Principal Component Analysis (PCA)

Suppose the matrix X is already given in R, then you can compute the scatter matrix (S) simply by:

$$S = X'X$$

```
X <- matrix(c(1,-3,1,-3,-1,2,1,2,3), ncol=3, byrow = TRUE)
X
```

```
##      [,1] [,2] [,3]
## [1,]    1   -3    1
## [2,]   -3   -1    2
## [3,]    1    2    3
```

```
S <- t(X) %*% X
S
```

```
##      [,1] [,2] [,3]
## [1,]   11    2   -2
## [2,]    2   14    1
## [3,]   -2    1   14
```

If you want to compute the eigenvectors and eigenvalues of the matrix

```
M <- matrix(c(1,-3,1,-3,-1,2,1,2,3) , ncol=3, byrow = TRUE)
M
```

```
##      [,1] [,2] [,3]
## [1,]    1   -3    1
## [2,]   -3   -1    2
## [3,]    1    2    3
```

```
eigen_of_M <- eigen(M)
eigen_of_M
```

```
## eigen() decomposition
## $values
## [1]  3.872983  3.000000 -3.872983
##
## $vectors
##            [,1]       [,2]       [,3]
## [1,] -0.1938227  0.8164966  0.5438438
## [2,]  0.4722473 -0.4082483  0.7812271
## [3,]  0.8598926  0.4082483 -0.3064605
```

Now, we enter the dataset into R, and use prcomp() to carry out PCA.

- We first do centering but not normalization.

```
Y <- matrix(c(-3, 2.2, 4, 1.1, -5.8, -2, 1.9, 3, 1.2, -4.1,
              0, 2.1, 1, 1.2, 0, 1, 1.7, 0, 1.2, 2.1,
              2, 1.8, -1, 1.2, 3.7, 4, 2.3, -3, 0, 7.9,
              6, 2.2, -5, 1.2, 12.3), ncol=5, byrow = TRUE)
Y
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   -3  2.2    4  1.1 -5.8
## [2,]   -2  1.9    3  1.2 -4.1
## [3,]    0  2.1    1  1.2  0.0
## [4,]    1  1.7    0  1.2  2.1
## [5,]    2  1.8   -1  1.2  3.7
## [6,]    4  2.3   -3  0.0  7.9
## [7,]    6  2.2   -5  1.2 12.3
```

```
pca.noscale <- prcomp(Y,center = TRUE)
pca.noscale
```

```
## Standard deviations (1, .., p=5):
## [1] 7.821103e+00 4.402721e-01 1.990404e-01 8.008254e-02 4.129542e-16
##
## Rotation (n x k) = (5 x 5):
##               PC1         PC2        PC3        PC4           PC5
## [1,]  0.407114397 -0.02325611 -0.2587597  0.5164886 -7.071068e-01
## [2,]  0.007412688 -0.31162831  0.8573340  0.4096478  9.992007e-16
## [3,] -0.407114397  0.02325611  0.2587597 -0.5164886 -7.071068e-01
## [4,] -0.019921330  0.94836412  0.2557574  0.1865389  1.621966e-16
## [5,]  0.817351777  0.04910795  0.2562293 -0.5136838  1.498801e-15
```

Next, we do both centering and normalization.

- scale = TRUE => do centering and do normalization

```
pca.scale <- prcomp(Y, scale = TRUE)
pca.scale
```

```
## Standard deviations (1, .., p=5):
## [1] 1.819324e+00 1.119328e+00 6.608955e-01 1.950016e-02 8.069628e-17
##
## Rotation (n x k) = (5 x 5):
##             PC1        PC2         PC3         PC4           PC5
## [1,]  0.5324006 -0.2215414 -0.02341776  0.40855919  7.071068e-01
## [2,]  0.2442453  0.6930274 -0.67801871  0.01865143  1.110223e-16
## [3,] -0.5324006  0.2215414  0.02341776 -0.40855919  7.071068e-01
## [4,] -0.2999653 -0.6111549 -0.73225922  0.01751951 -5.551115e-17
## [5,]  0.5324136 -0.2191838 -0.05468348 -0.81578426 -2.109424e-15
```

Quiz

For this question, you may use R or Python. Your answer must be accurate to at least 4 decimal places.

What is the sum of all eigenvalues of

```
q <- matrix(c(13.3, -6.4, 1.8, -17.6, -6.4, -7.7, 1.8, -10.1, 1.8,
             1.8, -4.4, 27.9, -17.6, 10.1, 27.9, 5.2),
           ncol=4, byrow = TRUE)
q
```

```
##       [,1] [,2] [,3]  [,4]
## [1,]  13.3 -6.4  1.8 -17.6
## [2,]  -6.4 -7.7  1.8 -10.1
## [3,]   1.8  1.8 -4.4  27.9
## [4,] -17.6 10.1 27.9   5.2
```

```
eigen_of_q <- eigen(q)
eigen_of_q
```

```
## eigen() decomposition
## $values
## [1]  35.333241 -30.169024  10.375158  -9.139374
##
## $vectors
##             [,1]       [,2]       [,3]        [,4]
## [1,] -0.51034728 -0.3125313  0.6638405 -0.24640538
## [2,] -0.07171063 -0.3940240 -0.3405695 -0.95383540
## [3,]  0.47465202  0.6571854  0.5968120  0.16441278
## [4,]  0.71351854 -0.5614060  0.2952009  0.04950608
```

```
eigen_of_q$values
```

```
## [1]  35.333241 -30.169024  10.375158  -9.139374
```

```
35.333241+-30.169024 + 10.375158 + -9.139374
```

```
## [1] 6.400001
```

```
sum(eigen_of_q$values)
```

```
## [1] 6.4
```