Problem Set 3

Janette Avelar

4/27/2022

Question 1

Run a components analysis on all 19 items. Be sure to generate the reproduced correlation matrix. Use principal components extraction to extract any factors with an eigenvalue greater than 1, and *varimax* orthogonal rotation.

This is the original correlation matrix with heatmap visualization:

```
# get a correlation matrix
# focus on 19 items--therefore, we'll drop first 2 cols
q1data_mod <- q1data %>%
  select(c(3:21))
# create cor
og_cor <- data.frame(cor(q1data_mod, use = "pairwise.complete.obs"))

round(og_cor, 2)
```

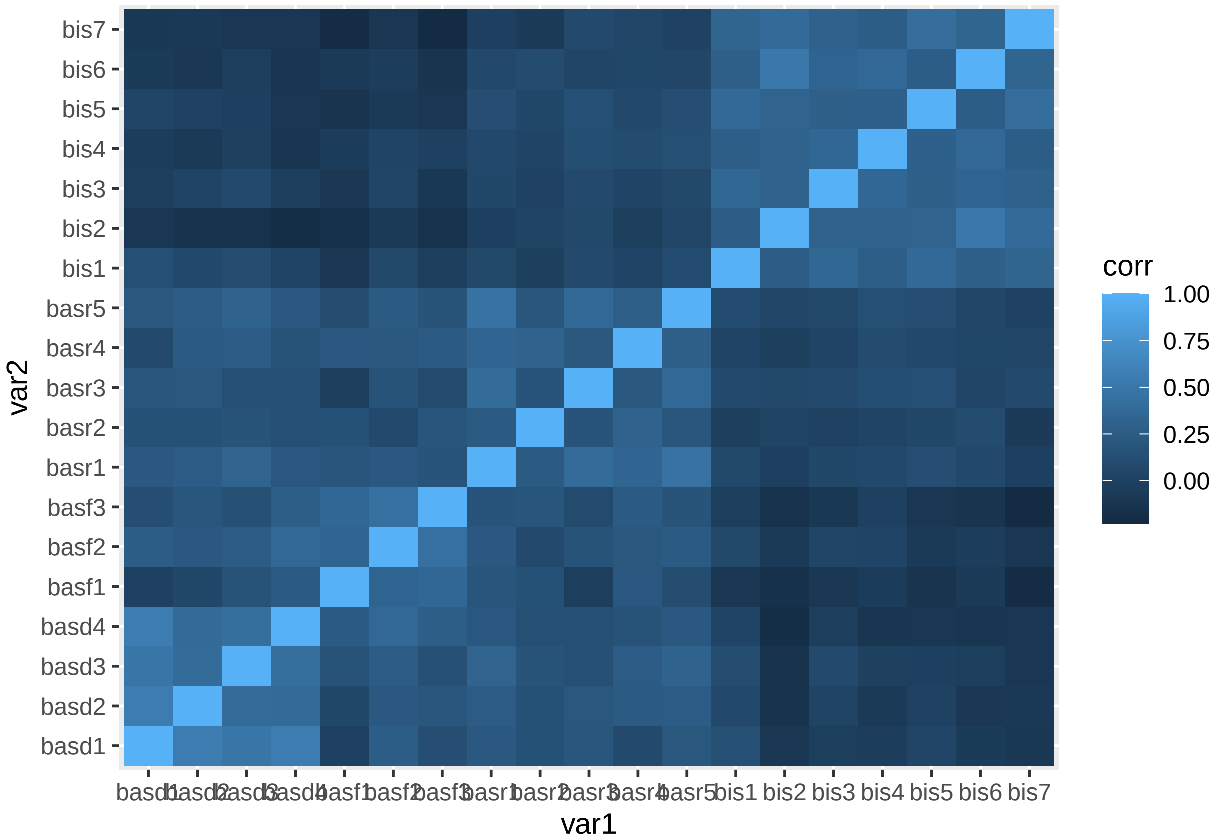
```
##        bis1  bis2  bis3  bis4  bis5  bis6  bis7 basd1 basd2 basd3 basd4 basr1
## bis1   1.00  0.26  0.35  0.28  0.38  0.29  0.33  0.13  0.07  0.11  0.03  0.07
## bis2   0.26  1.00  0.31  0.31  0.33  0.51  0.39 -0.11 -0.14 -0.15 -0.19 -0.01
## bis3   0.35  0.31  1.00  0.36  0.29  0.33  0.31 -0.03  0.02  0.09 -0.04  0.06
## bis4   0.28  0.31  0.36  1.00  0.29  0.37  0.27 -0.04 -0.08 -0.01 -0.12  0.07
## bis5   0.38  0.33  0.29  0.29  1.00  0.27  0.42  0.04  0.01 -0.01 -0.10  0.12
## bis6   0.29  0.51  0.33  0.37  0.27  1.00  0.33 -0.06 -0.09 -0.02 -0.12  0.07
## bis7   0.33  0.39  0.31  0.27  0.42  0.33  1.00 -0.10 -0.08 -0.10 -0.10 -0.01
## basd1  0.13 -0.11 -0.03 -0.04  0.04 -0.06 -0.10  1.00  0.54  0.50  0.55  0.22
## basd2  0.07 -0.14  0.02 -0.08  0.01 -0.09 -0.08  0.54  1.00  0.39  0.39  0.25
## basd3  0.11 -0.15  0.09 -0.01 -0.01 -0.02 -0.10  0.50  0.39  1.00  0.43  0.33
## basd4  0.03 -0.19 -0.04 -0.12 -0.10 -0.12 -0.10  0.55  0.39  0.43  1.00  0.21
## basr1  0.07 -0.01  0.06  0.07  0.12  0.07 -0.01  0.22  0.25  0.33  0.21  1.00
## basr2 -0.01  0.02  0.01  0.02  0.06  0.10 -0.06  0.15  0.15  0.18  0.13  0.24
## basr3  0.08  0.07  0.08  0.13  0.14  0.03  0.09  0.20  0.21  0.13  0.13  0.40
## basr4  0.02 -0.01  0.03  0.10  0.07  0.05  0.05  0.08  0.24  0.26  0.17  0.34
## basr5  0.10  0.04  0.06  0.13  0.12  0.05  0.00  0.20  0.26  0.32  0.22  0.46
## basf1 -0.11 -0.17 -0.10 -0.06 -0.13 -0.07 -0.21  0.00  0.06  0.17  0.24  0.19
## basf2  0.07 -0.08  0.04  0.03 -0.07 -0.04 -0.10  0.27  0.22  0.25  0.36  0.22
## basf3 -0.02 -0.15 -0.10  0.00 -0.11 -0.13 -0.23  0.12  0.19  0.14  0.28  0.18
##       basr2 basr3 basr4 basr5 basf1 basf2 basf3
## bis1  -0.01  0.08  0.02  0.10 -0.11  0.07 -0.02
## bis2   0.02  0.07 -0.01  0.04 -0.17 -0.08 -0.15
## bis3   0.01  0.08  0.03  0.06 -0.10  0.04 -0.10
```

```
## bis4   0.02  0.13  0.10  0.13 -0.06  0.03  0.00
## bis5   0.06  0.14  0.07  0.12 -0.13 -0.07 -0.11
## bis6   0.10  0.03  0.05  0.05 -0.07 -0.04 -0.13
## bis7  -0.06  0.09  0.05  0.00 -0.21 -0.10 -0.23
## basd1  0.15  0.20  0.08  0.20  0.00  0.27  0.12
## basd2  0.15  0.21  0.24  0.26  0.06  0.22  0.19
## basd3  0.18  0.13  0.26  0.32  0.17  0.25  0.14
## basd4  0.13  0.13  0.17  0.22  0.24  0.36  0.28
## basr1  0.24  0.40  0.34  0.46  0.19  0.22  0.18
## basr2  1.00  0.18  0.31  0.20  0.15  0.09  0.19
## basr3  0.18  1.00  0.22  0.37 -0.03  0.17  0.09
## basr4  0.31  0.22  1.00  0.28  0.21  0.21  0.23
## basr5  0.20  0.37  0.28  1.00  0.11  0.24  0.18
## basf1  0.15 -0.03  0.21  0.11  1.00  0.33  0.36
## basf2  0.09  0.17  0.21  0.24  0.33  1.00  0.44
## basf3  0.19  0.09  0.23  0.18  0.36  0.44  1.00
```

```r
# create a heatmap to visualize

## convert the correlation matrix to long format
og_cor_long <- og_cor %>%
  rownames_to_column(var = "var1") %>%
  gather(key = "var2", value = "corr", -var1)

# plot
ggplot(data = og_cor_long, aes(x = var1, y = var2, fill = corr)) +
  geom_tile()
```



Next, we'll run an unrotated PCA and extract eigenvalues.

```
# note we have NA values
# useful to figure out where they are in our modified dataset to figure out what to do with them

# first let's figure out how many
sum(is.na(q1data_mod)) #76 NAs
```

```
## [1] 76
```

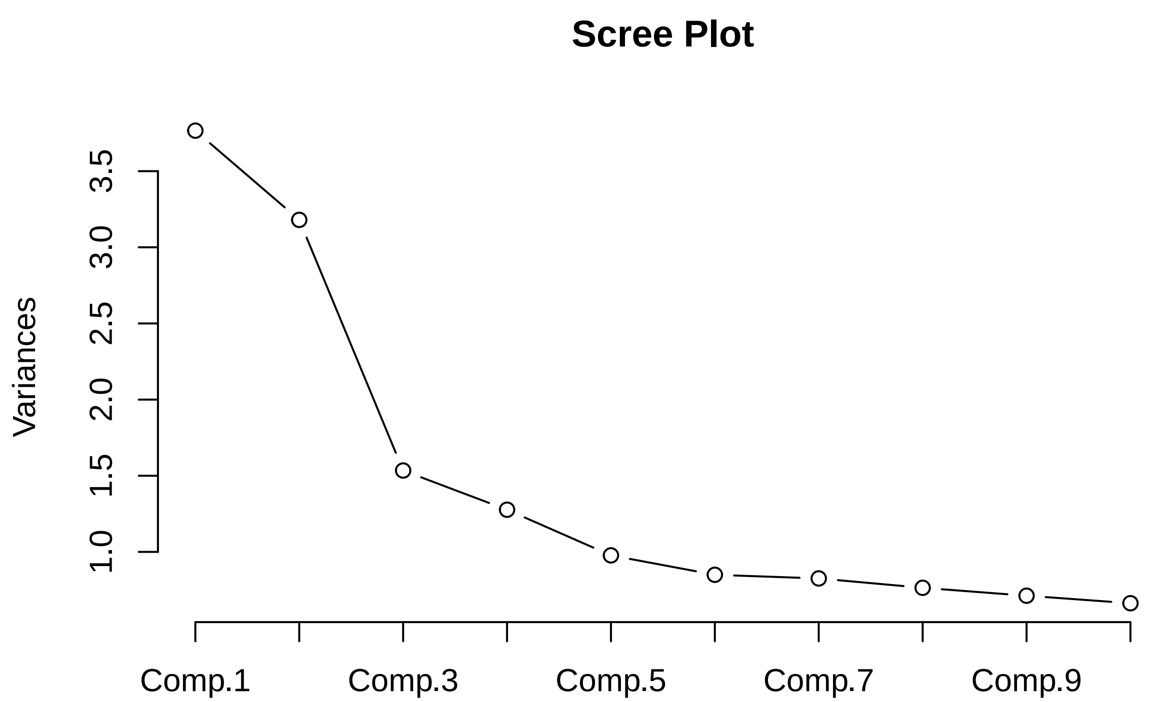
```
#that's a lot of missing values... are they full rows or individual observations?
which(is.na(q1data_mod))
```

```
##  [1]    2   83  208  209  343  424  549  550  684  765  890  891 1025 1106 1231
## [16] 1232 1366 1447 1572 1573 1707 1788 1913 1914 2048 2129 2254 2255 2389 2470
## [31] 2595 2596 2730 2811 2936 2937 3071 3152 3277 3278 3412 3493 3618 3619 3753
## [46] 3834 3959 3960 4094 4175 4300 4301 4435 4516 4641 4642 4776 4857 4982 4983
## [61] 5117 5198 5323 5324 5458 5539 5664 5665 5799 5880 6005 6006 6140 6221 6346
## [76] 6347
```

```
#from cursory look, looks like the missing NAs are full rows
#I've elected to drop them
q1data_mod <- q1data_mod %>%
  na.omit()
```

```r
# now we can run PCA
pca <- princomp(q1data_mod, cor = TRUE)

# scree plot to visualize what's happening
plot(pca, type = "lines", main = "Scree Plot")
```




```r
# extract eigenvalues:
eigens <- eigen(og_cor)
eigval <- eigens$values
eigvec <- eigens$vectors
```

At first glance, from the scree plot, it looks like we may have 5 components that explain most of the variance. However, only 4 of these have eigenvalues greater than 1.

The next step is to calculate our weighted component loadings.

```r
# Components 1-4 (eigval > 1)
pca$loadings[,1:4]
```

```
##            Comp.1     Comp.2      Comp.3       Comp.4
## bis1  0.008378518 0.33987664  0.18456667  0.259474250
## bis2 -0.145566825 0.35714142 -0.08829756  0.058073334
## bis3 -0.034919747 0.34481391  0.05192937  0.228145661
## bis4 -0.039733665 0.33901516 -0.15588085  0.147532958
## bis5 -0.044127474 0.36424916  0.06360562 -0.059107564
```

```
## bis6  -0.085565124  0.36317895 -0.11151328  0.135779898
## bis7  -0.133206293  0.35258874  0.08405520  0.005158542
## basd1  0.321554852  0.04173399  0.46730170  0.052770209
## basd2  0.322294609  0.03411510  0.31934774 -0.049325428
## basd3  0.336603929  0.05981459  0.23189756  0.050435044
## basd4  0.346478332 -0.03721238  0.23749343  0.223503074
## basr1  0.294872010  0.15539429 -0.16384835 -0.310071835
## basr2  0.195014560  0.08252673 -0.24122296 -0.216562130
## basr3  0.202230822  0.18311232 -0.05735976 -0.419873314
## basr4  0.247567347  0.11949270 -0.29383219 -0.156663303
## basr5  0.278363391  0.17170407 -0.11401036 -0.274201231
## basf1  0.210397749 -0.10952160 -0.40105921  0.325265058
## basf2  0.296633124  0.01459476 -0.15302004  0.393241229
## basf3  0.267202078 -0.07671809 -0.32425188  0.310337966
```

```
# Compute weighted loadings:
# Weighted loadings = loadings * sqrt(eigenvalue)
loadings <- pca$loadings[,1:4]

# sqrt of the variance/eigenvalue
weights <- vec2diag(pca$sdev[1:4])
# remember: Multiplying by this matrix = multiplying each column of your loadings by sdev.

# component loadings
loadings[, 1:4] %*% weights
```

```
##              [,1]        [,2]        [,3]         [,4]
## bis1   0.01625918  0.60609262  0.22861560  0.293207058
## bis2  -0.28248399  0.63688042 -0.10937077  0.065623126
## bis3  -0.06776454  0.61489712  0.06432291  0.257805613
## bis4  -0.07710634  0.60455637 -0.19308359  0.166712900
## bis5  -0.08563287  0.64955546  0.07878582 -0.066791810
## bis6  -0.16604593  0.64764698 -0.13812719  0.153431889
## bis7  -0.25849740  0.62876175  0.10411592  0.005829176
## basd1  0.62400275  0.07442307  0.57882855  0.059630571
## basd2  0.62543831  0.06083651  0.39556369 -0.055737953
## basd3  0.65320668  0.10666571  0.28724255  0.056991825
## basd4  0.67236874 -0.06635981  0.29417393  0.252559469
## basr1  0.57222257  0.27711034 -0.20295261 -0.350382554
## basr2  0.37844125  0.14716764 -0.29879355 -0.244716171
## basr3  0.39244498  0.32653914 -0.07104932 -0.474458715
## basr4  0.48042412  0.21308802 -0.36395857 -0.177030230
## basr5  0.54018629  0.30619512 -0.14122022 -0.309848612
## basf1  0.40829356 -0.19530684 -0.49677653  0.367550964
## basf2  0.57564016  0.02602644 -0.18954000  0.444364340
## basf3  0.51852687 -0.13680924 -0.40163826  0.350683283
```

```
loadings(principal(og_cor, nfactors = 19, rotate = "none"))
```

```
##
## Loadings:
##       PC1    PC2    PC3    PC4    PC5    PC6    PC7    PC8    PC9    PC10
## bis1         0.606 -0.229  0.293 -0.112  0.243         -0.118 -0.304
```

```
## bis2  -0.282  0.637  0.109         0.118 -0.234  0.310  0.256
## bis3           0.615         0.258        -0.140 -0.281 -0.282         0.490
## bis4           0.605  0.193  0.167 -0.101 -0.214 -0.126 -0.398  0.131 -0.327
## bis5           0.650                       0.454               -0.302 -0.123
## bis6  -0.166  0.648  0.138  0.153  0.310 -0.340         0.213        -0.105
## bis7  -0.258  0.629 -0.104                0.355         0.294  0.262  0.125
## basd1  0.624        -0.579         0.110         0.191               -0.107
## basd2  0.625        -0.396                             -0.144  0.197 -0.129
## basd3  0.653  0.107 -0.287         0.203 -0.122 -0.392
## basd4  0.672        -0.294  0.253                       0.215         0.101
## basr1  0.572  0.277  0.203 -0.350 -0.130        -0.206  0.195 -0.157
## basr2  0.378  0.147  0.299 -0.245  0.600         0.275 -0.241 -0.254  0.226
## basr3  0.392  0.327        -0.474 -0.397 -0.107  0.269                0.276
## basr4  0.480  0.213  0.364 -0.177  0.240  0.335 -0.171         0.476 -0.106
## basr5  0.540  0.306  0.141 -0.310 -0.249 -0.170 -0.150  0.100 -0.150 -0.209
## basf1  0.408 -0.195  0.497  0.368               -0.219  0.276 -0.155
## basf2  0.576         0.190  0.444 -0.300         0.198         0.105  0.129
## basf3  0.519 -0.137  0.402  0.351 -0.148  0.143  0.289 -0.222        -0.146
##       PC11   PC12   PC13   PC14   PC15   PC16   PC17   PC18   PC19
## bis1  -0.223 -0.368 -0.264  0.164  0.129
## bis2  -0.180  0.112  0.128                0.297  0.106  0.292
## bis3  -0.147  0.202  0.169                                     0.103
## bis4   0.416                                            0.145
## bis5   0.147  0.316  0.108 -0.240                      -0.177
## bis6                -0.218               -0.145        -0.384
## bis7   0.161 -0.116  0.116  0.134 -0.118 -0.276  0.229
## basd1  0.144                                                   0.386
## basd2 -0.280  0.364         0.195  0.110 -0.220               -0.167
## basd3        -0.142        -0.221                0.403        -0.122
## basd4  0.268         0.102  0.215 -0.198  0.202 -0.236        -0.203
## basr1               -0.248 -0.105 -0.378                0.228
## basr2  0.111 -0.115  0.123               -0.146
## basr3  0.159        -0.234         0.166  0.150  0.154 -0.149
## basr4 -0.112 -0.131                       0.206 -0.148
## basr5 -0.117 -0.120  0.438  0.212  0.131
## basf1  0.162  0.261 -0.151  0.222  0.251
## basf2                      -0.378  0.179 -0.237 -0.123
## basf3 -0.149                      -0.331         0.244 -0.113
##
##                  PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8   PC9  PC10
## SS loadings    3.766 3.180 1.534 1.277 0.977 0.850 0.826 0.764 0.712 0.663
## Proportion Var 0.198 0.167 0.081 0.067 0.051 0.045 0.043 0.040 0.037 0.035
## Cumulative Var 0.198 0.366 0.446 0.514 0.565 0.610 0.653 0.693 0.731 0.766
##                 PC11  PC12  PC13  PC14  PC15  PC16  PC17  PC18  PC19
## SS loadings    0.627 0.595 0.575 0.516 0.502 0.470 0.446 0.422 0.297
## Proportion Var 0.033 0.031 0.030 0.027 0.026 0.025 0.023 0.022 0.016
## Cumulative Var 0.799 0.830 0.860 0.888 0.914 0.939 0.962 0.984 1.000
```

With the loadings calculated, we can reproduce our correlation matrix.

```
# first multiply our eigenvectors and eigenvalues by t()
rep_cor <- round(eigvec[,1:4] %*% diag(eigens$values[1:4]) %*% t(eigvec[,1:4]), 2)
# extract residuals
# og cor - 4 best eigens
```

```r
resid_cor <- round(og_cor - rep_cor, 2)
resid_cor
```

```
##        bis1  bis2  bis3  bis4  bis5  bis6  bis7 basd1 basd2 basd3 basd4 basr1
## bis1   0.49 -0.12 -0.11 -0.09 -0.01 -0.11 -0.07 -0.08 -0.05 -0.05 -0.08  0.04
## bis2  -0.12  0.50 -0.11 -0.13 -0.09  0.03 -0.07  0.08  0.04 -0.01  0.06 -0.02
## bis3  -0.11 -0.11  0.55 -0.05 -0.10 -0.11 -0.10 -0.09  0.01  0.04 -0.04  0.03
## bis4  -0.09 -0.13 -0.05  0.56 -0.08 -0.09 -0.11  0.06  0.02  0.02 -0.01 -0.03
## bis5  -0.01 -0.09 -0.10 -0.08  0.56 -0.14 -0.02  0.00 -0.01 -0.04 -0.01 -0.02
## bis6  -0.11  0.03 -0.11 -0.09 -0.14  0.51 -0.11  0.07  0.04  0.05  0.04  0.01
## bis7  -0.07 -0.07 -0.10 -0.11 -0.02 -0.11  0.53 -0.05  0.00 -0.03  0.08 -0.01
## basd1 -0.08  0.08 -0.09  0.06  0.00  0.07 -0.05  0.27 -0.08 -0.09 -0.05 -0.02
## basd2 -0.05  0.04  0.01  0.02 -0.01  0.04  0.00 -0.08  0.45 -0.14 -0.13 -0.06
## basd3 -0.05 -0.01  0.04  0.02 -0.04  0.05 -0.03 -0.09 -0.14  0.48 -0.10  0.00
## basd4 -0.08  0.06 -0.04 -0.01 -0.01  0.04  0.08 -0.05 -0.13 -0.10  0.39 -0.01
## basr1  0.04 -0.02  0.03 -0.03 -0.02  0.01 -0.01 -0.02 -0.06  0.00 -0.01  0.43
## basr2  0.03  0.02  0.03 -0.06  0.00  0.06 -0.02  0.09  0.01  0.02  0.04 -0.16
## basr3  0.03  0.00  0.03  0.03 -0.06 -0.05  0.00  0.00 -0.05 -0.11  0.03 -0.10
## basr4  0.02 -0.04  0.00 -0.03 -0.01 -0.03  0.08 -0.01  0.06  0.04  0.01 -0.13
## basr5  0.03  0.00  0.00  0.01 -0.04 -0.03 -0.04 -0.06 -0.06 -0.01  0.00 -0.07
## basf1  0.01 -0.01 -0.02 -0.07  0.10  0.00  0.07  0.03  0.03  0.05  0.01  0.04
## basf2 -0.04  0.02 -0.04 -0.05  0.01 -0.06  0.05 -0.01 -0.04 -0.10 -0.08  0.00
## basf3  0.04  0.02 -0.05 -0.01  0.08 -0.06  0.03  0.02  0.05 -0.09 -0.05 -0.04
##       basr2 basr3 basr4 basr5 basf1 basf2 basf3
## bis1   0.03  0.03  0.02  0.03  0.01 -0.04  0.04
## bis2   0.02  0.00 -0.04  0.00 -0.01  0.02  0.02
## bis3   0.03  0.03  0.00  0.00 -0.02 -0.04 -0.05
## bis4  -0.06  0.03 -0.03  0.01 -0.07 -0.05 -0.01
## bis5   0.00 -0.06 -0.01 -0.04  0.10  0.01  0.08
## bis6   0.06 -0.05 -0.03 -0.03  0.00 -0.06 -0.06
## bis7  -0.02  0.00  0.08 -0.04  0.07  0.05  0.03
## basd1  0.09  0.00 -0.01 -0.06  0.03 -0.01  0.02
## basd2  0.01 -0.05  0.06 -0.06  0.03 -0.04  0.05
## basd3  0.02 -0.11  0.04 -0.01  0.05 -0.10 -0.09
## basd4  0.04  0.03  0.01  0.00  0.01 -0.08 -0.05
## basr1 -0.16 -0.10 -0.13 -0.07  0.04  0.00 -0.04
## basr2  0.69 -0.15 -0.06 -0.17 -0.03 -0.08 -0.02
## basr3 -0.15  0.51 -0.15 -0.10  0.01  0.13  0.07
## basr4 -0.06 -0.15  0.56 -0.15 -0.06 -0.06 -0.07
## basr5 -0.17 -0.10 -0.15  0.50 -0.01  0.03 -0.01
## basf1 -0.03  0.01 -0.06 -0.01  0.41 -0.16 -0.21
## basf2 -0.08  0.13 -0.06  0.03 -0.16  0.43 -0.09
## basf3 -0.02  0.07 -0.07 -0.01 -0.21 -0.09  0.43
```

We now have our loadings from the first 4 vectors, corresponding to 4 components that explain the most variance (eigvalues > 1). Now we want to use this to consider our data, so we'll rotate using varimax orthogonal rotation.

```r
# specifying varimax
pca_var <- principal(q1data_mod, rotate = "varimax", nfactors = 4, missing = TRUE)
summary(pca_var)
```

```
##
```

```
## Factor analysis with Call: principal(r = q1data_mod, nfactors = 4, rotate = "varimax", missing = TRUE
##
## Test of the hypothesis that 4 factors are sufficient.
## The degrees of freedom for the model is 101  and the objective function was  1.05
## The number of observations was  337  with Chi Square =  342.37  with prob <  0.0000000000000000000000
##
## The root mean square of the residuals (RMSA) is  0.07
```

```r
pca_var$loadings
```

```
##
## Loadings:
##       RC2    RC1    RC4    RC3
## bis1   0.655  0.266
## bis2   0.665 -0.215        -0.103
## bis3   0.666
## bis4   0.629 -0.120  0.120  0.110
## bis5   0.603         0.175 -0.213
## bis6   0.681 -0.140
## bis7   0.636               -0.254
## basd1         0.850
## basd2         0.700  0.245
## basd3         0.664  0.244  0.153
## basd4 -0.103  0.704         0.311
## basr1         0.201  0.717  0.105
## basr2                0.537  0.160
## basr3         0.156  0.662 -0.142
## basr4                0.596  0.281
## basr5         0.232  0.658
## basf1 -0.146         0.120  0.742
## basf2         0.333  0.112  0.663
## basf3 -0.115  0.116  0.173  0.718
##
##                  RC2   RC1   RC4   RC3
## SS loadings    3.020 2.557 2.280 1.900
## Proportion Var 0.159 0.135 0.120 0.100
## Cumulative Var 0.159 0.294 0.414 0.514
```

```r
pca_var$residual
```

```
##               bis1         bis2          bis3        bis4          bis5
## bis1   0.494151905 -0.118208722 -0.1081955337 -0.09230344 -0.0150888489
## bis2  -0.118208722  0.498317758 -0.1075011375 -0.12613374 -0.0979281513
## bis3  -0.108195534 -0.107501138  0.5467083331 -0.04963204 -0.1017711785
## bis4  -0.092303441 -0.126133743 -0.0496320421  0.56349175 -0.0847279244
## bis5  -0.015088849 -0.097928151 -0.1017711785 -0.08472792  0.5600763686
## bis6  -0.111723620  0.027857076 -0.1090471725 -0.08756947 -0.1465390588
## bis7  -0.067591209 -0.077241070 -0.1067776604 -0.11471746 -0.0152802019
## basd1 -0.075091423  0.081436582 -0.0812340676  0.06040431  0.0008758808
## basd2 -0.047676440  0.045312359  0.0161535523  0.01955562 -0.0098361565
## basd3 -0.048274540 -0.010104603  0.0326868285  0.02519367 -0.0426556246
## basd4 -0.083419386  0.053980444 -0.0338100191 -0.01066383 -0.0094152853
## basr1  0.038670422 -0.024155318  0.0303784075 -0.03585433 -0.0190993641
```

```
## basr2  0.030901779  0.020872453  0.0270056035 -0.05221104 -0.0001851440
## basr3  0.033707066 -0.007077752  0.0345071860  0.02519799 -0.0675533662
## basr4  0.022817659 -0.042048909 -0.0005317469 -0.03057618 -0.0067051821
## basr5  0.025944763  0.007031271  0.0018363642  0.01206587 -0.0392918794
## basf1  0.006220071 -0.005298065 -0.0108247513 -0.06415493  0.0931910532
## basf2 -0.044408318  0.020741431 -0.0399002226 -0.05579833  0.0042084049
## basf3  0.042625284  0.016253323 -0.0430769133 -0.01510524  0.0773632155
##              bis6         bis7         basd1        basd2        basd3
## bis1  -0.111723620 -0.067591209 -0.0750914229 -0.047676440 -0.048274540
## bis2   0.027857076 -0.077241070  0.0814365816  0.045312359 -0.010104603
## bis3  -0.109047172 -0.106777660 -0.0812340676  0.016153552  0.032686829
## bis4  -0.087569474 -0.114717460  0.0604043097  0.019555618  0.025193666
## bis5  -0.146539059 -0.015280202  0.0008758808 -0.009836157 -0.042655625
## bis6   0.510361673 -0.101911978  0.0678906863  0.036604795  0.047752976
## bis7  -0.101911978  0.526963655 -0.0427385454 -0.001636852 -0.028058728
## basd1  0.067890686 -0.042738545  0.2664834806 -0.076773379 -0.088720448
## basd2  0.036604795 -0.001636852 -0.0767733787  0.445548489 -0.130527823
## basd3  0.047752976 -0.028058728 -0.0887204479 -0.130527823  0.476187111
## basd4  0.039529453  0.084176409 -0.0515005831 -0.130072342 -0.102392773
## basr1  0.015926499 -0.014758621 -0.0210187757 -0.060902544  0.002696499
## basr2  0.064457101 -0.023058977  0.0883496527  0.008347695  0.016210023
## basr3 -0.050501637 -0.006093848 -0.0012106129 -0.056200014 -0.114193402
## basr4 -0.028101912  0.078698262 -0.0111834163  0.062938345  0.033730627
## basr5 -0.032861884 -0.032846568 -0.0554386332 -0.060295203 -0.006145042
## basf1 -0.001019909  0.064324183  0.0226025663  0.031654592  0.046930214
## basf2 -0.052869150  0.047863809 -0.0105747711 -0.040822813 -0.094940552
## basf3 -0.064064576  0.031420391  0.0178105906  0.057097951 -0.089389172
##              basd4        basr1        basr2        basr3         basr4
## bis1  -0.083419386  0.038670422  0.030901779  0.033707066  0.0228176593
## bis2   0.053980444 -0.024155318  0.020872453 -0.007077752 -0.0420489087
## bis3  -0.033810019  0.030378407  0.027005603  0.034507186 -0.0005317469
## bis4  -0.010663832 -0.035854335 -0.052211040  0.025197989 -0.0305761819
## bis5  -0.009415285 -0.019099364 -0.000185144 -0.067553366 -0.0067051821
## bis6   0.039529453  0.015926499  0.064457101 -0.050501637 -0.0281019121
## bis7   0.084176409 -0.014758621 -0.023058977 -0.006093848  0.0786982622
## basd1 -0.051500583 -0.021018776  0.088349653 -0.001210613 -0.0111834163
## basd2 -0.130072342 -0.060902544  0.008347695 -0.056200014  0.0629383453
## basd3 -0.102392773  0.002696499  0.016210023 -0.114193402  0.0337306269
## basd4  0.393192065 -0.005480072  0.034306409  0.028974074  0.0145069034
## basr1 -0.005480072  0.431813493 -0.166245952 -0.100204196 -0.1288706514
## basr2  0.034306409 -0.166245952  0.685960312 -0.149742072 -0.0577339858
## basr3  0.028974074 -0.100204196 -0.149742072  0.509200048 -0.1524475563
## basr4  0.014506903 -0.128870651 -0.057733986 -0.152447556  0.5599806216
## basr5 -0.002197189 -0.071605236 -0.169182363 -0.100398282 -0.1476336769
## basf1  0.003286874  0.035246515 -0.034907804  0.008819500 -0.0579816746
## basf2 -0.077700243  0.002125533 -0.083380081  0.137669242 -0.0660493687
## basf3 -0.051307570 -0.034155818 -0.020069074  0.069047544 -0.0691920035
##              basr5        basf1        basf2        basf3
## bis1   0.025944763  0.006220071 -0.044408318  0.042625284
## bis2   0.007031271 -0.005298065  0.020741431  0.016253323
## bis3   0.001836364 -0.010824751 -0.039900223 -0.043076913
## bis4   0.012065873 -0.064154928 -0.055798332 -0.015105241
## bis5  -0.039291879  0.093191053  0.004208405  0.077363216
## bis6  -0.032861884 -0.001019909 -0.052869150 -0.064064576
```

```
## bis7  -0.032846568  0.064324183  0.047863809  0.031420391
## basd1 -0.055438633  0.022602566 -0.010574771  0.017810591
## basd2 -0.060295203  0.031654592 -0.040822813  0.057097951
## basd3 -0.006145042  0.046930214 -0.094940552 -0.089389172
## basd4 -0.002197189  0.003286874 -0.077700243 -0.051307570
## basr1 -0.071605236  0.035246515  0.002125533 -0.034155818
## basr2 -0.169182363 -0.034907804 -0.083380081 -0.020069074
## basr3 -0.100398282  0.008819500  0.137669242  0.069047544
## basr4 -0.147633677 -0.057981675 -0.066049369 -0.069192004
## basr5  0.498494008 -0.007689842  0.029358394 -0.008889081
## basf1 -0.007689842  0.413270974 -0.155497193 -0.206820310
## basf2  0.029358394 -0.155497193  0.434575957 -0.085952047
## basf3 -0.008889081 -0.206820310 -0.085952047  0.428121058
```

```r
# calculating variance explained
round(eigens$values[1:4] / sum(eigens$values[1:4]), 4)
```

```
## [1] 0.3860 0.3259 0.1572 0.1309
```

How many components emerge, and how much variance do they explain all together? We had 4 components emerge, explaining 38.6%, 32.59%, 15.72%, and 13.09% of the variance respectively. If we add that up, our 4 components explain 99% of the variance.

Are there any residual correlations (i.e., the difference between observed and reproduced) that are large (e.g., >0.2)?

Yes. There are residual correlations greater than .2, all of which occur along the diagonal when the individual item is correlated with itself. However, this isn't meaningful to interpretation.

Do they tend to cluster on one or the other component?

There is one residual correlation of -.21 between items basf1 and basf3. When looking at the component loadings, these items mapped onto RC4 and RC3, which correspond to reward seeking and fun seeking, respectively.

Based on the component loadings in the rotated solution, label each of the components.
* Component 1 (RC2) = Avoidant Behaviors (Avoidant + Opposite Fun/Drive) * Component 2 (RC1) = Drive (Drive + Mostly not Avoidant + Some Reward/Fun)
* Component 3 (RC4) = Reward Seeking (Reward + Some Drive)
* Component 4 (RC3) = Fun Seeking (Fun + Some Drive + Opposite Avoidant)

Question 4

Context: A researcher hypothesizes that fluid intelligence is composed of two underlying factors:
* Sequential processing
* Simultaneous processing She collected 3 measures of the first and 5 measures of the second, using 200 participants.

Using R, run a basic CFA model whereby `sequential` is indicated by the first 3 items (`handmov`, `numbrec`, and `wordord`) and `simultaneous` is indicated by the last 5 items (`gesclos`, `triangle`, `spatmem`, `matanalg`, and `photser`).

```r
#establish our model
model <- '
SEQ  =~ handmov + numbrec + wordord
```

```r
SIM =~ gesclos + triangle + spatmem + matanalg + photser

SEQ ~~ SIM
'
fit <- cfa(model, sample.cov = q4data, sample.nobs = 200)
summary(fit, fit.measures = TRUE)
```

```
## lavaan 0.6-9 ended normally after 43 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        17
##
##   Number of observations                           200
##
## Model Test User Model:
##
##   Test statistic                                38.325
##   Degrees of freedom                                19
##   P-value (Chi-square)                           0.005
##
## Model Test Baseline Model:
##
##   Test statistic                               498.336
##   Degrees of freedom                                28
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.959
##   Tucker-Lewis Index (TLI)                       0.939
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -3779.041
##   Loglikelihood unrestricted model (H1)      -3759.878
##
##   Akaike (AIC)                                7592.082
##   Bayesian (BIC)                              7648.153
##   Sample-size adjusted Bayesian (BIC)         7594.295
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.071
##   90 Percent confidence interval - lower         0.038
##   90 Percent confidence interval - upper         0.104
##   P-value RMSEA <= 0.05                          0.132
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.072
##
## Parameter Estimates:
```

```
##
##   Standard errors                              Standard
##   Information                                  Expected
##   Information saturated (h1) model           Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   SEQ =~
##     handmov           1.000
##     numbrec           1.147    0.181    6.341    0.000
##     wordord           1.388    0.219    6.340    0.000
##   SIM =~
##     gesclos           1.000
##     triangle          1.445    0.227    6.352    0.000
##     spatmem           2.029    0.335    6.062    0.000
##     matanalg          1.212    0.212    5.717    0.000
##     photser           1.727    0.265    6.521    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   SEQ ~~
##     SIM               1.271    0.324    3.918    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .handmov           8.664    0.938    9.237    0.000
##    .numbrec           1.998    0.414    4.831    0.000
##    .wordord           2.902    0.604    4.801    0.000
##    .gesclos           5.419    0.585    9.261    0.000
##    .triangle          3.426    0.458    7.479    0.000
##    .spatmem           9.997    1.202    8.320    0.000
##    .matanalg          5.105    0.578    8.838    0.000
##    .photser           3.482    0.537    6.482    0.000
##     SEQ               2.838    0.838    3.389    0.001
##     SIM               1.834    0.530    3.459    0.001
```

Report all the path weights (with SEs):
From `sequential`:
* handmov 1.000
* numbrec 1.147(0.181)
* wordord 1.388(0.219)

From `simultaneous`:
* gesclos 1.000
* triangle 1.445(0.227)
* spatmem 2.029(0.335)
* matanalg 1.212(0.212)
* photser 1.727(0.265)

Report the covariance between the factors:
1.271(0.324)

Report (any) one fit index: Comparative Fit Index (CFI) = 0.959

Note: I tried to install the {semPlots} package, but it kept crashing RStudio for me. Once I got it to stop doing that, it kept prompting me to install {igraphs} which I did multiple times but still couldn't get semPlots to open. Are there any other alternatives you know of to plot these types of models using a different package and/or any tips you can give me if other students have run into this issue?