Assignment4_AbhishekPandit

Abhishek Pandit

10 November 2019

R Markdown

First, we load the data and scale it correctly.

```
countries <- read.csv('countries.csv') %>%
  column_to_rownames('X') %>%
  as_tibble() %>%
  mutate_if(is.numeric, scale)
```

```
## Warning: package 'bindrcpp' was built under R version 3.5.2
```

MAIN QUESTIONS

How do CFA and EFA differ?

Overall, the goal of FA is to be explicit about the latent features underlying a dataset and to attempt to capture them based on covariances between all the variables in the dataset.

Exploratory Factor Analysis is atheoretical, and seeks to determine the number of latent dimensions are needed to account for the covariation among selected indicators.

Confirmatory Factor Analysis comes into play when the reseachers already have a theory in place regarding the numberand nature ofthe latent constructs and relationships. It then sets out to verify if this selected if the number of factors and input (measured) feature loadings conform to this theory.

Fit three exploratory factor analysis models initialized at 2, 3, and 4 factors.

First, we get some understanding through the scree plot.

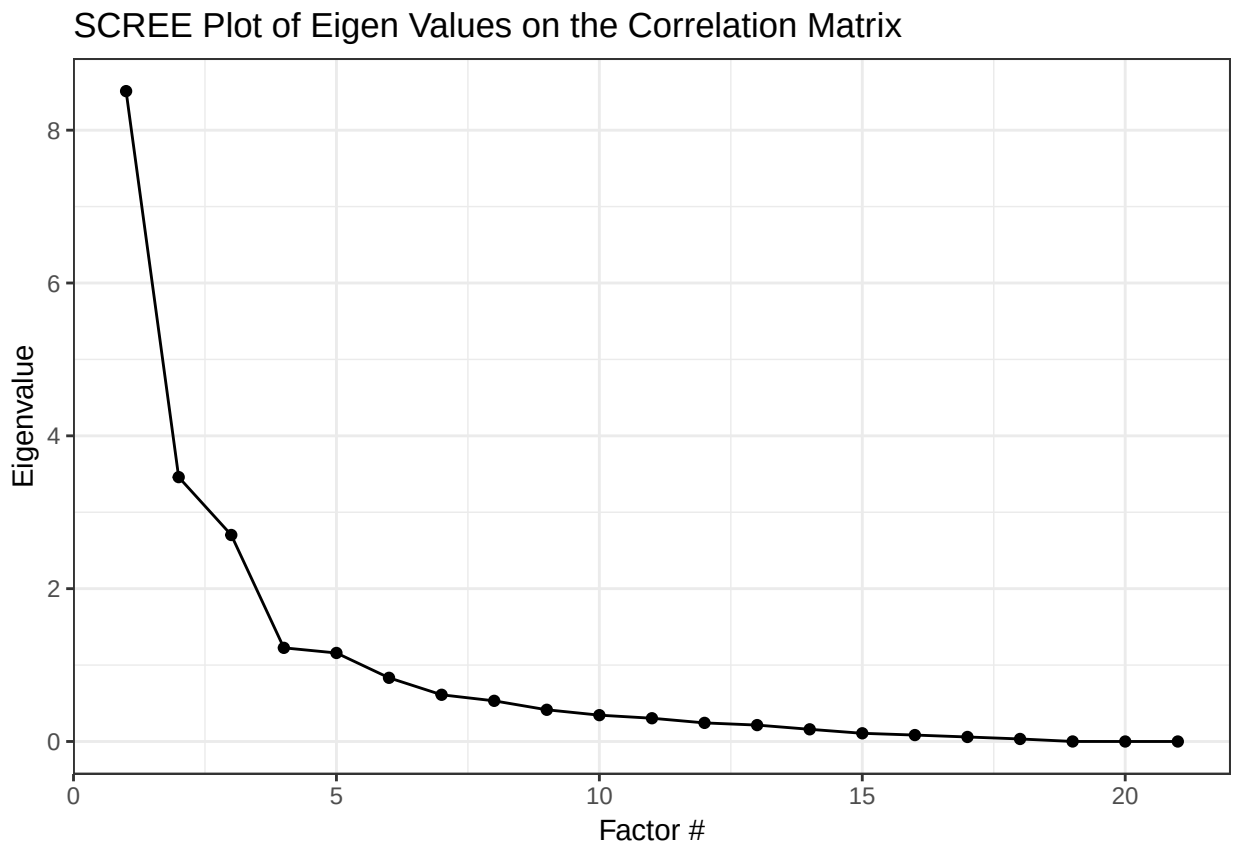
```
# First, (view, then) store the correlation matrix
round(cor(countries), 4)
```

```
##             idealpoint  polity polity2   democ   autoc   unreg physint
## idealpoint      1.0000  0.6085  0.6085  0.6667 -0.4843  0.1186  0.5199
## polity          0.6085  1.0000  1.0000  0.9738 -0.9550  0.3512  0.3214
## polity2         0.6085  1.0000  1.0000  0.9738 -0.9550  0.3512  0.3214
## democ           0.6667  0.9738  0.9738  1.0000 -0.8625  0.3896  0.3912
## autoc          -0.4843 -0.9550 -0.9550 -0.8625  1.0000 -0.2735 -0.2049
## unreg           0.1186  0.3512  0.3512  0.3896 -0.2735  1.0000  0.0841
## physint         0.5199  0.3214  0.3214  0.3912 -0.2049  0.0841  1.0000
## speech          0.4587  0.6490  0.6490  0.6366 -0.6137  0.2975  0.4010
## new_empinx      0.5715  0.8343  0.8343  0.8283 -0.7760  0.3416  0.4829
## wecon           0.3082  0.2617  0.2617  0.3356 -0.1447 -0.0227  0.4111
## wopol           0.3592  0.5092  0.5092  0.4684 -0.5221  0.0681  0.1448
## wosoc           0.5658  0.4288  0.4288  0.5108 -0.2879  0.1578  0.5112
## elecsd          0.5018  0.8475  0.8475  0.8433 -0.7858  0.3489  0.3360
## gdp.pc.wdi      0.4886  0.2930  0.2930  0.4007 -0.1293  0.0254  0.5116
## gdp.pc.un       0.4767  0.2827  0.2827  0.3909 -0.1193  0.0247  0.5093
## pop.wdi        -0.1123 -0.0174 -0.0174 -0.0011  0.0373  0.0061 -0.2275
## amnesty        -0.5107 -0.3008 -0.3008 -0.3664  0.1913 -0.0421 -0.6545
## statedept      -0.6046 -0.3911 -0.3911 -0.4724  0.2541 -0.0196 -0.7969
## milper         -0.0681 -0.0665 -0.0665 -0.0458  0.0882 -0.0020 -0.2218
```

```
## cinc           0.0262  0.0177  0.0177  0.0467  0.0216  0.0170 -0.1222
## domestic9     -0.0724  0.1039  0.1039  0.0741 -0.1345  0.2192 -0.4357
##              speech new_empinx   wecon   wopol   wosoc  elecsd gdp.pc.wdi
## idealpoint   0.4587     0.5715  0.3082  0.3592  0.5658  0.5018     0.4886
## polity       0.6490     0.8343  0.2617  0.5092  0.4288  0.8475     0.2930
## polity2      0.6490     0.8343  0.2617  0.5092  0.4288  0.8475     0.2930
## democ        0.6366     0.8283  0.3356  0.4684  0.5108  0.8433     0.4007
## autoc       -0.6137    -0.7760 -0.1447 -0.5221 -0.2879 -0.7858    -0.1293
## unreg        0.2975     0.3416 -0.0227  0.0681  0.1578  0.3489     0.0254
## physint      0.4010     0.4829  0.4111  0.1448  0.5112  0.3360     0.5116
## speech       1.0000     0.7831  0.1569  0.2901  0.3378  0.6903     0.3082
## new_empinx   0.7831     1.0000  0.2723  0.5110  0.4939  0.8504     0.3310
## wecon        0.1569     0.2723  1.0000  0.3143  0.6562  0.2351     0.4719
## wopol        0.2901     0.5110  0.3143  1.0000  0.4149  0.4371     0.0159
## wosoc        0.3378     0.4939  0.6562  0.4149  1.0000  0.3959     0.5037
## elecsd       0.6903     0.8504  0.2351  0.4371  0.3959  1.0000     0.2971
## gdp.pc.wdi   0.3082     0.3310  0.4719  0.0159  0.5037  0.2971     1.0000
## gdp.pc.un    0.3021     0.3216  0.4654  0.0047  0.4925  0.2904     0.9994
## pop.wdi     -0.0969    -0.1702 -0.1245  0.0381 -0.0671 -0.0658    -0.0579
## amnesty     -0.2848    -0.3535 -0.3393 -0.0585 -0.4288 -0.3174    -0.5360
## statedept   -0.3770    -0.4872 -0.4399 -0.1007 -0.5041 -0.3879    -0.5795
## milper      -0.1359    -0.2330 -0.1732 -0.0358 -0.0915 -0.1018    -0.0330
## cinc        -0.0460    -0.1101 -0.0942  0.0197 -0.0076  0.0028     0.1314
## domestic9   -0.0217    -0.0180 -0.1108  0.0809 -0.1048  0.0395    -0.1377
##             gdp.pc.un pop.wdi amnesty statedept  milper    cinc domestic9
## idealpoint     0.4767 -0.1123 -0.5107   -0.6046 -0.0681  0.0262   -0.0724
## polity         0.2827 -0.0174 -0.3008   -0.3911 -0.0665  0.0177    0.1039
## polity2        0.2827 -0.0174 -0.3008   -0.3911 -0.0665  0.0177    0.1039
## democ          0.3909 -0.0011 -0.3664   -0.4724 -0.0458  0.0467    0.0741
## autoc         -0.1193  0.0373  0.1913    0.2541  0.0882  0.0216   -0.1345
## unreg          0.0247  0.0061 -0.0421   -0.0196 -0.0020  0.0170    0.2192
## physint        0.5093 -0.2275 -0.6545   -0.7969 -0.2218 -0.1222   -0.4357
## speech         0.3021 -0.0969 -0.2848   -0.3770 -0.1359 -0.0460   -0.0217
## new_empinx     0.3216 -0.1702 -0.3535   -0.4872 -0.2330 -0.1101   -0.0180
## wecon          0.4654 -0.1245 -0.3393   -0.4399 -0.1732 -0.0942   -0.1108
## wopol          0.0047  0.0381 -0.0585   -0.1007 -0.0358  0.0197    0.0809
## wosoc          0.4925 -0.0671 -0.4288   -0.5041 -0.0915 -0.0076   -0.1048
## elecsd         0.2904 -0.0658 -0.3174   -0.3879 -0.1018  0.0028    0.0395
## gdp.pc.wdi     0.9994 -0.0579 -0.5360   -0.5795 -0.0330  0.1314   -0.1377
## gdp.pc.un      1.0000 -0.0577 -0.5337   -0.5747 -0.0336  0.1326   -0.1374
## pop.wdi       -0.0577  1.0000  0.3146    0.2421  0.8898  0.8961    0.0635
## amnesty       -0.5337  0.3146  1.0000    0.7439  0.3511  0.2516    0.4018
## statedept     -0.5747  0.2421  0.7439    1.0000  0.2455  0.1405    0.4361
## milper        -0.0336  0.8898  0.3511    0.2455  1.0000  0.9399    0.0949
## cinc           0.1326  0.8961  0.2516    0.1405  0.9399  1.0000    0.0782
## domestic9     -0.1374  0.0635  0.4018    0.4361  0.0949  0.0782    1.0000
```

```r
countrycor <- (cor(countries))

# Next, generate the eigenvalues
ev <- eigen(countrycor) # store EVs on the correlation matrix

qplot(y = ev$values,
      main = 'SCREE Plot of Eigen Values on the Correlation Matrix',
```

```
    xlab = 'Factor #',
    ylab = 'Eigenvalue') +
  geom_line() +
  theme_bw()
```



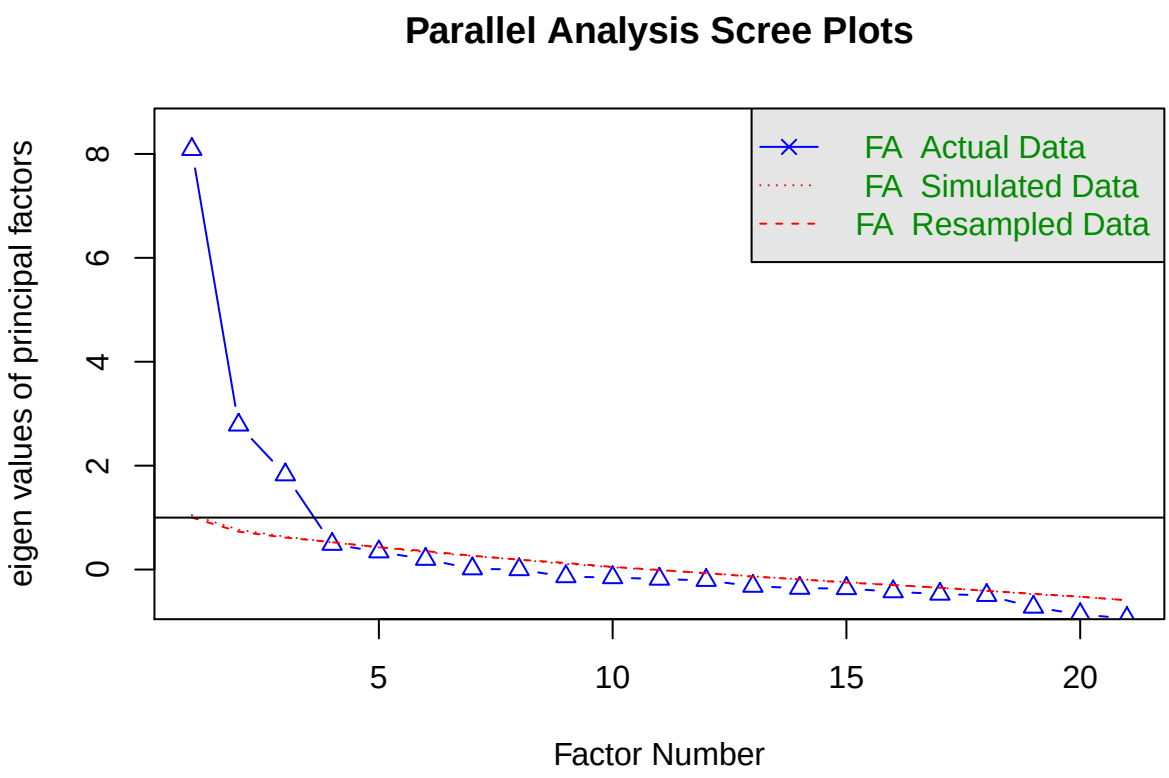
We see the eigenvalues largely levelling after 4 components. So it would make sense to test factor models with 1, 2, 3 and 4 factor models. We will exclude 1 for now.

We also consider the psych package's parallel plot

```
parallel <- fa.parallel(countries, fm = 'minres', fa = 'fa')
```

```
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was
## done

## 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 extraction method.

## In factor.scores, the correlation matrix is singular, an approximation is used
```

```
## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was
## done
```




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

This plot similarly suggests a point of inflection at 4 factors (the point where the gap between simulated data and actual data tends to be minimum.) Thus, the ideal number of factors would lie between 2 and 4

We now proceed with the actual Factor Analysis modelling.

```
# 2-Factor Model
factan.2 <- fa(countries,
               nfactors = 2)
```

```
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was
## done

## 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 extraction method.
```

```
## In factor.scores, the correlation matrix is singular, an approximation is used
```

```
## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was
## done
```

```
#3-Factor Model
factan.3 <- fa(countries,
               nfactors = 3)
```

```
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was
## done
```

```
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was
## done
```

```
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was
## done
```

```
## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was
## done
```

```
## 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 extraction method.
```

```
## In factor.scores, the correlation matrix is singular, an approximation is used
```

```
## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was
## done
```

```
# 4-Factor Model
factan.4 <- fa(countries,
               nfactors = 4)
```

```
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was
## done
```

```
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was
## done
```

```
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was
## done
```

```
## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was
## done
```

```
## 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 extraction method.
```

```
## In factor.scores, the correlation matrix is singular, an approximation is used
```

```
## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was
## done
```

Present the loadings from these solutions and discuss in substantive terms. How does each fit? What sense does this give you of the underlying dimensionality of the space? And so on?

In general, the factor loadings (much like regression coefficients) tell us the degree of correlation between the underlying factor and the observed explanatory variables.

We test out the three factor models respectively.

```
# 2 Factor Model
factan.2$loadings
```

```
##
## Loadings:
##            MR1    MR2
## idealpoint  0.449  0.429
## polity      0.995
## polity2     0.995
## democ       0.931
## autoc      -0.969  0.159
## unreg       0.412 -0.131
## physint             0.782
## speech      0.631  0.154
## new_empinx  0.802  0.197
## wecon               0.509
## wopol       0.551
## wosoc       0.286  0.497
## elecsd      0.852
## gdp.pc.wdi          0.673
## gdp.pc.un           0.671
## pop.wdi     0.204 -0.476
## amnesty            -0.821
## statedept          -0.849
## milper      0.158 -0.468
## cinc        0.211 -0.366
## domestic9   0.288 -0.479
##
##                  MR1   MR2
## SS loadings    6.523 4.527
## Proportion Var 0.311 0.216
## Cumulative Var 0.311 0.526
```

```
factan.2$fit
```

```
## [1] 0.8687218
```

The fit score (0.86) at the bottom suggests that the model is successful in explaining the underlying correlation matrix. Now we examine these with the cutoff of 0.3.

```
print(factan.2$loadings,cutoff = 0.3)
```

```
##
## Loadings:
##            MR1    MR2
## idealpoint  0.449  0.429
## polity      0.995
## polity2     0.995
## democ       0.931
## autoc      -0.969
## unreg       0.412
## physint             0.782
## speech      0.631
## new_empinx  0.802
## wecon               0.509
## wopol       0.551
```

```
## wosoc              0.497
## elecsd      0.852
## gdp.pc.wdi         0.673
## gdp.pc.un          0.671
## pop.wdi           -0.476
## amnesty           -0.821
## statedept         -0.849
## milper            -0.468
## cinc              -0.366
## domestic9         -0.479
##
##                  MR1   MR2
## SS loadings    6.523 4.527
## Proportion Var 0.311 0.216
## Cumulative Var 0.311 0.526
```

We now see that under this approach, largely political variables (polity, democ) load onto the first factor, while largely economic variables (GDP, milper) load onto the second factor.

We apply a similar approach to 3 factors.

```
# 3 Factor Model
factan.3$fit
```

```
## [1] 0.9468818
```

```
factan.3$loadings
```

```
##
## Loadings:
##            MR1    MR2    MR3
## idealpoint  0.432  0.468
## polity      0.992
## polity2     0.992
## democ       0.910  0.144
## autoc      -0.994  0.191
## unreg       0.413 -0.129
## physint            0.737 -0.136
## speech      0.646  0.128
## new_empinx  0.840  0.131 -0.125
## wecon              0.518
## wopol       0.552
## wosoc       0.263  0.547
## elecsd      0.858
## gdp.pc.wdi         0.856  0.158
## gdp.pc.un          0.853  0.157
## pop.wdi                   0.892
## amnesty           -0.715  0.243
## statedept         -0.803  0.144
## milper                    0.949
## cinc                      0.999
## domestic9   0.269 -0.443
##
##                  MR1   MR2   MR3
## SS loadings    6.466 4.275 2.881
## Proportion Var 0.308 0.204 0.137
## Cumulative Var 0.308 0.512 0.649
```

The fit is even higher, at 0.94. We again use the cutoff of 0.3

```
print(factan.3$loadings,cutoff = 0.3)
```

```
##
## Loadings:
##             MR1    MR2    MR3
## idealpoint  0.432  0.468
## polity      0.992
## polity2     0.992
## democ       0.910
## autoc      -0.994
## unreg       0.413
## physint             0.737
## speech      0.646
## new_empinx  0.840
## wecon               0.518
## wopol       0.552
## wosoc               0.547
## elecsd      0.858
## gdp.pc.wdi          0.856
## gdp.pc.un           0.853
## pop.wdi                    0.892
## amnesty            -0.715
## statedept          -0.803
## milper                     0.949
## cinc                       0.999
## domestic9          -0.443
##
##                  MR1   MR2   MR3
## SS loadings    6.466 4.275 2.881
## Proportion Var 0.308 0.204 0.137
## Cumulative Var 0.308 0.512 0.649
```

The relationship among political variables continues, but we now have the 'economic' factor split into 2- pop wdi, milper, and cinc on one hand, and amnesty, statedept, domestic9. This could be thought of as per-capita level vs macroeconomic variables.

Now with 4 Factors:

```
factan.4$fit
```

```
## [1] 0.9597186
```

```
factan.4$loadings
```

```
##
## Loadings:
##             MR1    MR3    MR4    MR2
## idealpoint  0.467          0.214 -0.294
## polity      0.995
## polity2     0.995
## democ       0.922          0.127
## autoc      -0.986          0.146
## unreg       0.405                 0.165
## physint     0.119                -0.761
## speech      0.658                -0.109
```

```
## new_empinx  0.855               -0.145
## wecon       0.105         0.390 -0.170
## wopol       0.555
## wosoc       0.300         0.350 -0.239
## elecsd      0.865
## gdp.pc.wdi                0.986
## gdp.pc.un                 0.979
## pop.wdi             0.923
## amnesty             0.177 -0.197  0.602
## statedept  -0.137         -0.139  0.783
## milper              0.965
## cinc                0.981  0.111
## domestic9   0.247         0.204  0.757
##
##                  MR1   MR3   MR4   MR2
## SS loadings    6.605 2.811 2.426 2.370
## Proportion Var 0.315 0.134 0.116 0.113
## Cumulative Var 0.315 0.448 0.564 0.677
```

As may be expected, the fit does rise- but only marginally from the fit for 3 factors (from 94 to 95%). Most of the loading on the second factor now seem weak- with nothing above 0.75.

We finally apply the same cutoff process to improve ease of interpretation.

```
print(factan.4$loadings,cutoff = 0.3)
```

```
##
## Loadings:
##            MR1    MR3    MR4    MR2
## idealpoint  0.467
## polity      0.995
## polity2     0.995
## democ       0.922
## autoc      -0.986
## unreg       0.405
## physint                       -0.761
## speech      0.658
## new_empinx  0.855
## wecon                  0.390
## wopol       0.555
## wosoc       0.300      0.350
## elecsd      0.865
## gdp.pc.wdi             0.986
## gdp.pc.un              0.979
## pop.wdi             0.923
## amnesty                       0.602
## statedept                     0.783
## milper              0.965
## cinc                0.981
## domestic9                     0.757
##
##                  MR1   MR3   MR4   MR2
## SS loadings    6.605 2.811 2.426 2.370
## Proportion Var 0.315 0.134 0.116 0.113
## Cumulative Var 0.315 0.448 0.564 0.677
```

As had seemed to be the case, we now have a fourth factor with 'weak' loadings from wecon and wecon (weak with respect to other stronger loadings observed under fewer factors).

Overall, from this process it would seem that the dimensionality of the data is in fact either 3 or 4 underlying latent variables. From the large gain in fit between 2 and 3 factors (relative to the gain between 3 and 4 factors), I would posit that there are in fact 3 underlying factors in this data,

We will likewise explore this finding in the next section.

Rotate the 3-factor solution using any oblique method you would like and present a visual of the unrotated and rotated versions side-by-side. How do these differ? And why does this matter (or not)?

We begin with the unrotated version.

```
## Initial (unrotated) factor solution
nonrotated.factors <- fa(cor(countries),
        fm = "pa", # communalities along the diagonal (total variation across features)
        nfactors = 3,
        rotate = "none",
        residuals = TRUE)
```

```
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was
## done
```

```
## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was
## done
```

```
## 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 extraction method.
```

```
## In factor.scores, the correlation matrix is singular, an approximation is used
```

```
## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was
## done
```

```
# loadings / structure
nonrotated.factors$loadings
```

```
##
## Loadings:
##            PA1    PA2    PA3
## idealpoint  0.726         0.162
## polity      0.897  0.366 -0.188
## polity2     0.897  0.366 -0.188
## democ       0.925  0.292
## autoc      -0.778 -0.418  0.319
## unreg       0.283  0.216 -0.139
## physint     0.610 -0.434  0.259
## speech      0.693  0.120 -0.108
## new_empinx  0.884  0.136 -0.196
## wecon       0.445 -0.260  0.212
## wopol       0.456  0.236 -0.132
## wosoc       0.627 -0.158  0.238
## elecsd      0.822  0.263 -0.163
## gdp.pc.wdi  0.558 -0.321  0.543
## gdp.pc.un   0.548 -0.323  0.543
## pop.wdi    -0.176  0.676  0.574
```

```
## amnesty    -0.563  0.517 -0.184
## statedept  -0.671  0.468 -0.283
## milper     -0.217  0.680  0.641
## cinc                0.659  0.733
## domestic9           0.373 -0.213
##
##                  PA1   PA2   PA3
## SS loadings    8.258 3.202 2.512
## Proportion Var 0.393 0.152 0.120
## Cumulative Var 0.393 0.546 0.665
```

```
# Plot unrotated factor pattern
nonrot.pattern <- as.data.frame(nonrotated.factors$loadings[1:8,])
```

Plotting this gives us:

```
nonrot.plot<- xyplot(PA2 ~ PA1, data = nonrot.pattern,
       aspect = 1,
       xlim = c(-.1, 1.2),
       ylim = c(-.5, .8),
       panel = function (x, y) {
         panel.segments(c(0, 0), c(0, 0),
             c(1, 0), c(0, 1), col = "gray")
         panel.text(1, 0, labels = "Initial\n(unrotated)\nfactor 1",
                    cex = .65, pos = 3, col = "gray")
         panel.text(0, .7, labels = "Initial\n(unrotated)\nfactor 2",
                    cex = .65, pos = 4, col = "gray")
         panel.segments(rep(0, 8), rep(0, 8), x, y,
            col = "black")
         panel.text(x[-7], y[-7], labels = rownames(nonrot.pattern)[-7],
            pos = 4, cex = .75)
         panel.text(x[7], y[7], labels = rownames(nonrot.pattern)[7],
                    pos = 1, cex = .75)
       },
       main = "Unrotated Factor Pattern",
       xlab = "",
       ylab = "",
       scales = list(x = list(at = c(0, 1)),
                     y = list(at = c(-.4, 0, .6)))
)
```

In the oblique rotated version, the points appear more closely spaced. We choose the 'oblimin' rotation.

```
oblimin.factors <- fa(cor(countries[,-1]),
                      nfactors = 3,
                      fm = "pa",
                      rotate = "oblimin")
```

```
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was
## done
```

```
## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was
## done
```

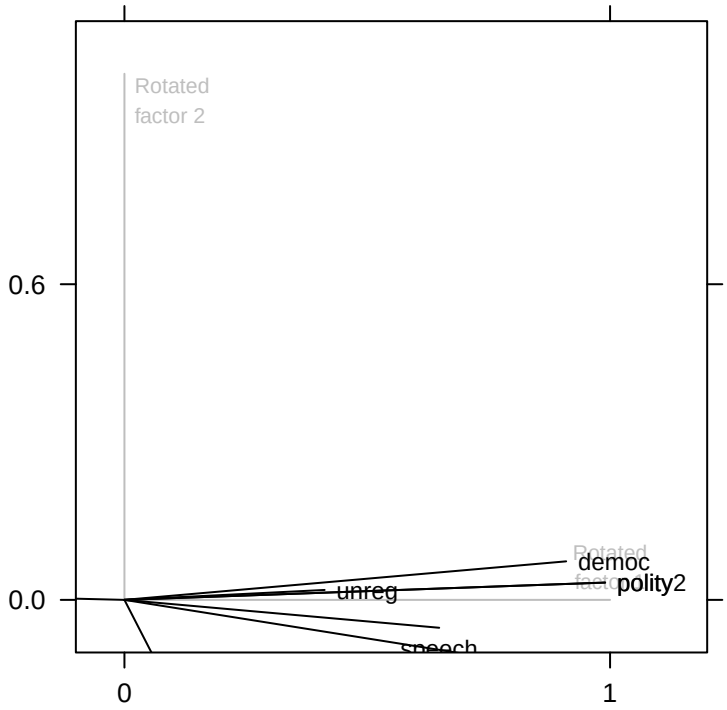
```
## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs
## = np.obs, : The estimated weights for the factor scores are probably
## incorrect. Try a different factor extraction method.
```

```
## In factor.scores, the correlation matrix is singular, an approximation is used
## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was
## done
```

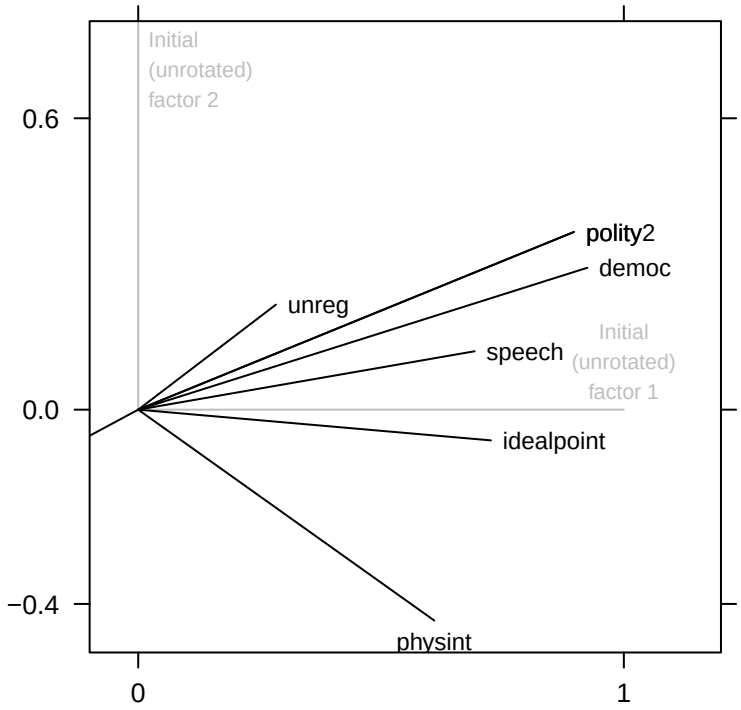
```
# Plot the factor pattern
oblique.pattern <- as.data.frame(oblimin.factors$loadings[1:8,])

oblique.plot <- xyplot(PA2 ~ PA1, data = oblique.pattern,
       aspect = 1,
       xlim = c(-.1, 1.2),
       ylim = c(-.1, 1.1),
       panel = function (x, y) {
         panel.segments(c(0, 0), c(0, 0),
                        c(1, 0), c(0, 1), col = "gray")
         panel.text(1, 0, labels = "Rotated\nfactor 1",
                    cex = .65, pos = 3, col = "gray")
         panel.text(0, .95, labels = "Rotated\nfactor 2",
                    cex = .65, pos = 4, col = "gray")
         panel.segments(rep(0, 8), rep(0, 8), x, y,
                        col = "black")
         panel.text(x[-7], y[-7], labels = rownames(oblique.pattern)[-7],
                    pos = 4, cex = .75)
         panel.text(x[7], y[7], labels = rownames(oblique.pattern)[7],
                    pos = 1, cex = .75)
       },
       main = "Oblique (oblimin) Rotated Factor Pattern",
       xlab = "",
       ylab = "",
       scales = list(x = list(at = c(0, 1)),
                     y = list(at = c(-.4, 0, .6)))
)
```

Now plotting them side by side,

```
par(mfcol=c(2,1))
oblique.plot
```


Oblique (oblimin) Rotated Factor Pattern
Rotated
factor 2
0.6
0.0
democ
polity2
unreg
0
1

```
nonrot.plot
```



After this rotation, the separation into 3 factors seems less clear, due to almost flattening out of the view into 2 dimensions. From this viewpoint, the researcher could be misled into underestimating the spread of factors in the original plot.

Principal Components Analysis

What is the statistical difference between PCA and FA? Describe the basic construction of each approach using equations and then point to differences that exist across these two widely used methods for reducing dimensionality.

Factor Analysis- Mathematical Construction

We get factors/components (F) that we assumed to be the cause of the n observed indicators (Xn), e.g., $X_1 = b_1F + d_1U_1$ $X_2 = b_2F + d_2U_2$ . . . $X_n = b_nF + d_nU_n$

The same may be expanded for more than one Factor. Critically, factors are correlated to with error terms, and error terms themselves are uncorrelated. $cov(F; U_1) = cov(F; U_2) = cov(U_1; U_2) = 0$

Principal Components- Mathematical Construction

We get components/factors that are outcomes built from linear combinations of the k features (X) in the dataset so as to ensure that all components are mutually orthogonal.
$C_1 = L_1X_1 + L_2X_2 + ... + L_kX_k$

This method makes no assumptions of latent structure, components, rotation, etc.

Differences between Factor Analysis and PCA: 1. Interpretation- The factors in factor analysis are conceptualized as "real world" entities such as depression, anxiety, and disturbed thought. This is in contrast to principal components analysis (PCA), where the components are simply geometrical abstractions that may not map easily onto real world phenomena. 2. Causality- In factor analysis, the latent factor(s), is/are causing the responses on the four measured Y variables.There is no such causal relation expected between

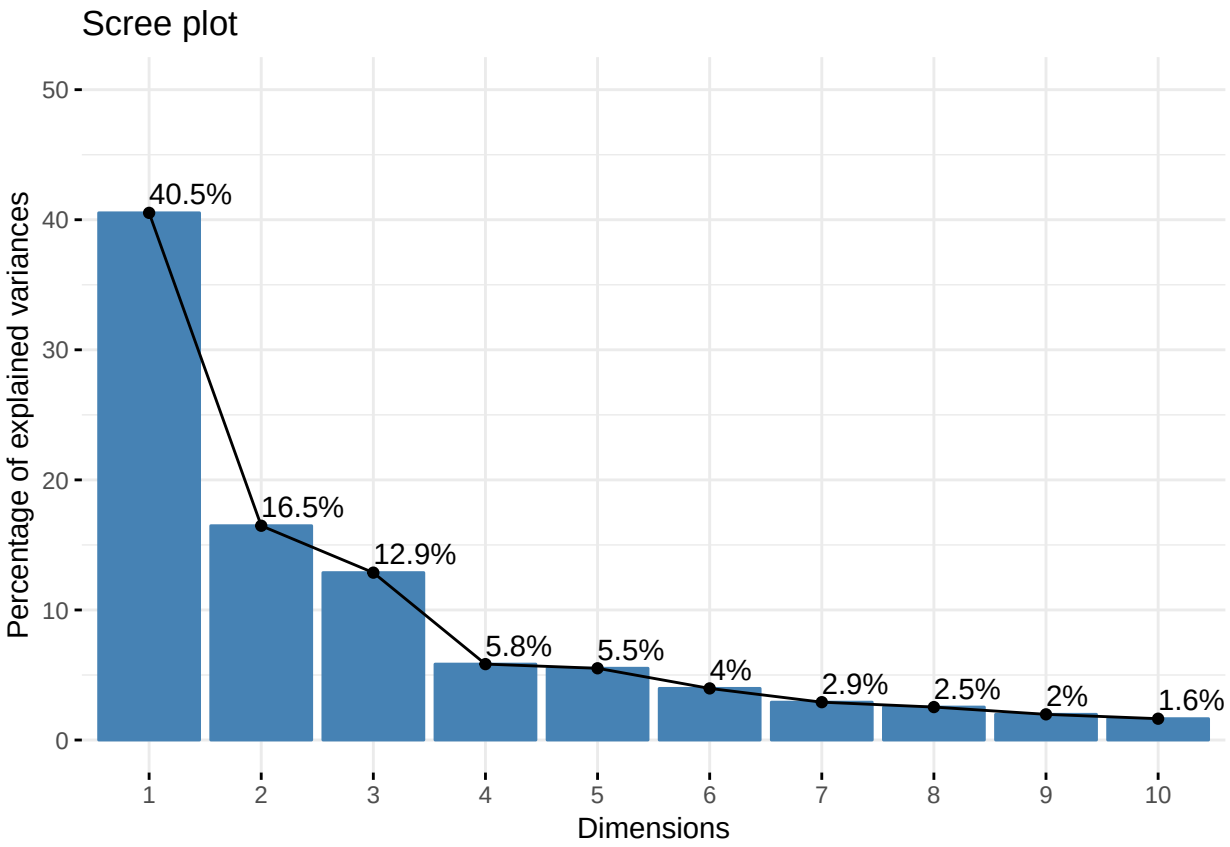
a Principal Component and its constituent variables 3. Goals- The goal is to uncover latency, while PCA extracting actual values

Fit a PCA model. Present the proportion of explained variance across the first 10 components. What do these values tell you substantively (e.g., how many components likely characterize these data?)?

```
res.pca <- prcomp(countries, scale = TRUE)
get_eig(res.pca)
```

```
##          eigenvalue variance.percent cumulative.variance.percent
## Dim.1  8.510913e+00     4.052816e+01                     40.52816
## Dim.2  3.459670e+00     1.647462e+01                     57.00278
## Dim.3  2.702458e+00     1.286885e+01                     69.87162
## Dim.4  1.225737e+00     5.836844e+00                     75.70847
## Dim.5  1.158450e+00     5.516427e+00                     81.22489
## Dim.6  8.333739e-01     3.968447e+00                     85.19334
## Dim.7  6.112287e-01     2.910613e+00                     88.10395
## Dim.8  5.321391e-01     2.533996e+00                     90.63795
## Dim.9  4.150076e-01     1.976227e+00                     92.61418
## Dim.10 3.446022e-01     1.640963e+00                     94.25514
## Dim.11 3.043095e-01     1.449093e+00                     95.70423
## Dim.12 2.434553e-01     1.159311e+00                     96.86354
## Dim.13 2.147082e-01     1.022420e+00                     97.88596
## Dim.14 1.595778e-01     7.598945e-01                     98.64586
## Dim.15 1.073515e-01     5.111976e-01                     99.15705
## Dim.16 8.416652e-02     4.007930e-01                     99.55785
## Dim.17 5.927696e-02     2.822712e-01                     99.84012
## Dim.18 3.317877e-02     1.579942e-01                     99.99811
## Dim.19 3.962048e-04     1.886689e-03                    100.00000
## Dim.20 5.403853e-31     2.573263e-30                    100.00000
## Dim.21 1.080510e-32     5.145287e-32                    100.00000
```

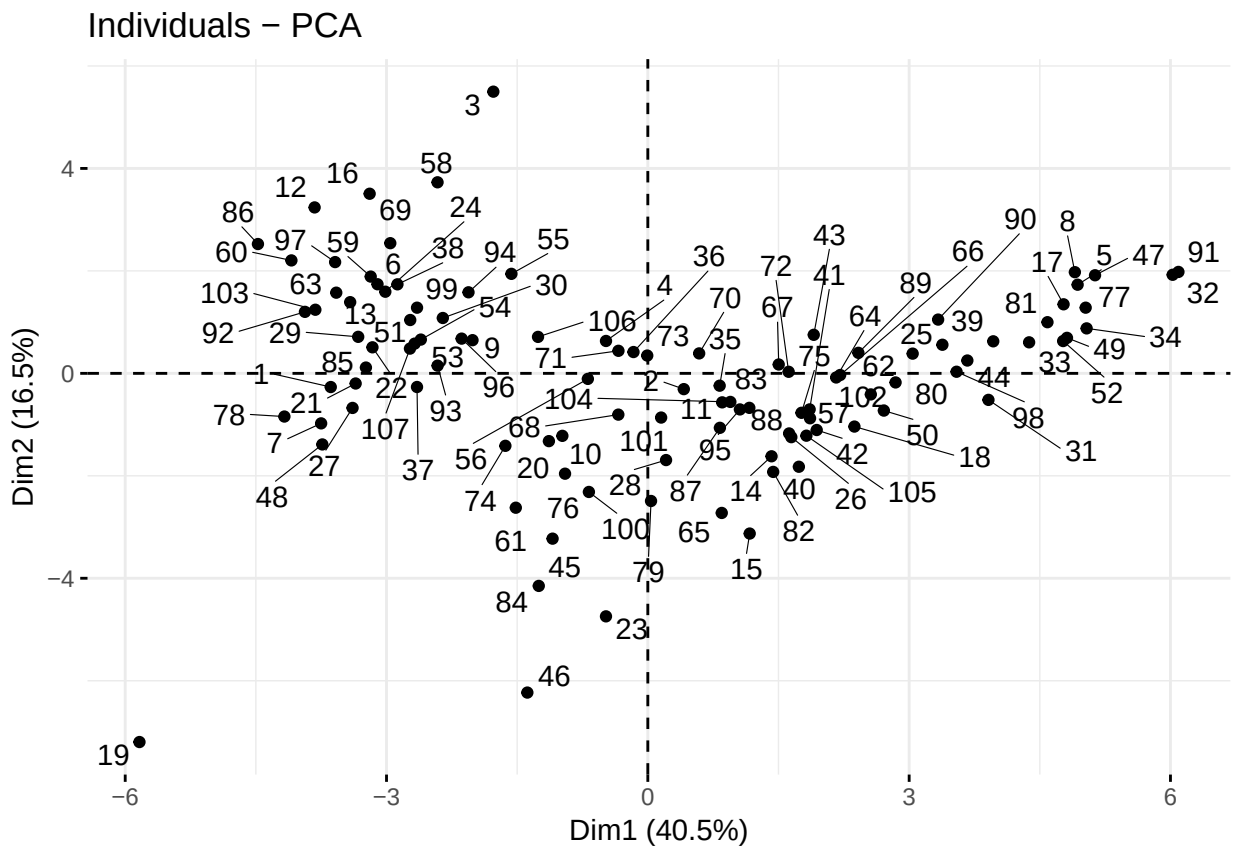
```
fviz_screeplot(res.pca, addlabels = TRUE, ylim = c(0, 50))
```



Based on the numbers presented above, we find that the first 10 components account for 94.2% of the variance. The remainder would be explained by the 4 remaining components.

Present a biplot of the PCA fit from the previous question. Describe what you see (e.g., which countries are clustered together?

```
fviz_pca_ind(res.pca,
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE # Avoid text overlapping (slow if many points)
             )
```



We see about 3-4 clusers with a few marked outliers like China.

The first cluster appears to be of several countries in the Middle East, Islamic former Soviet nations in Central Asia as well as some from Africa (Congo, Rwanda) and others. While the common features are not entirely clear, they do seem to involve oppressive regimes with dictatorships and authoritarian leaders. Interestingly, China is not in the cluster.

The second cluster near the centre involves many South American countries, non-Islamic former Soviet nations. Their uniting traits seem less clear.

Finally, a third cluster on the right seems to match our intuition of liberal democracies with high standards of living, including several European nations which score highly on the Human Development Index.

Which input features are doing the bulk of the explaining? How do you know this?

First, we verify if the biplot (with 2 principal components) adequately summarizes the variance in the data.

Indeed, the first 2 components explain 60% of the variance. Now we check for variables contributing to these high scores.

```
var <- get_pca_var(res.pca)
var$coord
```

```
##                Dim.1      Dim.2       Dim.3       Dim.4       Dim.5
## idealpoint  0.75574120  0.0766019 -0.15621211 -0.01041329  0.05182585
## polity      0.88233042 -0.3905207  0.13132259  0.03614164  0.01171423
## polity2     0.88233042 -0.3905207  0.13132259  0.03614164  0.01171423
## democ       0.91388914 -0.3089785  0.02861504  0.01464104 -0.04665534
## autoc      -0.77129911  0.4659439 -0.25487586 -0.06132097 -0.08691305
## unreg       0.30711777 -0.3070881  0.14938712  0.12247394 -0.47217517
```

```
## physint     0.63310053  0.4910660 -0.18826178  0.21364198  0.15136008
## speech      0.72219942 -0.1660694  0.11736120  0.23749728 -0.09494681
## new_empinx  0.88575456 -0.1814741  0.19131545  0.08589383  0.03175919
## wecon       0.48139052  0.3282174 -0.20701845 -0.59480263  0.13386041
## wopol       0.48873284 -0.3084056  0.11644050 -0.42779015  0.48671372
## wosoc       0.66292371  0.1963410 -0.23758738 -0.43015312  0.11667840
## elecsd      0.82917182 -0.3067658  0.13651309  0.11174469 -0.02802852
## gdp.pc.wdi  0.57365329  0.3923582 -0.50818283 -0.11519418 -0.38986093
## gdp.pc.un   0.56373319  0.3962443 -0.50945471 -0.10955196 -0.39780363
## pop.wdi    -0.17818761 -0.5905566 -0.71085216  0.08004851  0.14695047
## amnesty    -0.58508324 -0.5701812  0.09241488 -0.21687728 -0.02334766
## statedept  -0.68533416 -0.5120235  0.19911282 -0.21873701 -0.11459053
## milper     -0.21676518 -0.5686834 -0.74986819  0.09785998  0.07573166
## cinc       -0.08368972 -0.5315018 -0.81666325  0.08633278  0.02834588
## domestic9  -0.09579610 -0.5018765  0.19395564 -0.46552374 -0.51828460
##                   Dim.6        Dim.7        Dim.8        Dim.9
## idealpoint -0.102816267 -0.443453991 -0.275888066 -0.073468760
## polity     -0.101763070 -0.062870672  0.163995675 -0.023953354
## polity2    -0.101763070 -0.062870672  0.163995675 -0.023953354
## democ      -0.035297142 -0.087004502  0.158012436 -0.032647822
## autoc       0.180390287  0.026416227 -0.158812250  0.010717028
## unreg       0.706517594 -0.085554674  0.065303395  0.162705749
## physint     0.211760398 -0.001702566 -0.148939188 -0.123637268
## speech      0.004001029  0.389212328 -0.351578019 -0.131361244
## new_empinx  0.031049738  0.183882722 -0.145570739  0.008387747
## wecon       0.160244974  0.201248060  0.266380311 -0.214272566
## wopol       0.044507562 -0.013589482 -0.208475344  0.389460754
## wosoc       0.288504030 -0.026069486 -0.093702906 -0.091446943
## elecsd     -0.033456939  0.154002613  0.062102023  0.015778526
## gdp.pc.wdi -0.209642060  0.108713035  0.003410841  0.177461282
## gdp.pc.un  -0.210426864  0.113406851  0.006049836  0.179955406
## pop.wdi     0.075896275  0.032524315  0.056717883 -0.014781904
## amnesty    -0.019094106  0.236420171 -0.103762030 -0.129392219
## statedept  -0.037427972  0.121390035 -0.017721185  0.146811344
## milper      0.045846882 -0.047035697 -0.016537019 -0.061411433
## cinc        0.008690375  0.001531831 -0.039513996  0.005079603
## domestic9  -0.115626023 -0.205228033 -0.193812279 -0.189220073
##                  Dim.10       Dim.11       Dim.12       Dim.13
## idealpoint -0.225318351  0.072509240 -0.171024161 -0.108534790
## polity     -0.014776872  0.042602664 -0.002012630  0.075634249
## polity2    -0.014776872  0.042602664 -0.002012630  0.075634249
## democ      -0.037631473  0.044378513  0.001071620  0.010722128
## autoc      -0.016200624 -0.036912861  0.005875775 -0.154304115
## unreg      -0.015084937  0.061355871 -0.091852640 -0.012492901
## physint     0.269596179  0.200842162  0.097301375  0.115605527
## speech     -0.057230980 -0.155255152 -0.189047562  0.098190225
## new_empinx  0.022071971  0.020975976  0.105227400 -0.077038859
## wecon       0.058106295 -0.012854705 -0.220075294 -0.073956635
## wopol       0.161438064  0.024534627 -0.063595765 -0.009384415
## wosoc      -0.257385719 -0.114641729  0.266314724  0.107715890
## elecsd      0.018444559 -0.103843978  0.146640424 -0.314782625
## gdp.pc.wdi -0.018234464  0.067460486  0.005843573  0.023678223
## gdp.pc.un  -0.011200241  0.067810153  0.007640011  0.022874159
## pop.wdi     0.046323250 -0.104928781 -0.018010015  0.058559706
```

```
## amnesty    -0.124310620  0.407869179  0.034491378 -0.052539951
## statedept  -0.154831577 -0.057069860  0.000110894  0.111457768
## milper      0.006047945  0.002326027 -0.001554101 -0.018014830
## cinc        0.045987196 -0.002146836  0.007639820 -0.046328253
## domestic9   0.273567212 -0.094692952  0.053611118  0.007063254
##                 Dim.14       Dim.15       Dim.16       Dim.17
## idealpoint -0.074548600  0.055249626 -0.001379217  0.034217192
## polity      0.004128221 -0.006219959  0.011361664 -0.020757710
## polity2     0.004128221 -0.006219959  0.011361664 -0.020757710
## democ       0.017188708  0.041586275  0.062130631 -0.138592315
## autoc       0.013232377  0.068077003  0.055751716 -0.134567751
## unreg       0.018062865 -0.014228773 -0.006196791  0.025953688
## physint    -0.183264833  0.024848636  0.040897043  0.000507691
## speech      0.025231771 -0.056688182  0.048906988 -0.009990730
## new_empinx  0.054100233  0.144381496 -0.194148113 -0.036489556
## wecon      -0.058741181  0.014801391 -0.041636533  0.006754492
## wopol       0.040470203 -0.038620428  0.033691351 -0.014139234
## wosoc       0.040361534 -0.041372768  0.015630113  0.017038955
## elecsd     -0.117033437 -0.036337295  0.092736913  0.044438118
## gdp.pc.wdi  0.008673785  0.001369589  0.008766455  0.005475817
## gdp.pc.un   0.005845743  0.002760984  0.011499957  0.006907472
## pop.wdi     0.070076572  0.200519462  0.100373945  0.047977655
## amnesty     0.062820655 -0.001795036  0.050668353  0.020093060
## statedept  -0.289643047  0.069684863 -0.021732361 -0.025387770
## milper     -0.038721800 -0.145663551 -0.048219986 -0.095737695
## cinc       -0.030292722 -0.043831293 -0.088045977  0.052126288
## domestic9   0.018682745  0.004438596  0.005254640  0.001274524
##                 Dim.18        Dim.19        Dim.20        Dim.21
## idealpoint  0.0147825294  1.706889e-04 -4.679351e-31  1.487452e-32
## polity     -0.0121285281 -1.028949e-05 -9.136171e-17  8.842621e-17
## polity2    -0.0121285281 -1.028949e-05  6.173377e-16 -1.912624e-17
## democ      -0.0518161983 -1.228626e-04 -3.083048e-16 -4.062069e-17
## autoc      -0.0405931860 -1.373449e-04  2.363911e-16  3.114570e-17
## unreg       0.0077199128 -4.555640e-06 -8.352631e-32 -1.081922e-32
## physint     0.0002175328 -1.272251e-04  2.065842e-31 -5.770251e-33
## speech     -0.0137326599 -1.770841e-05  2.550422e-32  8.655376e-33
## new_empinx  0.0273327125  1.594106e-04 -4.080675e-31 -2.019588e-32
## wecon       0.0052459939  5.118652e-05 -2.907481e-31 -1.442563e-33
## wopol      -0.0021974995  3.095389e-05  3.570591e-32 -2.019588e-32
## wosoc      -0.0063485912  1.047019e-04  8.161350e-32 -5.770251e-33
## elecsd      0.0145388112 -8.286643e-05  1.606766e-31 -1.731075e-32
## gdp.pc.wdi  0.0182468971 -1.408410e-02  5.228365e-31  1.182901e-31
## gdp.pc.un   0.0107841130  1.405024e-02 -4.896810e-31 -1.442563e-31
## pop.wdi     0.0389942052  4.100540e-05 -1.115810e-31  1.658947e-32
## amnesty     0.0006556237  1.506383e-05 -4.335717e-32 -1.154050e-32
## statedept  -0.0019078395 -4.470309e-05 -8.161350e-32 -5.770251e-33
## milper      0.0951919236  3.839841e-04 -8.512033e-32  8.294735e-33
## cinc       -0.1265258801 -3.931781e-04  1.761385e-31 -1.172082e-32
## domestic9   0.0031335329 -3.336781e-05  7.810667e-32 -3.606407e-34
```

Observing just the top two dimensions, we see the largest contributions (of 0.75 or more) emerging from idealpoint, polity, polity2, democracy, speech, elecsd and empinx. Autocracy also exerts a strong negative influence. It seems that this component refers to democratic rule of law with its associated freedoms.

The last 4 variables contribute to the second dimension (all -0.50 and below), with others exerting milder

influences.

Bonus Question (5 points):

Fit a sparse PCA model and a probabilistic PCA model. Compare these results substantively. What does each tell you and why do these distinctions matter in terms of inference (or not)?

As per R documentation for the respective packages, we know that: 1) "The Sparse PCA Model approach provides better interpretability for the principal components in high-dimensional data settings, since the principal components are formed as a linear combination of only a few of the original variables (rather than all of them, which is normally the case)."

2) "Probabilistic PCA combines an EM approach for PCA with a probabilistic model. The EM approach is based on the assumption that the latent variables as well as the noise are normal distributed."

Reference:

```
#install.packages('sparsepca')
#install.packages('ppca')
```