4.0 Multidimensional Scaling Assignment

Problem 1

Use the Olympic Heptathlon data, be sure to remove PNG as an outlier and the final scores variable:

```
data("heptathlon",package="HSAUR2")
mydata <- heptathlon[-25,-8]
```

a) Create a scaled distance matrix for observations

This is the scaled distance matrix for the first two coordinates. We can look at the eigenvalues printed or calculate the cumulative proportion of the eigenvalues (commented out) to see that two coordinates is an appropriate number to examine.

```
cmd <- cmdscale(dist(scale(mydata)), k = 2,  eig = T)
# cumsum(cmd$eig)/sum(cmd$eig)
cmd
```

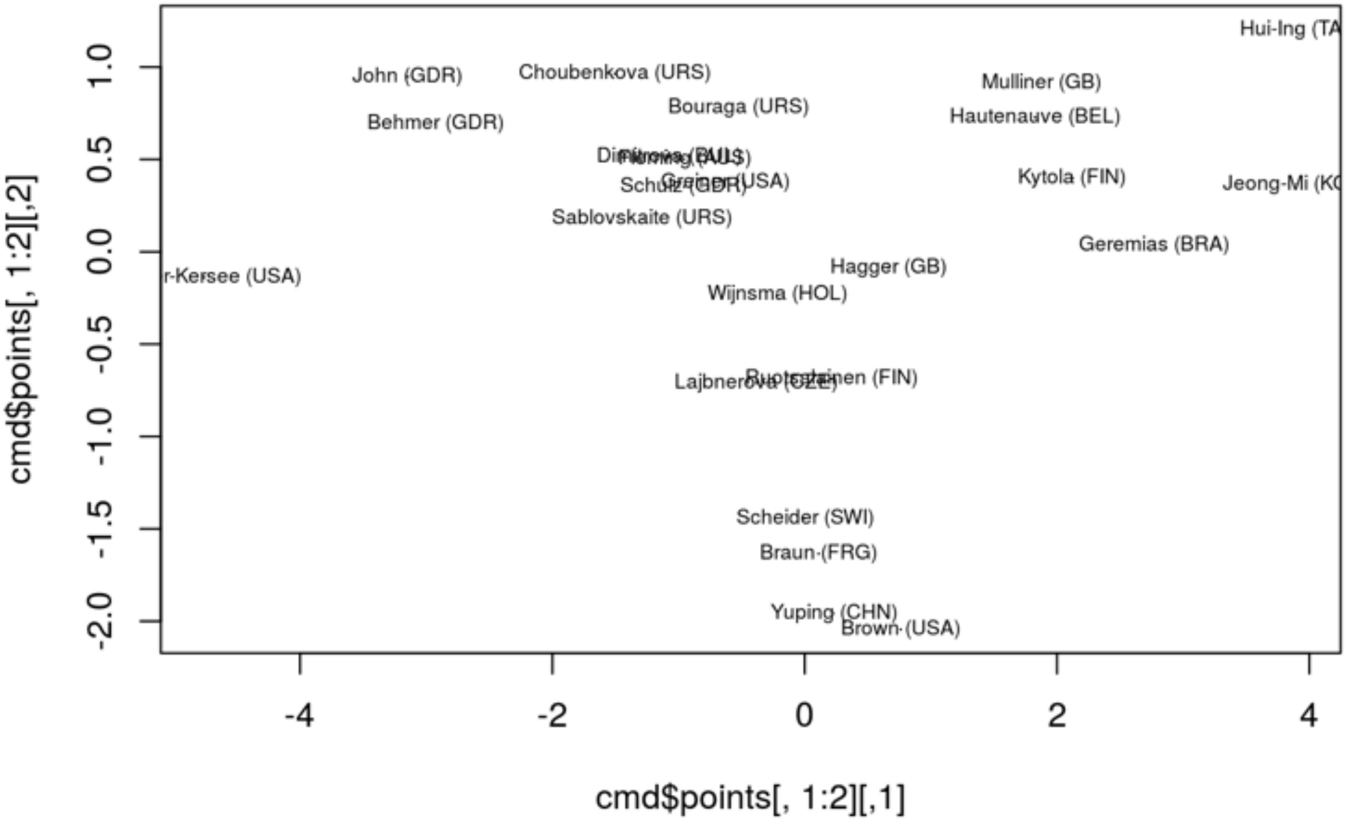
```
## $points
##                              [,1]        [,2]
## Joyner-Kersee (USA) -4.757530189 -0.13986143
## John (GDR)          -3.147943402  0.94859029
## Behmer (GDR)        -2.926184760  0.69534239
## Sablovskaite (URS)  -1.288135516  0.17900713
## Choubenkova (URS)   -1.503450994  0.96177329
## Schulz (GDR)        -0.958467101  0.35121643
## Fleming (AUS)       -0.953445060  0.49982537
## Greiner (USA)       -0.633239267  0.37592917
## Lajbnerova (CZE)    -0.381571974 -0.71213459
## Bouraga (URS)       -0.522322004  0.77688861
## Wijnsma (HOL)       -0.217701500 -0.23369645
## Dimitrova (BUL)     -1.075984276  0.51552998
## Scheider (SWI)       0.003014986 -1.44688825
## Braun (FRG)          0.109183759 -1.63595645
## Ruotsalainen (FIN)   0.208868056 -0.68866173
## Yuping (CHN)         0.232507119 -1.95999641
## Hagger (GB)          0.659520046 -0.08775813
## Brown (USA)          0.756854602 -2.04292201
## Mulliner (GB)        1.880932819  0.91530324
## Hautenauve (BEL)     1.828170404  0.72629699
## Kytola (FIN)         2.118203163  0.39921397
## Geremias (BRA)       2.770706272  0.03463584
## Hui-Ing (TAI)        3.901166920  1.20175472
## Jeong-Mi (KOR)       3.896847898  0.36656804
## 
## $eig
##  [1]  9.944377e+01  2.067687e+01  1.908406e+01  1.073543e+01  6.861410e+00
##  [6]  2.619143e+00  1.579319e+00  2.040683e-14  4.388356e-15  3.315435e-15
## [11]  2.630528e-15  1.983402e-15  6.458129e-16  2.294581e-16  1.850502e-16
## [16] -2.883364e-16 -3.398533e-16 -1.286904e-15 -1.316577e-15 -3.094201e-15
## [21] -4.230624e-15 -4.245227e-15 -4.393542e-15 -8.699585e-15
## 
## $x
## NULL
## 
## $ac
## [1] 0
## 
## $GOF
## [1] 0.7460909 0.7460909
```

b) Perform graphical MDS analysis on the “distance” matrix of a

Label the points using the row names (set an appropriate cex (size) for a better view). Who is the most similar athlete to Scheider(SWI)?

This is the plot of the scaled distances, from this we can see that Braun from FRG (West Germany) is the closest toe Scheider.

```
plot(cmd$points[,1:2], pch = ".")
text(cmd$points[,1:2], labels = rownames(mydata), cex = 0.6)
```



c) Create a distance matrix from a correlation matrix, explain why this represents distance between variables

Create a correlation matrix, convert it to a distance matrix by computing 1-correlation This is a distance matrix created by performing 1 - the correlation matrix of the data. The resulting distances can be understood relative to their correlations, with a distance of zero being perfect correlation. Therefore, higher distances in this matrix represent lower correlations and vice versa.

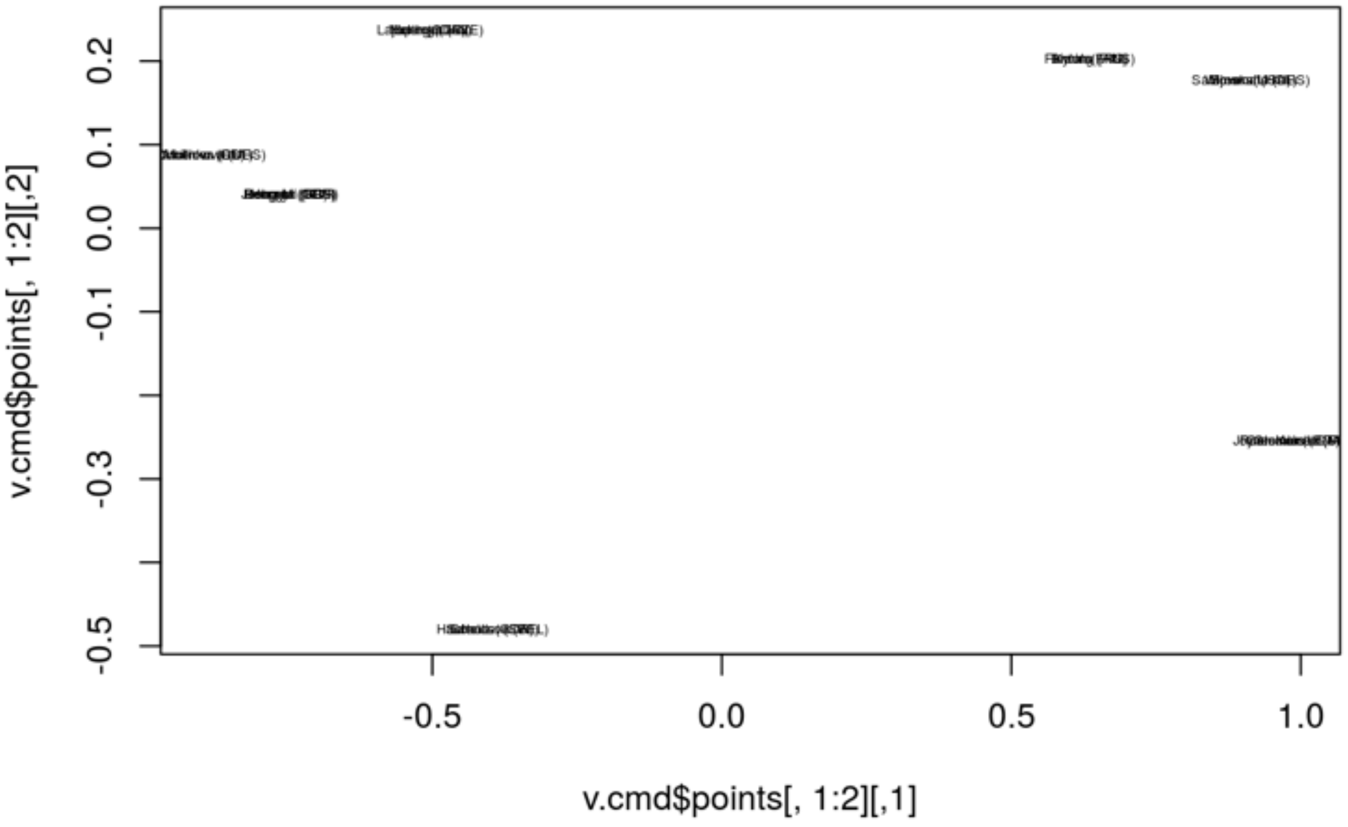
```
v.dist <- 1-cor(mydata)
v.dist
```

```
##            hurdles  highjump      shot   run200m  longjump   javelin   run800m
## hurdles  0.0000000 1.5817409 1.7666860 0.1699629 1.8893472 1.3324779 0.4412206
## highjump 1.5817409 0.0000000 0.5353146 1.3909024 0.3373090 0.6519207 1.1523350
## shot     1.7666860 0.5353146 0.0000000 1.6694330 0.2159620 0.6569667 1.4082925
## run200m  0.1699629 1.3909024 1.6694330 0.0000000 1.8106176 1.4707969 0.4268098
## longjump 1.8893472 0.3373090 0.2159620 1.8106176 0.0000000 0.7129174 1.5233809
## javelin  1.3324779 0.6519207 0.6569667 1.4707969 0.7129174 0.0000000 1.2559348
## run800m  0.4412206 1.1523350 1.4082925 0.4268098 1.5233809 1.2559348 0.0000000
```

d) Perform graphical MDS analysis on the correlation-distance matrix

Label the points using the column names (set an appropriate cex (size) for a better view). What variables are more similar (related) to each other? This is the MDS plot of the correlation-distance matrix created above. Unfortunately it is difficult to tell which variables are related regardless of how the text is scaled, but there are obvious clusters and the ones near the bottom would be the most closely related.

```
v.cmd <- cmdscale(v.dist, eig = T)
plot(v.cmd$points[,1:2], pch = ".")
text(v.cmd$points[,1:2], labels = rownames(mydata), cex = .4)
```



Problem 2

Use the TTU grad student exit survey data. Two variables of interest are FacTeaching, a 1, 2, 3, 4, 5 ratings of teaching at TTU by the student, and COL, the college from which the student graduated.

```
grad <- read.csv("http://westfall.ba.ttu.edu/isqs6348/Rdata/pgs.csv", header = T)
```

```
install.packages("ca")
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
```

```
library(ca)
```

a) Construct the contingency table showing counts of students in all combinations of these two variables

```
attach(grad)

tbl = table(COL, FacTeaching)
tbl
```

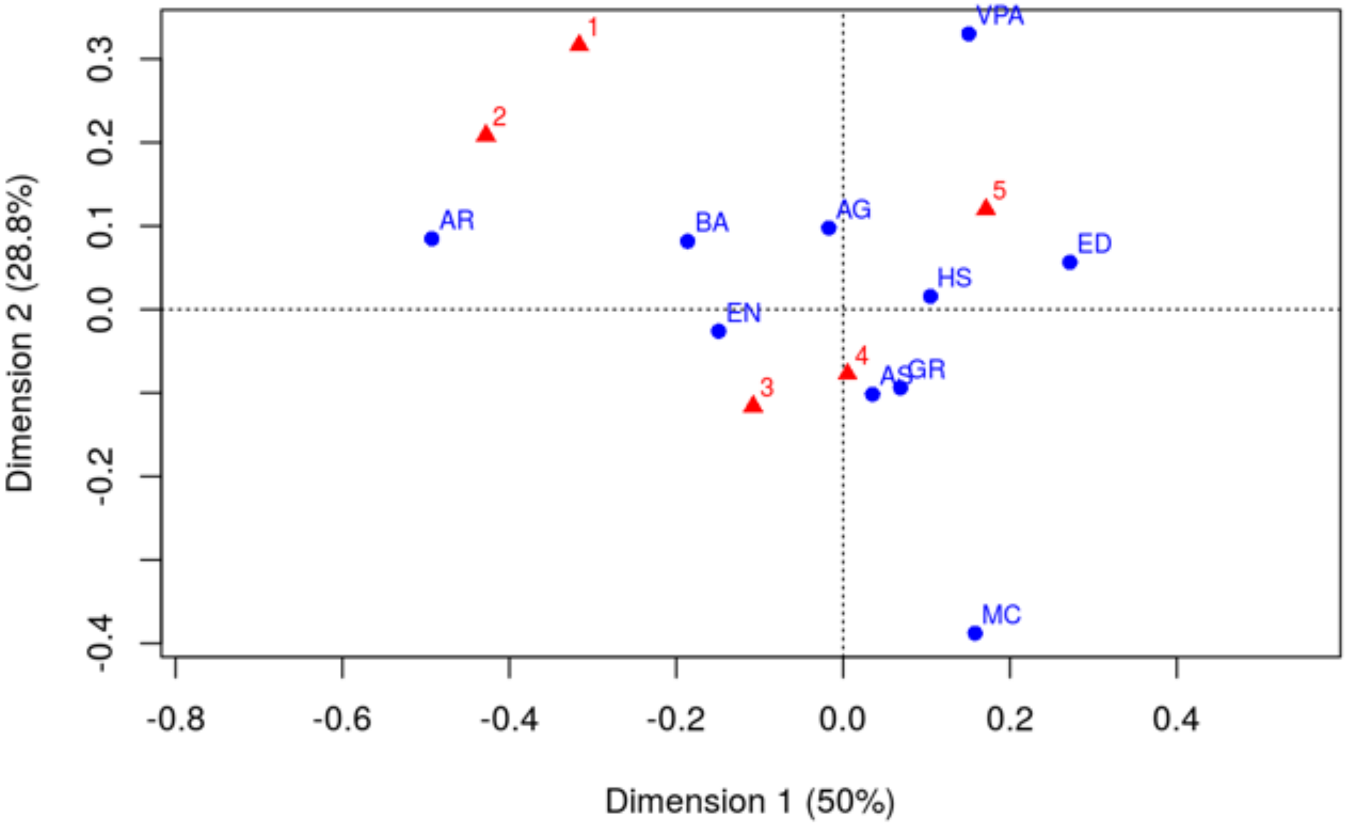
```
##       FacTeaching
## COL      1   2   3   4   5
##   AG     4  15  26  78  56
##   AR     3   4   6  16   4
##   AS    12  24 124 290 171
##   BA     9  28  44 116  66
##   DUAL   0   0   2   0   0
##   ED     3   6  26 113  93
##   EN     5  36  65 168  86
##   GR     0   3   8  27  15
##   HS     1   5  17  41  33
##   MC     0   0   3  25   6
##   VPA    4   7  10  37  44
```

b) Construct the correspondence analysis (CA) plot and comment on the outlier in the previous problem

Remove the outlier data you discovered and re-construct the CA plot. Using the `ca()` function, we create a CA plot of the evaluation data. One obvious outlier is “Dual,” which has an extremely low count. This may be because students who are dual majors are counted in other categories, or there are just very few of them. We can remove it before executing the final plot.

```
grad.ca <- ca(tbl)

# remove column 5, dual
grad.ca <- ca(tbl2 <- tbl[-5, ])
plot(grad.ca)
```



c) Analysis of three selected colleges

Pick three colleges in your graph, two of which are close to each other, and the third of which is far from your first two. Find the three conditional distributions of rating for your three colleges, and interpret the distance between the graph points in terms of "distances" between those three conditional distributions.

R-Documentation: - `prop.table` returns conditional proportions given margins, i.e. entries of x, divided by the appropriate marginal sums.

1. First we create the proportion table with the `prop.table()` function:

```
tbl2 <- tbl[-5, ]

# create proportion table
prop.tbl2 <- prop.table(tbl2)
round(prop.tbl2, 3)
```

```
##      FacTeaching
## COL       1     2     3     4     5
##   AG  0.002 0.008 0.013 0.039 0.028
##   AR  0.002 0.002 0.003 0.008 0.002
##   AS  0.006 0.012 0.063 0.146 0.086
##   BA  0.005 0.014 0.022 0.058 0.033
##   ED  0.002 0.003 0.013 0.057 0.047
##   EN  0.003 0.018 0.033 0.085 0.043
##   GR  0.000 0.002 0.004 0.014 0.008
##   HS  0.001 0.003 0.009 0.021 0.017
##   MC  0.000 0.000 0.002 0.013 0.003
##   VPA 0.002 0.004 0.005 0.019 0.022
```

2. Let’s examine an apparent cluster of three schools: BA, EN, and MC BA and EN appear close, while MC is extremely far from both.

```
prop.table(tbl2, 1)[c(4, 6, 9), ]
```

```
##     FacTeaching
## COL          1          2          3          4          5
##   BA 0.03422053 0.10646388 0.16730038 0.44106464 0.25095057
##   EN 0.01388889 0.10000000 0.18055556 0.46666667 0.23888889
##   MC 0.00000000 0.00000000 0.08823529 0.73529412 0.17647059
```

The distances between BA and EN are extremely close between all scores with the exception of 1, and BA is visibly close to 1 on the graph. MC is far on most of the scores, with most of the proportion centered on 4. It is weighted significantly towards higher scores in general.

Problem 3

Use the Daily stock returns data set. The columns are companies; Man1, Man2, Man3 are manufacturing companies; Serv1, Serv2, Serv3, Serv4 are service companies.The numbers (returns) are all small, close to zero, and this causes some numerical problems with the computer analyses. (As a general rule, numerical algorithms are designed for stability when the numbers are “nice,” like 12.3, 4.56, 2.38 rather than .000123, .0000456, .0000238.) So multiply everything in the data set by 100 first. (This converts returns to % scale; eg, 0.013 becomes 1.3%).

```
stock <- read.csv("https://bit.ly/3egKiMU")
stock = stock*100
```

a) Perform an exploratory factor analysis (EFA) using two factors.

To perform a basic EFA, we will use the `factanal()` function and pass in `factors = 2` to specify the number of factors.

```
stock.fa<- factanal(stock, factors = 2)
stock.fa
```

```
##
## Call:
## factanal(x = stock, factors = 2)
##
## Uniquenesses:
##  Man1  Man2  Man3 Serv1 Serv2 Serv3 Serv4
## 0.764 0.513 0.520 0.900 0.714 0.788 0.574
##
## Loadings:
##       Factor1 Factor2
## Man1  0.462   0.148
## Man2  0.674   0.180
## Man3  0.676   0.150
## Serv1 0.131   0.288
## Serv2 0.128   0.519
## Serv3 0.131   0.441
## Serv4 0.136   0.639
##
##                Factor1 Factor2
## SS loadings      1.195   1.032
## Proportion Var   0.171   0.147
## Cumulative Var   0.171   0.318
##
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 5.1 on 8 degrees of freedom.
## The p-value is 0.747
```

b) Interpret the p-value reported in your EFA.

The null hypothesis is that the number of factors is sufficient for presenting the data. With a p-value of 0.75, we fail to reject the null hypothesis.

c) What are the factors (latent variables) in this model? Name them.

The factors are the types of companies, namely manufacturing and service. This is apparent from the clustered loadings of each company type.

d) Write the EFA regression model for variable Man1.

For example: Man1 = a f1 + b f2 + e. Based on the outputs, what are a and b? Note: e is an unknown error term random variable (called u in the lecture video). The most important thing we know about e is the variance of e, which we call it uniqueness, also we know that e is normally distributed with the mean of zero. Long story short, do not plug a value for e because it's a random variable, not a constant value. The regression model for Man1 uses the coefficients of the loadings and the additional error term. $Man1 = 0.46f_1 + 0.15f_2 + e$

e) In the model of part d, determine the variance of the error term e.

Knowing that the variance of the scaled variable is unit, we can solve for the error term and establish that the variance of the error term (uniqueness) is one minus the sum of squared loadings:

$$Var(e_i) = 1 - (\lambda_{i1}^2 + \lambda_{i2}^2 + \ldots + \lambda_{ik}^2)$$

We can calculate this in R and arrive at a uniqueness of 0.76:

```
1 - sum(stock.fa$loadings[1,1:2]^2)
```

```
## [1] 0.764247
```

f) What is the correlation between f2 and Serve2?

From the analysis above we can find this at 0.52.

g) Compare the EFA approximated correlation matrix versus the actual correlation matrix. Report RMSE. What do you conclude?

To perform the extimated correlation, we use the formula:

$$\hat{\Sigma} = \Lambda\Lambda^T + \Psi$$

Multiply the loadings matrix by its transpose and add the covariance matrix of error. Then we can perform the RMSE on that result and the original correlation matrix. From the result of 0.01 it appears the matrices have low error and are therefore very similar

```
s.loadings = stock.fa$loadings[,1:2]
corHat <- s.loadings %*% t(s.loadings) + diag(stock.fa$uniquenesses)

corr <- cor(stock)
rmse = sqrt(mean((corHat-corr)^2))
rmse
```

```
## [1] 0.01009255
```

Problem 4

Perform factor analysis on questions 22-35 of TTU web survey data:

```
ttuweb <- read.csv("https://bit.ly/3oNr5qX")
mydata <- ttuweb[,22:35]
```

a) There are some missing values in this data. Find the correlation matrix based on pair-wise deletion.

Use this correlation matrix as an input for EFA.

The nulls can be dropped by performing a correlation and passing in pairwise for the `use` argument:

```
cor(mydata, use = "pairwise.complete.obs")
```

```
##            Q22        Q23         Q24        Q25         Q26         Q27
## Q22  1.0000000  0.6140032  0.55843782  0.5557452  0.51726130  0.49654159
## Q23  0.6140032  1.0000000  0.65906191  0.5962273  0.54752476  0.54748415
## Q24  0.5584378  0.6590619  1.00000000  0.5664979  0.51626946  0.48058691
## Q25  0.5557452  0.5962273  0.56649794  1.0000000  0.47081220  0.45866847
## Q26  0.5172613  0.5475248  0.51626946  0.4708122  1.00000000  0.76733895
## Q27  0.4965416  0.5474841  0.48058691  0.4586685  0.76733895  1.00000000
## Q28  0.5419000  0.5720596  0.53301613  0.4809110  0.74443194  0.68690731
## Q29  0.5170146  0.5041127  0.44420526  0.4521453  0.69510444  0.65943926
## Q30 -0.2519846 -0.1622736 -0.14204850 -0.2135915 -0.26099585 -0.23336630
## Q31 -0.3677561 -0.2843945 -0.28287994 -0.3038377 -0.33374258 -0.28686368
## Q32 -0.2716450 -0.2932138 -0.26050696 -0.1715154 -0.29453330 -0.27865468
## Q33 -0.2636431 -0.2207740 -0.22789015 -0.2545225 -0.19391366 -0.23251598
## Q34 -0.1823547 -0.1033739 -0.04788993 -0.1846743 -0.09898865 -0.03584522
## Q35 -0.3476101 -0.2432658 -0.22627100 -0.3212392 -0.27708454 -0.29297916
##             Q28         Q29        Q30        Q31        Q32        Q33
## Q22  0.54189996  0.51701459 -0.2519846 -0.3677561 -0.2716450 -0.2636431
## Q23  0.57205960  0.50411274 -0.1622736 -0.2843945 -0.2932138 -0.2207740
## Q24  0.53301613  0.44420526 -0.1420485 -0.2828799 -0.2605070 -0.2278902
## Q25  0.48091101  0.45214534 -0.2135915 -0.3038377 -0.1715154 -0.2545225
## Q26  0.74443194  0.69510444 -0.2609959 -0.3337426 -0.2945333 -0.1939137
## Q27  0.68690731  0.65943926 -0.2333663 -0.2868637 -0.2786547 -0.2325160
## Q28  1.00000000  0.82562526 -0.2011958 -0.2834717 -0.2322904 -0.1767991
## Q29  0.82562526  1.00000000 -0.1727148 -0.3318058 -0.2496653 -0.1705341
## Q30 -0.20119578 -0.17271482  1.0000000  0.4982871  0.4050433  0.3854891
## Q31 -0.28347173 -0.33180580  0.4982871  1.0000000  0.5502768  0.4514010
## Q32 -0.23229036 -0.24966527  0.4050433  0.5502768  1.0000000  0.5941093
## Q33 -0.17679910 -0.17053414  0.3854891  0.4514010  0.5941093  1.0000000
## Q34 -0.09349773 -0.09684238  0.4175193  0.4109468  0.3870482  0.4171537
## Q35 -0.22589995 -0.19105965  0.3742362  0.4287879  0.3587126  0.4011587
##             Q34        Q35
## Q22 -0.18235469 -0.3476101
## Q23 -0.10337389 -0.2432658
## Q24 -0.04788993 -0.2262710
## Q25 -0.18467431 -0.3212392
## Q26 -0.09898865 -0.2770845
## Q27 -0.03584522 -0.2929792
## Q28 -0.09349773 -0.2258999
## Q29 -0.09684238 -0.1910596
## Q30  0.41751930  0.3742362
## Q31  0.41094685  0.4287879
## Q32  0.38704822  0.3587126
## Q33  0.41715366  0.4011587
## Q34  1.00000000  0.3696030
## Q35  0.36960299  1.0000000
```

b) Perform EFA suggesting two common factors. How would you name those factors?

You find corr in part a, then you can perform EFA based on the corr matrix given knowing the size of the original data. This is the expected outcome

fter performing EFA, the two factors group around types of questions. Factor one appears to be attitudes towards the university, and Factor two is related to opinions about the website.

```
cleandata = na.omit(mydata)
cleandata.fa <- factanal(cleandata, factors = 2)
cleandata.fa
```

```
##
## Call:
## factanal(x = cleandata, factors = 2)
##
## Uniquenesses:
##   Q22   Q23   Q24   Q25   Q26   Q27   Q28   Q29   Q30   Q31   Q32   Q33   Q34
## 0.533 0.515 0.583 0.607 0.297 0.380 0.198 0.292 0.611 0.452 0.488 0.506 0.621
##   Q35
## 0.648
##
## Loadings:
##     Factor1 Factor2
## Q22  0.615  -0.297
## Q23  0.665  -0.205
## Q24  0.615  -0.195
## Q25  0.573  -0.254
## Q26  0.823  -0.162
## Q27  0.773  -0.148
## Q28  0.893
## Q29  0.838
## Q30 -0.156   0.604
## Q31 -0.287   0.683
## Q32 -0.198   0.688
## Q33 -0.109   0.694
## Q34          0.615
## Q35 -0.206   0.557
##
##                Factor1 Factor2
## SS loadings      4.502   2.767
## Proportion Var   0.322   0.198
## Cumulative Var   0.322   0.519
##
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 289.33 on 64 degrees of freedom.
## The p-value is 2.15e-30
```

c) Perform EFA suggesting three common factors. How would you name those factors?

The first two factors are the same as before, but there is a third factor with a cluster of questions 22-25 which appear to be about value of the university.

```
cleandata.fa <- factanal(cleandata, factors = 3)
cleandata.fa
```

```
##
## Call:
## factanal(x = cleandata, factors = 3)
##
## Uniquenesses:
##   Q22   Q23   Q24   Q25   Q26   Q27   Q28   Q29   Q30   Q31   Q32   Q33   Q34
## 0.451 0.313 0.366 0.472 0.316 0.402 0.157 0.209 0.591 0.440 0.484 0.512 0.614
##   Q35
## 0.647
##
## Loadings:
##     Factor1 Factor2 Factor3
## Q22  0.367  -0.258   0.590
## Q23  0.351  -0.139   0.738
## Q24  0.296  -0.124   0.729
## Q25  0.296  -0.201   0.632
## Q26  0.710  -0.186   0.380
## Q27  0.655  -0.168   0.375
## Q28  0.839  -0.102   0.358
## Q29  0.845  -0.122   0.249
## Q30 -0.131   0.623
## Q31 -0.220   0.694  -0.174
## Q32 -0.139   0.691  -0.137
## Q33          0.683  -0.140
## Q34          0.621
## Q35 -0.100   0.546  -0.211
##
##                Factor1 Factor2 Factor3
## SS loadings      2.882   2.727   2.416
## Proportion Var   0.206   0.195   0.173
## Cumulative Var   0.206   0.401   0.573
##
## Test of the hypothesis that 3 factors are sufficient.
## The chi square statistic is 128.28 on 52 degrees of freedom.
## The p-value is 2.19e-08
```

d) What rotation method is used in factanal as a default method? Explain what that rotation does?

Rotation changes the factor loadings with the goal of making them easier to interpret. The default rotation method is varimax.

e) Repeat part b (EFA with two factors) without rotation (inside factanal put rotation = “none”)

Will you end up with the same names for your factors?

This is the EFA with no rotation rather than the default of Varimax. The names would remain the same, as the underlying math is not changed.

```
cleandata = na.omit(mydata)
cleandata.fa <- factanal(cleandata, factors = 2, rotation = "none")
cleandata.fa
```

```
## 
## Call:
## factanal(x = cleandata, factors = 2, rotation = "none")
## 
## Uniquenesses:
##   Q22   Q23   Q24   Q25   Q26   Q27   Q28   Q29   Q30   Q31   Q32   Q33   Q34 
## 0.533 0.515 0.583 0.607 0.297 0.380 0.198 0.292 0.611 0.452 0.488 0.506 0.621 
##   Q35 
## 0.648 
## 
## Loadings:
##     Factor1 Factor2
## Q22  0.681         
## Q23  0.695         
## Q24  0.645         
## Q25  0.626         
## Q26  0.827   0.139 
## Q27  0.776   0.135 
## Q28  0.862   0.245 
## Q29  0.812   0.221 
## Q30 -0.359   0.510 
## Q31 -0.510   0.537 
## Q32 -0.428   0.574 
## Q33 -0.347   0.611 
## Q34 -0.247   0.564 
## Q35 -0.389   0.448 
## 
##                Factor1 Factor2
## SS loadings      5.345   1.924
## Proportion Var   0.382   0.137
## Cumulative Var   0.382   0.519
## 
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 289.33 on 64 degrees of freedom.
## The p-value is 2.15e-30
```