INSH5301 Intro Computational Statistics

Ali Banijamali

04/06/2020

For this homework you'll need to use some real world data to answer some research questions using multiple regression. The data, along with the data description, can be downloaded from the course material section in Blackboard. You can also downlod this dataset using the AER package.

AER package: Just Run library(AER) first, and then data("CollegeDistance")

For this homework use the bfi dataset (25 personality items thought to boil down to a few core personality types) from the psych package. You can load the data using, for instance, data(bfi) after loading the psych package; you may need to clean it a bit first with na.omit() to remove the observations with na items, or else impute those missing items. It might also help to use scale() on your dataset before analysis. scale() takes all your variables (columns) and rescales them to have a mean of 0 and a sd of 1, so that you can more easily compare all your factors or clusters to see which are larger or smaller.

For the factor analysis, you may use any of the methods covered in the lesson – they should all produce similar results, though princomp and prcomp might be simplest. You don't have to interpret everything, say, fa() outputs, which produce a lot of output – is easier to use str() to examine the output of your function and find the quantities you want.

Also, because some of the methods when deciding the number of factors and number of clusters are not objective, don't worry about not getting the right number. But provide an explanation to why you choose a the particular number, your reasoning should be based on the described methodologies in the learning module.

```r
# Required Packages:
# install.packages("psych")
library(psych)

# Reading the data:
bfi <- data.frame(bfi)

# Removing NAs:
bfi <- na.omit(bfi)

# Scaling to mean=0, sd=1:
# (We could also write a simple function to do this, we used rescale from psych package)
bfi_scaled <- data.frame(apply(bfi, 2, FUN=function(x){rescale(x, mean=0, sd=1)}))
colnames(bfi_scaled) <- colnames(bfi) # Adding col.names

# I am throwing away gender, education and age because I think we want
# to cluster the behaviors not age, gender and education
```

```r
bfi_scaled <- bfi_scaled[1:25]
```

After running a factor analysis or PCA, be sure to discuss and interpret the results:

ANS.

```r
# PCA using singular value decomposition (SVD):
pca.SVD <- prcomp(bfi_scaled)
# 1st component:
pca.SVD$rotation[ ,1][order(pca.SVD$rotation[ ,1])] # order() for sorting
```

```
##          E2          N4          C5          C4          E1          N1
## -0.28697697 -0.24569198 -0.23036582 -0.21728787 -0.19954871 -0.18867456
##          N2          N3          N5          A1          O5          O2
## -0.18298575 -0.18081192 -0.16606019 -0.10521298 -0.09884416 -0.09617060
##          O4          C2          C3          O1          C1          O3
## -0.04068912  0.15460703  0.15768579  0.15966293  0.15990507  0.18772797
##          A4          A2          A3          E5          E3          A5
##  0.19534967  0.21411726  0.24417964  0.24919230  0.25577592  0.26665860
##          E4
##  0.28063193
```

```r
# 2nd component:
pca.SVD$rotation[ ,2][order(pca.SVD$rotation[ ,2])]
```

```
##          E1          O5          E2          A1          C3          O2
## -0.13164003 -0.03610246 -0.02435069 -0.02086753  0.02855128  0.06506317
##          A4          C1          C4          E4          C5          A5
##  0.08128399  0.08397180  0.09081069  0.10941536  0.11464585  0.11868968
##          C2          O1          O3          E5          O4          A2
##  0.11965427  0.12739072  0.18682163  0.18774466  0.19075492  0.19435996
##          A3          E3          N4          N5          N2          N1
##  0.19738145  0.21747051  0.28372463  0.30348850  0.39483119  0.39805944
##          N3
##  0.40758215
```

```r
# 3rd component:
pca.SVD$rotation[ ,3][order(pca.SVD$rotation[ ,3])]
```

```
##          O2          C4          O5          E4          A5          A3
## -0.31347563 -0.29649657 -0.28170324 -0.25158783 -0.20871666 -0.20368897
##          C5          A2          A4          N5          E3          N1
## -0.17437239 -0.16555821 -0.16101138 -0.06053846 -0.06047071  0.02535326
##          N3          N2          N4          E5          A1          E2
##  0.03276540  0.06794496  0.07347423  0.07752656  0.12294013  0.15800444
##          O3          O4          E1          O1          C3          C2
##  0.17910595  0.18940617  0.19524364  0.23049690  0.23766890  0.31672021
##          C1
##  0.34818699
```

```r
# PCA using covariance/eigenvector:
pca.COV <- princomp(bfi_scaled)
# 1st component:
pca.COV$loadings[ ,1][order(pca.COV$loadings[ ,1])]
```

```
##          E4          A5          E3          E5          A3          A2
## -0.28063193 -0.26665860 -0.25577592 -0.24919230 -0.24417964 -0.21411726
##          A4          O3          C1          O1          C3          C2
## -0.19534967 -0.18772797 -0.15990507 -0.15966293 -0.15768579 -0.15460703
##          O4          O2          O5          A1          N5          N3
##  0.04068912  0.09617060  0.09884416  0.10521298  0.16606019  0.18081192
##          N2          N1          E1          C4          C5          N4
##  0.18298575  0.18867456  0.19954871  0.21728787  0.23036582  0.24569198
##          E2
##  0.28697697
```

```
# 2nd component:
pca.COV$loadings[ ,2][order(pca.COV$loadings[ ,2])]
```

```
##          N3          N1          N2          N5          N4          E3
## -0.40758215 -0.39805944 -0.39483119 -0.30348850 -0.28372463 -0.21747051
##          A3          A2          O4          E5          O3          O1
## -0.19738145 -0.19435996 -0.19075492 -0.18774466 -0.18682163 -0.12739072
##          C2          A5          C5          E4          C4          C1
## -0.11965427 -0.11868968 -0.11464585 -0.10941536 -0.09081069 -0.08397180
##          A4          O2          C3          A1          E2          O5
## -0.08128399 -0.06506317 -0.02855128  0.02086753  0.02435069  0.03610246
##          E1
##  0.13164003
```

```
# 3rd component:
pca.COV$loadings[ ,3][order(pca.COV$loadings[ ,3])]
```

```
##          O2          C4          O5          E4          A5          A3
## -0.31347563 -0.29649657 -0.28170324 -0.25158783 -0.20871666 -0.20368897
##          C5          A2          A4          N5          E3          N1
## -0.17437239 -0.16555821 -0.16101138 -0.06053846 -0.06047071  0.02535326
##          N3          N2          N4          E5          A1          E2
##  0.03276540  0.06794496  0.07347423  0.07752656  0.12294013  0.15800444
##          O3          O4          E1          O1          C3          C2
##  0.17910595  0.18940617  0.19524364  0.23049690  0.23766890  0.31672021
##          C1
##  0.34818699
```

```
# In order to interpret these, let's look at the keys in bfi table:
bfi.keys
```

```
## $agree
## [1] "-A1" "A2"  "A3"  "A4"  "A5"
##
## $conscientious
## [1] "C1"  "C2"  "C3"  "-C4" "-C5"
##
## $extraversion
## [1] "-E1" "-E2" "E3"  "E4"  "E5"
##
## $neuroticism
## [1] "N1" "N2" "N3" "N4" "N5"
##
## $openness
## [1] "O1"  "-O2" "O3"  "O4"  "-O5"
```

We can see that the 2 approaches produce the same result (only the order is different which is not really meaningful). In he first principal components, the extreme sides are E2 (Extraversion2) and N4 (Neuroticism4) from one side to A5 (Agree5) and E4 (Extraversion 4) from the other side [E2, N4, . . . , A5, E4]. On the second principal component we have E1 (Extraversion1) and O5 (Openness5) from one side to N1 (Neuroticism1) and N3 (Neuroticism3) on the other side [E1, O5, . . . , N1, N3].
bfi dataset represents 25 personality items in 5 general categories:
- Agree (A)
- Conscientious (C)
- Extraversion (E)
- Neuroticism (N)
- Openness (O)

which are thought to represent personality.
So according to the details of these codes, the first PC refers to the social skills of people w/ one side having people with no interest in approaching others and socializing and the other side people who make friends easily and who make others around them feel at ease.
The second important component is about emotional state of people with one side people who are indifferent about other people's feelings and are not talkative and the other side people who are more emotional and get angry easily and have frequent mood swings.

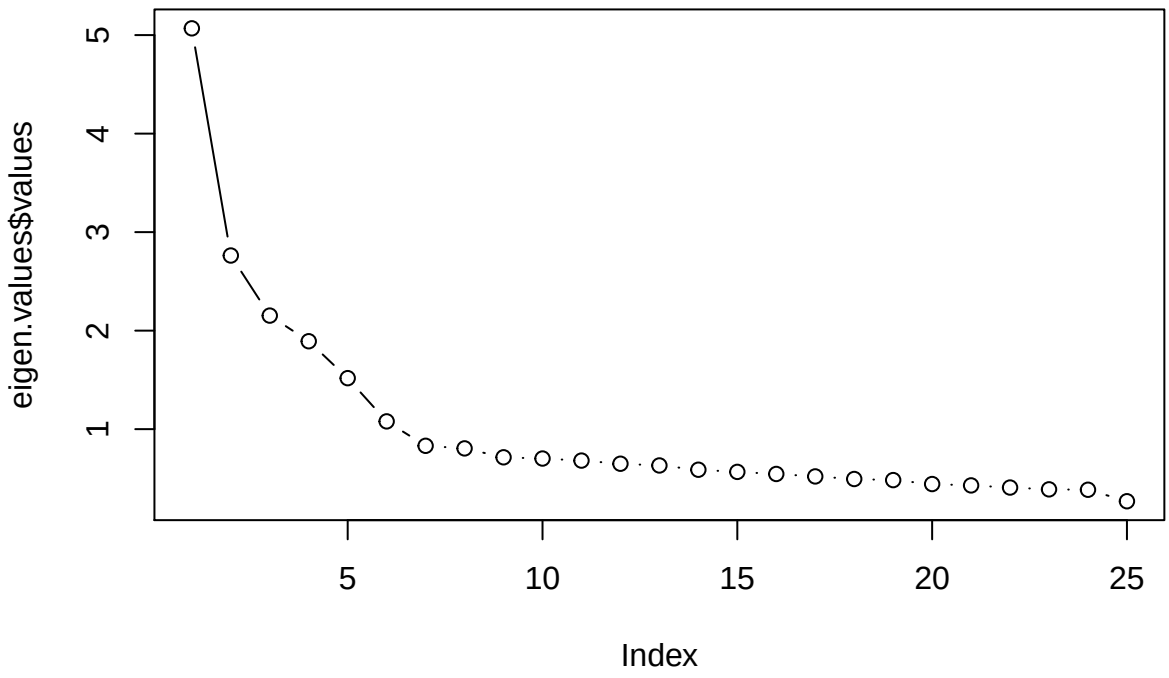
1. Examine the factor eigenvalues in the dataset. Plot these in a scree plot and use the "elbow" test to guess how many factors one should retain.

ANS.

```
# To get the elbow test, I am manually calculating the eigen values:

# PCA using manual method:
# 1. Find the covariance matrix:
cov.matrix <- cov(bfi_scaled)
# 2. Calculating the eigen values:
eigen.values <- eigen(cov.matrix)
# 3. The PC's:
# eigen1 <- eigenm$vectors[,1]

# 4. The scree plot:
plot(eigen.values$values, type="b")
```

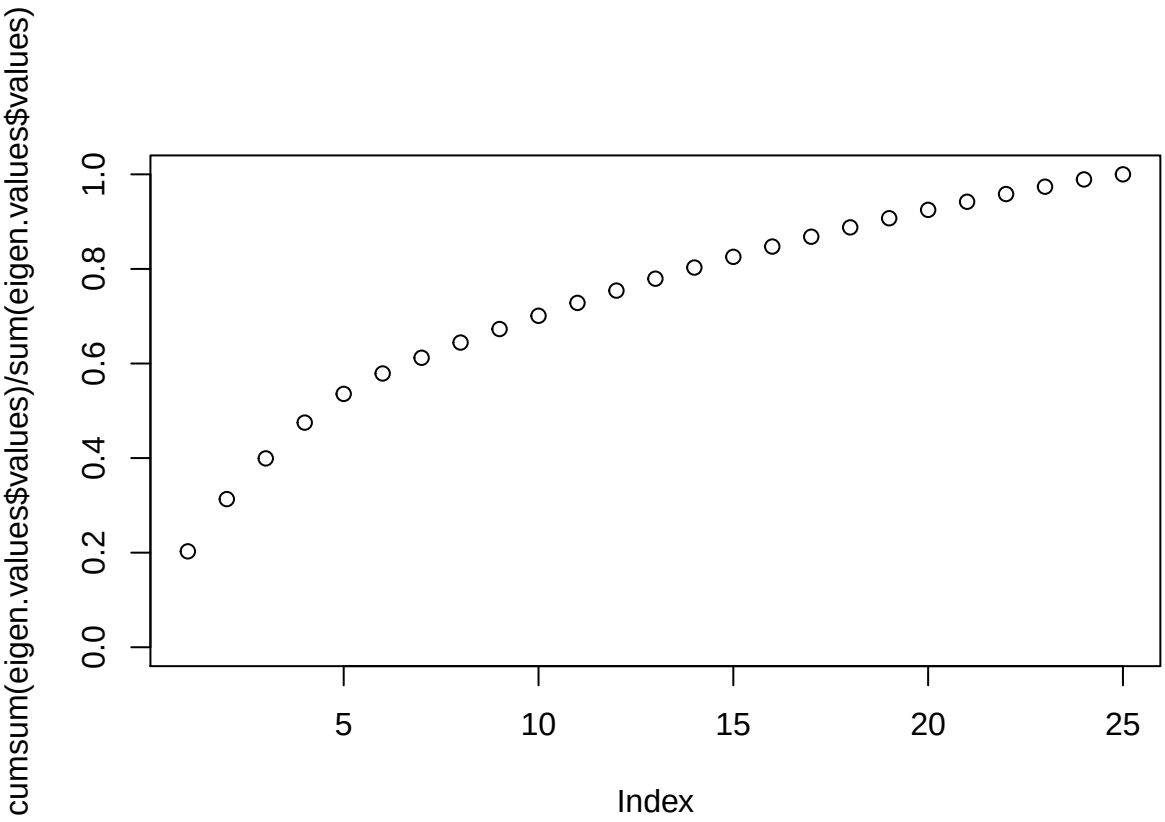


I would say that from 7th principal component, the curve gets flat, so at that point on we dn't want the PCs. If we wanted to go by the other method to keep only eigen values above 1, we would have done the same. So, we'll say the points from 7th principal component are probably just noise.

2. How many factors are needed to explain 50 percent of the total variance in the dataset?

ANS.
To answer this question, we have to use the cumulative variance

```
plot(cumsum(eigen.values$values)/sum(eigen.values$values), ylim=c(0,1))
```



I would say, somewhere between 4th and 4th component, explains 50 percent of the total variance of the data.

3. Examine the loadings of the factors on the variables (sometimes called the "rotation" in the function output) – ie, the projection of the factors on the variables – focusing on just the first one or two factors. Sort the variables by their loadings, and try to interpret what the first one or two factors "mean." This may require looking more carefully into the dataset to understand exactly what each of the variables were measuring. You can find more about the data in the psych package using ?psych.

ANS.
I have completely answered this part before Question 1. (I could copy and paste it here again but it was redundant)

Next perform a cluster analysis with the same dataset.

4. First use k-means and examine the centers using two and three centers. How are they similar to and different from the factor loadings of the first couple factors?

ANS.

```
set.seed(1)
# Running kmeans w/ 2 centers and 25 random start:
kmeans2.bfi <- kmeans(bfi_scaled, centers=2, nstart=25)
```

```
kmeans2.centroids <- kmeans2.bfi$centers

kmeans2.topvars_centroid1 <- kmeans2.centroids[1, order(kmeans2.centroids[1, ])]
kmeans2.topvars_centroid1
```

```
##          E2          N4          C5          C4          E1          N1
## -0.48598456 -0.40581189 -0.36588286 -0.34916948 -0.34411900 -0.32243896
##          N2          N3          N5          A1          O5          O2
## -0.30435328 -0.29844546 -0.25834400 -0.17487016 -0.17171722 -0.15372774
##          O4          C2          C1          C3          O1          O3
## -0.03901529  0.20411492  0.20623777  0.21101543  0.25523565  0.30802986
##          A4          A2          E5          A3          E3          A5
##  0.31578114  0.35636692  0.38851850  0.41409109  0.43071349  0.45911976
##          E4
##  0.46547511
```

```
kmeans2.topvars_centroid2 <- kmeans2.centroids[2, order(kmeans2.centroids[2, ])]
kmeans2.topvars_centroid2
```

```
##          E4          A5          E3          A3          E5          A2
## -0.57741302 -0.56952933 -0.53429188 -0.51367211 -0.48194980 -0.44206638
##          A4          O3          O1          C3          C1          C2
## -0.39172049 -0.38210518 -0.31661496 -0.26176062 -0.25583403 -0.25320067
##          O4          O2          O5          A1          N5          N3
##  0.04839772  0.19069633  0.21301194  0.21692311  0.32047081  0.37021592
##          N2          N1          E1          C4          C5          N4
##  0.37754444  0.39997939  0.42687307  0.43313809  0.45387072  0.50340192
##          E2
##  0.60285460
```

```
# Running kmeans w/ 3 centers and 25 random start:
kmeans3.bfi <- kmeans(bfi_scaled, centers=3, nstart=25)

kmeans3.centroids <- kmeans3.bfi$centers

kmeans3.topvars_centroid1 <- kmeans3.centroids[1, order(kmeans3.centroids[1, ])]
kmeans3.topvars_centroid1
```

```
##          E4          E3          A3          A5          E5          A2
## -0.86095868 -0.85202834 -0.81903389 -0.79200860 -0.79132977 -0.71556499
##          A4          O3          O1          C2          C3          C1
## -0.58436508 -0.56925264 -0.44424068 -0.38432302 -0.37429611 -0.37230071
##          O4          N3          O2          N1          N2          N5
##  0.03220906  0.09905386  0.10937457  0.11834123  0.13212769  0.13898040
##          O5          A1          N4          C4          C5          E1
##  0.22460812  0.23546645  0.40567625  0.41880655  0.45121007  0.69166447
##          E2
##  0.75491351
```

```
kmeans3.topvars_centroid2 <- kmeans3.centroids[2, order(kmeans3.centroids[2, ])]
kmeans3.topvars_centroid2
```

```
##         N4         N3         N1         N2         E2         N5
## -0.6528027 -0.6227137 -0.6214409 -0.6134448 -0.5278221 -0.5121771
##         C5         C4         E1         A1         O2         O4
## -0.4880386 -0.4261516 -0.3122541 -0.2066443 -0.2040121 -0.1796505
```

```
##         O5         C2         C1         O1         C3         O3
## -0.1718240  0.1537441  0.1838138  0.2230236  0.2339815  0.2387011
##         A2         A4         E5         A3         E3         A5
##  0.2980642  0.3112422  0.3314147  0.3470127  0.3829768  0.4466571
##         E4
##  0.4770411
```

```
kmeans3.topvars_centroid3 <- kmeans3.centroids[3, order(kmeans3.centroids[3, ])]
kmeans3.topvars_centroid3
```

```
##          E1          E2          O5          C3          A1          C1
## -0.23029309  0.01674889  0.02617452  0.03143428  0.06544973  0.10139836
##          O1          A4          A5          C2          E4          O2
##  0.11569971  0.12657401  0.13596715  0.15628960  0.15999063  0.18526545
##          C4          O3          O4          C5          A2          E3
##  0.20043144  0.21575550  0.22597490  0.25739137  0.27406180  0.28608701
##          E5          A3          N4          N5          N2          N1
##  0.30051192  0.30530514  0.53822529  0.59802883  0.74992160  0.77489754
##          N3
##  0.79562678
```

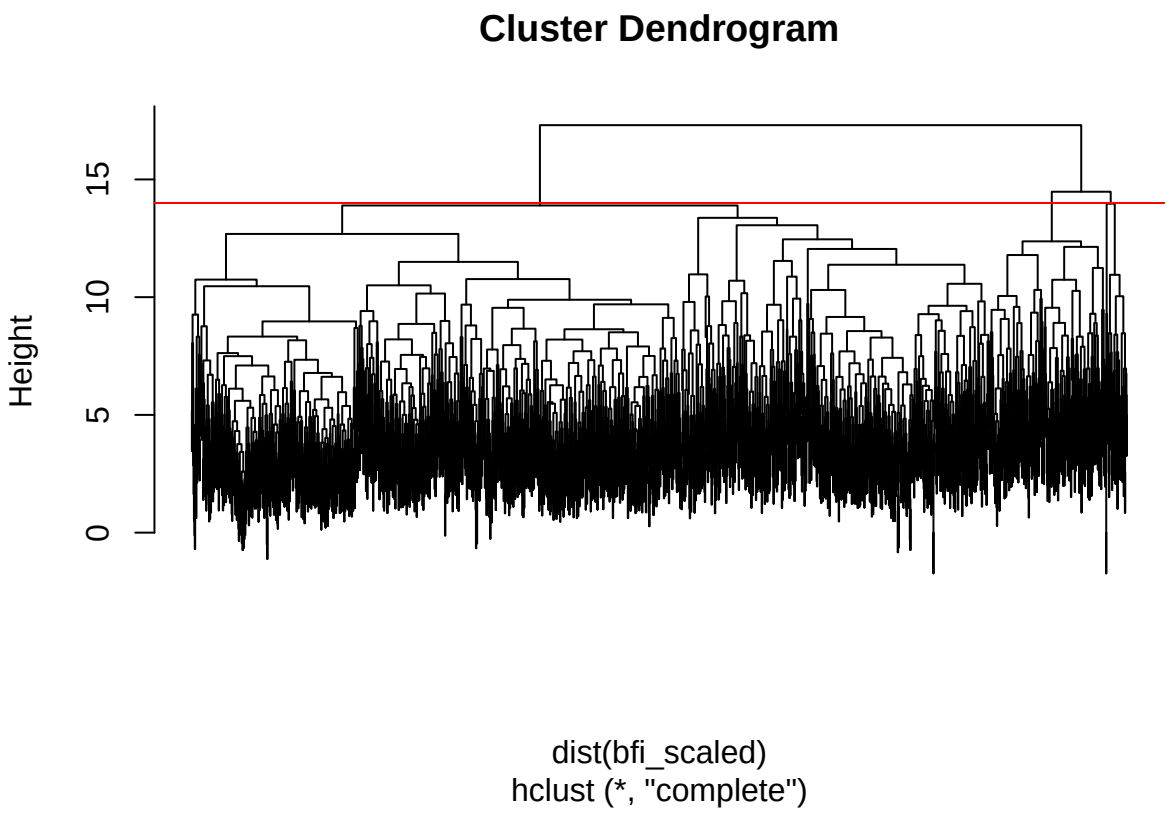
For 2 clusters, the clusters are the same w/ different order and the same as one of the factors in the previous part.
The first 2 PCs were: [E2, N4, . . . , A5, E4] and [E1, O5, . . . , N1, N3]
Here for 2 clusters model, we have: [E2, N4, . . . , A5, E4] and [E4, A5, . . . , N4, E2]
and for the 3 cluster model: [E4, E3, . . . , E1, E2], [N4, N3, . . . , A5, E4] and [E1, E2, . . . , N1, N3]
Generally, there is a strong overlap between clusters and factors, but a key difference is that factors are inherently dimensional and oppositional: there are two directions for every factor, and we often see clear oppositions at either end.
Clusters are less oppositional: we can talk about variables that score highly , but it is less illuminating to look at the variables that score weakly, since those are just things that aren't near the cluster, which is more of a grab-bag when the cluster is some specific set of variables.
That's why instead of looking at the whole ranges, we should look at the tails of the above distributions which have higher scores:
For 2 cluster model: [E4, A5, E3, . . .] and [E2, N4, C5, . . .]
and for the 3 cluster model: [E2, E1, C5, . . .], [E4, A5, E3, . . .] and [N3, N1, N2, . . .]
As we can see, 2 clusters are similar in 2 and 3 models.
Here, the 3 cluster model is:
Cluster 1: Find it difficult to approach others, Don't talk a lot, Waste my time, . . .
Cluster 2: Make friends easily, Make people feel at ease, Know how to captivate people, . . .
Cluster 3: Have frequent mood swings, Get angry easily, Get irritated easily, . . .
Cluster 1 anti-social people, Cluster 2 supportive, social people and Cluster 3 sensitive and attention-payer! people.
and the 2 cluster is:
Cluster 1: Make friends easily, Make people feel at ease, Know how to captivate people, . . .
Cluster 2: Find it difficult to approach others, Often feel blue, Waste my time, . . .
Clearly, 2 clusters are social vs antisocial people.

5. Next use hierarchical clustering. Print the dendrogram, and use that to guide your choice of the number of clusters. Use cutree to generate a list of which clusters each observation belongs to. Aggregate the data by cluster and then examine those centers (the aggregate means) as you did in (4). Can you interpret all of them meaningfully using the methods from (4) to look at the centers?

ANS.

```
# 1. Hierarchical clustering:
bfi.hier <- hclust(dist(bfi_scaled), method="complete")
# I changed the method to complete, because the results for average were
# too dense and unrocognizable

# 2. Cluster dendrogram pot:
plot(bfi.hier, labels=F)
abline(a=14, b=0, col="red")
```




```
# 3. Using cutree to generate the list of clusters of the observations:
hier.clusters <- as.vector(cutree(bfi.hier, h=14))
hier.info <- cbind(bfi[26:28], hier_clusters=hier.clusters)
head(hier.info)
```

```
##       gender education age hier_clusters
## 61623      2         3  21             1
## 61629      1         2  19             2
## 61634      1         1  21             1
```

```
## 61640      1         1  17             1
## 61661      1         5  68             1
## 61664      2         2  27             1
```

```
# 4. Aggregating to find the centroids:
hier.clust <- cbind(bfi_scaled, hier_clusters=hier.clusters)
hier.centroids <- aggregate(hier.clust[1:25], by=list(hier.clust$hier_clusters), FUN=mean)
hier.centroids
```

```
##   Group.1          A1         A2         A3         A4         A5
## 1       1 -0.05777335  0.1438697  0.1595687  0.1035063  0.1473632
## 2       2  0.28151633 -0.6677104 -0.7858912 -0.5151658 -0.6793014
## 3       3  0.63903571 -1.7717500 -1.7198161 -1.0864272 -1.8397856
##           C1           C2          C3         C4          C5          E1
## 1  0.1202638  0.121512166  0.08814495 -0.1061676 -0.08984416 -0.08512309
## 2 -0.7988927 -0.839190079 -0.50420172  0.7145077  0.60146444  0.34794697
## 3 -0.1782216 -0.006867887 -0.57076546  0.1072517  0.10800619  1.30326682
##           E2         E3         E4         E5         N1          N2
## 1 -0.1195248  0.1350979  0.1406725  0.1476389 -0.0632771 -0.05296484
## 2  0.6172664 -0.6955643 -0.6204629 -0.8377135  0.4940002  0.42009466
## 3  1.1334762 -1.2926697 -1.9077655 -0.9928191 -0.3048639 -0.29090669
##            N3          N4          N5         O1          O2         O3
## 1 -0.03710169 -0.07474281 -0.02798384  0.1242417 -0.0652780  0.1391360
## 2  0.36811914  0.48897410  0.28415691 -0.9114216  0.5713866 -0.8936340
## 3 -0.60340684  0.15151334 -0.49031678  0.2818628 -0.6487646 -0.3719133
##             O4          O5
## 1  0.004918112 -0.08397274
## 2 -0.119232990  0.65498967
## 3  0.461169206 -0.40143520
```

```
# 5. Looking at the centers:
tail(t(hier.centroids[1, order(hier.centroids[1, ])]))
```

```
##                 1
## E4      0.1406725
## A2      0.1438697
## A5      0.1473632
## E5      0.1476389
## A3      0.1595687
## Group.1 1.0000000
```

```
tail(t(hier.centroids[2, order(hier.centroids[2, ])]))
```

```
##                 2
## O2      0.5713866
## C5      0.6014644
## E2      0.6172664
## O5      0.6549897
## C4      0.7145077
## Group.1 2.0000000
```

```
tail(t(hier.centroids[3, order(hier.centroids[3, ])]))
```

```
##                 3
## O1      0.2818628
## O4      0.4611692
## A1      0.6390357
```

```
## E2      1.1334762
## E1      1.3032668
## Group.1 3.0000000
```

Looking at clusters, for cluster 1 we have: [A3, E5, A5, ...] -> Know how to comfort others, Take charge, Make people feel at ease, ...
cluster 2: [C4, O5, E2, ...] -> Do things in a half-way manner, Will not probe deeply into a subject, Find it difficult to approach others, ...
Cluster 3: [E1, E2, A1, ...] -> Don't talk a lot, Find it difficult to approach others, Indifferent to the feelings of others, ...
Cluster 1 supportive social people, Cluster 2 in non-interested and insensitive people, Cluster 3 antisocial indifferent people.
Here 2 group is similar to kmeans cluster (the supportive and socail group and antisocail group), and 1 is the exact opposite: the insensitive and non-interested people.

6. From the factor and cluster analysis, what can you say more generally about what you have learned about your data?

ANS.
To answer this part let's add the information that we threw away at the beggining. Let's see how these clusters are related to education, age and gender. As we saw in the hierarchical model, 3 clusters seem reasonable. So, I am going to comment based on the results of kmeans for 3 custer:

```
# Analysis using dplyr package:
library(tidyverse)
# Cluster Info:
info <- cbind(bfi[26:28], clusters=kmeans3.bfi$cluster)

# Grouping by Clusters:
info.gr <- group_by(info, clusters)

# Percentage in each cluster:
summarize(info.gr,
          perc_in_cluster=n()*100/nrow(info))
```

```
## # A tibble: 3 x 2
##   clusters perc_in_cluster
##      <int>           <dbl>
## 1        1            28.7
## 2        2            42.0
## 3        3            29.3
```

```
# Percentage of men in cluster 1:
nrow(info[(info$gender==1 & info$clusters==1), ])*100/nrow(info[info$clusters==1, ])
```

```
## [1] 43.30218
```

```
# Percentage of men in cluster 2:
nrow(info[(info$gender==1 & info$clusters==2), ])*100/nrow(info[info$clusters==2, ])
```

```
## [1] 31.09691
```

```
# Percentage of men in cluster 3:
nrow(info[(info$gender==1 & info$clusters==3), ])*100/nrow(info[info$clusters==3, ])
```

```
## [1] 25.19084
```

```r
# Average age in each cluster:
summarize(info.gr,
          ave_age=mean(age))
```

```
## # A tibble: 3 x 2
##   clusters ave_age
##      <int>   <dbl>
## 1        1    28.5
## 2        2    30.8
## 3        3    28.6
```

```r
# Percentage of education lvl for Cluster (1):
# Highschool:
nrow(info[(info$education==1 & info$clusters==1), ])*100/nrow(info[info$clusters==1, ])
```

```
## [1] 11.21495
```

```r
# Finished HS:
nrow(info[(info$education==2 & info$clusters==1), ])*100/nrow(info[info$clusters==1, ])
```

```
## [1] 11.37072
```

```r
# Some College:
nrow(info[(info$education==3 & info$clusters==1), ])*100/nrow(info[info$clusters==1, ])
```

```
## [1] 41.90031
```

```r
# College Grad:
nrow(info[(info$education==4 & info$clusters==1), ])*100/nrow(info[info$clusters==1, ])
```

```
## [1] 18.53583
```

```r
# Grad Degree:
nrow(info[(info$education==5 & info$clusters==1), ])*100/nrow(info[info$clusters==1, ])
```

```
## [1] 16.97819
```

```r
# Percentage of education lvl for Cluster (2):
# Highschool:
nrow(info[(info$education==1 & info$clusters==2), ])*100/nrow(info[info$clusters==2, ])
```

```
## [1] 7.454739
```

```r
# Finished HS:
nrow(info[(info$education==2 & info$clusters==2), ])*100/nrow(info[info$clusters==2, ])
```

```
## [1] 10.86262
```

```r
# Some College:
nrow(info[(info$education==3 & info$clusters==2), ])*100/nrow(info[info$clusters==2, ])
```

```
## [1] 51.11821
```

```r
# College Grad:
nrow(info[(info$education==4 & info$clusters==2), ])*100/nrow(info[info$clusters==2, ])
```

```
## [1] 14.58999
```

```r
# Grad Degree:
nrow(info[(info$education==5 & info$clusters==2), ])*100/nrow(info[info$clusters==2, ])
```

```
## [1] 15.97444
```

```r
# Percentage of education lvl for Cluster (3):
# Highschool:
nrow(info[(info$education==1 & info$clusters==3), ])*100/nrow(info[info$clusters==3, ])
```

```
## [1] 8.549618
```

```r
# Finished HS:
nrow(info[(info$education==2 & info$clusters==3), ])*100/nrow(info[info$clusters==3, ])
```

```
## [1] 11.45038
```

```r
# Some College:
nrow(info[(info$education==3 & info$clusters==3), ])*100/nrow(info[info$clusters==3, ])
```

```
## [1] 50.22901
```

```r
# College Grad:
nrow(info[(info$education==4 & info$clusters==3), ])*100/nrow(info[info$clusters==3, ])
```

```
## [1] 13.74046
```

```r
# Grad Degree:
nrow(info[(info$education==5 & info$clusters==3), ])*100/nrow(info[info$clusters==3, ])
```

```
## [1] 16.03053
```

Comparison of education and age wasn't very useful. However, the analysis shows that clusters 2 and 3 are majority women (w/ %69 and %75 women respectively). However again, we have to take into account that %68 of the total participants in the survey were women, so it might be more interesting that cluster 1 had a greater percentage of men.
The details of each category was already discussed so I didn't talk about them again here. But the whole point of difference between factors and clusters is that factors are a range (here ranges of mood from social to isolated OR from indifferent to sensitive), but clusters are one head of this range. In the case of this study we can say the personality of people is first explained by their social abilities and then by how emotional and sensitive they are.