ps4 becky lau

FA Q1 EFA is concerned about how many latent dimensions are needed to acount for covariation among the indicators. It is atheoretical, and makes use of factor loadings to infer the factor structure. CFA, on the other hand, is used to determine if the number of factors and input feature loadings conform to what is expected based on established theory.

FA Q2

# load libraries  
library(tidyverse)

## -- Attaching packages ------------------------------------------------------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1 v purrr 0.3.2  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 1.0.0 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## -- Conflicts ---------------------------------------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(ggfortify)  
library(lattice) # for X,Y plot  
library(psych) # for fa function

##   
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

library(GPArotation) # for some rotation options  
library(ggplot2) # for quick plot ("qplot")  
library(readr)  
countries <- read\_csv("countries.csv")

## Warning: Missing column names filled in: 'X1' [1]

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## X1 = col\_character()  
## )

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

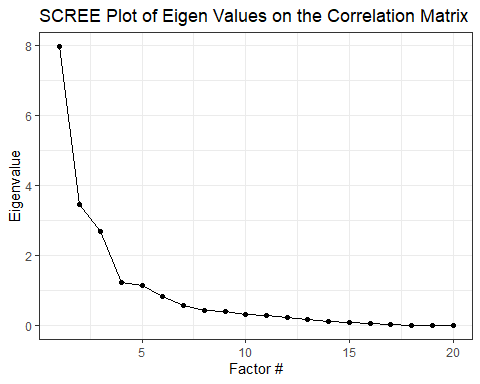
df <- countries %>% select(-X1) %>% scale()  
  
# First, (view, then) store the correlation matrix  
round(cor(df[,-1]), 4) # drop first col and calculate all correlations

## polity polity2 democ autoc unreg physint speech  
## polity 1.0000 1.0000 0.9738 -0.9550 0.3512 0.3214 0.6490  
## polity2 1.0000 1.0000 0.9738 -0.9550 0.3512 0.3214 0.6490  
## democ 0.9738 0.9738 1.0000 -0.8625 0.3896 0.3912 0.6366  
## autoc -0.9550 -0.9550 -0.8625 1.0000 -0.2735 -0.2049 -0.6137  
## unreg 0.3512 0.3512 0.3896 -0.2735 1.0000 0.0841 0.2975  
## physint 0.3214 0.3214 0.3912 -0.2049 0.0841 1.0000 0.4010  
## speech 0.6490 0.6490 0.6366 -0.6137 0.2975 0.4010 1.0000  
## new\_empinx 0.8343 0.8343 0.8283 -0.7760 0.3416 0.4829 0.7831  
## wecon 0.2617 0.2617 0.3356 -0.1447 -0.0227 0.4111 0.1569  
## wopol 0.5092 0.5092 0.4684 -0.5221 0.0681 0.1448 0.2901  
## wosoc 0.4288 0.4288 0.5108 -0.2879 0.1578 0.5112 0.3378  
## elecsd 0.8475 0.8475 0.8433 -0.7858 0.3489 0.3360 0.6903  
## gdp.pc.wdi 0.2930 0.2930 0.4007 -0.1293 0.0254 0.5116 0.3082  
## gdp.pc.un 0.2827 0.2827 0.3909 -0.1193 0.0247 0.5093 0.3021  
## pop.wdi -0.0174 -0.0174 -0.0011 0.0373 0.0061 -0.2275 -0.0969  
## amnesty -0.3008 -0.3008 -0.3664 0.1913 -0.0421 -0.6545 -0.2848  
## statedept -0.3911 -0.3911 -0.4724 0.2541 -0.0196 -0.7969 -0.3770  
## milper -0.0665 -0.0665 -0.0458 0.0882 -0.0020 -0.2218 -0.1359  
## cinc 0.0177 0.0177 0.0467 0.0216 0.0170 -0.1222 -0.0460  
## domestic9 0.1039 0.1039 0.0741 -0.1345 0.2192 -0.4357 -0.0217  
## new\_empinx wecon wopol wosoc elecsd gdp.pc.wdi gdp.pc.un  
## polity 0.8343 0.2617 0.5092 0.4288 0.8475 0.2930 0.2827  
## polity2 0.8343 0.2617 0.5092 0.4288 0.8475 0.2930 0.2827  
## democ 0.8283 0.3356 0.4684 0.5108 0.8433 0.4007 0.3909  
## autoc -0.7760 -0.1447 -0.5221 -0.2879 -0.7858 -0.1293 -0.1193  
## unreg 0.3416 -0.0227 0.0681 0.1578 0.3489 0.0254 0.0247  
## physint 0.4829 0.4111 0.1448 0.5112 0.3360 0.5116 0.5093  
## speech 0.7831 0.1569 0.2901 0.3378 0.6903 0.3082 0.3021  
## new\_empinx 1.0000 0.2723 0.5110 0.4939 0.8504 0.3310 0.3216  
## wecon 0.2723 1.0000 0.3143 0.6562 0.2351 0.4719 0.4654  
## wopol 0.5110 0.3143 1.0000 0.4149 0.4371 0.0159 0.0047  
## wosoc 0.4939 0.6562 0.4149 1.0000 0.3959 0.5037 0.4925  
## elecsd 0.8504 0.2351 0.4371 0.3959 1.0000 0.2971 0.2904  
## gdp.pc.wdi 0.3310 0.4719 0.0159 0.5037 0.2971 1.0000 0.9994  
## gdp.pc.un 0.3216 0.4654 0.0047 0.4925 0.2904 0.9994 1.0000  
## pop.wdi -0.1702 -0.1245 0.0381 -0.0671 -0.0658 -0.0579 -0.0577  
## amnesty -0.3535 -0.3393 -0.0585 -0.4288 -0.3174 -0.5360 -0.5337  
## statedept -0.4872 -0.4399 -0.1007 -0.5041 -0.3879 -0.5795 -0.5747  
## milper -0.2330 -0.1732 -0.0358 -0.0915 -0.1018 -0.0330 -0.0336  
## cinc -0.1101 -0.0942 0.0197 -0.0076 0.0028 0.1314 0.1326  
## domestic9 -0.0180 -0.1108 0.0809 -0.1048 0.0395 -0.1377 -0.1374  
## pop.wdi amnesty statedept milper cinc domestic9  
## polity -0.0174 -0.3008 -0.3911 -0.0665 0.0177 0.1039  
## polity2 -0.0174 -0.3008 -0.3911 -0.0665 0.0177 0.1039  
## democ -0.0011 -0.3664 -0.4724 -0.0458 0.0467 0.0741  
## autoc 0.0373 0.1913 0.2541 0.0882 0.0216 -0.1345  
## unreg 0.0061 -0.0421 -0.0196 -0.0020 0.0170 0.2192  
## physint -0.2275 -0.6545 -0.7969 -0.2218 -0.1222 -0.4357  
## speech -0.0969 -0.2848 -0.3770 -0.1359 -0.0460 -0.0217  
## new\_empinx -0.1702 -0.3535 -0.4872 -0.2330 -0.1101 -0.0180  
## wecon -0.1245 -0.3393 -0.4399 -0.1732 -0.0942 -0.1108  
## wopol 0.0381 -0.0585 -0.1007 -0.0358 0.0197 0.0809  
## wosoc -0.0671 -0.4288 -0.5041 -0.0915 -0.0076 -0.1048  
## elecsd -0.0658 -0.3174 -0.3879 -0.1018 0.0028 0.0395  
## gdp.pc.wdi -0.0579 -0.5360 -0.5795 -0.0330 0.1314 -0.1377  
## gdp.pc.un -0.0577 -0.5337 -0.5747 -0.0336 0.1326 -0.1374  
## pop.wdi 1.0000 0.3146 0.2421 0.8898 0.8961 0.0635  
## amnesty 0.3146 1.0000 0.7439 0.3511 0.2516 0.4018  
## statedept 0.2421 0.7439 1.0000 0.2455 0.1405 0.4361  
## milper 0.8898 0.3511 0.2455 1.0000 0.9399 0.0949  
## cinc 0.8961 0.2516 0.1405 0.9399 1.0000 0.0782  
## domestic9 0.0635 0.4018 0.4361 0.0949 0.0782 1.0000

dfcor <- (cor(df[,-1]))   
  
  
# Next, generate the eigenvalues  
ev <- eigen(dfcor) # store EVs on the correlation matrix  
ev$values # recall we re looking for EVs over 1.

## [1] 7.971794e+00 3.454111e+00 2.679856e+00 1.225666e+00 1.156714e+00  
## [6] 8.287256e-01 5.612043e-01 4.378709e-01 4.046079e-01 3.096707e-01  
## [11] 2.801924e-01 2.220646e-01 1.678275e-01 1.151692e-01 8.417283e-02  
## [16] 6.502640e-02 3.490669e-02 4.198263e-04 2.002490e-15 -7.313782e-16

# Next, generate Scree plot  
qplot(y = ev$values,   
 main = 'SCREE Plot of Eigen Values on the Correlation Matrix',   
 xlab = 'Factor #',   
 ylab = 'Eigenvalue') +  
 geom\_line() +   
 theme\_bw()



# Next, fit the factor analysis model with 2,3,4 factors  
factan.2 <- fa(df,   
 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

summary(factan.2) # inspect output

##   
## Factor analysis with Call: fa(r = df, 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

factan.3 <- fa(df,   
 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

summary(factan.3) # inspect output

##   
## Factor analysis with Call: fa(r = df, 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

factan.4 <- fa(df,   
 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

summary(factan.4) # inspect output

##   
## Factor analysis with Call: fa(r = df, 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

FA Q3

## ROTATION  
r.factan.rotate <- fa(df,   
 nfactors = 3,   
 rotate = "varimax") #rotation choice, orthogonal

## 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

summary(r.factan.rotate )

##   
## Factor analysis with Call: fa(r = df, nfactors = 3, rotate = "varimax")  
##   
## 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

FA Q4 The rotated factor pattern seem to fit the two factor structure better, with physint clearly loading on factor 1, and maybe also idealpoint. Nothing was close to factor 2 in the unrotated factor pattern.

## VISUALIZATION  
# Manual plotting non-rotation vs. rotation  
## Initial (unrotated) factor solution  
nonrotated.factors <- fa(cor(df),  
 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.

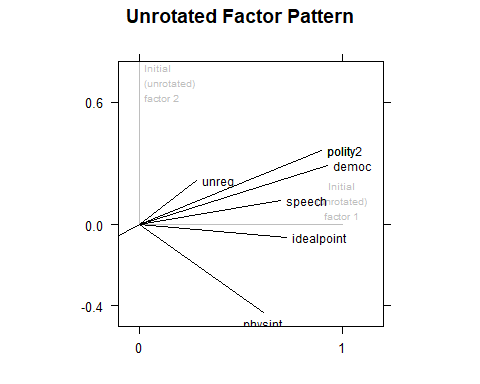
## In factor.scores, the correlation matrix is singular, an approximation is used

## 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 (xyplot from lattice package)  
nonrot.pattern <- as.data.frame(nonrotated.factors$loadings[1:8,])  
  
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)))  
)



# Orthogonal (varimax) rotated factor solution  
varimax.factors <- fa(cor(df),  
 fm = "pa",  
 nfactors = 3,  
 rotate = "varimax",  
 smc = TRUE)

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

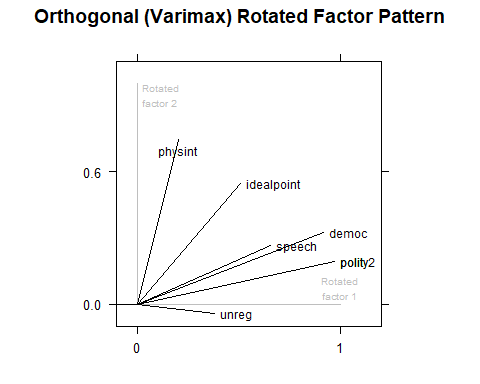
## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was  
## done

## 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

# Plot the factor pattern  
orthog.pattern <- as.data.frame(varimax.factors$loadings[1:8,])  
  
xyplot(PA2 ~ PA1, data = orthog.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(orthog.pattern)[-7],  
 pos = 4, cex = .75)  
 panel.text(x[7], y[7], labels = rownames(orthog.pattern)[7],  
 pos = 1, cex = .75)  
 },  
 main = "Orthogonal (Varimax) Rotated Factor Pattern",  
 xlab = "",  
 ylab = "",  
 scales = list(x = list(at = c(0, 1)),  
 y = list(at = c(-.4, 0, .6)))  
)



PCA Q1

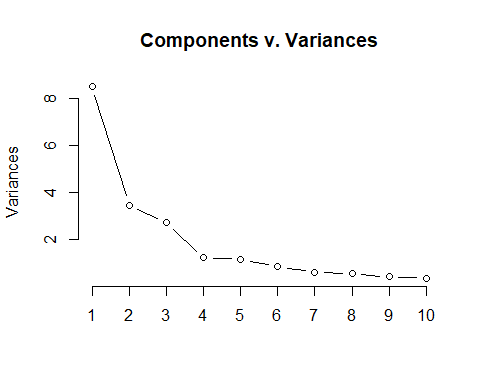
PCA and FA are both variable reduction techniques. However, there are key differences. In PCA, principal components account for maximal amount of variance of observed variables, while in FA, factors account for common variance in the data. PCA minimizes the sum of squared perpendicular distance to the component axis, while FA estimates factors which influence responses on observed variables. In PCA, component scores are a linear combination of observed variables, weighted by eigenvectors, while in FA, the observed variables ar elinear combinations of the underlying and unique factors.

PCA Q 2 It looks like the first three principal components are most effectively characterizing the data, because there is a sharp drop in variance explained after the third PC.

# when creating more components, less variance being explained.   
pca.out <- prcomp(df)  
  
summary(pca.out)

## 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 7.605e-16 2.858e-16  
## Proportion of Variance 0.00002 0.000e+00 0.000e+00  
## Cumulative Proportion 1.00000 1.000e+00 1.000e+00

# Plot PCA object as scree  
plot(pca.out,   
 type="l", # change to "b" for a barplot  
 main = "Components v. Variances")



PCA Q 3

Countries such as Libya, Saudi Arabia, and Bahrain are more autocratic. Within this category, countries such as Guatamela is on the other side of the extreme from the origin, which means they are likely comparatively less autocratic. Political Terror score are on the opposite side of the origin from variables like physical integrity, women’s economic rights, and GDP, indicating that perhaps countries with more political terror (e.g. Iran 48 and China 19) also has less physical integrity, women’s economic rights, and lower GDP comparatively, than countries like Canada (17). The lines are overlapping a lot so it is a bit hard to read, but based on the length of the lines from the origin, autocracy, polity score and physical integrity seem to be explaining a lot of the variance.

autoplot(pca.out,   
 shape = F,   
 loadings.label = T) +   
 theme\_bw()

