Tammy Glazer Homework 4

11/11/2019

Factor Analysis

Question 1: How do CFA and EFA differ?

Exploratory factor analysis explores how many latent dimensions are needed to account for covariation among indicators. It is a statistical method used to uncover the underlying structure of a large set of variables, and is used to identify relationships between the variables. It should be used when a researcher has no a priori hypothesis about factors or patterns of measured variables, and the only a priori assumption is that some indicator is associated with any factor. It is atheoretical, and makes use of factor loadings to intuit the factor structure of data. In EFA, all measured variables are related to every latent variable.

On the other hand, confirmatory factor analysis sets out to determine if the number of factors and measured input feature loadings conform to what is expected on the basis of a pre-established theory. In other words, researchers can specify the number of factors required in the data and which measured variable is related to which latent variable. It is used to test a hypothesis that a specified number of latent factors can account for the covariation in indicators.

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

```
library(tidyverse)
library(psych)
library(lattice)
library(plotly)
library(factoextra)
library(sparsepca)
library(MetabolAnalyze)
```

```
data <- read_csv('countries.csv', col_names=TRUE)
data <- data %>%
        rename(country = X1) %>%
        mutate_at(vars(-country), as.numeric) %>%
        mutate_if(is.numeric, funs(scale))

model2 <- fa(data[-1],
             nfactors = 2)
summary(model2)
```

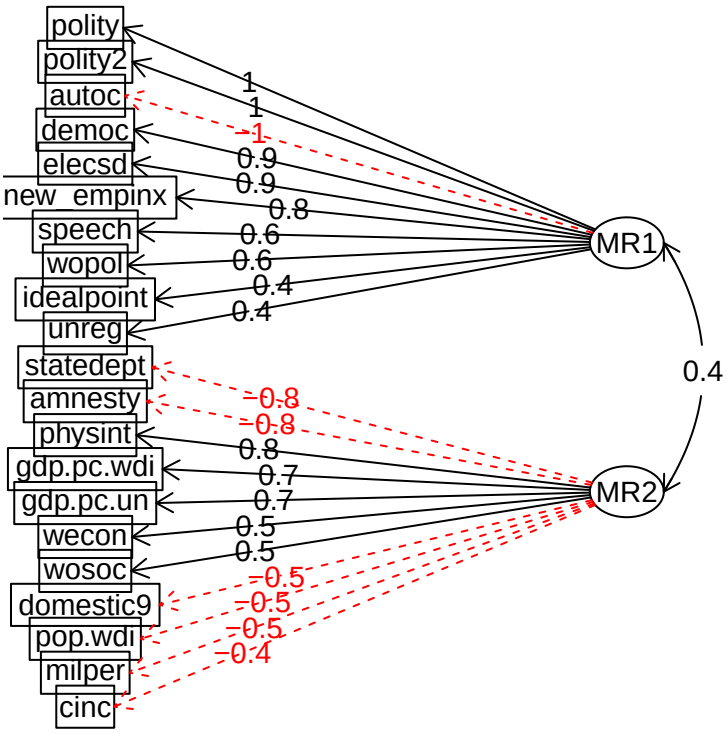
```
##
## Factor analysis with Call: fa(r = data[-1], nfactors = 2)
##
## Test of the hypothesis that 2 factors are sufficient.
## The degrees of freedom for the model is 169  and the objective function was  51.41
```

```
## The number of observations was  107  with Chi Square =  4978.17  with prob <  0
##
## The root mean square of the residuals (RMSA) is  0.12
## The df corrected root mean square of the residuals is  0.14
##
## Tucker Lewis Index of factoring reliability =  0.081
## RMSEA index =  0.543  and the 10 % confidence intervals are  0.506 NA
## BIC =  4188.46
##  With factor correlations of
##      MR1  MR2
## MR1 1.00 0.41
## MR2 0.41 1.00
```

```
model2$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.diagram(model2)
```




```
model3 <- fa(data[-1],
             nfactors = 3)
summary(model3)
```

```
##
## Factor analysis with Call: fa(r = data[-1], nfactors = 3)
##
## Test of the hypothesis that 3 factors are sufficient.
## The degrees of freedom for the model is 150  and the objective function was  46.65
## The number of observations was  107  with Chi Square =  4486.65  with prob <  0
##
## The root mean square of the residuals (RMSA) is  0.06
## The df corrected root mean square of the residuals is  0.07
##
## Tucker Lewis Index of factoring reliability =  0.06
## RMSEA index =  0.549  and the 10 % confidence intervals are  0.509 NA
## BIC =  3785.72
##  With factor correlations of
##       MR1   MR2   MR3
## MR1  1.00  0.38 -0.05
## MR2  0.38  1.00 -0.12
## MR3 -0.05 -0.12  1.00
```

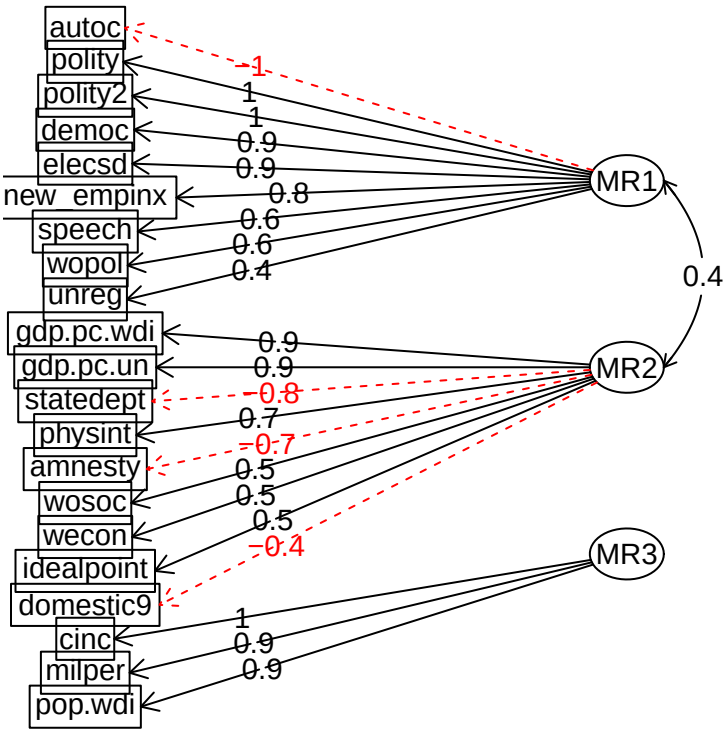
```
model3$loadings
```

```
##
## 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.diagram(model3)
```

Factor Analysis




```
model4 <- fa(data[-1],
             nfactors = 4)
summary(model4)
```

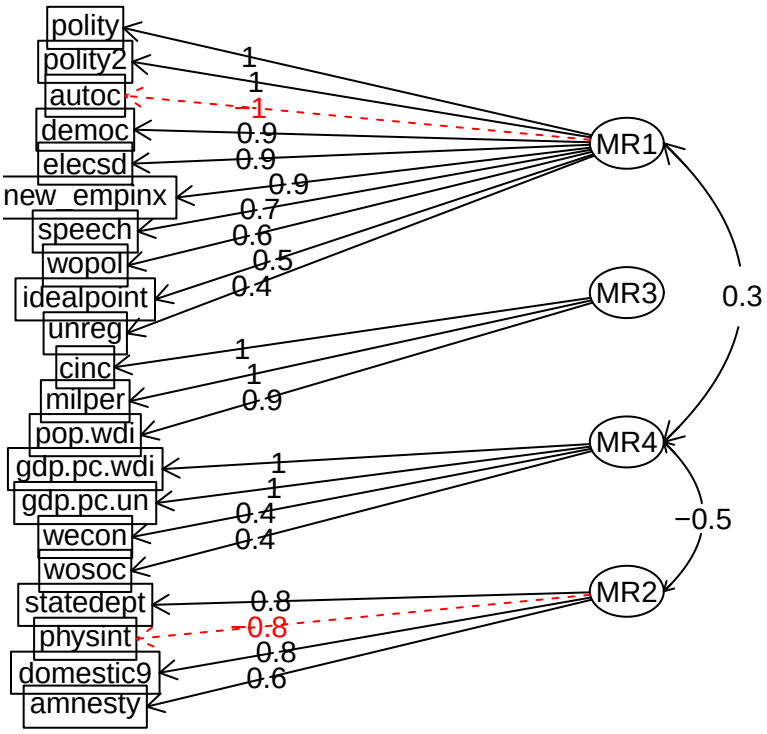
```
##
## Factor analysis with Call: fa(r = data[-1], nfactors = 4)
##
## Test of the hypothesis that 4 factors are sufficient.
## The degrees of freedom for the model is 132  and the objective function was  43.56
## The number of observations was  107  with Chi Square =  4160.02  with prob <  0
##
## The root mean square of the residuals (RMSA) is  0.04
## The df corrected root mean square of the residuals is  0.05
##
## Tucker Lewis Index of factoring reliability =  0
## RMSEA index =  0.566  and the 10 % confidence intervals are  0.523 NA
## BIC =  3543.21
##  With factor correlations of
##       MR1   MR3   MR4   MR2
## MR1  1.00 -0.06  0.32 -0.29
## MR3 -0.06  1.00  0.00  0.23
## MR4  0.32  0.00  1.00 -0.52
## MR2 -0.29  0.23 -0.52  1.00
```

```
model4$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
```

```
fa.diagram(model4)
```



When the EDA model was initialized at 2, I was testing the null hypothesis that 2 factors are sufficient to predict the data. The chi-square goodness-of-fit test of whether the model fits perfectly has a high score and a p-value of zero, which indicates that we can reject the null hypothesis. In other words, 2 factors do not seem to predict the data well and are not fully capturing the variance in the variable set. RMSA strives for a score below 0.08 to indicate good-fit (this model has a score of 0.12) and RMSEA is a parsimony-adjusted index. Lower Bayesian Information Criterion (BIC) scores indicate better-fitting models, while this model has a high score of 4188.

Factor loading coefficients relate the latent variable to the observed feature. Ideally, there will be strong (>0.8) loadings, since the square of the factor loading can be translated as item reliability. However, this model has largely low- and mid-range loadings. The cumulative proportion of the variance explained is 0.3 for factor 1 and 0.5 for factor 2, indicating that likely more than two factors describe the underlying dimensionality of the space. That said, a factor is worth keeping if the sum of squared loadings is greater than 1. In this case, both factors have high SS loadings, indicating that each may be describing more than one latent dimension.

When the EDA model was initialized at 3, the chi-square goodness-of-fit test has a high score and p-value zero, which indicates that we can again reject the null hypothesis that 3 factors seem to predict the data well. RMSA has a lower score of 0.06 and the BIC score is slightly lower at 3786, potentially indicating better-fit than the 2-factor model. In this case, 6 loadings on the first factor are have values at or above 0.8 as well as all 3 loadings on the third factor, indicating potentially strong relationships between latent variables and the observed features for these factors. The cumulative proportion of the variance explained is 0.3 for factor 1, 0.5 for factor 2, and 0.6 for factor 3. Overall, this model seems to better fit the data than the first.

Finally, when the EDA model was initialized at 4, the chi-square goodness-of-fit test had a higher score than the prior model with a p-value of less than zero, which again indicates that we can reject the null hypothesis that 4 factors seem to predict the data well. RMSA has an even lower score of 0.04 and the BIC score is slightly lower at 3543, potentially indicating better-fit than the prior 2 models. In this case, 6 loadings on the first factor have values at or above 0.8, as well as all 3 loadings on the third factor, 2 loadings on the

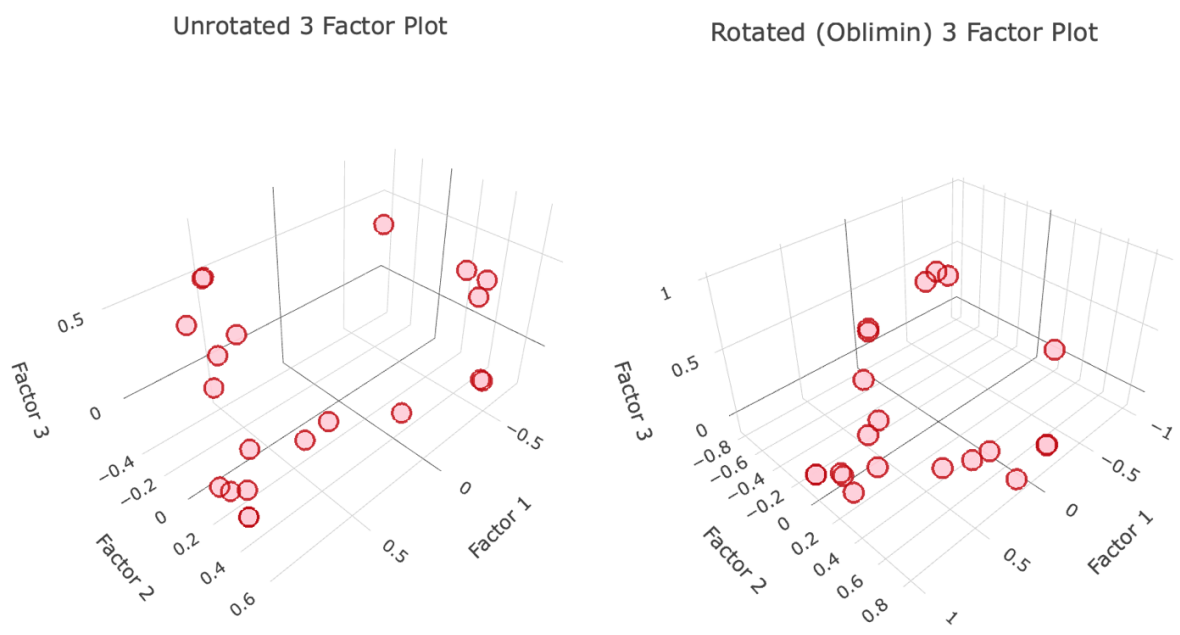
fourth factor, and 3 loadings on the second factor. This builds a case that the latent variables have strong relationships with the observed features. The cumulative proportion of the variance explained is 0.3 for factor 1, 0.4 for factor 3, 0.5 for factor 4, and 0.7 for factor 2. Overall, this model seems to best fit the data, and indicate that there are at least 4 underlying latent variables that define the space, though there may be additional unobservables explaining variance in the variable set since we rejected the null hypothesis.

Question 3: 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
factors <- fa(cor(data[,-1]),
           fm = "pa",
           nfactors = 3,
           rotate = "none",
           residuals = TRUE)
pattern <- as.data.frame(factors$loadings[1:21,])
p_un <- plot_ly(pattern, x= ~PA1, y= ~PA2, z= ~PA3,
        marker = list(size = 8,
                      color = 'rgba(255, 182, 193, .9)',
                      line = list(color = 'rgba(152, 0, 0, .8)',
                                  width = 2))) %>%
        layout(title = "Unrotated 3 Factor Plot",
               scene = list(xaxis = list(title = 'Factor 1'),
                            yaxis = list(title = 'Factor 2'),
                            zaxis = list(title = 'Factor 3')))

# Orthogonal (oblimin) rotated factor solution
factors <- fa(cor(data[,-1]),
                fm = "pa",
                nfactors = 3,
                rotate = "oblimin")
pattern <- as.data.frame(factors$loadings[1:21,])
p_rot <- plot_ly(pattern, x= ~PA1, y= ~PA2, z= ~PA3,
         marker = list(size = 8,
                      color = 'rgba(255, 182, 193, .9)',
                      line = list(color = 'rgba(152, 0, 0, .8)',
                                  width = 2))) %>%
         layout(title = "Rotated (Oblimin) 3 Factor Plot")

subplot(p_un, p_rot, nrows=1)
```



Because there are infinitely many solutions that are equally "good" in terms of variance explained, factor solutions are only identified up to a rotation. Rotating means moving the factors around in space so they explain the same amount of variance but so they are more easily interpretable and display "simple structure". All we are doing with rotation is changing the reference axes, which are arbitrary anyway, with a goal of making patterns more obvious, rather than changing patterns. Furthermore, an oblique method assumes that latent factors are correlated, and rotates axes based on the maximum correlation of loadings within factors across features. In this case, I apply oblimin rotation, which allows the factors to not be orthogonal and therefore can sometimes lead to more interpretable results.

In this case, many variables in the unrotated plot appear to be spread apart, with a majority of variables loading on factors 2 and 3 at low values, and a few variables loading on factor 1 at higher values. Meanwhile, the rotated plot has more tightly grouped variables, with a clear clustering of variables loading on factor 2 at low values, and a group of 3 variables loading on both factors 1 and 3. There are fewer points in the middle of the graph, making it easier to distinguish where each point lies. It is generally important to view the factors in different ways in order to draw and test conclusions, but it is important to remember that they ultimately convey the same information.

Principal Components Analysis

Question 1: 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 is a set of techniques for uncovering the latent (unobserved) dimensionality of a set of input features. We are trying uncover if a latent factor can account for covariation among a set of observed features. The focus is uncovering latent dimensionality, and we are trying to come up with the simplest model that explains the observed covariance in the population, based on a sample. It examines the interrelationships among a large number of input features and attempts to explain them in terms of their common underlying dimensions. This can be used to help bypass multicollinearity issues, or to improve the reliability of aggregated scales or high-dimensional feature spaces. In factor analysis, the variation in individual input features

is not solely determined by latent factors, but variation between input features is completely determined by a common factor.

Suppose:
Xi,. . . ,Xp = a set of p observable variables
F1 = a single underlying feature (for example)
Ui = a distinct unobserved error term for each X that is uncorrelated with Y

Then:
X1 = b1F1 + d1U1
Xp = b2F1 + d2U2
where b = the covariance between F1 and Xi (factor loading)
and the covariance between F1 and any observable Ui = 0

In factor analysis, we get factors or components that are assumed to be the cause of observed indicators. We typically assume the latent variables in FA are Gaussian, which allows us to assert independence, and we use FA to reorder rather than reduce complexity. The square of the factor loadings, hˆ2, is the communality of input features. The flow of factor analysis is as follows: 1) Start with a feature space to reduce; 2) Assume a mathematical combination of the features that maximizes the variance you can predict across all features (first factor); 3) Create a new combination from the residual variance that orthogonally maximizes the variance you can predict (second factor); 4) Continue until all the variance is accounted for; 5) Select the minimal number of factors that captures the most amount of variance; 6) Interpret the factors. In factor analysis, knowledge of correlation among observed features does not lead to knowledge of underlying strucure, and multiple underlying structures can lead to the same observed correlations. Ultimately, CFA makes sense when there is a clear expectation requiring some hypothesis to be tested.

Meanwhile, in PCA we get components or factors that are outcomes built from combinations of the observed items with no assumptions around latent structure, components, rotation, etc. PCA stops short of imposing any distributional assumptions on latent factors, allowing for true statistical independence. PCA is most often used over FA in statistics, and provides a low-dimensional representation of a dataset that captures as much information as possible. Its focus is explicitly reducing dimensionality (eg. from 2 dimensions to 1), and it allows us to represent each data point via scores on a new component (eg. z1). In this case, loadings refer to the amount each component is weighted towards each variable. PCA works by finding a sequence of linear combinations across p features that have maximal variance and are mutually uncorrelated. Maximal variance defines some line that is as close as possible to the data via Euclidean distance. For instance, the first component represents the direction along which observations vary the most, and each succeeding component has the highest variance possible under the constraint that it is orthogonal to the preceding components.

Suppose:
Xi,. . . ,Xp = a set of p observable variables

Then:
Comp1 = L1X1 + L2X2 + . . . + LkXk

And Suppose:
Sigma = Loadings of the first component

Then:
The first PC of a set of features is the normalized linear combination of the features that has the largest variance (optimal loading vector for),
Z1 = Sigma11X1 + Sigma21X2 + . . . + Sigmap1Xp

Note that PCA is subject to the normalized constraint, so the optimization problem is defined as: 1/n x Summation(from i to n) of Ziˆ2. As many as min(n-1, p) principal components can be computed, and standardization is vital for PCA. PCs are constrained to be uncorrelated. When the number of PCs is sufficiently large, PC score vectors and loading vectors can provide a good approximation of the data.

Question 2: Fit a PCA model. Present the proportion of explained variance aross the first 10 components. What do these values tell you substantively (eg. how many components likely characterize these data)?

```
# For explanatory purposes, I demonstrate the PCA steps below
# And conclude with the prcomp function

# covariance matrix
data_cov <- data %>%
            select(-country) %>%
            cov()

# calculate and store eigenvectors
data_eigen <- eigen(data_cov)

# extract first ten loadings
phi <- data_eigen$vectors[, 1:10]
phi <- -phi
colnames(phi) <- c("PC1", "PC2", "PC3", "PC4", "PC5", "PC6",
                   "PC7", "PC8", "PC9", "PC10")

# Calculate Principal Components scores
PC1 <- as.matrix(select_if(data, is.numeric)) %*% phi[,1]
PC2 <- as.matrix(select_if(data, is.numeric)) %*% phi[,2]
PC3 <- as.matrix(select_if(data, is.numeric)) %*% phi[,3]
PC4 <- as.matrix(select_if(data, is.numeric)) %*% phi[,4]
PC5 <- as.matrix(select_if(data, is.numeric)) %*% phi[,5]
PC6 <- as.matrix(select_if(data, is.numeric)) %*% phi[,6]
PC7 <- as.matrix(select_if(data, is.numeric)) %*% phi[,7]
PC8 <- as.matrix(select_if(data, is.numeric)) %*% phi[,8]
PC9 <- as.matrix(select_if(data, is.numeric)) %*% phi[,9]
PC10 <- as.matrix(select_if(data, is.numeric)) %*% phi[,10]

# Create data frame with Principal Components scores
PC <- tibble(
  country = data$country,
  PC1 = PC1[,1],
  PC2 = PC2[,1],
  PC3 = PC3[,1],
  PC4 = PC4[,1],
  PC5 = PC5[,1],
  PC6 = PC6[,1],
  PC7 = PC7[,1],
  PC8 = PC8[,1],
  PC9 = PC9[,1],
  PC10 = PC10[,1])

# Leverage function
pca_full <- prcomp(data[-1],
                   scale=TRUE)
```

```r
df <- data.frame(summary(pca_full)$importance)
df[2,1:10]
```

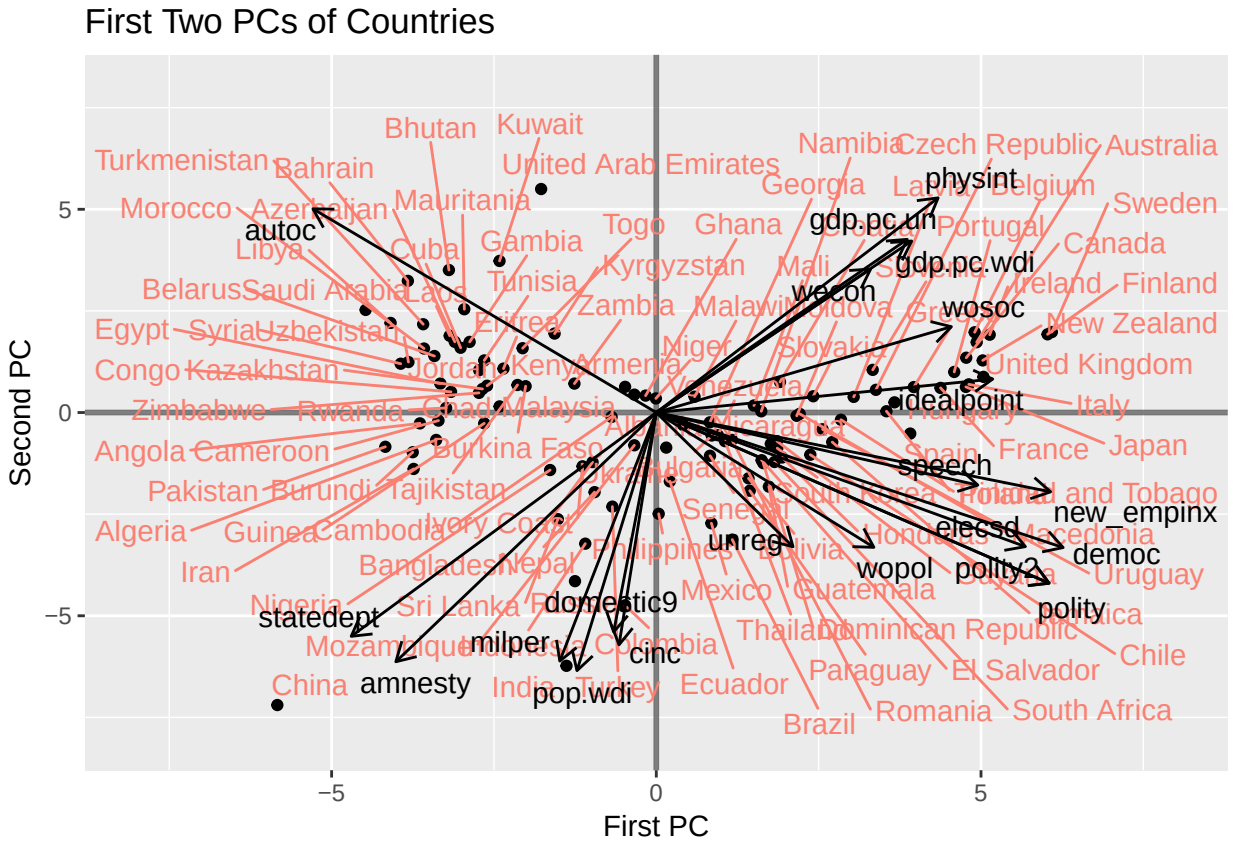
```
##                             PC1     PC2     PC3     PC4     PC5     PC6
## Proportion of Variance 0.40528 0.16475 0.12869 0.05837 0.05516 0.03968
##                             PC7     PC8     PC9    PC10
## Proportion of Variance 0.02911 0.02534 0.01976 0.01641
```

These values (proportion of explained variance) describe how much of the variance in the data is captured by the principal components (ie. the square of the component loadings). In other words, these values tell you the extent to which each individual principal component (which is an outcome built from combinations of the observed items) accounts for true differences in the dataset. For instance, the value of 0.41 for PC1 indicates that the first principal component captures over 40% of the total variation in a dataset for which we originally had 21 different variables. To compute the cumulative portion of the variance explained by the first 10 PCs, the individual portions can be summed. Therefore, in substantive terms, the first 3 components together explain nearly 70% of the total variation in the dataset, while the first 10 components together explain nearly 94% of the total variation and likely characterize these data, making the use of 21 variables unnecessary in many analyses since similar results can likely be achieved using only these 10 or even just 3.

Question 3: Present a biplot of the PCA fit from the previous question. Describe what you see (eg. which countries are clustered together? Which input features are doing the bulk of the explaining? How do you know this?)

```r
phi_df <- (phi * 20) %>%
          as.data.frame %>%
          rownames_to_column(var = "variable")
phi_df$variable <- colnames(data[-1])

ggplot(PC, aes(PC1, PC2)) +
       geom_vline(xintercept = 0, size = 1, color = "black", alpha = .5) +
       geom_hline(yintercept = 0, size = 1, color = "black", alpha = .5) +
       geom_point() +
       xlim(-8, 8) +
       ylim(-8, 8) +
       ggrepel::geom_text_repel(aes(label = country), color="salmon") +
       geom_segment(data = phi_df,
                    aes(x = 0, y = 0,
                        xend = PC1,
                        yend = PC2),
                    arrow = arrow(length = unit(0.03, "npc")),
                    color = "black") +
       ggrepel::geom_text_repel(data = phi_df,
                                aes(x = PC1, y = PC2, label = variable),
                                color = "black") +
       labs(title = "First Two PCs of Countries",
            x = "First PC",
            y = "Second PC")
```



This biplot isolates the relationship between countries and input features with the first two principal components (PC1 and PC2). Specifically, a country that falls far from 0 on the x-axis is highly associated with PC1 (high loading on PC1), and a country that falls far from the 0 on the y-axis is highly associated with PC2 (high loading on PC2). According to this plot, There are two noticable clusters of countries. The first cluster is located in the direction of the 1st PC (negative values on the x-axis) and includes countries such as Cuba, Tunisia, Laos, Syria, Libya, Morocco, and Uzbekistan. Since "autoc" closely aligns with this first cluster, a potential commonality of these countries may be autocratic governance or tendancies. The bulk of the 1st PC is explained by a combination of several input variables, which are pointing towards high positive and negative values on the x-axis. These include "autoc", "statedept", "speech", and "elecsd".

There is also a second cluster located closer to the center of the plot, which includes countries like Honduras, Paraguay, Dominican Republic, South Africa, and Hondoras. A potential commonality of these countries may be, for instance, that they have limits on free speech or high levels of electoral self-determination. It appears that the input features "autoc", "physint", "statedept", "domestic", and "cinc" are doing a bulk of the explaining of the second PC, since they are the variables that point towards high magnitude (positive and negative) values on the y-axis. It is interesting to note features that explain a bulk of both PC1 and PC2, which include "autoc" and "statedept", since they are distant from zero on both axes. Potentially less explantory input features (among the top 10 most explanatory) include "unreg" and "wopol", because they fall closer to 0 on both axes than the other input variables. It is interesting to note that a few countries fall very far from the clusters, including China, UAE, and Finland. Overall, the angles between the input vectors tell us how characteristics correlate with one another. When two vectors form a small angle, they are positively correlated, when they form a right angle, they are not likely to be correlated, and then they form a large angle, they are negatively correlated.

Bonus Question: 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)?

```
pca_sparse <- spca(data[-1],
                   verbose=FALSE)
loadings <- pca_sparse$loadings[,1:3]
colnames(loadings) <- c("PC1", "PC2", "PC3")
rownames(loadings) <- colnames(data[-1])
round(loadings, digits=2)
```

```
##              PC1   PC2   PC3
## idealpoint  0.29  0.02 -0.04
## polity      0.36 -0.20  0.08
## polity2     0.36 -0.20  0.08
## democ       0.36 -0.16  0.05
## autoc      -0.31  0.23 -0.11
## unreg       0.06 -0.13  0.04
## physint     0.25  0.32 -0.15
## speech      0.23 -0.05  0.01
## new_empinx  0.23 -0.04  0.10
## wecon       0.07  0.24 -0.14
## wopol       0.11 -0.09  0.03
## wosoc       0.17  0.17 -0.12
## elecsd      0.26 -0.14  0.02
## gdp.pc.wdi  0.16  0.28 -0.24
## gdp.pc.un   0.15  0.28 -0.24
## pop.wdi    -0.01 -0.23 -0.49
## amnesty    -0.11 -0.29  0.06
## statedept  -0.26 -0.25  0.09
## milper     -0.04 -0.30 -0.50
## cinc       -0.03 -0.25 -0.52
## domestic9  -0.04 -0.31  0.11
```

```
pca_sparse$sdev[1:5]
```

```
## [1] 2.916711 1.858990 1.642841 1.105568 1.074855
```

A sparse PCA is an algorithm for error reconstruction when there is a lot of sparsity and/or missing values. It uses LASSO (on a reformulated regression version of the optimization problem) to estimate new components with sparse loadings better. This approach provides better interpretability for the principal components in a high-dimensional dataset, since the PCs are formed as linear combinations of only a few of the original variables. Inspecting the loadings, is it evident that the values are generally very low, with a majority of the variation in PC1 explained by four variables (with values over a cutoff of +/-0.3), and a majority of the variation in PC2 explaind by only two variables. While many loadings are very close to 0, they are still included, indicating that while each PC is technically a linear combination of all input variables, several can be removed for interpretation. In this case, there are several variables that don't strongly load onto any of the first 3 PCs, including "unreg", "wopol", and "wosoc". Because the first PC has a relatively high standard deviation, however, we may want to focus attention on the second two or refine the model.

```r
pca_prob <- ppca.metabol(data[-1], printout=FALSE)
prob_loadings <- pca_prob$loadings
colnames(prob_loadings) <- c("PC1", "PC2")
prob_loadings
```

```
##                    PC1         PC2
## idealpoint -0.65234607 -0.06975498
## polity     -0.76161625  0.35561473
## polity2    -0.76161625  0.35561473
## democ      -0.78885734  0.28136100
## autoc       0.66577546 -0.42429635
## unreg      -0.26510011  0.27963958
## physint    -0.54648423 -0.44717293
## speech     -0.62339323  0.15122556
## new_empinx -0.76457193  0.16525332
## wecon      -0.41553010 -0.29888026
## wopol      -0.42186789  0.28083930
## wosoc      -0.57222721 -0.17879138
## elecsd     -0.71573043  0.27934604
## gdp.pc.wdi -0.49517013 -0.35728794
## gdp.pc.un  -0.48660723 -0.36082674
## pop.wdi     0.15380925  0.53777072
## amnesty     0.50503631  0.51921659
## statedept   0.59157162  0.46625722
## milper      0.18710891  0.51785267
## cinc        0.07223988  0.48399445
## domestic9   0.08268996  0.45701719
```

```r
pca_prob$BIC
```

```
## [1] -5683.526 -5407.333
```

Probabilistic PCA treats the optimization problem of mapping/representing the principal subspace as a latent variable model, and therefore uses the maximum-likelihood estimator and expectation-maximization to derive parameter estimates to probabilistically map the space. It does this rather than simply maximizing feature variance, while minimizing observation distances. This method is also useful because it can be used on incomplete data, and is not heavily skewed by outliers. However, it is based on the assumption that the latent variable as well as the noise are normally distributed. Finally, similar to sparse PCA, it captures dominant correlations with only a few parameters. In this case, the there are only two PCs in the optimal model, selected by the BIC. It is interesting to note that these loadings are largely negative, and have much higher values than the sparse PCA model. In the sparse PCA model, the first five input variables explained a majority of the variation in the first PC, whereas there is a more level distribution between a variety of variables in the probabilistic pca model. Furthermore, in this model, very few loadings are close to zero, as compared to most variables in sparse model.

Ultimately, these distinctions are open to interpretation and should be viewed by a domain expert in order to determine which is most appropriate for the data. In this case, because there is no missing data and the dataset is relatively small, I would not necessarily gravitate to sparse PCA. The primary distinction between the two is that the sparse model provides a clear set of components, which are strictly linear combinations of the input variables, while the probabilistic model assumes latency in the data and converges on only two principal components with a slightly less interpretable output.