Assignment 4

Janelle Morano

27 April 2022

Conduct a Principal Component Analysis

```
# explore the major modes of variability in a reconstruction of the Palmer
# Drought Severity Index (PDSI) across North America, a monthly measure of the
# balance between atmospheric moisture supply and demand and takes into account
# regional climatology and short-term hydrologic persistence.  Data: The Living
# Blended Drought Atlas (LBDA), a paleoclimate reconstruction of average summer
# (June-August) PDSI over North America based on 1845 tree ring chronologies
# spatially complete over North America from 1473 to 2005 columns are locations
# rows are time values are the PDSI values


# a) Load LBDA.RData and lon.lat.RData
load("/Users/janellemorano/Git/Reference-R-scripts/Envtl-Multivariate-Stats/data/LBDA.RData")
load("/Users/janellemorano/Git/Reference-R-scripts/Envtl-Multivariate-Stats/data/lon.lat.RData")
# ls()

# 1. Conduct a Principal Component Analysis b) Conduct a PCA on the covariance
# matrix (prcomp()) on the LBDA matrix (all values at all locations and times)
pca <- prcomp(LBDA, center = TRUE, scale = TRUE)

# summary(pca) #ids the PCs and their stdev

# Remember for PCA...U = XW X = the data (LBDA matrix) W = weights; columns are
# EOFs/eigenvectors/principle axes; elements are loadings U = columns are
# PCAs/transformed X values; elements are scores total variance (sum of
# diagonals of cov matrix) is the sum of the eigenvalues

# names(pca) pca$x #U matrix pca$rotation #W matrix pca$sdev #sqrt of
# eigenvalues/lambdas of covariance matrix; sdev^2 are the lambdas variance
# explained by the PC = sdev^2/ sum(sdev^2)
```

c) Calculate and report the variance explained by each of the first 10 EOFs

```
var_explained <- pca$sdev^2/sum(pca$sdev^2)
print(var_explained[1:10])
```

```
##  [1] 0.16959594 0.12073789 0.10540613 0.06773072 0.04671775 0.03903215
##  [7] 0.03581427 0.03140932 0.02721475 0.02388850
```

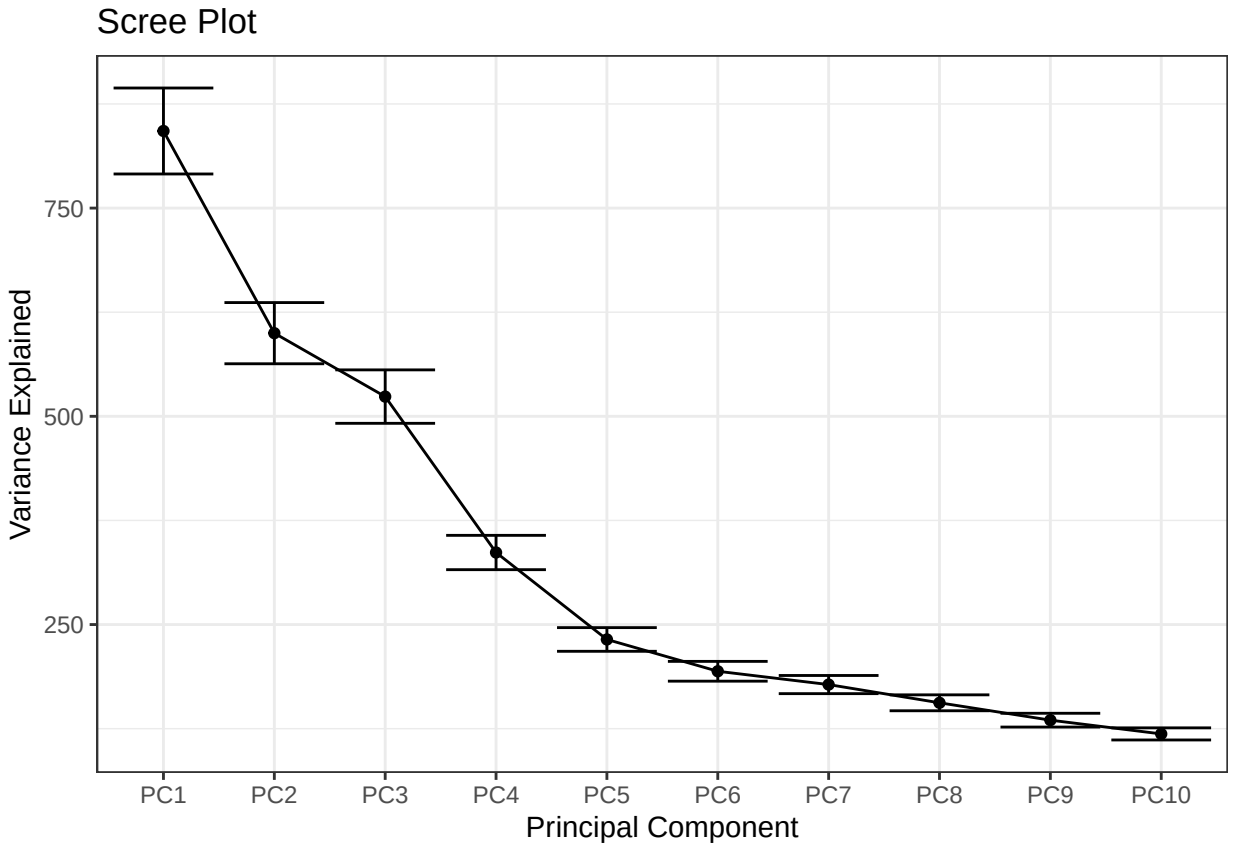
d) Scree plot and +/- 1 standard error for the first 10 eigenvalues

```
# scree plot is the variance explained or proportion of variation by each PC
# Put eigenvalues into df
```

```r
eigen_df <- data.frame(PC= paste0("PC",1:533), #there are 533 PCs
                       EOF=pca$sdev^2)
# print(eigen_df[1:10,])

# add SE
# SE = lambda(i) * sqrt(2/n)
SE <- pca$sdev^2 * sqrt(2/length(pca$sdev))
# add to df
eigen_df$SE <- SE
head(eigen_df)
```

```
##    PC      EOF       SE
## 1 PC1 842.5526 51.61173
## 2 PC2 599.8258 36.74317
## 3 PC3 523.6577 32.07738
## 4 PC4 336.4862 20.61193
## 5 PC5 232.0938 14.21723
## 6 PC6 193.9117 11.87833
```

```r
# Plot
library(ggplot2)
library(dplyr)
# pca$sdev by cols of U pca$x OR var_explained_df$var_explained by var_explained_df$PC
scree <- eigen_df[1:10,]
# order PCs
scree <- scree %>%
  arrange(desc(EOF)) %>%
  mutate(PC=factor(PC, levels=PC))

ggplot(data = scree, aes(x = PC, y = EOF, group = 1)) +
  geom_line() +
  geom_point() +
  geom_errorbar(aes(ymin=EOF-SE, ymax=EOF+SE)) +
  xlab("Principal Component") +
  ylab("Variance Explained") +
  ggtitle("Scree Plot") +
  theme_bw()
```

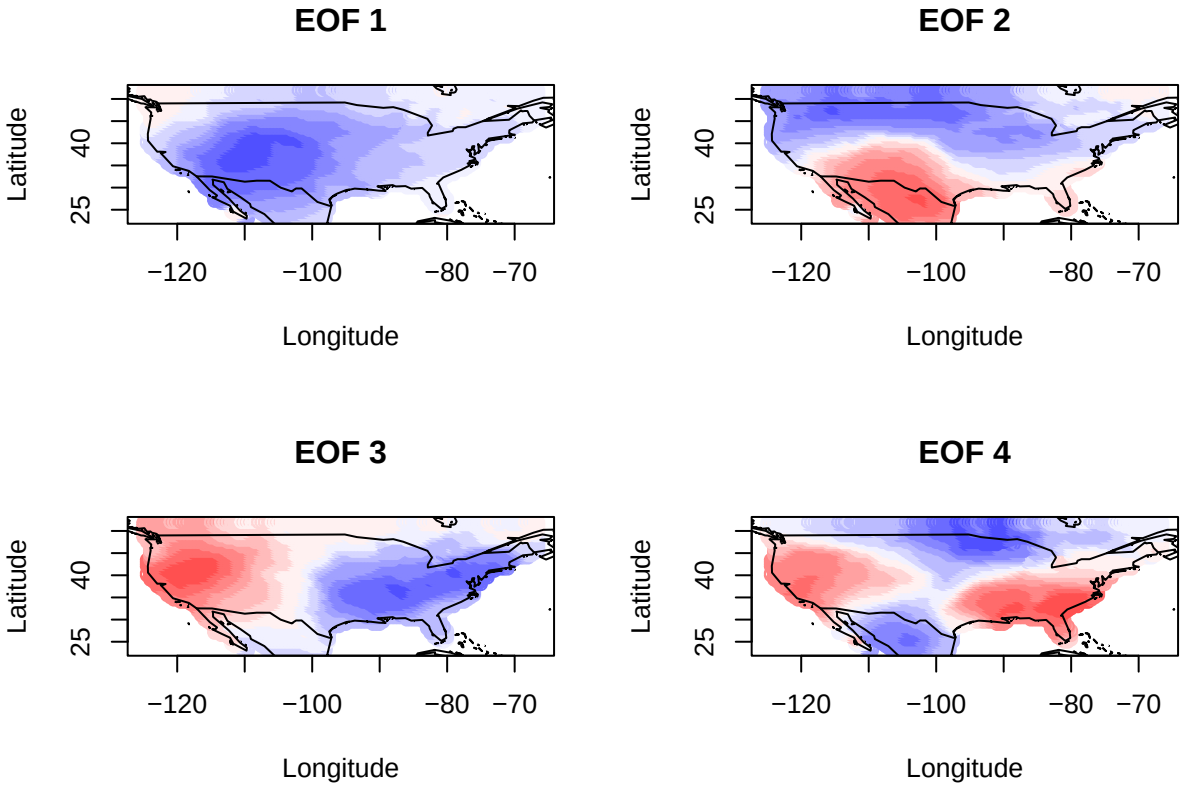


North's Rule is to make a scree plot and look for where the SE of eigenvalues are not overlapping, because when the SEs are overlapping, they are not different from each other and the uncertainty between the values is too great and they should be ignored. Looking at this graph, PCs 6-10 are not different from each other. PC5 is barely different from PC6. PCs1-4 are more different from the other PCs, so keep the first 4.

e) Plot EOF (elements of which are loadings) patterns on map

```
# loadings = weights/EOFs/W matrix Retain the 4 EOFs from pca$rotation into
# cur_EOF
cur_EOF <- pca$rotation[, 1:4]
# Now have the first 4 EOFs for each row (location) of data

# make vector of color breaks
mycol_list <- list()
for (i in 1:4) {
    mybreaks <- c(seq(-0.04, -0.001, by = 0.004), seq(0.001, 0.04, by = 0.004))
    mycut <- cut(cur_EOF[, i], breaks = mybreaks, label = FALSE)
    mycolpal <- colorRampPalette(c("red", "white", "blue"))
    mycol <- mycolpal(length(mybreaks))[mycut]
    mycol_list[[i]] <- mycol
}

# f) Plot each EOF on map In a multi-panel figure, plot each EOF using the
# plot() function, the information in lon.lat, and the vector of colors in
# mycol. Add a map of countries by adding map('world',add=T)
library(maps)
par(mfrow = c(2, 2))
plot(x = lon.lat$lon, y = lon.lat$lat, col = mycol_list[[1]], xlab = "Longitude",
```

```
    ylab = "Latitude", main = "EOF 1")
map("world", add = T)

plot(x = lon.lat$lon, y = lon.lat$lat, col = mycol_list[[2]], xlab = "Longitude",
    ylab = "Latitude", main = "EOF 2")
map("world", add = T)

plot(x = lon.lat$lon, y = lon.lat$lat, col = mycol_list[[3]], xlab = "Longitude",
    ylab = "Latitude", main = "EOF 3")
map("world", add = T)

plot(x = lon.lat$lon, y = lon.lat$lat, col = mycol_list[[4]], xlab = "Longitude",
    ylab = "Latitude", main = "EOF 4")
map("world", add = T)
```




```
par(mfrow = c(1, 1))
```

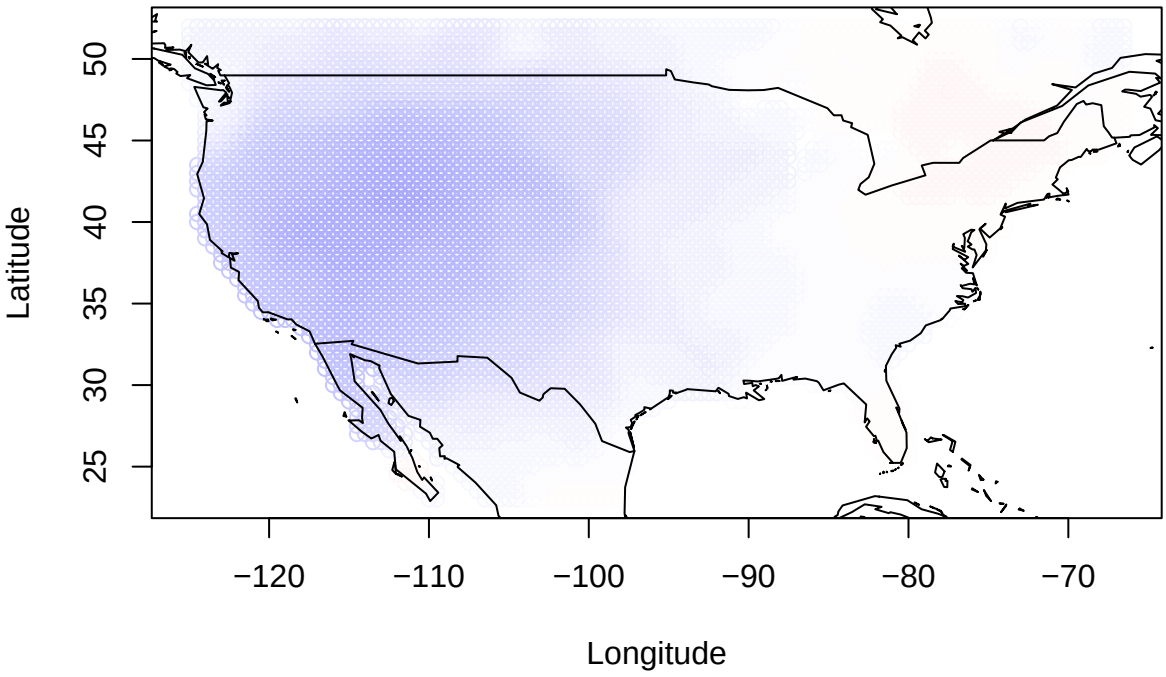
2) Create a Parsimonious Forecast Model

```
# g) forecast PDSI in next year for each EOF

# Save the forecast predictions in a vector
PC.forecast <- c()

for (i in 1:4) {
    # select PCAs (U matrix)
    PC <- pca$x[, i]
```

```
    # g) fit a simple linear regression model between lead and lag versions of
    # given PC x = t-1
    x <- PC[1:(length(PC) - 1)]
    # y: t
    y <- PC[2:length(PC)]
    # put x & y values in df
    df <- data.frame(x, y)
    # fit model
    PC.lm <- lm(y ~ x, data = df)

    # h) forecast for each of the retained PCs in 2005 (the last PC value) for
    # the time step n+1 (=2006)
    forecast <- predict(PC.lm, newdata = data.frame(x = PC[length(PC)]))

    # # Export the predictions for 2006
    PC.forecast[i] <- forecast
}

# See predictions
PC.forecast
```

```
## [1]  9.127986  1.068795 -4.495005 -1.723320
```

```
# i) Using the synthesis equation and the vector of 1-step ahead forecasts,
# develop a 1-step ahead forecast for the PDSI for all of the grid cells.

# Synthesis equation = X = U %*% t(W) X = output U = forecast t(W) = transpose
# of original loadings (ie pca$rotation)
forecast.PDSI <- PC.forecast %*% t(pca$rotation[, 1:4])

# j) Plot forecasts on a map make vector of color breaks of forecast.PDSI, from
# -1 to 1
mybreaks <- mybreaks <- c(seq(-1, -0.01, by = 0.02), seq(0.01, 1, by = 0.02))
mycut <- cut(forecast.PDSI, breaks = mybreaks, label = FALSE)  # something is wrong is going on here
mycolpal <- colorRampPalette(c("red", "white", "blue"))
mycol <- mycolpal(length(mybreaks))[mycut]

# Now plot it
plot(x = lon.lat$lon, y = lon.lat$lat, col = mycol, xlab = "Longitude", ylab = "Latitude",
    main = "Forecast of 2006 PDSI")
map("world", add = T)
```



Rotate the EOFs and Derive Rotated Principal Components

```
# k-m) Rotate EOFs (use 1st 4 EOFs/weights/W matrix of original pca$rotation)
my.varimax <- varimax(pca$rotation[, 1:4])

# Find rotated PCs: multiply the P LBDA by REOFs (reofs) W* = W %* %T new set
# of rotated EOFs = original EOFs %*% rotation matrix
rotated.PCs <- pca$rotation[, 1:4] %*% my.varimax$rotmat

# Rotated PCs
cor(rotated.PCs)
```

```
##            [,1]       [,2]       [,3]       [,4]
## [1,]  1.0000000 -0.2168429 -0.2596567 -0.2528773
## [2,] -0.2168429  1.0000000 -0.2506434 -0.2440993
## [3,] -0.2596567 -0.2506434  1.0000000 -0.2922945
## [4,] -0.2528773 -0.2440993 -0.2922945  1.0000000
```

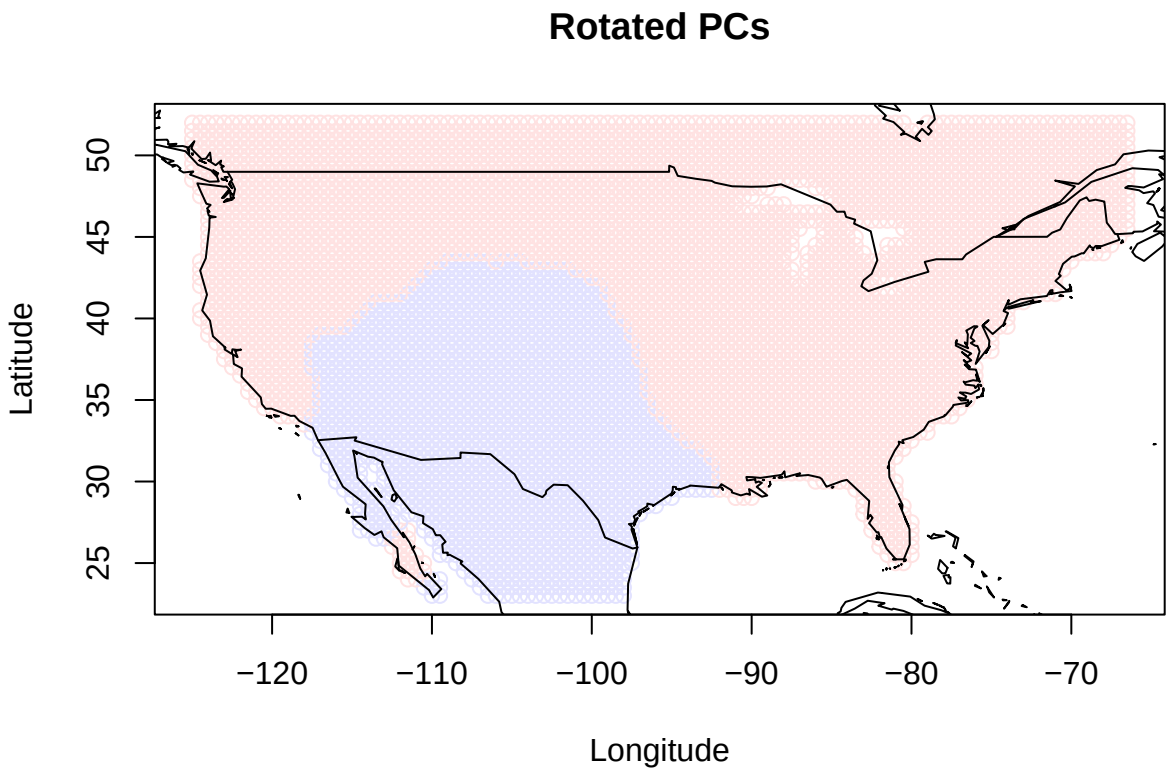
```
# original PCs
cor(pca$x[, 1:4])
```

```
##                PC1           PC2           PC3           PC4
## PC1  1.000000e+00 -6.665943e-16 -5.711740e-16 1.404113e-16
## PC2 -6.665943e-16  1.000000e+00  3.345416e-17 4.597398e-16
## PC3 -5.711740e-16  3.345416e-17  1.000000e+00 4.333023e-16
## PC4  1.404113e-16  4.597398e-16  4.333023e-16 1.000000e+00
```

m) Report, compare, and discuss the correlation matrix of the rotated PCs with the correlation matrix of the original PC's. What is the largest difference in the correlation structure of the original PCs and the rotated PCs?

The rotated EOFs are now no longer uncorrelated and are now following the same correlation pattern (all negative values). The original PCs showed that they were uncorrelated and not following the same pattern across PCs (negative and positive values).

```
# Plot each rotated EOF on a map.  make vector of color breaks of rotated.PC,
# from -1 to 1
mybreaks <- mybreaks <- c(seq(-1, -0.01, by = 0.2), seq(0.01, 1, by = 0.2))
mycut <- cut(rotated.PCs, breaks = mybreaks, label = FALSE)  # something is wrong is going on here
mycolpal <- colorRampPalette(c("red", "white", "blue"))
mycol <- mycolpal(length(mybreaks))[mycut]

# Now plot it
plot(x = lon.lat$lon, y = lon.lat$lat, col = mycol, xlab = "Longitude", ylab = "Latitude",
    main = "Rotated PCs")
map("world", add = T)
```



n) Discuss the differences between these plots and those for the original EOFs.

In the original EOFs, there are 4 different patterns, and most of them show a dipole response, where one area is opposite the other (i.e. one is wet while the other is dry and vice versa), EOF1 is showing that the west-southwest region is most similar, in that when it is wet OR dry, the area follows the same, bipolar pattern. EOF2 is showing that the northern region is more similar to each other and follows an opposite pattern than the southwest (and Florida up through the Carolinas). So when the north is wet, the southwest is dry and vice versa. EOF3 is showing the west and east are opposite each other, and EOF4 is showing the west and east are similiar to each other and different from the north and south.

The orthogonal EOF is showing that a dipolar pattern is dominant, where the southwest is opposite of the rest of the country (i.e. wet when the other is dry, and vice versa).