Day2 exercise solutions

Ali Movasati

Sept. 23rd, 2024

```
# Set global code chunk options
knitr::opts_chunk$set(warning = FALSE)
```

```
# load required libraries
library(ggplot2)
library(magrittr)
library(dplyr)
library(tibble)
library(maps)
library(fields)


# define functions
`%notin%` <- Negate(`%in%`)
```

Problem 1

```
# load the data and inspect it

protein <- read.table("/Users/alimos313/Documents/studies/phd/university/courses/stat-modelling/day2/da

str(protein)
```

```
## 'data.frame':    25 obs. of  10 variables:
##  $ Country  : chr  "Albania" "Austria" "Belgium" "Bulgaria" ...
##  $ RedMeat  : num  10.1 8.9 13.5 7.8 9.7 10.6 8.4 9.5 18 10.2 ...
##  $ WhiteMeat: num  1.4 14 9.3 6 11.4 10.8 11.6 4.9 9.9 3 ...
##  $ Eggs     : num  0.5 4.3 4.1 1.6 2.8 3.7 3.7 2.7 3.3 2.8 ...
##  $ Milk     : num  8.9 19.9 17.5 8.3 12.5 25 11.1 33.7 19.5 17.6 ...
##  $ Fish     : num  0.2 2.1 4.5 1.2 2 9.9 5.4 5.8 5.7 5.9 ...
##  $ Cereals  : num  42.3 28 26.6 56.7 34.3 21.9 24.6 26.3 28.1 41.7 ...
##  $ Starch   : num  0.6 3.6 5.7 1.1 5 4.8 6.5 5.1 4.8 2.2 ...
##  $ Nuts     : num  5.5 1.3 2.1 3.7 1.1 0.7 0.8 1 2.4 7.8 ...
##  $ Fr.Veg   : num  1.7 4.3 4 4.2 4 2.4 3.6 1.4 6.5 6.5 ...
```

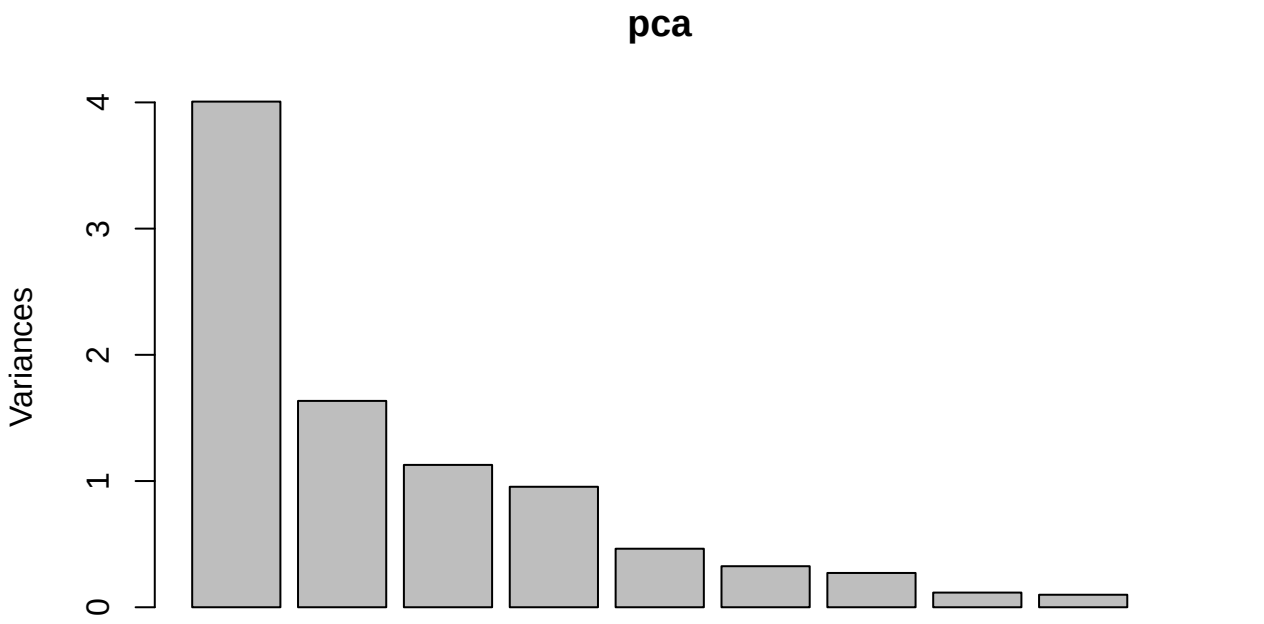
1.A)

```
# perfrom PCA and visualize the results

## prepare the dataset
protein %<>% column_to_rownames(var = "Country") %>%
```

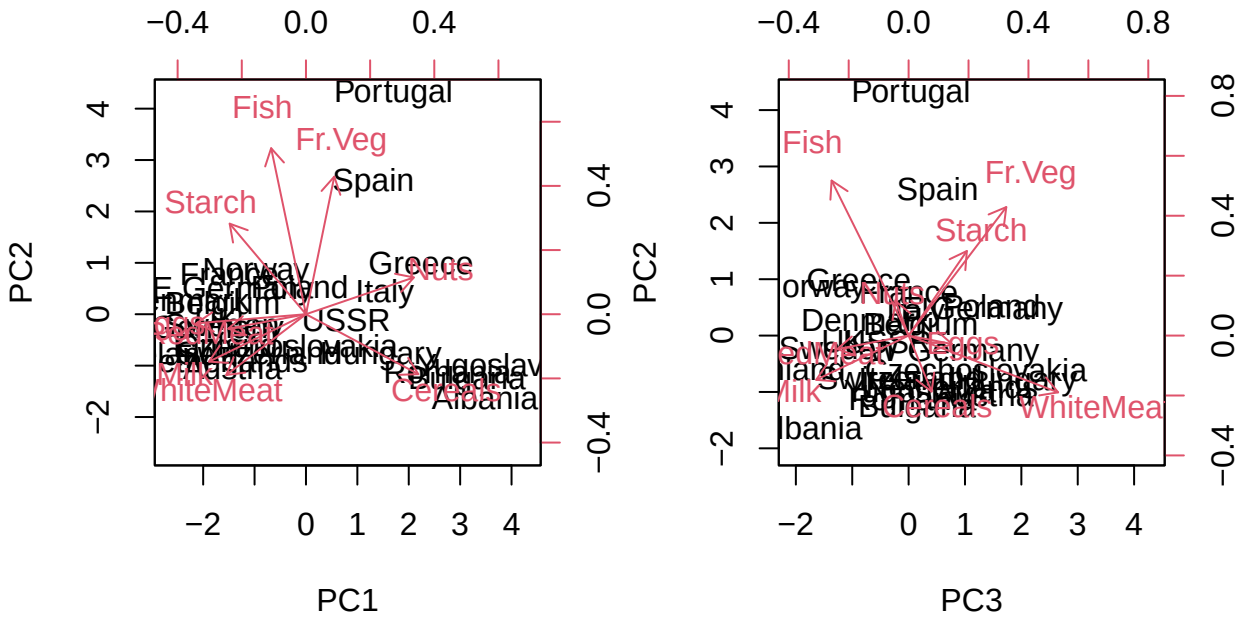
```
        as.matrix()


## perform PCA
pca <- prcomp(protein, scale=TRUE)


str(pca, give.attr=FALSE)
```

```
## List of 5
##  $ sdev    : num [1:9] 2.002 1.279 1.062 0.977 0.681 ...
##  $ rotation: num [1:9, 1:9] -0.303 -0.311 -0.427 -0.378 -0.136 ...
##  $ center  : Named num [1:9] 9.83 7.9 2.94 17.11 4.28 ...
##  $ scale   : Named num [1:9] 3.35 3.69 1.12 7.11 3.4 ...
##  $ x       : num [1:25, 1:9] 3.49 -1.42 -1.62 3.13 -0.37 ...
```

```
## plot the PCs and the amount of variation how much of them can explain in the data
plot(pca)
```




```
## How much variance is explained by the 1st, the 1st and 2nd, the 1st, 2nd and 3rd PCs.
# min(which(round(cumsum(pca$sdev^2)/sum( pca$sdev^2), 3)>= 0.75))


## plot biplot
par(mfcol=c(1,2))
biplot(pca, scale = 0)

## to plot other principal components, use, for example:
biplot(pca, scale = 0, choices=c(3, 2))
```



« **Comments** »

- We performed PCA to reduce the dimensionality of the data. Two first principal components (PCs) explain 0.627 of variance in the data. When checking the loadings from the rotation matrix of the PCA object and visualize them along the PCA scores in a biplot, we can appreciate that the first PC is mainly influenced by Cereals and Nuts, while the second PC is influenced by Fish and vegetables. We can also see a nice clustering of the countries with similar socio-economic profiles together.

1.B)

« **Comments** »

- We could identify 9 independent principal components (PCs) that is the same number of our independent variables (features)
- Each PC represents an axis in the transformed space of variables (features). They are found in a way to maximize explaining the variation in the data and are ordered based on a descending order (as can be seen in the screeplot)

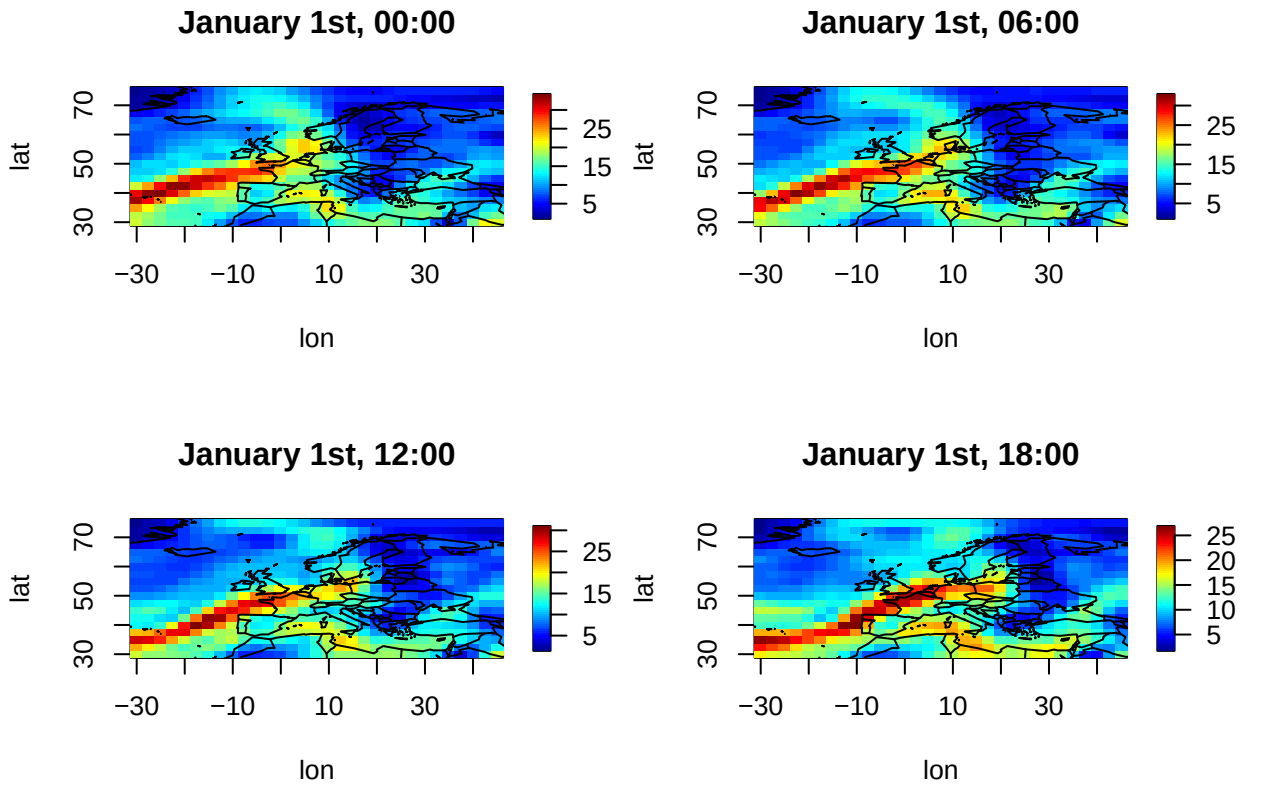
Problem 2

```
# load the data
loaded_obj <- load("/Users/alimos313/Documents/studies/phd/university/courses/stat-modelling/day2/data/p
```

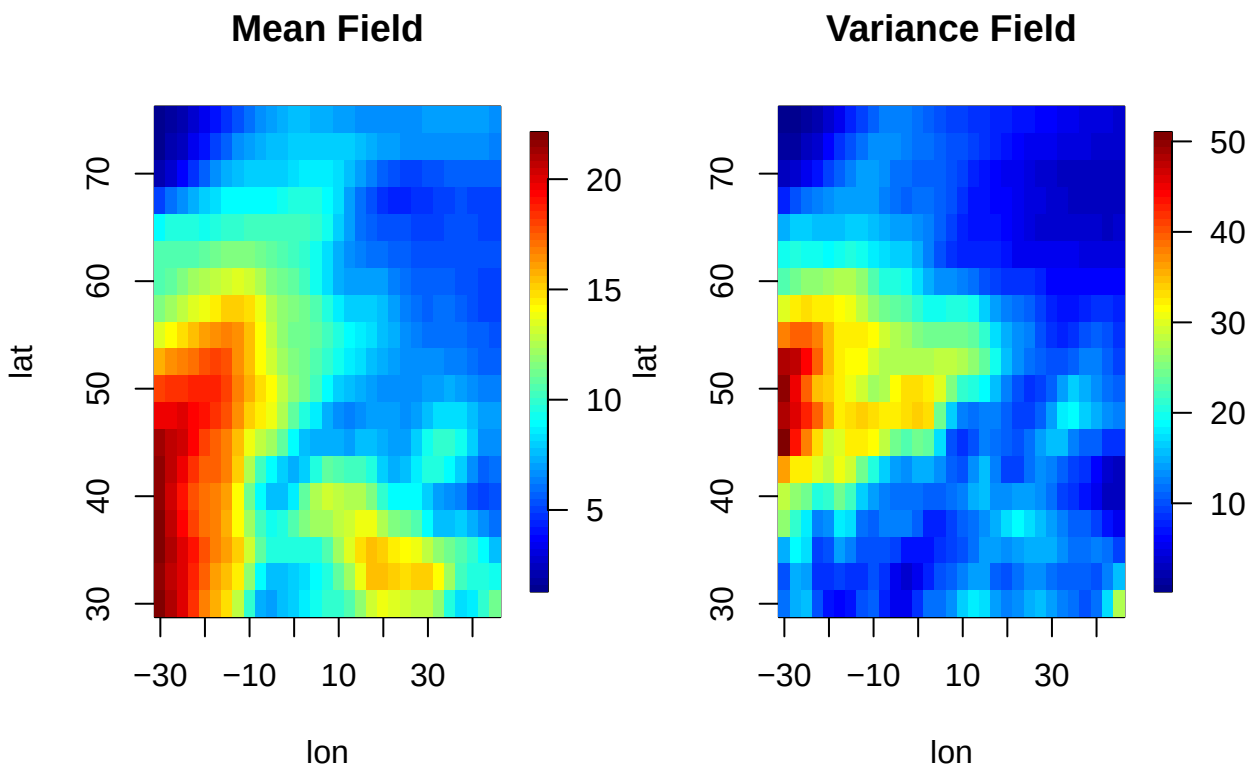
2.A)

```
par(mfcol=c(2,2))

times <- c("00:00", "06:00", "12:00", "18:00")

for (i in 1:4){
    time <- times[i]
    image.plot(lon, lat, pre[, , i], main = paste0("January 1st, ", time))
```

```
    map("world", add = T)
}
```

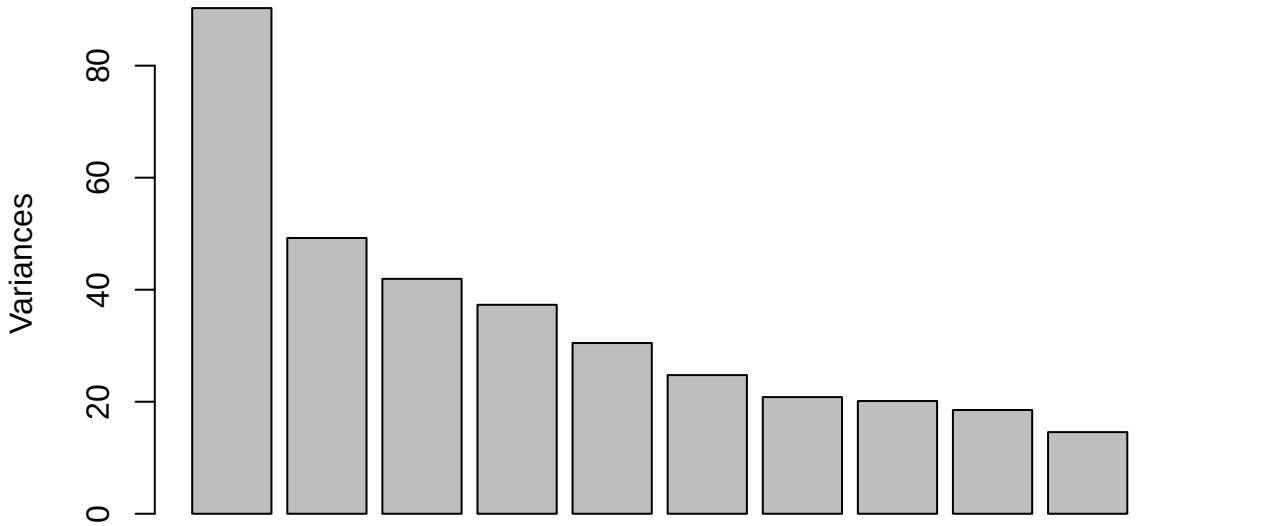



```
par(mfcol=c(1,2))

mean_values <- apply(pre, c(1, 2), mean)

image.plot(lon, lat, mean_values, main = paste0("Mean Field"))


variance_values <- apply(pre, c(1, 2), var)

image.plot(lon, lat, variance_values, main = paste0("Variance Field"))
```

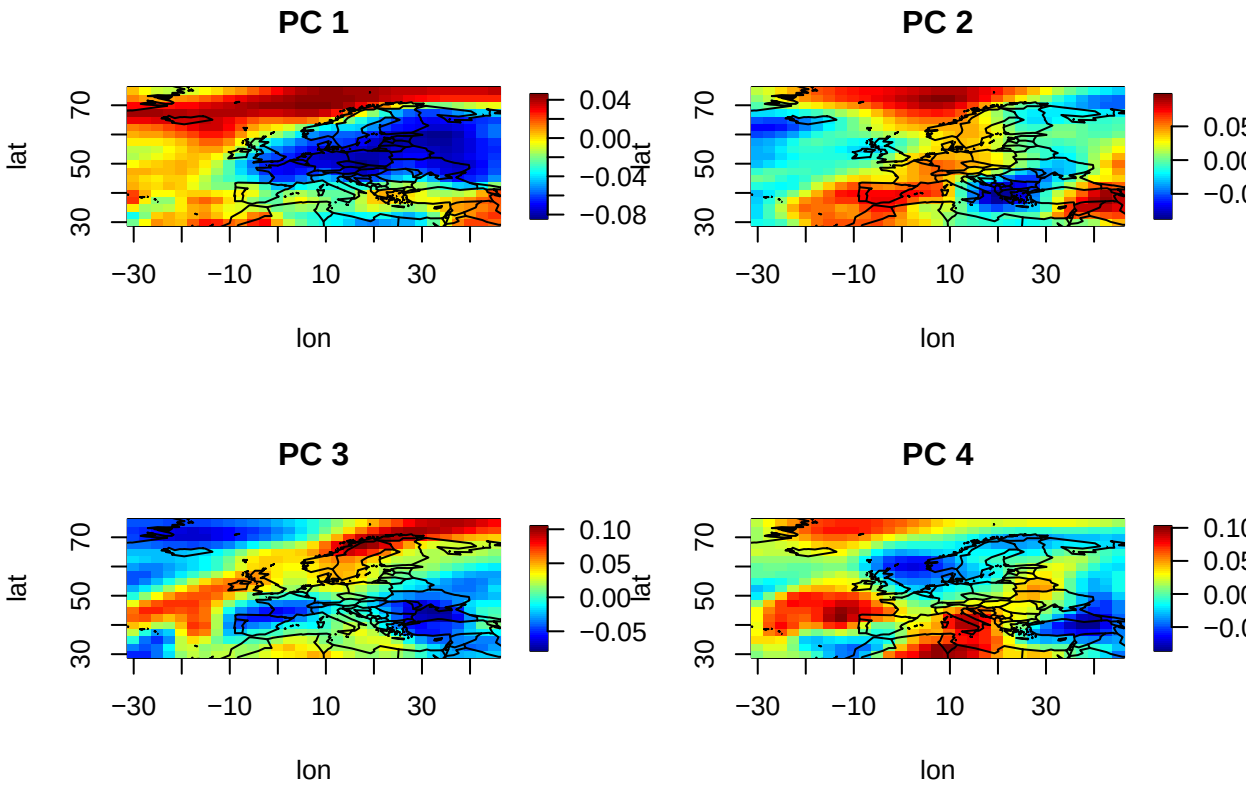


2.B)

```
## bring the data in the right format to perform PCA

pre2 <- t(array(pre, c(dim(pre)[1] * dim(pre)[2], dim(pre)[3])))


## perform PCA
pca <- prcomp(pre2, scale=TRUE)



## plot the PCs and the amount of variation how much of them can explain in the data

screeplot(pca)
```




```
## How much variance is explained by the 1st, the 1st and 2nd, the 1st, 2nd and 3rd PCs.
#round(cumsum(pca$sdev^2)/sum( pca$sdev^2), 3)



## Make a Plot for each loading

par(mfrow = c(2, 2))
for (i in 1:4) {
    loading <- pca$rotation[,i]
    image.plot(lon, lat, matrix(loading, nrow = 31), main = paste("PC", i))
    map("world", add = TRUE)
}
```

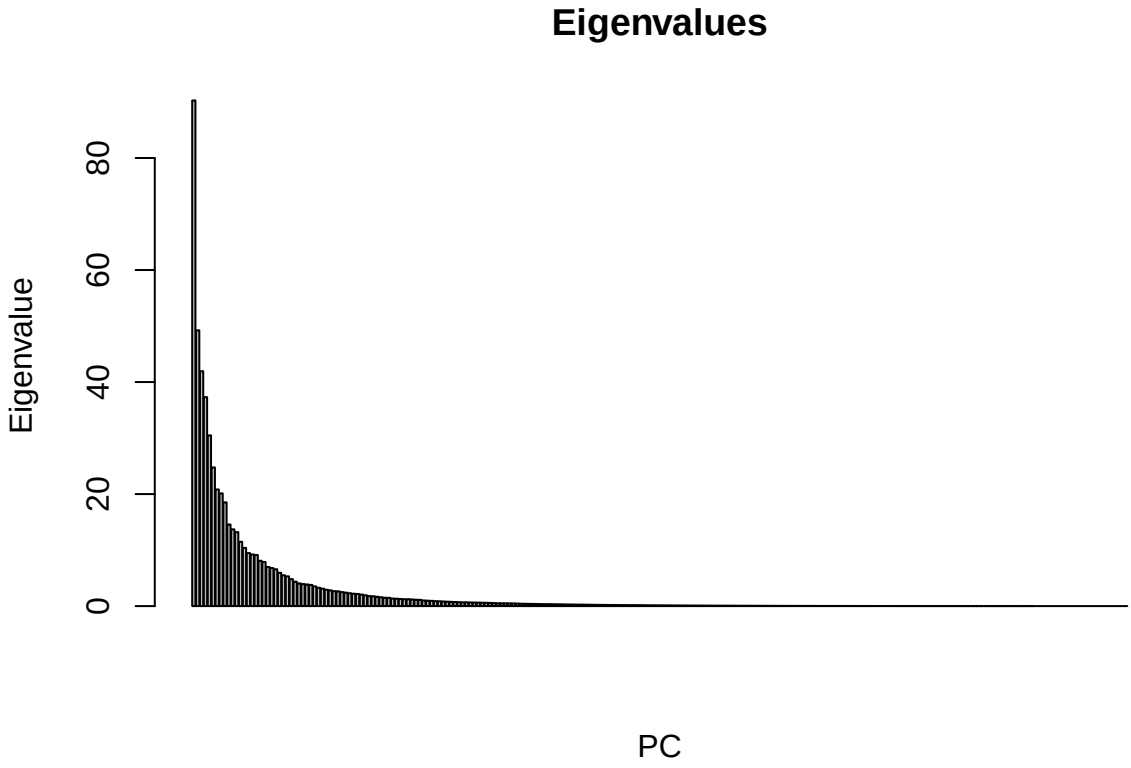


« **Comments** »

- First we needed to bring the data in the right format to perform PCA. We have 31*19=589 locations which are our variables. We have 240 time points that are our observations. Therefore, we rearrange the 3 dimensional matrix into a 2D matrix where the columns are the locations (variables or features) and the rows are time points (observations).
- We have spatial locations (variables)
- in this scenario we do not scale the data since they are all in the same unit and we are interested in the difference in variance that exist in the data

2.C)

```
## Calculate Eigenvalues from sdev
eigen_vals <- pca$sdev^2

## Display Eigenvalues
par(mfrow = c(1, 1))
barplot(eigen_vals, main = "Eigenvalues", xlab = "PC", ylab = "Eigenvalue")
```



« **Comments** »

- In Principal Component Analysis (PCA), the number of principal components (PCs) you can obtain is equal to the number of variables (or features) in your dataset, provided that the number of observations (samples) is greater than or equal to the number of variables. Here since our number of observations is lower than the number of dimensions, only the same number as our **observations (240)** are relevant!

2.D)

```
## let's apply North 'rule of thumb' to the result of the pca analysis to determine how many PCs should

eigen_vals <- pca$sdev^2

for (i in 1:(length(eigen_vals)-1)){
    if ((eigen_vals[i] - eigen_vals[i+1])/eigen_vals[i] < sqrt(2/length(eigen_vals))) {
        pc_opt <- i
        print(paste0("According to North's rule of thumb the first ", i, " PCs should be kept and the r
        break
    }
}
```

```
## [1] "According to North's rule of thumb the first 7 PCs should be kept and the rest should be trunca
```

```
print(paste0("With ", i, " PCs we can explain ", round(cumsum(pca$sdev^2)/sum( pca$sdev^2), 3)[pc_opt]
```

```
## [1] "With 7 PCs we can explain 0.501 of variance in the data!"
```

Problem 3

3.A)

```
# set parameters
mu <- c(2,5)
Sigma <- array( c(1, 0.5, 0.5, 1), c(2,2))

# eigen values of sigma
eigenSigma <- eigen(Sigma)

# eigen values
eigenvalues <- eigenSigma$values
eigenvalues
```

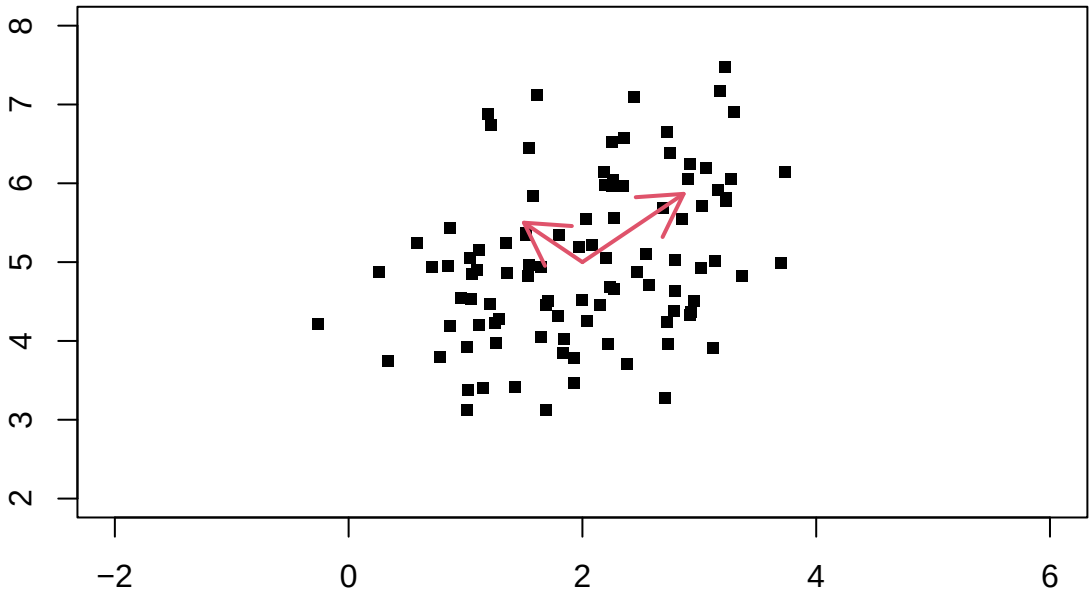
```
## [1] 1.5 0.5
```

```
# eigen vectors
eigenvectors <- eigenSigma$vectors
eigenvectors
```

```
##           [,1]       [,2]
## [1,] 0.7071068 -0.7071068
## [2,] 0.7071068  0.7071068
```

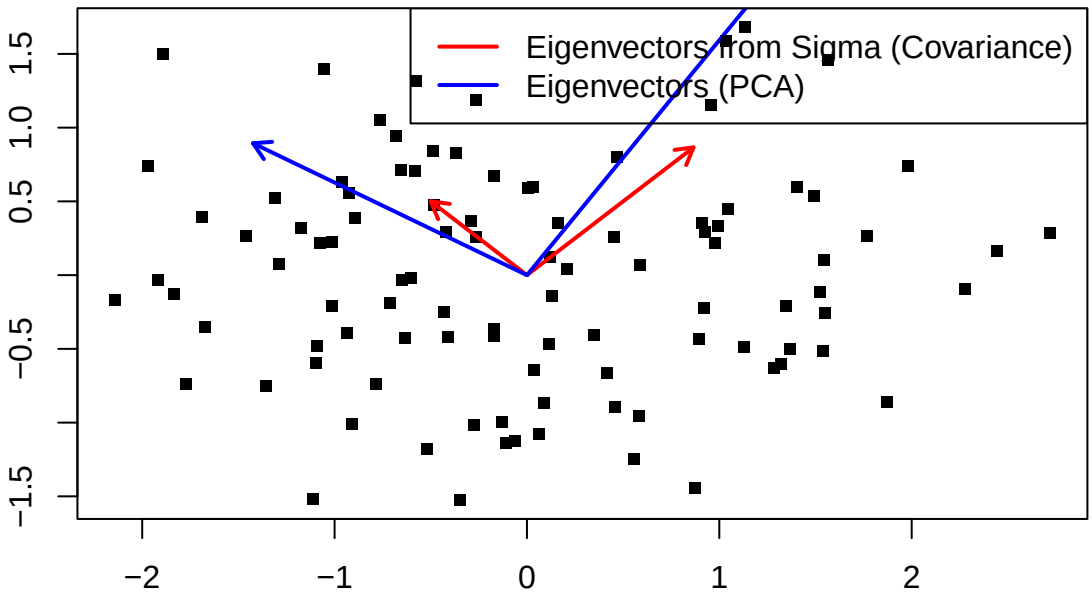
3.B&C)

```
set.seed(3)

sample <- rmvnorm(100, mean = mu, sigma=Sigma) # sample from bivariate


# scatter plot + eigenvectors

plot(sample, pch='.', xlab='', ylab='', xlim=c(-2, 6),ylim=c(2, 8), cex = 6)
arrows(2, 5, 2+sqrt(eigenvalues)*eigenvectors[1,], 5+sqrt(eigenvalues)*eigenvectors[2,], col = 2, lwd =
```



3.D)

```
set.seed(1)
# perform pcs
pca <- prcomp(sample, scale=FALSE)



plot(pca$x, pch='.', xlab='', ylab='', cex = 6)




arrows(0, 0, eigenvectors[1, 1] * sqrt(eigenvalues[1]), eigenvectors[2, 1] * sqrt(eigenvalues[1]),
       col="red", lwd=2, length=0.1)
arrows(0, 0, eigenvectors[1, 2] * sqrt(eigenvalues[2]), eigenvectors[2, 2] * sqrt(eigenvalues[2]),
       col="red", lwd=2, length=0.1)



loadings <- pca$rotation  # Get PCA loadings (eigenvectors)

arrows(0, 0, loadings[1, 1] * max(abs(pca$x[, 1])), loadings[2, 1] * max(abs(pca$x[, 1])),
       col="blue", lwd=2, length=0.1)
arrows(0, 0, loadings[1, 2] * max(abs(pca$x[, 2])), loadings[2, 2] * max(abs(pca$x[, 2])),
       col="blue", lwd=2, length=0.1)


legend("topright", legend=c("Eigenvectors from Sigma (Covariance)", "Eigenvectors (PCA)"), col=c("red",
```



« Comments »

- the descrepency may arise from two sources:

1. Data Variability: PCA uses the empirical covariance of the data, which may vary slightly from the theoretical covariance defined in sigma due to the randomness in the generated data.

2. Scaling and Centering: PCA works with centered data (mean-subtracted), while the eigenvalues of sigma represent the theoretical model, which assumes the data is centered.