
##  [1]  1  2  3  8 18 14 17 22 11  7 19 13 10 21  5 12  4 16  6  9 20 15
## Height (summary):
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   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
```

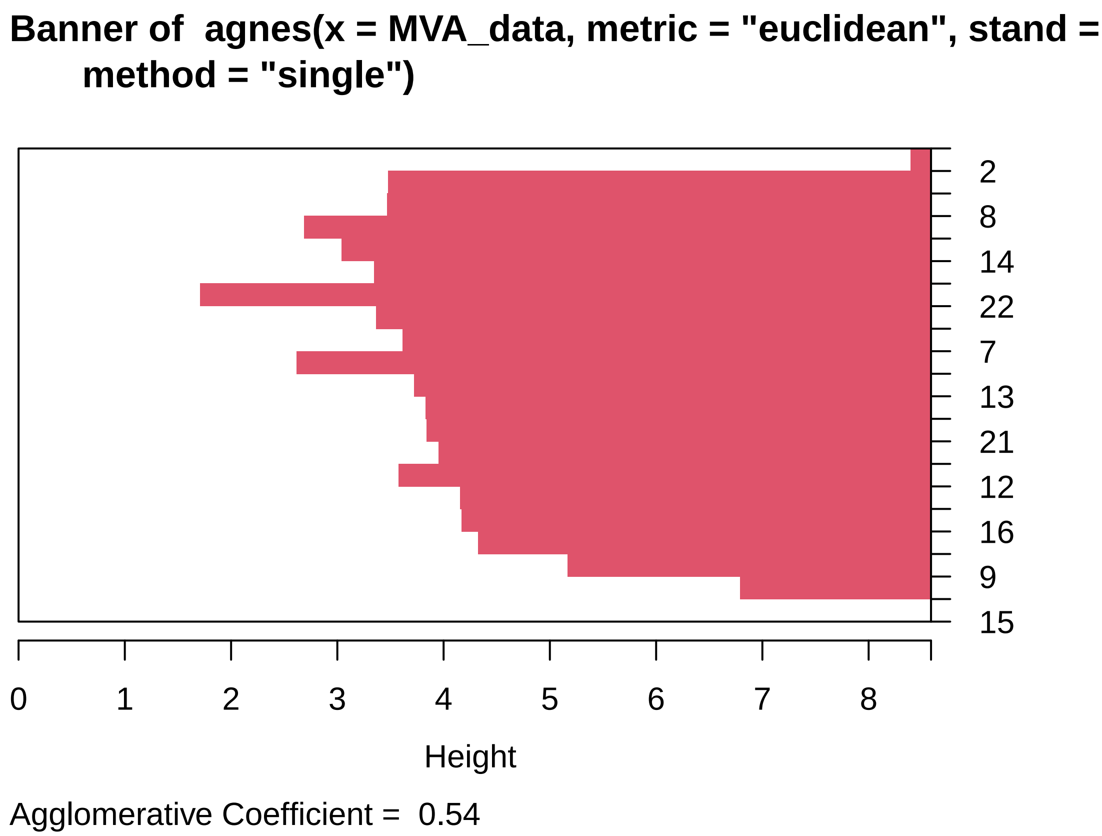
```
##       [,1] [,2]
##  [1,]  -17  -22
##  [2,]   -7  -19
##  [3,]   -8  -18
##  [4,]    3  -14
##  [5,]    4    1
##  [6,]    5  -11
##  [7,]   -3    6
##  [8,]   -2    7
##  [9,]   -5  -12
```

```
## [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
```

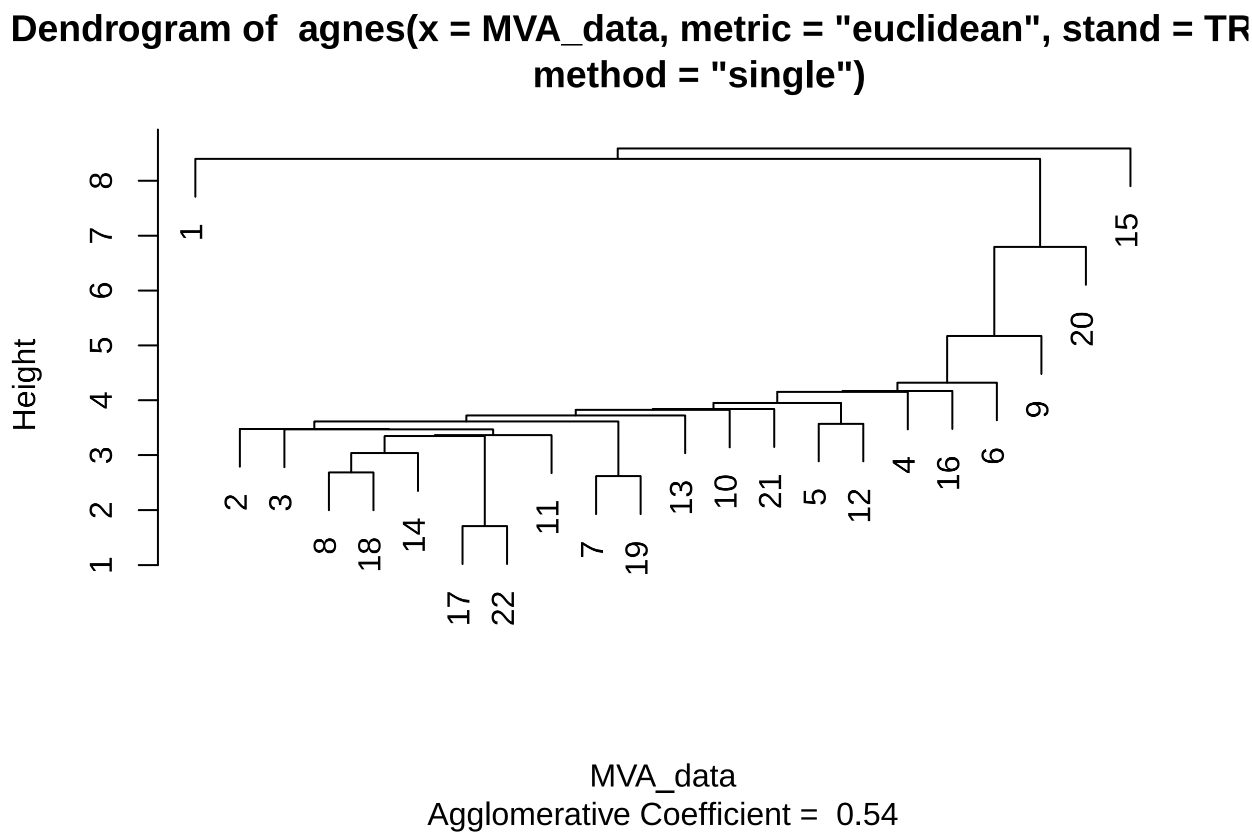
```
plot(as.dendrogram(agn.APP), xlab= "Distance between Countries",xlim=c(8,0), horiz = TRUE,main="Dendrog
```




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




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




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

#factor analysis

```r
library(psych)
```

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

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

```r
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
```

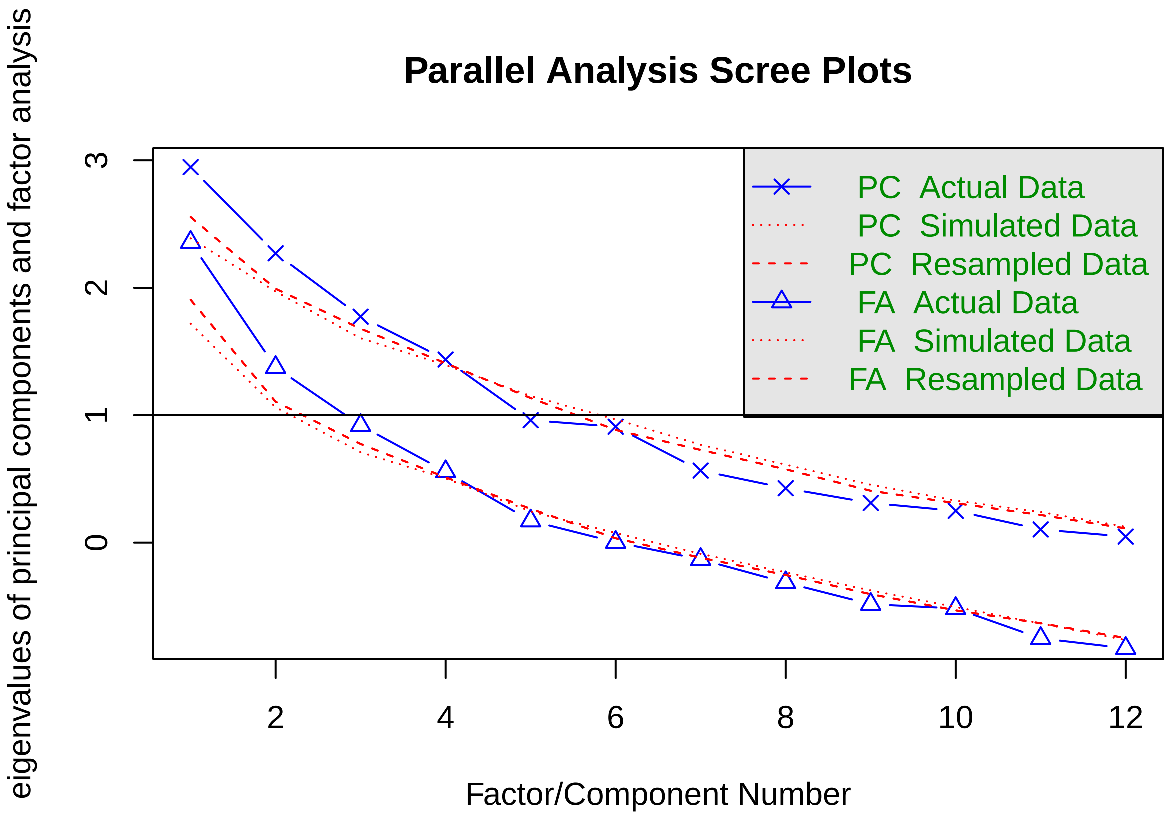
```r
round(fit.pc$values, 3)
```

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

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

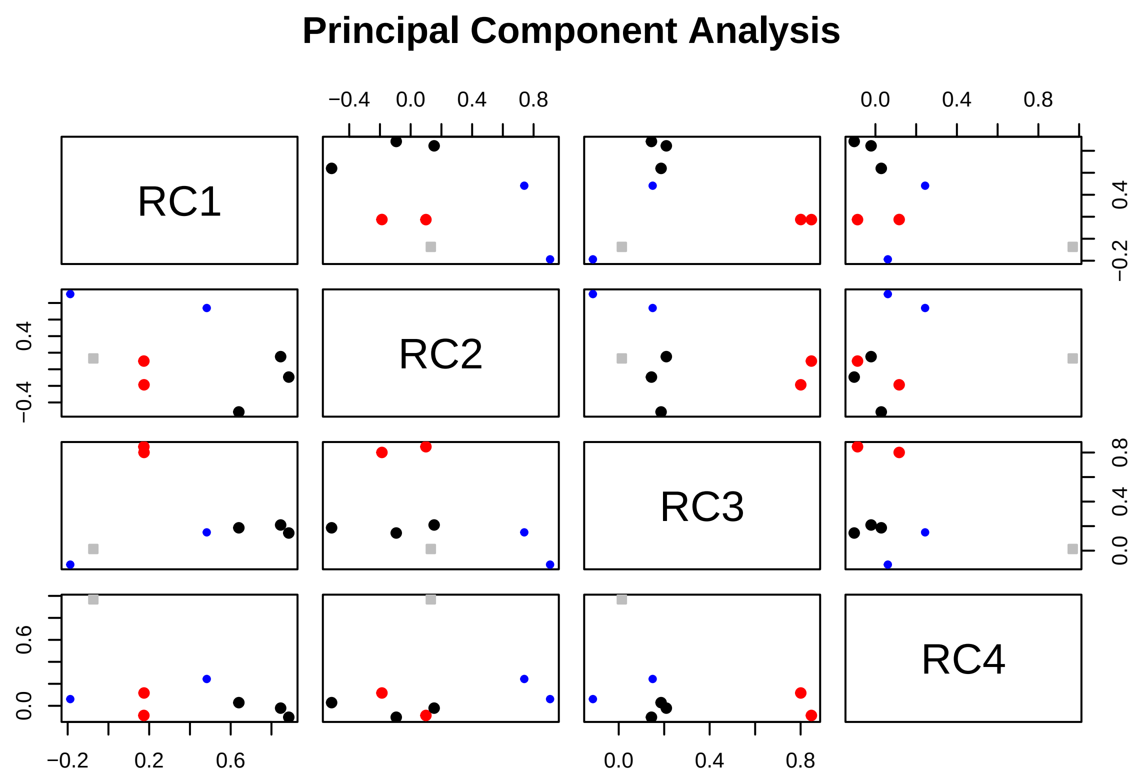
```
##
## Loadings:
##                                    RC1    RC2    RC3    RC4
## Instagram_value                     0.885         0.144 -0.104
## Linkedin_value                      0.174         0.847
## Snapchat_value                      0.845  0.153  0.209
## Twitter_value                      -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)
```

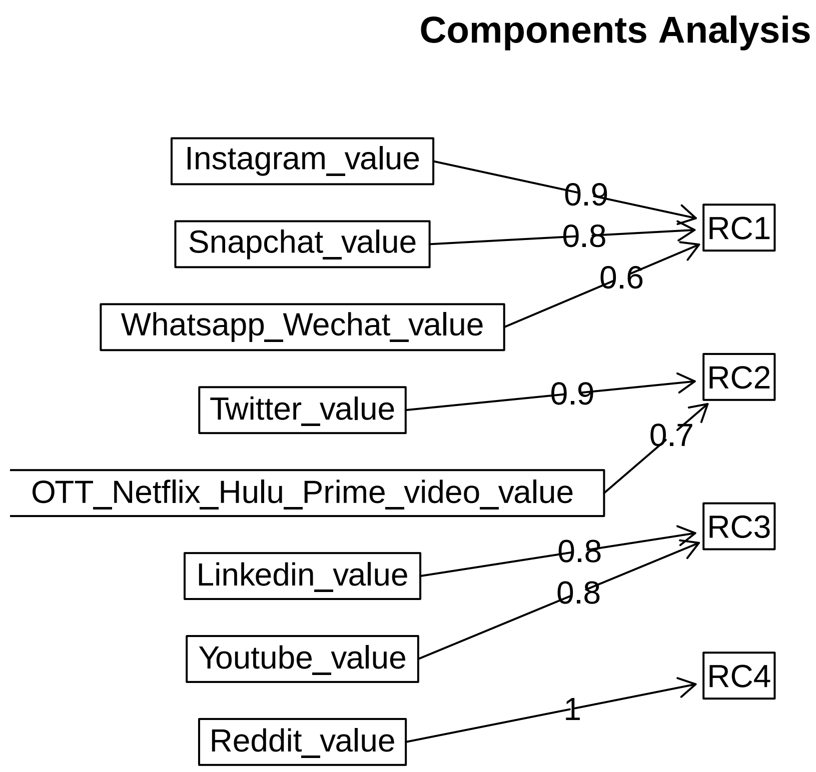



```
## Parallel analysis suggests that the number of factors =  3  and the number of components =  2
```

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



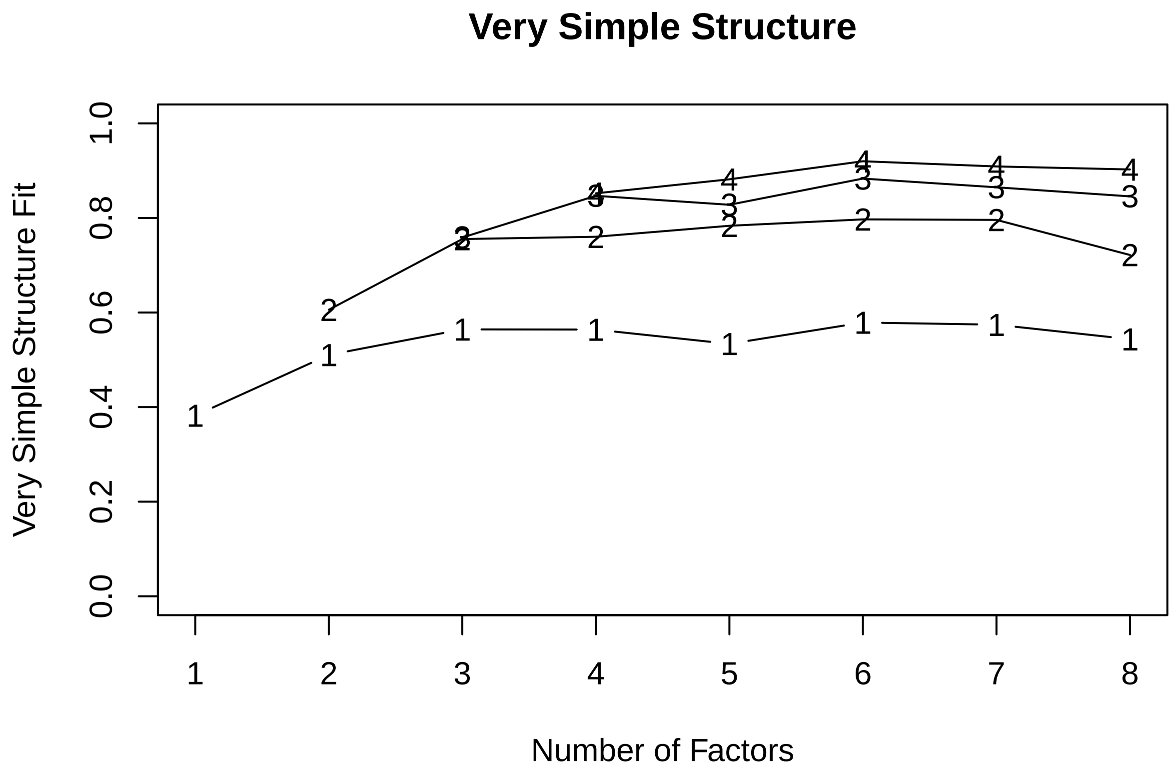

```
fa.diagram(fit.pc)
```




```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
```

Question 3. Application of different MVA models (10 points)

Multiregression model

Model development Running the following code, we build a multiple regression model based on rent house data. Its independent variables "Instagram_value", "Linkedin_value", "Snapchat_value", "Twitter_value", "Whatsapp_Wechat_value", "Youtube_value", "OTT_Netflix_Hulu_Prime_video_value", "Reddit_value", "job_interview_calls", "networking_done_with_coffee_chats", "learning_done_in_terms_of_items_crea
The dependent variable is "felt_the_entire_week".

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

```
## New names:
## Rows: 23 Columns: 33
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (15): ID, Instagram, Linkedin, Snapchat, Twitter, Whatsapp_Wechat, Yout... dbl
## (12): Instagram_value, Linkedin_value, Snapchat_value, Twitter_value, W... time
## (6): Hours_spent...3, Hours_spent...6, Hours_spent...9, Hours spent, H...
## 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...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 8.65 0.17
##  $ 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 0 ...
##  $ 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 Media" "Socia
##  $ job_interview_calls                       : num [1:22] 0 0 0 2 0 0 0 0 1 0 ...
##  $ networking_done_with_coffee_chats         : num [1:22] 0 1 0 0 2 0 2 0 0 0 ...
##  $ 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" ...
##  $ felt_the_entire_week                      : num [1:22] 3 3 4 4 3 5 4 4 3 2 ...
```

```r
fit <- lm(felt_the_entire_week ~ Instagram_value + Linkedin_value + Snapchat_value + Twitter_value + Wha
fit
```

```
##
## Call:
## lm(formula = felt_the_entire_week ~ Instagram_value + Linkedin_value +
##     Snapchat_value + Twitter_value + Whatsapp_Wechat_value +
##     Youtube_value + OTT_Netflix_Hulu_Prime_video_value + Reddit_value +
##     job_interview_calls + networking_done_with_coffee_chats +
##     learning_done_in_terms_of_items_created, data = APP_data)
##
## Coefficients:
##                             (Intercept)
##                                 3.38572
##                         Instagram_value
##                                -0.09842
##                          Linkedin_value
##                                 0.19780
##                          Snapchat_value
##                                -0.10269
##                           Twitter_value
##                                 0.24008
##                   Whatsapp_Wechat_value
##                                 0.10295
##                           Youtube_value
##                                 0.02357
##      OTT_Netflix_Hulu_Prime_video_value
##                                -0.05841
##                            Reddit_value
##                                -0.11598
##                     job_interview_calls
##                                 0.72837
##       networking_done_with_coffee_chats
##                                 0.04368
## learning_done_in_terms_of_items_created
##                                -0.28138
```

Model Acceptance In the summary of the model, we focus on R squared value, coefficients, and P-value of each coefficient. The R-squared value is 0.4434 and Adjust R-squared value is -0.1688. It shows there is a low proportion of variance in the dependent variable can be explained by the independent variables. Therefore, we use stepAIC to find an optimal model.

```r
summary(fit)
```

```
##
## Call:
## lm(formula = felt_the_entire_week ~ Instagram_value + Linkedin_value +
##     Snapchat_value + Twitter_value + Whatsapp_Wechat_value +
##     Youtube_value + OTT_Netflix_Hulu_Prime_video_value + Reddit_value +
##     job_interview_calls + networking_done_with_coffee_chats +
##     learning_done_in_terms_of_items_created, data = APP_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.97988 -0.38761 -0.06981  0.39474  1.51971
##
## Coefficients:
##                                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)                              3.38572    0.75072   4.510  0.00113 **
## Instagram_value                         -0.09842    0.10388  -0.947  0.36574
## Linkedin_value                           0.19780    0.12405   1.595  0.14190
## Snapchat_value                          -0.10269    0.20798  -0.494  0.63215
## Twitter_value                            0.24008    0.35856   0.670  0.51829
## Whatsapp_Wechat_value                    0.10295    0.08365   1.231  0.24660
## Youtube_value                            0.02357    0.14998   0.157  0.87827
## OTT_Netflix_Hulu_Prime_video_value      -0.05841    0.12293  -0.475  0.64488
## Reddit_value                            -0.11598    0.14469  -0.802  0.44141
## job_interview_calls                      0.72837    0.53742   1.355  0.20514
## networking_done_with_coffee_chats        0.04368    0.29401   0.149  0.88486
## learning_done_in_terms_of_items_created -0.28138    0.27268  -1.032  0.32642
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7985 on 10 degrees of freedom
## Multiple R-squared:  0.4434, Adjusted R-squared:  -0.1688
## F-statistic: 0.7242 on 11 and 10 DF,  p-value: 0.6985
```

```r
coefficients(fit)
```

```
##                             (Intercept)                         Instagram_value
##                              3.38572032                             -0.09842050
##                          Linkedin_value                          Snapchat_value
##                              0.19779691                             -0.10268982
##                           Twitter_value                   Whatsapp_Wechat_value
##                              0.24008117                              0.10295052
##                           Youtube_value      OTT_Netflix_Hulu_Prime_video_value
##                              0.02356621                             -0.05840958
##                            Reddit_value                     job_interview_calls
##                             -0.11598337                              0.72837196
##       networking_done_with_coffee_chats learning_done_in_terms_of_items_created
##                              0.04367604                             -0.28137960
```

```r
library(MASS)
step <- stepAIC(fit, direction="both")
```

```
## Start:  AIC=-3.25
## felt_the_entire_week ~ Instagram_value + Linkedin_value + Snapchat_value +
##     Twitter_value + Whatsapp_Wechat_value + Youtube_value + OTT_Netflix_Hulu_Prime_video_value +
##     Reddit_value + job_interview_calls + networking_done_with_coffee_chats +
##     learning_done_in_terms_of_items_created
##
##                                           Df Sum of Sq    RSS     AIC
## - networking_done_with_coffee_chats        1   0.01407 6.3896 -5.2001
## - Youtube_value                            1   0.01574 6.3913 -5.1944
## - OTT_Netflix_Hulu_Prime_video_value       1   0.14394 6.5195 -4.7575
## - Snapchat_value                           1   0.15543 6.5310 -4.7187
## - Twitter_value                            1   0.28583 6.6614 -4.2838
## - Reddit_value                             1   0.40968 6.7852 -3.8785
## - Instagram_value                          1   0.57231 6.9478 -3.3574
## <none>                                                 6.3755 -3.2486
## - learning_done_in_terms_of_items_created  1   0.67890 7.0544 -3.0225
## - Whatsapp_Wechat_value                    1   0.96566 7.3412 -2.1459
## - job_interview_calls                      1   1.17108 7.5466 -1.5387
## - Linkedin_value                           1   1.62100 7.9965 -0.2647
##
## Step:  AIC=-5.2
## felt_the_entire_week ~ Instagram_value + Linkedin_value + Snapchat_value +
##     Twitter_value + Whatsapp_Wechat_value + Youtube_value + OTT_Netflix_Hulu_Prime_video_value +
##     Reddit_value + job_interview_calls + learning_done_in_terms_of_items_created
##
##                                           Df Sum of Sq    RSS     AIC
## - Youtube_value                            1   0.02271 6.4123 -7.1221
## - Snapchat_value                           1   0.14326 6.5329 -6.7123
## - OTT_Netflix_Hulu_Prime_video_value       1   0.19634 6.5859 -6.5343
## - Twitter_value                            1   0.31760 6.7072 -6.1329
## - Reddit_value                             1   0.40306 6.7927 -5.8544
## - Instagram_value                          1   0.56822 6.9578 -5.3259
## <none>                                                 6.3896 -5.2001
## - learning_done_in_terms_of_items_created  1   0.68022 7.0698 -4.9746
## - Whatsapp_Wechat_value                    1   0.95246 7.3421 -4.1433
## - job_interview_calls                      1   1.18669 7.5763 -3.4524
## + networking_done_with_coffee_chats        1   0.01407 6.3755 -3.2486
## - Linkedin_value                           1   1.64753 8.0371 -2.1533
##
## Step:  AIC=-7.12
## felt_the_entire_week ~ Instagram_value + Linkedin_value + Snapchat_value +
##     Twitter_value + Whatsapp_Wechat_value + OTT_Netflix_Hulu_Prime_video_value +
##     Reddit_value + job_interview_calls + learning_done_in_terms_of_items_created
##
##                                           Df Sum of Sq    RSS     AIC
## - Snapchat_value                           1   0.17215 6.5845 -8.5392
## - OTT_Netflix_Hulu_Prime_video_value       1   0.17363 6.5859 -8.5343
## - Twitter_value                            1   0.30075 6.7131 -8.1137
## - Reddit_value                             1   0.38044 6.7928 -7.8541
## - Instagram_value                          1   0.56298 6.9753 -7.2707
```

```
## <none>                                               6.4123 -7.1221
## - learning_done_in_terms_of_items_created  1   0.75024 7.1626 -6.6878
## - Whatsapp_Wechat_value                    1   0.93212 7.3444 -6.1362
## + Youtube_value                            1   0.02271 6.3896 -5.2001
## + networking_done_with_coffee_chats        1   0.02104 6.3913 -5.1944
## - job_interview_calls                      1   1.32302 7.7353 -4.9953
## - Linkedin_value                           1   1.65446 8.0668 -4.0723
##
## Step:  AIC=-8.54
## felt_the_entire_week ~ Instagram_value + Linkedin_value + Twitter_value +
##     Whatsapp_Wechat_value + OTT_Netflix_Hulu_Prime_video_value +
##     Reddit_value + job_interview_calls + learning_done_in_terms_of_items_created
##
##                                           Df Sum of Sq    RSS     AIC
## - Reddit_value                             1   0.29735 6.8818 -9.5675
## <none>                                               6.5845 -8.5392
## - learning_done_in_terms_of_items_created  1   0.66755 7.2520 -8.4148
## - OTT_Netflix_Hulu_Prime_video_value       1   0.70119 7.2857 -8.3130
## - Twitter_value                            1   0.79954 7.3840 -8.0180
## + Snapchat_value                           1   0.17215 6.4123 -7.1221
## + Youtube_value                            1   0.05160 6.5329 -6.7123
## + networking_done_with_coffee_chats        1   0.00611 6.5784 -6.5597
## - Linkedin_value                           1   1.48267 8.0671 -6.0714
## - Instagram_value                          1   1.71262 8.2971 -5.4530
## - Whatsapp_Wechat_value                    1   1.79682 8.3813 -5.2309
## - job_interview_calls                      1   1.80555 8.3900 -5.2080
##
## Step:  AIC=-9.57
## felt_the_entire_week ~ Instagram_value + Linkedin_value + Twitter_value +
##     Whatsapp_Wechat_value + OTT_Netflix_Hulu_Prime_video_value +
##     job_interview_calls + learning_done_in_terms_of_items_created
##
##                                           Df Sum of Sq    RSS      AIC
## - learning_done_in_terms_of_items_created  1   0.42965 7.3115 -10.2352
## <none>                                               6.8818  -9.5675
## - Twitter_value                            1   0.76971 7.6515  -9.2350
## - OTT_Netflix_Hulu_Prime_video_value       1   0.80515 7.6870  -9.1333
## + Reddit_value                             1   0.29735 6.5845  -8.5392
## - Linkedin_value                           1   1.22294 8.1048  -7.9690
## + Snapchat_value                           1   0.08905 6.7928  -7.8541
## + networking_done_with_coffee_chats        1   0.01095 6.8709  -7.6026
## + Youtube_value                            1   0.00626 6.8756  -7.5875
## - Instagram_value                          1   1.44983 8.3317  -7.3616
## - job_interview_calls                      1   1.78437 8.6662  -6.4955
## - Whatsapp_Wechat_value                    1   1.91463 8.7965  -6.1673
##
## Step:  AIC=-10.24
## felt_the_entire_week ~ Instagram_value + Linkedin_value + Twitter_value +
##     Whatsapp_Wechat_value + OTT_Netflix_Hulu_Prime_video_value +
##     job_interview_calls
##
##                                           Df Sum of Sq    RSS      AIC
## - Twitter_value                            1   0.41685 7.7283 -11.0153
## - OTT_Netflix_Hulu_Prime_video_value       1   0.44635 7.7578 -10.9315
```

```
## <none>                                                 7.3115 -10.2352
## - Linkedin_value                          1   0.80531 8.1168  -9.9364
## + learning_done_in_terms_of_items_created  1   0.42965 6.8818  -9.5675
## - Instagram_value                         1   1.05710 8.3686  -9.2643
## + networking_done_with_coffee_chats       1   0.08350 7.2280  -8.4879
## - job_interview_calls                     1   1.37013 8.6816  -8.4564
## + Snapchat_value                          1   0.06601 7.2455  -8.4347
## + Reddit_value                            1   0.05944 7.2520  -8.4148
## + Youtube_value                           1   0.05443 7.2570  -8.3996
## - Whatsapp_Wechat_value                   1   1.55121 8.8627  -8.0023
##
## Step:  AIC=-11.02
## felt_the_entire_week ~ Instagram_value + Linkedin_value + Whatsapp_Wechat_value +
##     OTT_Netflix_Hulu_Prime_video_value + job_interview_calls
##
##                                          Df Sum of Sq    RSS      AIC
## - OTT_Netflix_Hulu_Prime_video_value      1   0.08759 7.8159 -12.7674
## - Linkedin_value                          1   0.71872 8.4470 -11.0590
## <none>                                                 7.7283 -11.0153
## - job_interview_calls                     1   1.02582 8.7541 -10.2734
## + Twitter_value                           1   0.41685 7.3115 -10.2352
## - Instagram_value                         1   1.10750 8.8358 -10.0691
## + Snapchat_value                          1   0.34127 7.3870 -10.0089
## - Whatsapp_Wechat_value                   1   1.16172 8.8900  -9.9345
## + Reddit_value                            1   0.12358 7.6047  -9.3700
## + Youtube_value                           1   0.09128 7.6370  -9.2767
## + learning_done_in_terms_of_items_created  1   0.07678 7.6515  -9.2350
## + networking_done_with_coffee_chats       1   0.02636 7.7020  -9.0905
##
## Step:  AIC=-12.77
## felt_the_entire_week ~ Instagram_value + Linkedin_value + Whatsapp_Wechat_value +
##     job_interview_calls
##
##                                          Df Sum of Sq    RSS     AIC
## - Linkedin_value                          1   0.64719 8.4631 -13.017
## <none>                                                 7.8159 -12.767
## + Snapchat_value                          1   0.42553 7.3904 -11.999
## - job_interview_calls                     1   1.19162 9.0075 -11.646
## - Whatsapp_Wechat_value                   1   1.27121 9.0871 -11.452
## + Reddit_value                            1   0.17296 7.6429 -11.260
## - Instagram_value                         1   1.36199 9.1779 -11.233
## + Youtube_value                           1   0.09129 7.7246 -11.026
## + OTT_Netflix_Hulu_Prime_video_value      1   0.08759 7.7283 -11.015
## + Twitter_value                           1   0.05809 7.7578 -10.931
## + learning_done_in_terms_of_items_created  1   0.04679 7.7691 -10.899
## + networking_done_with_coffee_chats       1   0.00387 7.8120 -10.778
##
## Step:  AIC=-13.02
## felt_the_entire_week ~ Instagram_value + Whatsapp_Wechat_value +
##     job_interview_calls
##
##                                          Df Sum of Sq     RSS     AIC
## <none>                                                  8.4631 -13.017
## + Linkedin_value                          1   0.64719  7.8159 -12.767
```

```
## - Instagram_value                          1    1.13255  9.5956 -12.254
## - Whatsapp_Wechat_value                    1    1.54766 10.0107 -11.322
## + Snapchat_value                           1    0.11643  8.3467 -11.322
## + Reddit_value                             1    0.11323  8.3499 -11.314
## + Twitter_value                            1    0.09719  8.3659 -11.271
## + learning_done_in_terms_of_items_created  1    0.08612  8.3770 -11.242
## + OTT_Netflix_Hulu_Prime_video_value       1    0.01606  8.4470 -11.059
## + Youtube_value                            1    0.00496  8.4581 -11.030
## + networking_done_with_coffee_chats        1    0.00080  8.4623 -11.019
## - job_interview_calls                      1    2.15468 10.6178 -10.027
```

```
fit2 <- lm(felt_the_entire_week ~ Instagram_value + Whatsapp_Wechat_value +
    job_interview_calls, data = APP_data)
```

Residual Analysis QQ plot is used in these residual analysis. We can conclude that most of residual points are located in a straight line. It satisfies normal distribution.

```
confint(fit2,level=0.95)
```

```
##                             2.5 %    97.5 %
## (Intercept)            2.54631223 3.8923436
## Instagram_value       -0.19785737 0.0297300
## Whatsapp_Wechat_value -0.01302441 0.1779132
## job_interview_calls    0.01182008 1.2592082
```

```
fitted(fit2)
```

```
##        1        2        3        4        5        6        7        8 
## 3.007549 2.914133 3.712836 4.475049 3.466661 4.012792 3.262572 3.455326 
##        9       10       11       12       13       14       15       16 
## 3.634724 3.287481 2.916761 3.241967 3.736560 2.996906 3.364320 3.678618 
##       17       18       19       20       21       22 
## 3.496136 3.220683 3.478091 3.222213 3.882471 3.536151
```

```
residuals(fit2)
```

```
##            1            2            3            4            5            6 
## -0.007549419  0.085866852  0.287163972 -0.475048787 -0.466661048  0.987207984 
##            7            8            9           10           11           12 
##  0.737427807  0.544674120 -0.634724119 -1.287481479  1.083239357 -0.241966718 
##           13           14           15           16           17           18 
##  0.263439858  0.003094405 -0.364320230  1.321381834 -0.496136217 -0.220683282 
##           19           20           21           22 
## -0.478090655 -0.222212542  0.117529247 -0.536150941
```

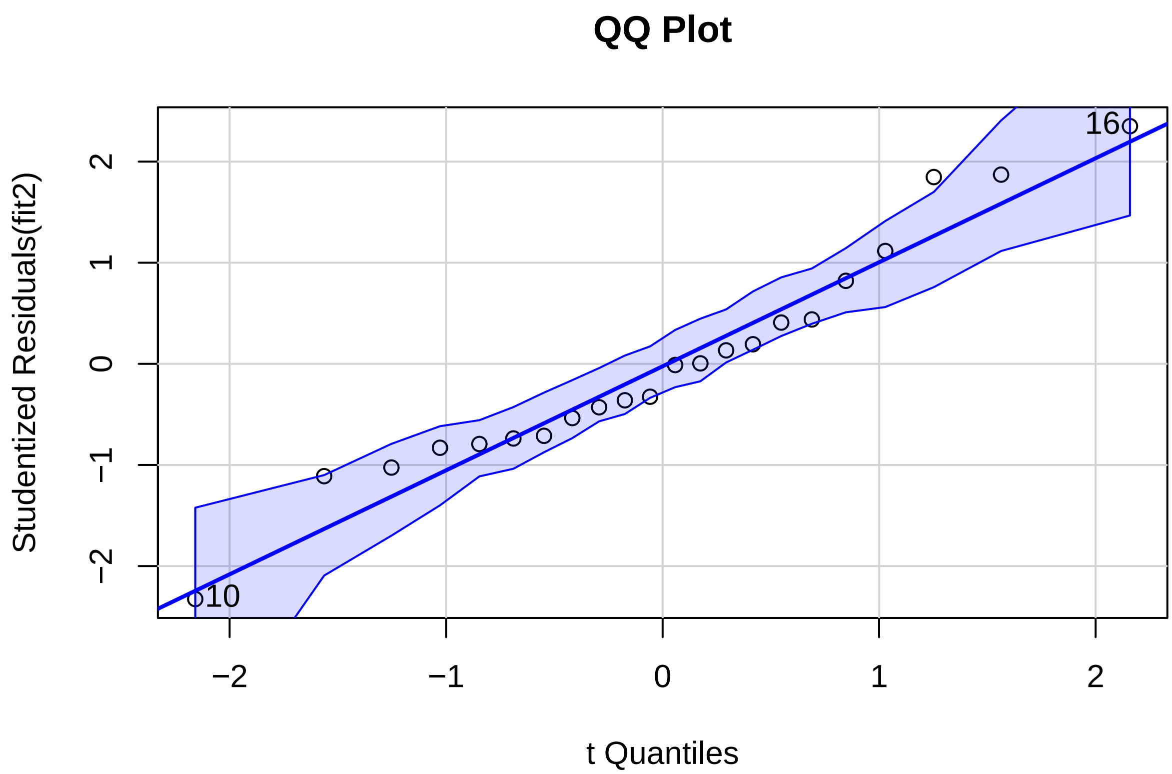
```
library(car)
```

```
## Loading required package: carData
```

```
## 
## Attaching package: 'car'
```

```
## The following object is masked from 'package:psych':
##
##     logit
```

```r
qqPlot(fit2, main="QQ Plot")
```




```
## [1] 10 16
```

Prediction We set a data point with Instagram_value = 5, Whatsapp_Wechat_value = 5 and, job_interview_calls = 0, then the feeling score of the entire week we predict is approximate to 3

```r
predict.lm(fit2, data.frame(Instagram_value = 5, Whatsapp_Wechat_value = 5, job_interview_calls = 0))
```

```
##        1
## 3.211231
```

Model Accuracy The accuracy is based on summary of the model and we also calculate the MSE and RMSE for the model. The MSE is 0.3846858 and RMSE is 0.6202304.

```r
summary(fit2)
```

```
##
```

```
## Call:
## lm(formula = felt_the_entire_week ~ Instagram_value + Whatsapp_Wechat_value +
##     job_interview_calls, data = APP_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2875 -0.4729 -0.1141  0.2812  1.3214
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            3.21933    0.32034  10.050 8.28e-09 ***
## Instagram_value       -0.08406    0.05416  -1.552   0.1381
## Whatsapp_Wechat_value  0.08244    0.04544   1.814   0.0863 .
## job_interview_calls    0.63551    0.29687   2.141   0.0462 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6857 on 18 degrees of freedom
## Multiple R-squared:  0.2612, Adjusted R-squared:  0.138
## F-statistic: 2.121 on 3 and 18 DF,  p-value: 0.1332
```

```r
predictions <- predict(fit2, APP_data)
mse <- mean((APP_data$felt_the_entire_week - predictions)^2)
rmse <- sqrt(mse)
cat("MSE: ", mse, "\n")
```

```
## MSE:  0.3846858
```

```r
cat("RMSE: ", rmse, "\n")
```

```
## RMSE:  0.6202304
```

lda model

Model development Running the following code, we build a linear discriminant analysis model to classify social media data. Its independent variables "Instagram_value", "Linkedin_value", "Snapchat_value", "Twitter_value", "Whatsapp_Wechat_value", "Youtube_value", "OTT_Netflix_Hulu_Prime_video_value", "Reddit_value", "job_interview_calls", "networking_done_with_coffee_chats", "learning_done_in_terms_of_items_crea
The dependent variable is "Tired_waking_up_in_morning".

```r
library(MASS)
library(ggplot2)
library(memisc)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'memisc'
```

```
## The following object is masked from 'package:car':
##
##     recode
```

```
## The following object is masked from 'package:magrittr':
##
##     %$%

## The following object is masked from 'package:ggplot2':
##
##     syms

## The following objects are masked from 'package:stats':
##
##     contr.sum, contr.treatment, contrasts

## The following object is masked from 'package:base':
##
##     as.array
```

```
library(ROCR)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:memisc':
##
##     collect, recode, rename, syms

## The following object is masked from 'package:car':
##
##     recode

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(klaR)
library(readr)
APP_data <- read_csv("Dataset/Social Media_cleaned.csv")
```

```
## New names:
## * `Hours_spent` -> `Hours_spent...3`
## * `Hours_spent` -> `Hours_spent...6`
## * `Hours_spent` -> `Hours_spent...9`
## * `Hours_spent` -> `Hours_spent...15`
## * `Hours_spent` -> `Hours_spent...18`
## * `Hours_spent` -> `Hours_spent...21`
## * `Hours_spent` -> `Hours_spent...24`
```

```
## Rows: 23 Columns: 33
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (15): ID, Instagram, Linkedin, Snapchat, Twitter, Whatsapp_Wechat, Yout...
## dbl  (12): Instagram_value, Linkedin_value, Snapchat_value, Twitter_value, W...
## time  (6): Hours_spent...3, Hours_spent...6, Hours_spent...9, Hours spent, H...
##
## 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.
```

```
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 8.65 0.17
##  $ 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 0 ...
##  $ 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 Media" "Socia
##  $ job_interview_calls                     : num [1:22] 0 0 0 2 0 0 0 0 1 0 ...
##  $ networking_done_with_coffee_chats       : num [1:22] 0 1 0 0 2 0 2 0 0 0 ...
##  $ 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" ...
##  $ felt_the_entire_week                    : num [1:22] 3 3 4 4 3 5 4 4 3 2 ...
```

```
APP_data$Tired_waking_up_in_morning <- as.factor(APP_data$Tired_waking_up_in_morning)
r <- lda(formula = Tired_waking_up_in_morning ~ Instagram_value + Linkedin_value + Snapchat_value + Twi
head(r$class)
```

```
## NULL
```

```
summary(r)
```

```
##         Length Class  Mode
## prior    2     -none- numeric
## counts   2     -none- numeric
## means   20     -none- numeric
```

```
## scaling 10     -none- numeric
## lev      2     -none- character
## svd      1     -none- numeric
## N        1     -none- numeric
## call     3     -none- call
## terms    3     terms  call
## xlevels  0     -none- list
```

Model Acceptance

In this model, we can see that the first linear discriminant explains all the between-group variance in the house data. Therefore, the model can be used to analyze the house data.

```
r$svd
```

```
## [1] 4.219726
```

```
(prop = r$svd^2/sum(r$svd^2))
```

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

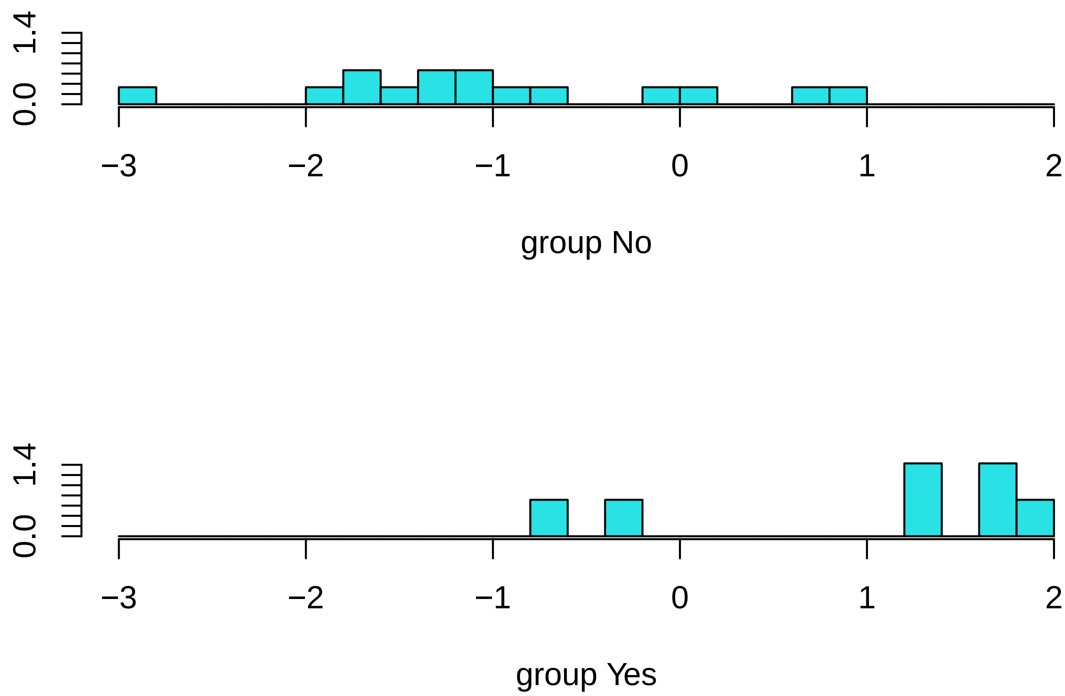
Residual Analysis

Since this model is a classification model, we focus on the posterior value of the model. The following code is to train the new model r3 and the model is used to test the model and display the predicted result and posterior probability. The plots of r1 and r3 shows how the model distinguishes between different furniture categories on training data

```
r2 <- lda(formula = Tired_waking_up_in_morning ~ Instagram_value + Linkedin_value + Snapchat_value + Twi
head(r2$posterior, 3)
```

```
##          No        Yes
## 1 0.9953703 0.00462970
## 2 0.9761574 0.02384258
## 3 0.1196170 0.88038297
```

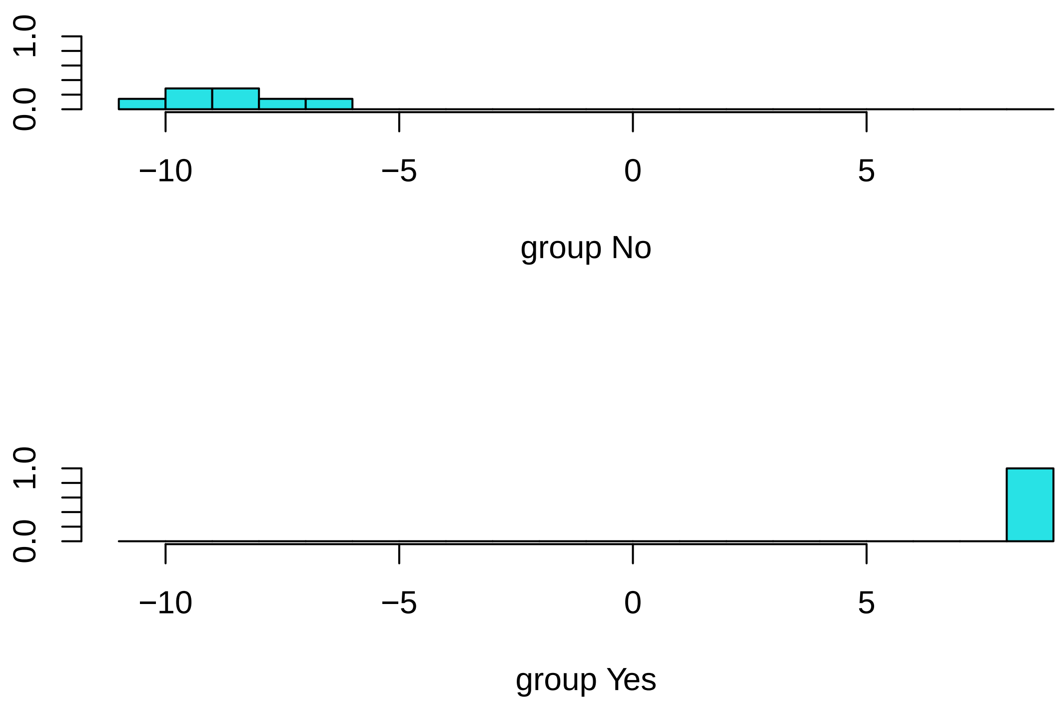
```
plot(r)
```




```
train <- sample(22, 10)
r3 <- lda(Tired_waking_up_in_morning ~ Instagram_value + Linkedin_value + Snapchat_value + Twitter_valu
          APP_data,
          prior = c(1,1)/2,
          subset = train)
```

```
## Warning in lda.default(x, grouping, ...): variables are collinear
```

```
plot(r3)
```




```
plda = predict(object = r3, # predictions
               newdata = APP_data[-train, ])
head(plda$class)
```

```
## [1] No  No  Yes No  No  No
## Levels: No Yes
```

```
head(plda$posterior, 6) # posterior prob.
```

```
##             No           Yes
## 1 1.000000e+00 3.025631e-207
## 2 1.000000e+00  5.999510e-46
## 3 8.754296e-06  9.999912e-01
## 4 1.000000e+00 2.848214e-313
## 5 1.000000e+00  3.952947e-22
## 6 1.000000e+00  7.989041e-87
```

```
head(plda$x, 3)
```

```
##           LD1
## 1 -27.7465809
## 2  -6.0757197
## 3   0.6795299
```

Prediction

The data will be predicted in the model and the predicted first linear discriminant scores of the are as follows.

```
r <- lda(Tired_waking_up_in_morning ~ Instagram_value + Linkedin_value + Snapchat_value + Twitter_value
          APP_data,
          prior = c(1,1)/2,)
prop.lda = r$svd^2/sum(r$svd^2)
plda <- predict(object = r,
                newdata = APP_data)
dataset = data.frame(furniture = APP_data[,"Tired_waking_up_in_morning"],lda = plda$x)
dataset$LD1
```

```
##  [1] -1.64940523 -1.50160901  0.43417209  0.07298424  1.88111904 -3.14085528
##  [7]  1.36594301 -0.18538993 -1.02890956  0.71923625 -2.10060140 -1.60142155
## [13]  1.41842173 -0.75351014 -1.35531594 -0.66500724 -0.69634235  1.42556247
## [19] -0.12856207 -1.01810093  1.70927156 -1.14444620
```

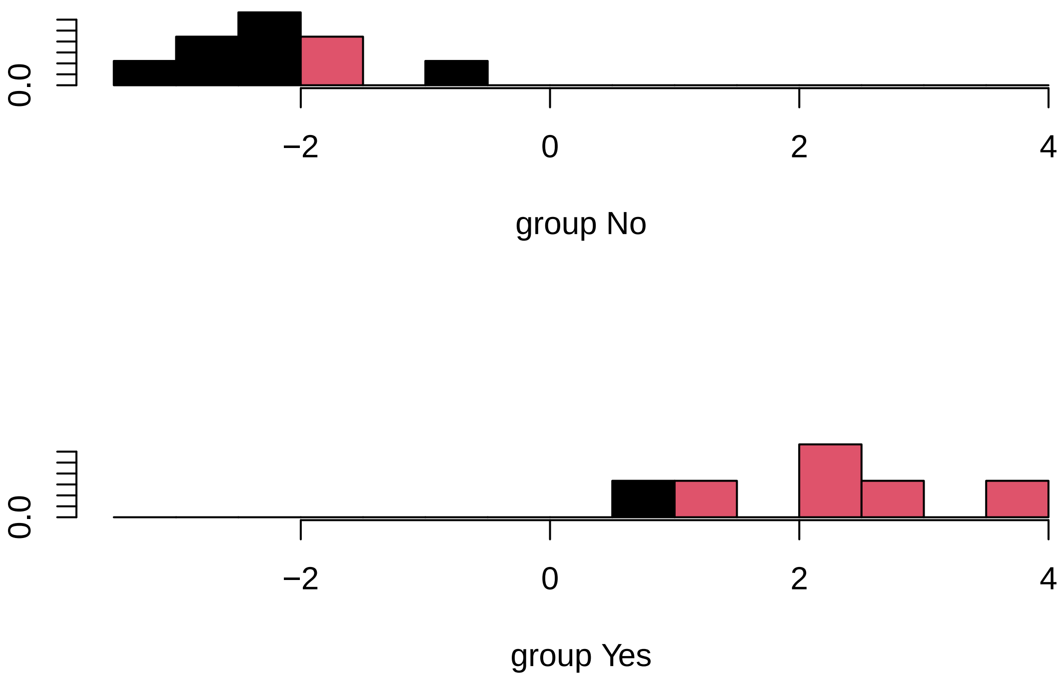
Model Accuracy

To observe the performance of the model, the test set is used to approximate accuracy.

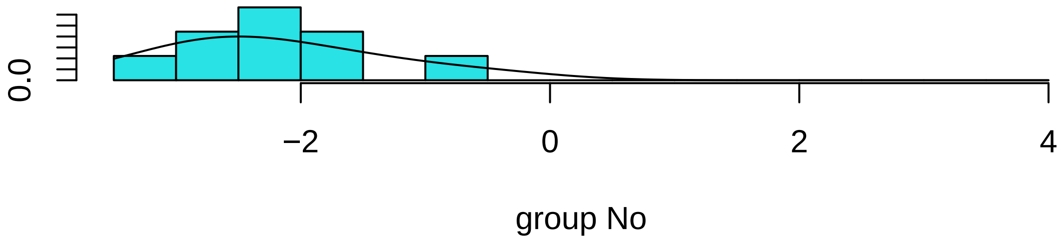
```
set.seed(101)
sample_n(APP_data,10)
```

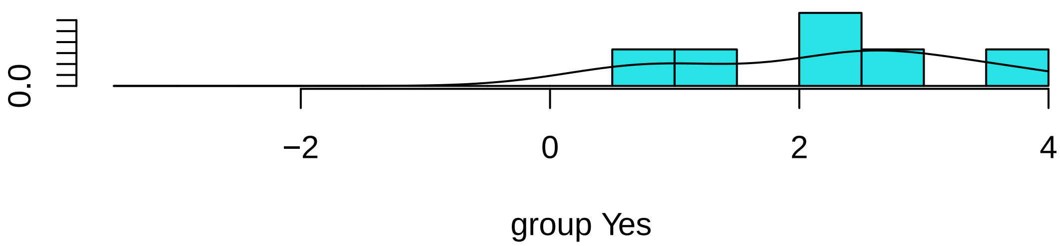
```
## # A tibble: 10 x 25
##    ID      Instagram Instagram_value Linkedin Linkedin_value Snapchat
##    <chr>   <chr>               <dbl> <chr>             <dbl> <chr>
##  1 yh2020  Yes                  8.65 Yes               10    Yes
##  2 hahah   Yes                  6    Yes                3    Yes
##  3 sss32   Yes                  9.8  Yes                0.8  No
##  4 Patty   Yes                  3.77 Yes                7    Yes
##  5 2134    Yes                  5.67 Yes                3.92 No
##  6 azhena  Yes                  8    Yes                2    No
##  7 vp1234  Yes                  7    Yes                5    yes
##  8 MVA37@S Yes                  6.8  Yes                1.92 Yes
##  9 AKIRA   Yes                  4.65 Yes                3.75 Yes
## 10 peace   Yes                  7.73 Yes                5.2  Yes
## # i 19 more variables: Snapchat_value <dbl>, Twitter <chr>,
## #   Twitter_value <dbl>, Whatsapp_Wechat <chr>, Whatsapp_Wechat_value <dbl>,
## #   Youtube <chr>, Youtube_value <dbl>, `OTT_Netflix_Hulu_Prime video` <chr>,
## #   OTT_Netflix_Hulu_Prime_video_value <dbl>, Reddit <chr>, Reddit_value <dbl>,
## #   `Application_type_Social media_OTT_Learning` <chr>,
## #   job_interview_calls <dbl>, networking_done_with_coffee_chats <dbl>,
## #   learning_done_in_terms_of_items_created <dbl>, Mood_Productivity <chr>, ...
```

```
training_sample <- sample(c(TRUE, FALSE), nrow(APP_data), replace = T, prob = c(0.75,0.25))
train <- APP_data[training_sample, ]
test <- APP_data[!training_sample, ]
lda.waking <- lda(Tired_waking_up_in_morning ~ Instagram_value + Linkedin_value + Snapchat_value + Twit
plot(lda.waking, col = as.integer(train$Tired_waking_up_in_morning))
```




```
# Sometime bell curves are better
plot(lda.waking, dimen = 1, type = "b")
```






```r
lda.train <- predict(lda.waking)
train$lda <- lda.train$class
table(train$lda,train$Tired_waking_up_in_morning)
```

```
##
##       No Yes
##   No   9   0
##   Yes  0   6
```

```r
# running accuracy on the training set shows how good the model is. It is not an indication of "true" a
lda.test <- predict(lda.waking,test)
test$lda <- lda.test$class
table(test$lda,test$Tired_waking_up_in_morning)
```

```
##
##       No Yes
##   No   1   0
##   Yes  5   1
```

Question 4 Model Insights (10 points)

We uses clustering techniques, indicating distinct groups or patterns within the data, which could be crucial for understanding different user behaviors and tailoring specific interventions or marketing strategies. The hierarchical clustering results underscore the diversity in social media usage among individuals, which might correlate with their professional networking activities and job-related outcomes.

Problem 5 Learnings and Takeaways (20 points)

Learnings: The most important thing learned is diversity in model evaluation: By evaluating model performance using different statistical metrics such as R-squared, adjusted R-squared, and F-statistics, you can gain a comprehensive understanding of the model's explanatory and predictive power. This helps in selecting the most suitable model for prediction or classification. Take aways: The necessity for careful selection of variables in model building, as shown by the stepwise regression outcomes. The application of clustering methods can reveal hidden patterns and segments within the data, which are crucial for targeted marketing and investment strategies in social media usage. PCA and factor analysis are powerful tools for reducing complexity in data, allowing easier interpretation and visualization.