UML - Problem Set 4

Jesus Pacheco

11/4/2019

Factor analysis

1. CFA versus EFA

The exploratory factor analysis (EFA) lets you identify a potential smaller number of features that may describe the data almost just as well (or closer to) as the complete dataset. Those features (factors) are supposed to account for most of the variation in the data. EFA does not make big assumptions about how many factors are needed to account for the variation, we can "choose" based on the outcome of this analysis, we just need a number smaller than the whole data (dimensionality reduction).

On the other hand, the confirmatory analysis (CFA) tests a certain prior or hypothesis, so we can test whether a predetermined number of latent factors account for the covariates on the data, based on those priors.

In a sense, in CFA a research question may already be formulated based on a theoretical understanding, while in EFA, the research question may emerge from what we see in the analysis.

For example, if we had an indicator or a proxy for a concept (say, wage and other compensations as proxy of household income), if we had a data on all sorts of income in a dataset, we could test the idea of salary and compensations being good indicators of household income (i.e. if it is accounting for the variation of the variable).

2. Scree Plot

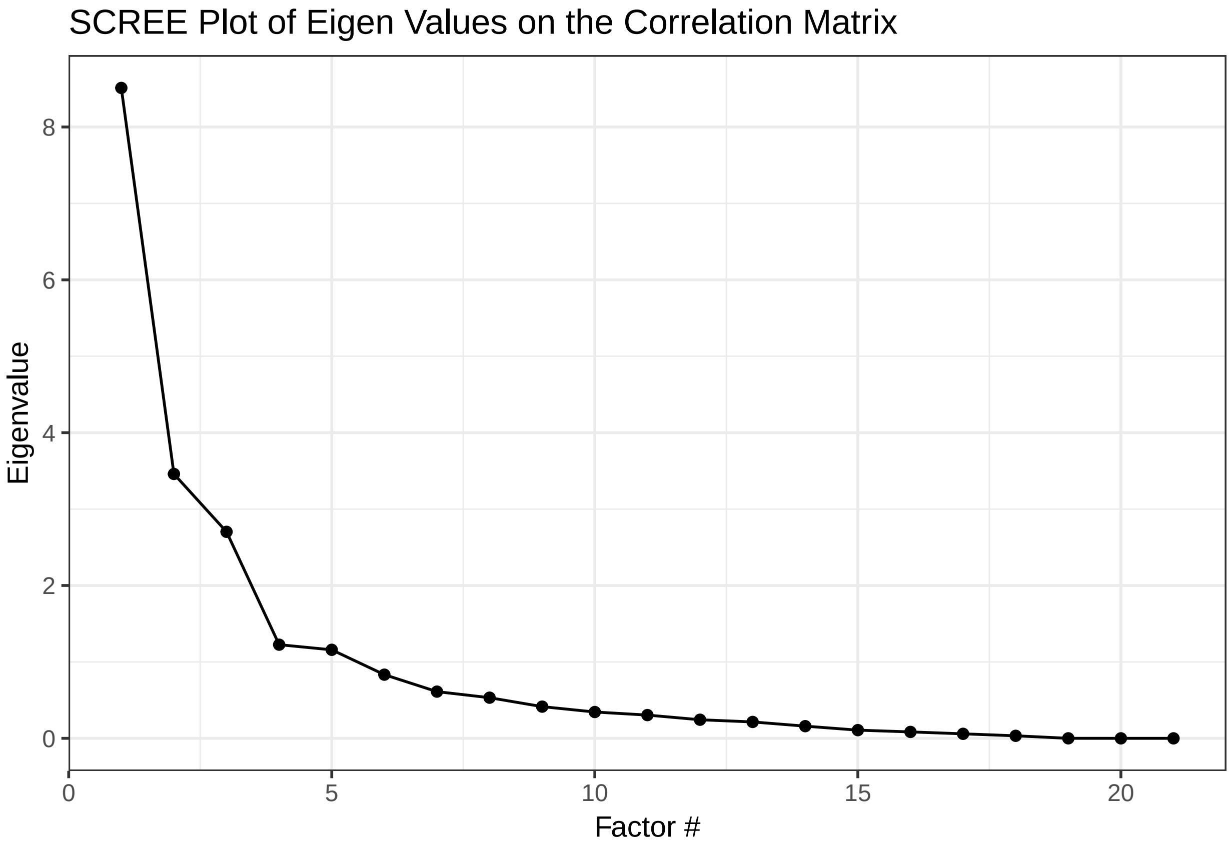
```r
#Setting up the dataset
countries <- as.data.frame( read_csv("https://raw.githubusercontent.com/macss-uml19/Problem-Set-4/maste
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   X1 = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```r
rownames(countries) <- countries[,1]
countries <- countries[,2:22]

countries_scaled <- data.frame(countries %>% scale())

#Correlation matrix and eigenvalues
cor_mat <- cor(countries_scaled)
#smooth_cor <- as.matrix(nearPD(cor_mat)$mat)
eig <- eigen(cor_mat)

#Scree plot
qplot(y = eig$values,
    main = 'SCREE Plot of Eigen Values on the Correlation Matrix',
    xlab = 'Factor #',
```

```r
    ylab = 'Eigenvalue') +
  geom_line() +
  theme_bw()
```




```r
#The 3 factor analyses models and the loadings
fa_2 <- fa(cor_mat,
                nfactors = 2)
```

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

```r
fa_3 <- fa(cor_mat,
                nfactors = 3)
```

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

```r
fa_4 <- fa(cor_mat,
                nfactors = 4)
```

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

```r
fa_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
```

```
fa_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
```

```
fa_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
```

From the scree plot we can see sharp decreases in the eigen values on 2 and 4 factors (elbows). The loadings from the two-factors analysis suggest that the polity and democracy/autocracy indexes are in the same latent factor, while the second one includes probably physical integrity and GDP measures. This is accounting for half of the variation on the data. The three-factor analysis includes one more latent variable that includes 'military capabilities', military and population, which makes sense as well, and is accounting for almost two thirds of the variation. The four-factor analysis includes one more factor with the amnesty variable with domestic conflict, but this additional factor does not have as strong loadings.

Overall, the dimensionality of the data can definitely be reduced, as expected by looking at the variable list since we have different measures of similar concepts like democracy, conflict, military, GDP, etc.

3. Rotation

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)?

```
#UNROTATED FACTOR ANALYSIS
unrotated_fa <- fa(cor_mat,
                   nfactors = 3,
                   rotate="none",
                   residuals =T)
```

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

```r
#unrotated_fa$loadings
unrotated_df <- as.data.frame(unrotated_fa$loadings[,])

fa_3_rot <- fa(cor_mat,
               nfactors = 3,
               rotate= "varimax")
```

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

```r
#ROTATED FACTOR ANALYSIS
rotated_fa <- fa(cor_mat,
                nfactors = 3,
                rotate="varimax",
                residuals = T)
```

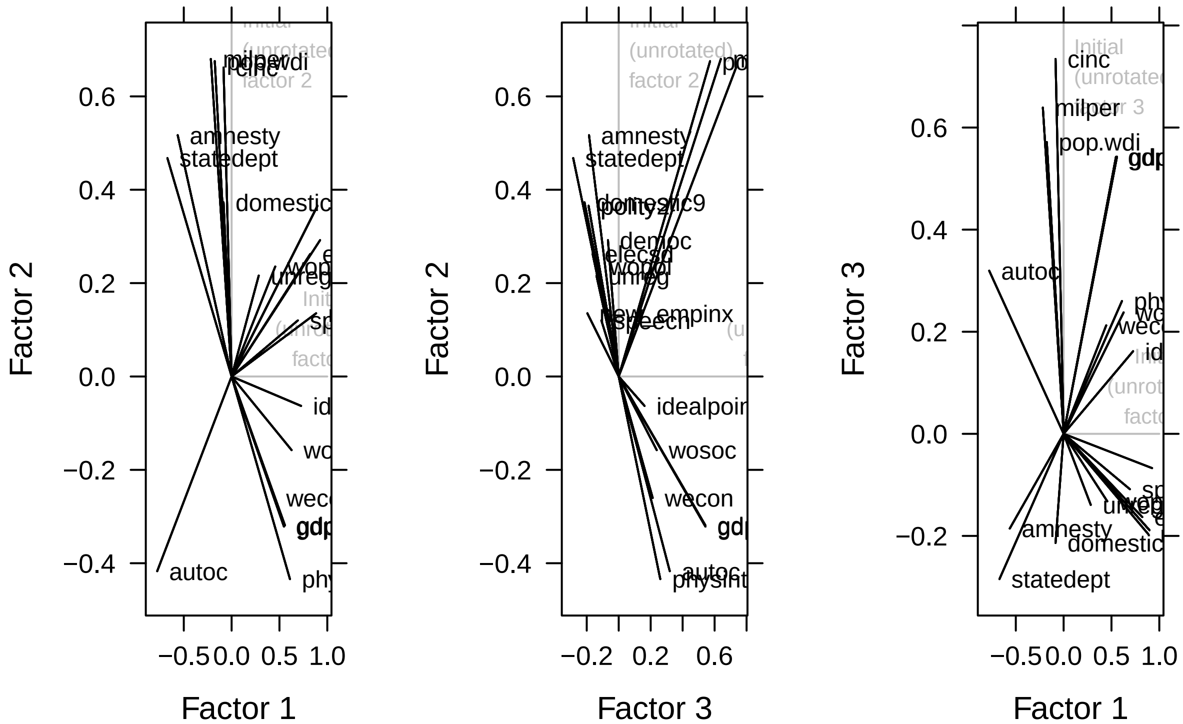
```
## In factor.scores, the correlation matrix is singular, an approximation is used
```

```r
#rotated_fa$loadings
rotated_df <- as.frame(rotated_fa$loadings[,])


#GRAPHS (BEAR WITH ME): I'M DOING THE 6 GRAPHS SEPARETELY
# 3 PLOTS WITH UNROTATED AXES
plot_1 <- xyplot(MR2 ~MR1, data = unrotated_df,
       #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, 106), rep(0, 106), x, y,
                        col = "black")
         panel.text(x, y, labels = rownames(unrotated_df),
                   pos = 4, cex = .75)
       },
       xlab = "Factor 1",
       ylab = "Factor 2"
)

plot_2 <-xyplot(MR2 ~ MR3, data = unrotated_df,
       #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 3",
                   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, 106), rep(0, 106), x, y,
                        col = "black")
         panel.text(x, y, labels = rownames(unrotated_df),
```

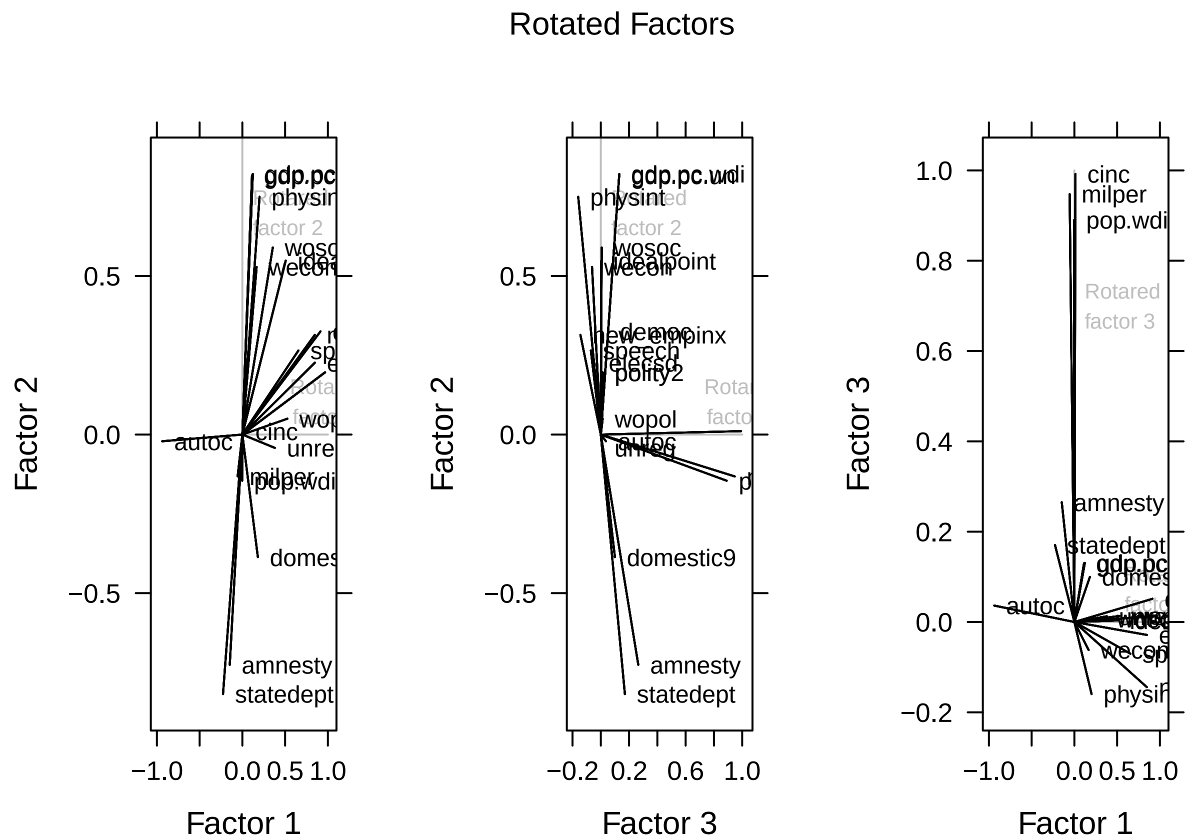
```
                 pos = 4, cex = .75)
       },
       xlab = "Factor 3",
       ylab = "Factor 2"
)

plot_3 <- xyplot(MR3 ~MR1, data = unrotated_df,
                 #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 3",
                              cex = .65, pos = 4, col = "gray")
                   panel.segments(rep(0, 106), rep(0, 106), x, y,
                                  col = "black")
                   panel.text(x, y, labels = rownames(unrotated_df)[],
                             pos = 4, cex = .75)
                 },
                 xlab = "Factor 1",
                 ylab = "Factor 3"
)
# 3 PLOTS WITH UNROTATED AXES
rplot_1 <- xyplot(MR2 ~MR1, data = rotated_df,
         #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 = "Rotared\nfactor 1",
                      cex = .65, pos = 3, col = "gray")
           panel.text(0, .7, labels = "Rotared\nfactor 2",
                      cex = .65, pos = 4, col = "gray")
           panel.segments(rep(0, 106), rep(0, 106), x, y,
                          col = "black")
           panel.text(x, y, labels = rownames(rotated_df),
                                    pos = 4, cex = .75)
                           },
                           xlab = "Factor 1",
                           ylab = "Factor 2"
)

rplot_2 <-xyplot(MR2 ~ MR3, data = rotated_df,
                #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 = "Rotared\nfactor 3",
                             cex = .65, pos = 3, col = "gray")
                  panel.text(0, .7, labels = "Rotared\nfactor 2",
```

```
                              cex = .65, pos = 4, col = "gray")
                  panel.segments(rep(0, 106), rep(0, 106), x, y,
                                 col = "black")
                  panel.text(x, y, labels = rownames(rotated_df),
                              pos = 4, cex = .75)
                },
                xlab = "Factor 3",
                ylab = "Factor 2"
)

rplot_3 <- xyplot(MR3 ~MR1, data = rotated_df,
                 #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 = "Rotated\nfactor 1",
                              cex = .65, pos = 3, col = "gray")
                   panel.text(0, .7, labels = "Rotared\nfactor 3",
                              cex = .65, pos = 4, col = "gray")
                   panel.segments(rep(0, 106), rep(0, 106), x, y,
                                  col = "black")
                   panel.text(x, y, labels = rownames(rotated_df),
                              pos = 4, cex = .75)
                 },
                 xlab = "Factor 1",
                 ylab = "Factor 3"
)

##PLOT THEM ALL
grid.arrange(plot_1, plot_2, plot_3, ncol=3, top ="Unrotated Factors")
```

Unrotated Factors




```r
grid.arrange(rplot_1, rplot_2, rplot_3, ncol=3, top ="Rotated Factors")
```



The graphs didn't quite displayed well in the pdf, but there's little difference between the visual interpretations from the rotated and unrotated axes. The plot between the factors 2 and 3 are the clearest in both cases. In most cases, rotating the axes is helpful to interpret visually the loadings, in my opinion, it wasn't super helpful in this case.

Principal Component Analysis

1. Differences between FA and PCA

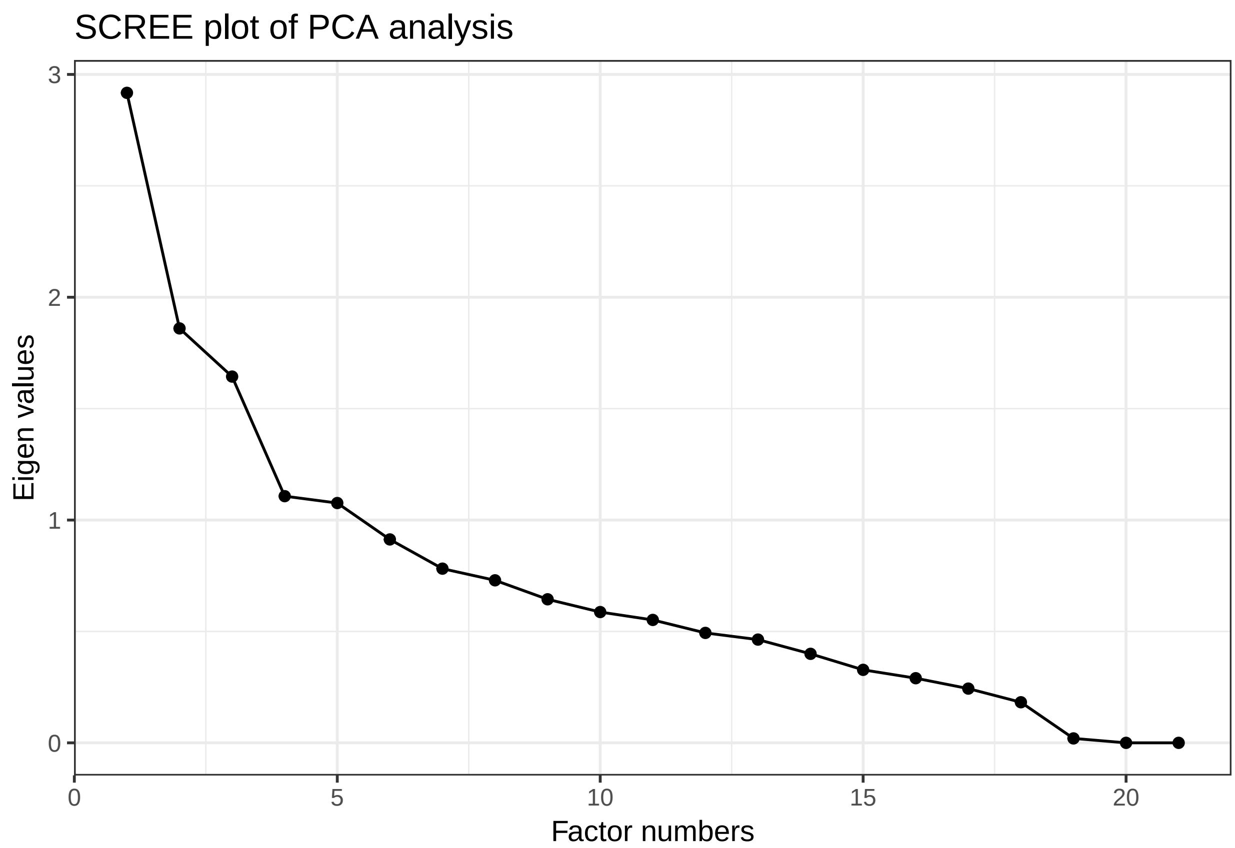
Describe the basic construction of each approach using equations and then point to differences that exist across these two widely used methods for reducing dimensionality. In the Factor Analysis, we are basically estimating a model of a component (what we previously called factors) causing some indicators that we actually observed plus some errors. In more formal words, components that are assumed to be the cause of observed indicators, so that: $X_1 = b_1F + d_iU_1$, $X_2 = b_2F + d_iU_2$ and so on... In PCA, the components are simply linear combinations of all the features, so that $F_1 = L_1X_1 + L_2X_2 + ... + L_kX_k$ In the PCA no distributional assumptions are made, as opposed to the FA where latent variables are assumed to be Gaussian (for factor independence). Due to the need of these assumptions, FA is, in general, less used in the social sciences than PCA. But if some hypothesis needs to be tested, FA could be more helpful.

2. Fitting a PCA model

```
pca_outcome<- prcomp(countries, scale = TRUE)
summary(pca_outcome)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6
## Standard deviation     2.9173 1.8600 1.6439 1.10713 1.07631 0.91289
## Proportion of Variance 0.4053 0.1648 0.1287 0.05837 0.05516 0.03968
## Cumulative Proportion  0.4053 0.5700 0.6987 0.75708 0.81225 0.85193
##                            PC7     PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.78181 0.72948 0.64421 0.58703 0.55164 0.49341
## Proportion of Variance 0.02911 0.02534 0.01976 0.01641 0.01449 0.01159
## Cumulative Proportion  0.88104 0.90638 0.92614 0.94255 0.95704 0.96864
##                           PC13   PC14    PC15    PC16    PC17    PC18
## Standard deviation     0.46337 0.3995 0.32765 0.29011 0.24347 0.18215
## Proportion of Variance 0.01022 0.0076 0.00511 0.00401 0.00282 0.00158
## Cumulative Proportion  0.97886 0.9865 0.99157 0.99558 0.99840 0.99998
##                           PC19      PC20      PC21
## Standard deviation     0.01990 8.602e-16 2.409e-16
## Proportion of Variance 0.00002 0.000e+00 0.000e+00
## Cumulative Proportion  1.00000 1.000e+00 1.000e+00
```

```
qplot(y=pca_outcome$sdev) + geom_line() +
  labs(title="SCREE plot of PCA analysis", y="Eigen values", x= "Factor numbers") +
  theme_bw()
```



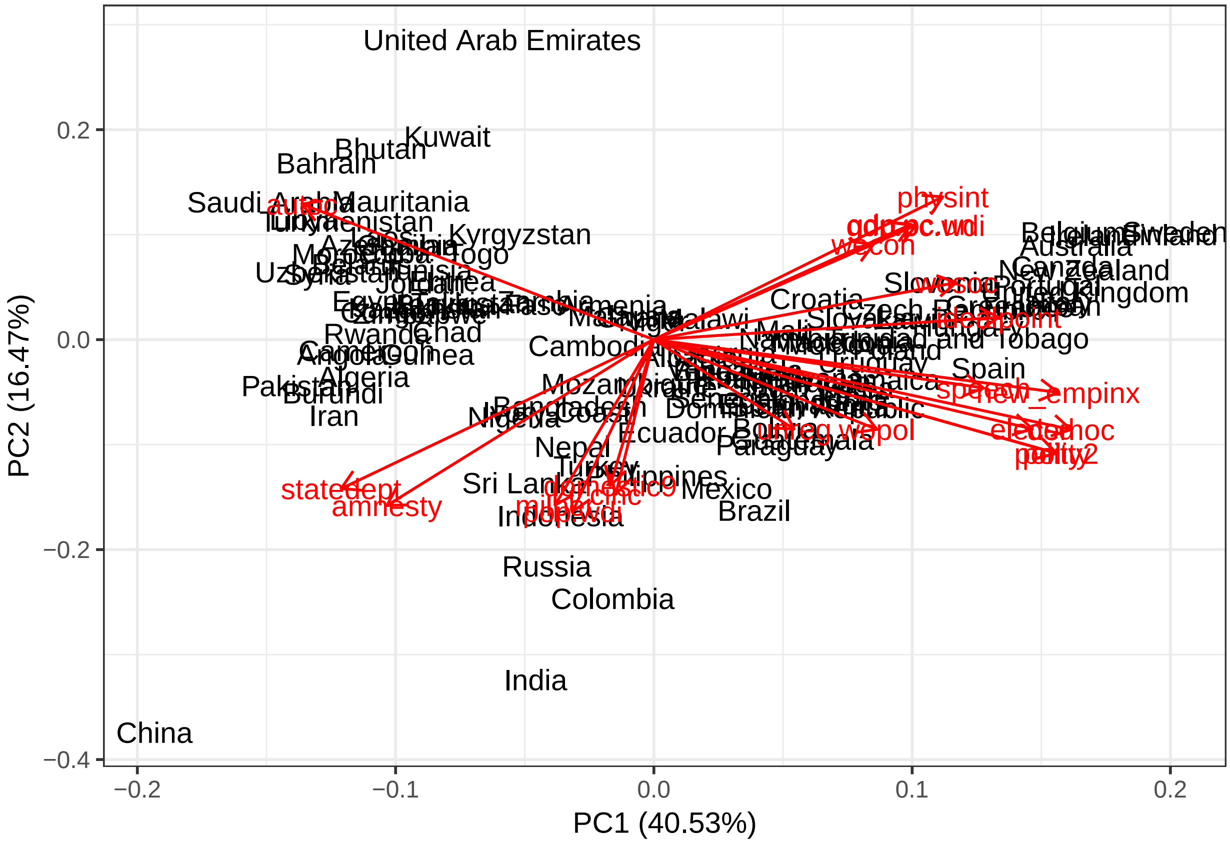

```
POV <- pca_outcome$sdev^2/sum(pca_outcome$sdev^2)
round(POV[1:10],4)
```

```
##  [1] 0.4053 0.1647 0.1287 0.0584 0.0552 0.0397 0.0291 0.0253 0.0198 0.0164
```

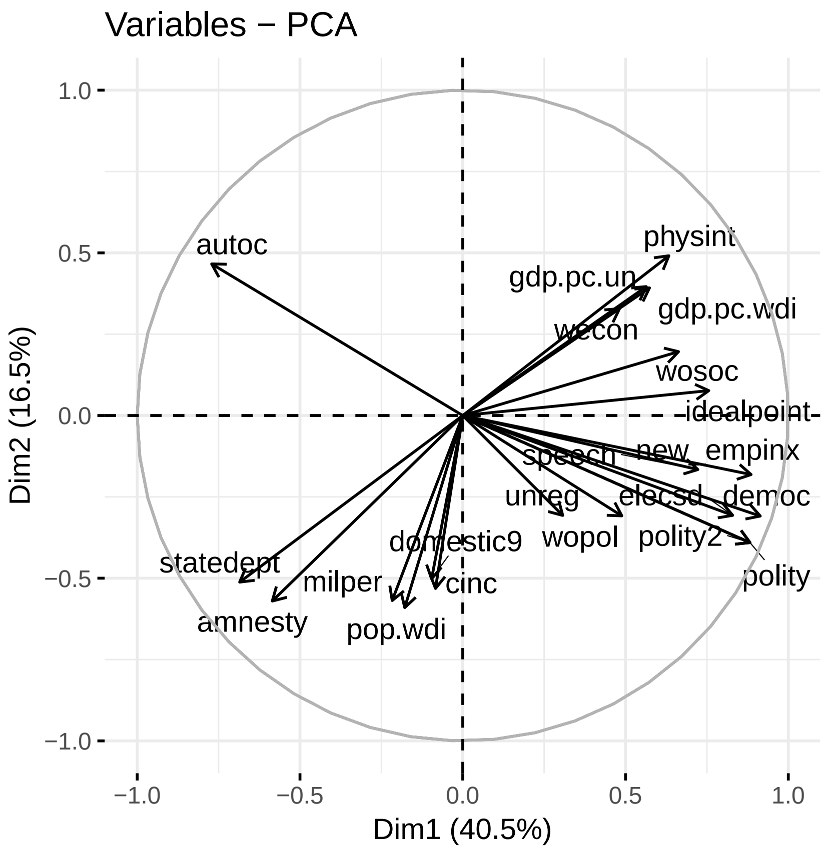
We can see from the scree plot an elbow in the fourth component. But the fifth eigen value is still greater than one. Looking at the proportion of variance explained we can see that the fourth and fifth component pretty much explain the same variation while the sixth one reduces significantly, hence there might be a case for five components.

3. Biplot

```r
autoplot(pca_outcome,
    shape = F,
    loadings.label = T,
    repel=T) +
  theme_bw()
```




```r
#Looking only at the components
fviz_pca_var(pca_outcome,repel = TRUE)
```



Similar to the one of FA interpretations, what we see in the PC1 and PC2 biplot is the features of democratic/civil rights indexes in the lower right hand (and autocratic in the upper left hand). In this sense, we see African and Middle Eastern countries clustered together, and European countries to the far right of the graph. We also some Latin American countries sort of clustered in the free/democratic quadrant but far from the GDP/economic rights factors. Features such as conflict, population and military are guiding the first dimension since they are closer to the vertical axis. Features such as "ideal point", freedom of speech and women social empowerment are doing the bulk of the explaining in the second component. For some reason, military and conflict are guiding more the variation in the indicators than social rights and economic conditions, it could be the case, that overall a bulk of countries are regularly scoring high in economic and social aspects (and low on both), but the conflict and military and conflict indicators are not following this pattern (explaining more of the variance).