Matrix Algebra Homework

Meghna Bajoria

25-09-2022

NOTE: Please do not use any of these functions - stats::prcomp(), FactoMiner::PCA(), ade4::dudi.pca(), ExPosition::epPCA() for completing this homework. You can use these functions to cross check your answers.

Question 1

Consider the inbuilt dataset `Seatbelts` . Please use `help()` to learn more about the data.

Task 1: Generate the covariance and correlation matrix. What can you say about the variables in the data

```
covariance_matrix <- cov(Seatbelts)
round(covariance_matrix)
```

```
##               DriversKilled drivers   front  rear     kms PetrolPrice VanKilled
## DriversKilled           644    6533    3141   745  -23944           0        38
## drivers                6533   83875   40995  8271 -378445          -2       511
## front                  3141   40995   30660  9025 -183855          -1       301
## rear                    745    8271    9025  6906   81306           0        37
## kms                  -23944 -378445 -183855 81306 8632133          14     -5322
## PetrolPrice               0      -2      -1     0      14           0         0
## VanKilled                38     511     301    37   -5322           0        13
## law                      -3     -42     -32     1     469           0         0
##               law
## DriversKilled  -3
## drivers       -42
## front         -32
## rear            1
## kms           469
## PetrolPrice     0
## VanKilled       0
## law             0
```

Covariance matrix indicates the direction of linear relationship between variables. From the above result, we can say that the variables which change positively with each other are (DriversKilled, drivers) and (kms, PetrolPrice) and (front, drivers).

```
correlation_matrix <- cor(Seatbelts)
correlation_matrix
```

```
##                DriversKilled    drivers      front        rear        kms
## DriversKilled     1.0000000  0.8888264  0.7067596  0.35335102 -0.3211016
## drivers           0.8888264  1.0000000  0.8084114  0.34366850 -0.4447631
## front             0.7067596  0.8084114  1.0000000  0.62022476 -0.3573823
## rear              0.3533510  0.3436685  0.6202248  1.00000000  0.3330069
## kms              -0.3211016 -0.4447631 -0.3573823  0.33300689  1.0000000
## PetrolPrice      -0.3866061 -0.4576675 -0.5392394 -0.13262721  0.3839004
## VanKilled         0.4070412  0.4853995  0.4724207  0.12175808 -0.4980356
## law              -0.3285051 -0.4452269 -0.5624455  0.02906753  0.4905494
##                PetrolPrice  VanKilled         law
## DriversKilled  -0.3866061  0.4070412 -0.32850510
## drivers        -0.4576675  0.4853995 -0.44522689
## front          -0.5392394  0.4724207 -0.56244554
## rear           -0.1326272  0.1217581  0.02906753
## kms             0.3839004 -0.4980356  0.49054938
## PetrolPrice     1.0000000 -0.2885584  0.39069323
## VanKilled      -0.2885584  1.0000000 -0.39494121
## law             0.3906932 -0.3949412  1.00000000
```

Correlation indicates both the strength and direction of the linear relationship between two variables. Numbers closer to 1 indicate high correlation so drivers, DriversKilled and front are highly correlated. VanKilled, kms are not correlated much.

Task 2: Check if the covariance matrix is orthognal

```
#transpose of covariance matrix is
transpose_cov_mat = t(covariance_matrix)
transpose_cov_mat
```

```
##                DriversKilled       drivers         front          rear
## DriversKilled  6.441386e+02  6.533134e+03  3.140834e+03   745.2613438
## drivers        6.533134e+03  8.387451e+04  4.099501e+04  8271.1764834
## front          3.140834e+03  4.099501e+04  3.065965e+04  9024.9594241
## rear           7.452613e+02  8.271176e+03  9.024959e+03  6905.9773124
## kms           -2.394370e+04 -3.784451e+05 -1.838551e+05 81306.4232112
## PetrolPrice   -1.194695e-01 -1.613852e+00 -1.149645e+00    -0.1341974
## VanKilled      3.757161e+01  5.112650e+02  3.008460e+02    36.7995201
## law           -2.714387e+00 -4.197941e+01 -3.206299e+01     0.7864311
##                          kms   PetrolPrice     VanKilled           law
## DriversKilled  -23943.69655 -0.1194695154  3.757161e+01  -2.714386998
## drivers       -378445.06621 -1.6138524988  5.112650e+02 -41.979412086
## front         -183855.12238 -1.1496454293  3.008460e+02 -32.062990838
## rear            81306.42321 -0.1341973554  3.679952e+01   0.786431065
## kms           8632133.14092 13.7333452831 -5.321710e+03 469.225676265
## PetrolPrice        13.73335  0.0001482509 -1.277804e-02   0.001548726
## VanKilled       -5321.71019 -0.0127780401  1.322707e+01  -0.467631981
## law               469.22568  0.0015487260 -4.676320e-01   0.105993674
```

```
#inverse of covariance matrix is
inverse_cov_mat = solve(covariance_matrix)
inverse_cov_mat
```

```
##                DriversKilled       drivers         front          rear
## DriversKilled  7.736360e-03 -6.441945e-04  2.968999e-05 -4.014583e-05
## drivers       -6.441945e-04  9.740092e-05 -9.084317e-05  7.035932e-05
## front          2.968999e-05 -9.084317e-05  4.228776e-04 -5.077998e-04
## rear          -4.014583e-05  7.035932e-05 -5.077998e-04  8.717233e-04
## kms           -4.544770e-06  1.954578e-07  5.556944e-06 -1.173056e-05
## PetrolPrice    1.260248e-01 -6.964661e-02  7.130730e-01 -6.418898e-01
## VanKilled     -4.586306e-04 -3.047254e-04  3.775767e-04 -1.615796e-03
## law           -3.148336e-02 -7.115011e-03  6.311573e-02 -7.905813e-02
##                         kms   PetrolPrice     VanKilled          law
## DriversKilled -4.544770e-06  1.260248e-01 -4.586306e-04 -0.031483356
## drivers        1.954578e-07 -6.964661e-02 -3.047254e-04 -0.007115011
## front          5.556944e-06  7.130730e-01  3.775767e-04  0.063115730
## rear          -1.173056e-05 -6.418898e-01 -1.615796e-03 -0.079058134
## kms            3.620554e-07 -1.193403e-03  7.250237e-05  0.000463551
## PetrolPrice   -1.193403e-03  1.063514e+04 -8.039937e-01 42.450618186
## VanKilled      7.250237e-05 -8.039937e-01  1.206517e-01  0.216858624
## law            4.635510e-04  4.245062e+01  2.168586e-01 23.773743846
```

As transpose of covariance matrix and its inverse is not equal, we can say that covariance matrix is not orthogonal.

Task 3: Compute the eigenvalues and eigenvectors for covariance and correlation matrix. What did you observe from your analysis?

```
#for covariance matrix
eigen_covariance = eigen(covariance_matrix)
#Eigen values for covariance matrix are
eigen_covariance$values
```

```
## [1] 8.653669e+06 8.887021e+04 1.017188e+04 1.382469e+03 1.290075e+02
## [6] 8.436161e+00 4.236133e-02 9.402641e-05
```

```
#Eigen vectors for covariance matrix are
eigen_covariance$vectors
```

```
##               [,1]          [,2]          [,3]          [,4]          [,5]
## [1,] -2.803988e-03 -6.918498e-02  4.581489e-02 -3.156923e-02  9.960454e-01
## [2,] -4.420087e-02 -8.506155e-01  5.110515e-01 -7.787800e-02 -8.518609e-02
## [3,] -2.149609e-02 -4.846563e-01 -7.335282e-01  4.757592e-01  1.509634e-02
## [4,]  9.326331e-03 -1.859958e-01 -4.455659e-01 -8.753904e-01 -2.015858e-02
## [5,]  9.987437e-01 -4.653636e-02  1.111680e-02  1.487509e-02 -4.599015e-04
## [6,]  1.595997e-06  1.489955e-05  2.217581e-05 -6.919467e-05 -1.935179e-05
## [7,] -6.175254e-04 -3.854373e-03 -3.271329e-03 -6.751971e-03  1.011423e-03
## [8,]  5.445045e-05  3.314404e-04  6.693543e-04 -4.054510e-03  1.212178e-03
##               [,6]          [,7]          [,8]
## [1,]  0.0013265015 -1.355558e-03  1.183774e-05
## [2,]  0.0020711901 -2.896868e-04 -6.551244e-06
## [3,]  0.0011078950  2.552946e-03  6.707090e-05
## [4,]  0.0080292325 -3.240857e-03 -6.038384e-05
## [5,] -0.0005745090  1.986889e-05 -1.120366e-07
## [6,] -0.0001126865 -3.999820e-03  9.999920e-01
## [7,] -0.9999198174  9.370268e-03 -7.551531e-05
## [8,]  0.0093952450  9.999386e-01  4.000389e-03
```

```
#for correlation matrix
eigen_correlation <- eigen(correlation_matrix)
#Eigen values for correlation matrix are
eigen_correlation$values
```

```
## [1] 4.03362903 1.57815295 0.73720569 0.64061615 0.56293459 0.30569022 0.09557333
## [8] 0.04619805
```

```
#Eigen vectors for correlation matrix are
eigen_correlation$vectors
```

```
##            [,1]        [,2]        [,3]       [,4]        [,5]       [,6]
## [1,] -0.4080022 -0.17871091 -0.25491038  0.4861386 -0.16113152 -0.3040074
## [2,] -0.4501636 -0.10462795 -0.18642612  0.3392325 -0.15088056 -0.0661405
## [3,] -0.4529427 -0.22338322  0.12423577 -0.1650995 -0.10215441  0.3127628
## [4,] -0.1833141 -0.68358056  0.04508572 -0.3320193  0.11033792  0.3288694
## [5,]  0.2878262 -0.55389084  0.12112979 -0.1335604 -0.08051612 -0.6998498
## [6,]  0.3171550 -0.09538963 -0.72129718 -0.1780651 -0.53759412  0.2062737
## [7,] -0.3240129  0.18802266 -0.51075334 -0.4886636  0.50574589 -0.3203784
## [8,]  0.3220408 -0.29780228 -0.29493683  0.4719727  0.61432282  0.2494962
##             [,7]        [,8]
## [1,]  0.60921245  0.10374436
## [2,] -0.69238591 -0.36060314
## [3,] -0.18355907  0.74853223
## [4,]  0.23231744 -0.46102396
## [5,] -0.21500732  0.18801435
## [6,] -0.02775146  0.07752596
## [7,] -0.01777660  0.03556438
## [8,] -0.12034254  0.20912771
```

Eigen value and Eigen vector for represent the direction in which the data is going to be selected. From results, we can see that the First eigen value will decide the direction of the data as it has more than 50% of the value.

Task 4: Find the squareroot of the covariance matrix using spectral decomposition method

```
covariance_vectors = eigen_covariance$vectors
sqrt = covariance_vectors %*% sqrt(diag(eigen_covariance$values)) %*% t(covariance_vectors)
sqrt
```

```
##               [,1]          [,2]          [,3]         [,4]          [,5]
## [1,] 12.9673017814   19.39747684   6.396194106 2.499876e+00 -7.249656e+00
## [2,] 19.3974768389  248.09333272  86.493099261 2.554054e+01 -1.175319e+02
## [3,]  6.3961941061   86.49309926 134.068473999 4.375776e+01 -5.699160e+01
## [4,]  2.4998759564   25.54053812  43.757756198 5.908899e+01  2.899760e+01
## [5,] -7.2496560247 -117.53192317 -56.991598705 2.899760e+01  2.934991e+03
## [6,] -0.0003549118   -0.00262414  -0.005123359 4.771715e-04  4.469229e-03
## [7,]  0.0849863074    0.90162263   0.715471265 5.399883e-01 -1.766567e+00
## [8,]  0.0140378683   -0.04606136  -0.171807992 8.427931e-02  1.538683e-01
##               [,6]          [,7]          [,8]
## [1,] -0.0003549118  0.0849863074  0.0140378683
## [2,] -0.0026241396  0.9016226255 -0.0460613640
## [3,] -0.0051233593  0.7154712655 -0.1718079916
## [4,]  0.0004771715  0.5399883286  0.0842793094
## [5,]  0.0044692286 -1.7665667665  0.1538683304
## [6,]  0.0097002013  0.0003086393 -0.0007740823
## [7,]  0.0003086393  2.9123958939 -0.0250266972
## [8,] -0.0007740823 -0.0250266972  0.2067645444
```

```
all.equal(sqrt %*% sqrt, covariance_matrix, check.attributes = FALSE)
```

```
## [1] TRUE
```

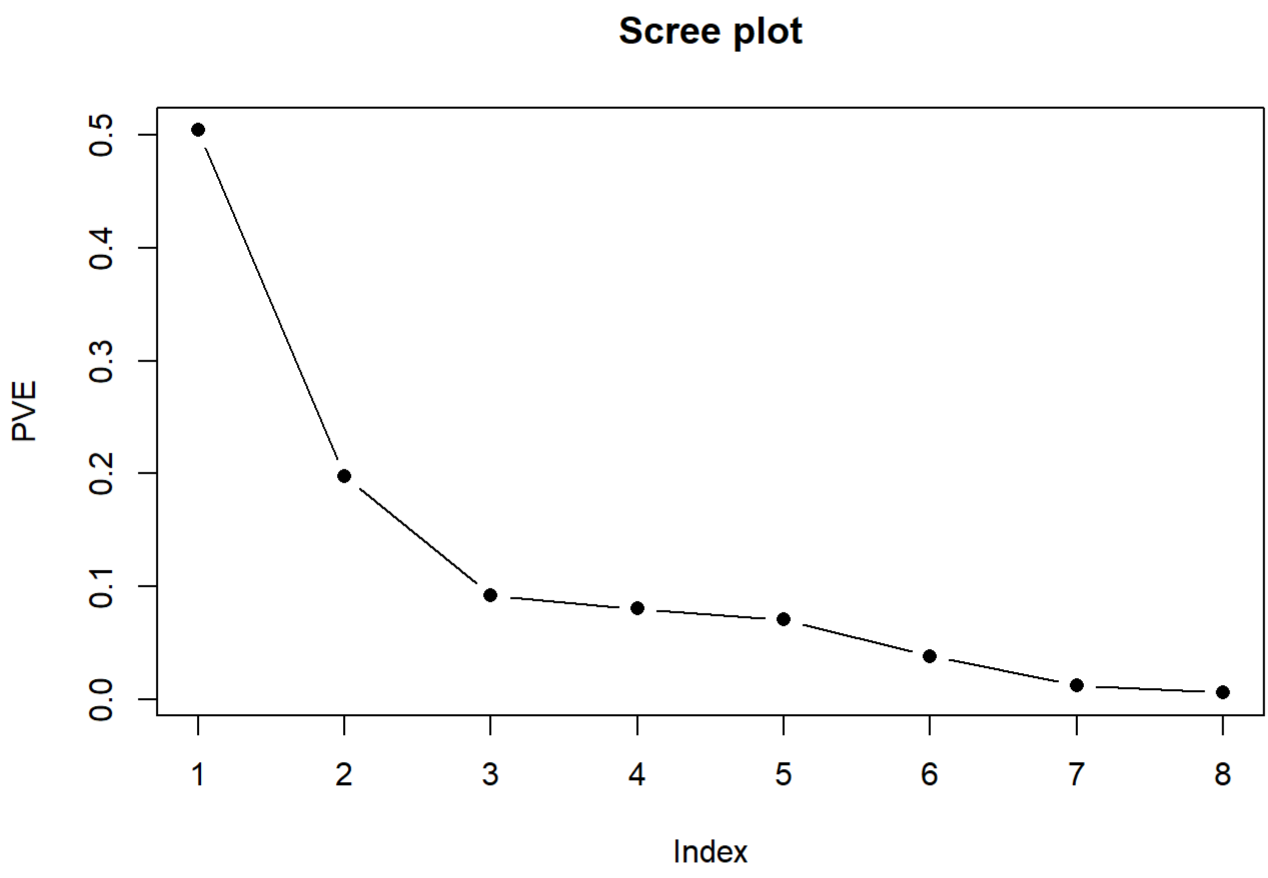
Here, the square root decomposition is equal to the covariance matrix

Task 5: Based on the eigen decomposition in task 3, determine how many principal components you would select to reduce feature dimensions yet capture atleast 85% of the variability in the data? Perform the analysis using the correlation matrix.

```
PVE = eigen_correlation$values/sum(eigen_correlation$values)
PVE
```

```
## [1] 0.504203628 0.197269118 0.092150711 0.080077019 0.070366823 0.038211278
## [7] 0.011946666 0.005774756
```

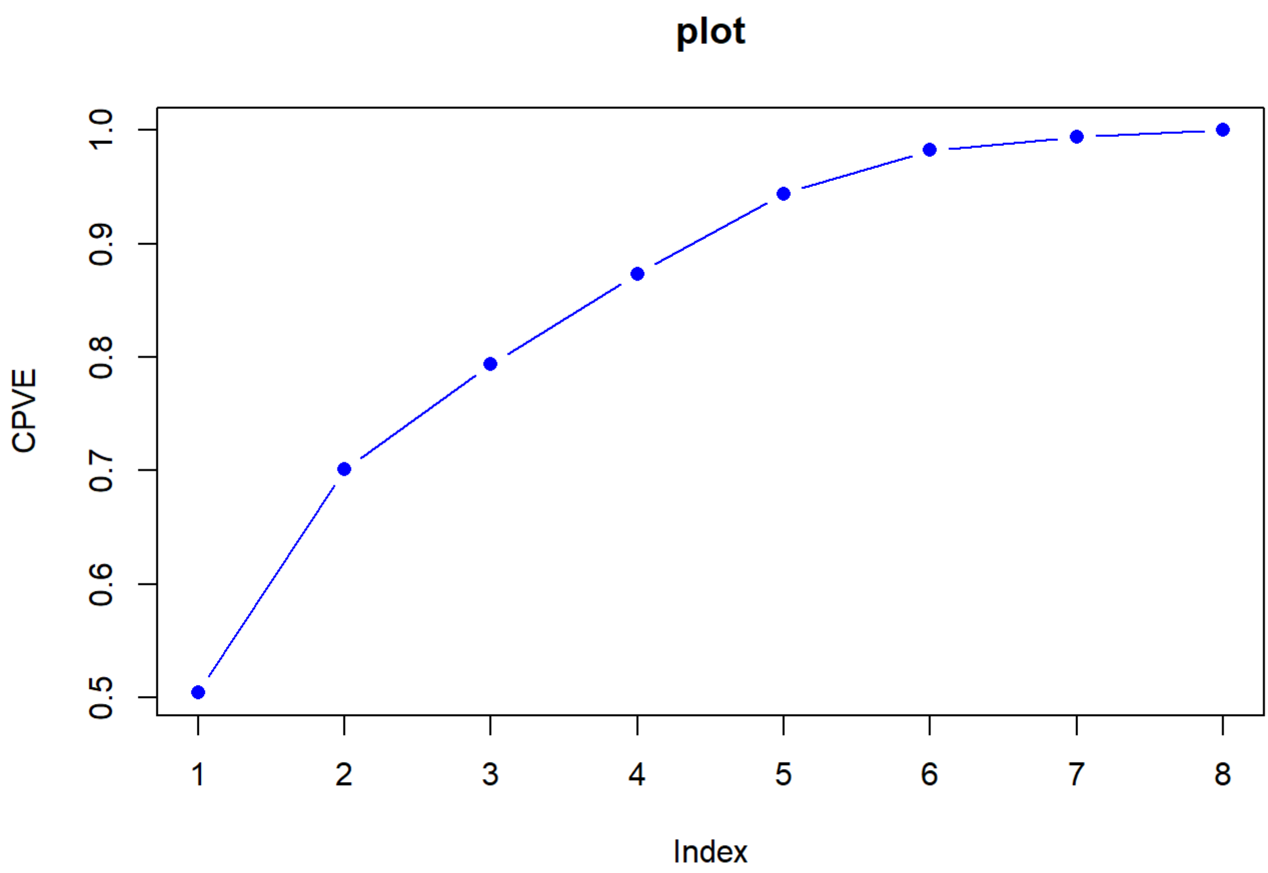
```
plot(PVE,type="b", col="black",pch=16,main = "Scree plot")
```




```
CPVE <- cumsum(PVE)
CPVE
```

```
## [1] 0.5042036 0.7014727 0.7936235 0.8737005 0.9440673 0.9822786 0.9942252
## [8] 1.0000000
```

```
plot(CPVE, type = "b", col="blue", pch=16, main = "plot")
```



We need to find the cumulative sum to find the number of pricnipal components needed to accumulate the sum to 85%. Hence, we need 4 principal components to capture at least 85% of the variability in the data.

Task 6: Compute the principal component vectors based on your selection in task 5. Comment on your interpretation of the PCs

```
evecs = eigen_correlation$vectors[,1:4]
colnames(evecs) = c("e1", "e2", "e3", "e4")
row.names(evecs) = colnames(Seatbelts)
evecs
```

```
##                      e1          e2          e3         e4
## DriversKilled -0.4080022 -0.17871091 -0.25491038  0.4861386
## drivers       -0.4501636 -0.10462795 -0.18642612  0.3392325
## front         -0.4529427 -0.22338322  0.12423577 -0.1650995
## rear          -0.1833141 -0.68358056  0.04508572 -0.3320193
## kms            0.2878262 -0.55389084  0.12112979 -0.1335604
## PetrolPrice    0.3171550 -0.09538963 -0.72129718 -0.1780651
## VanKilled     -0.3240129  0.18802266 -0.51075334 -0.4886636
## law            0.3220408 -0.29780228 -0.29493683  0.4719727
```

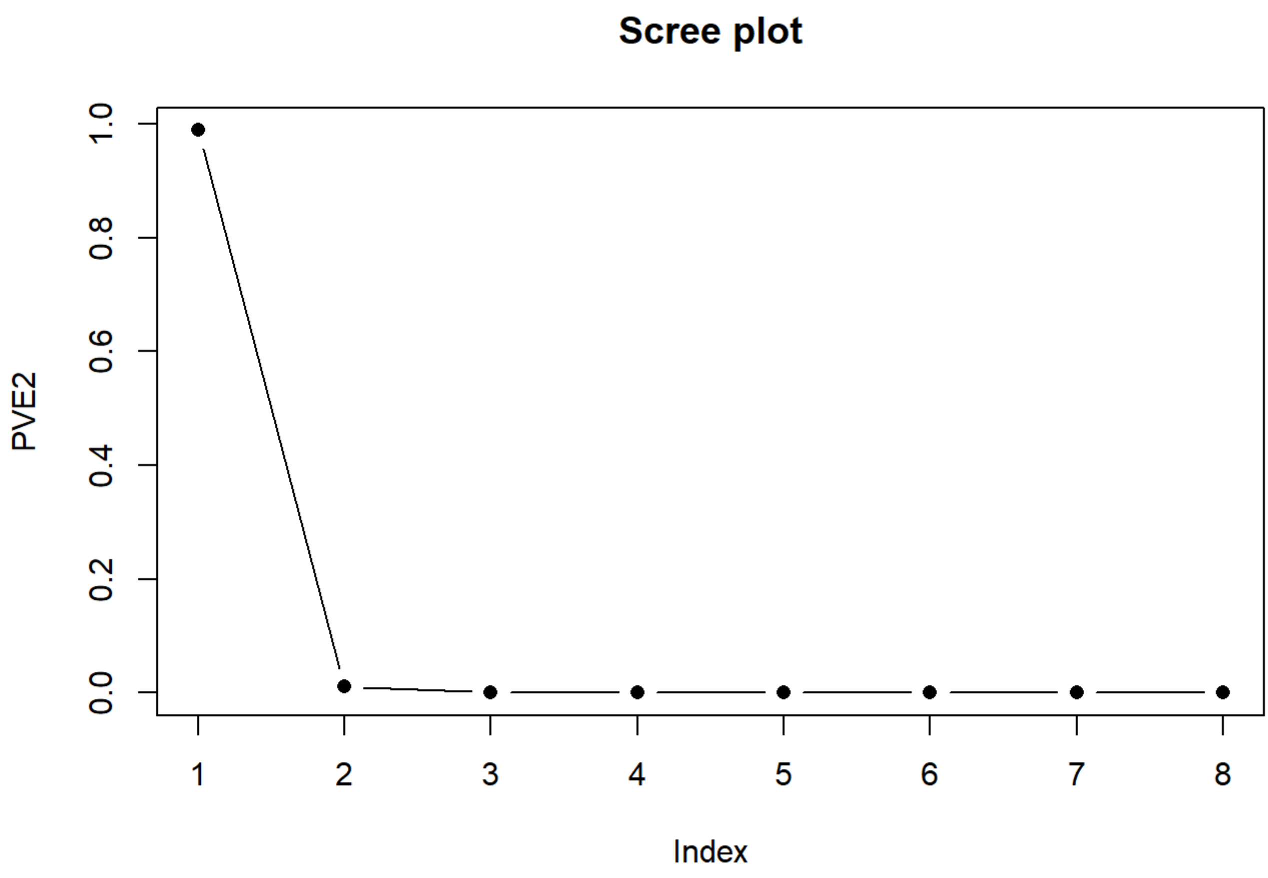
```
PC1 <- as.matrix(Seatbelts) %*% evecs[,1]
PC2 <- as.matrix(Seatbelts) %*% evecs[,2]
PC3 <- as.matrix(Seatbelts) %*% evecs[,3]
PC4 <- as.matrix(Seatbelts) %*% evecs[,4]
PC <- data.frame(PC1, PC2, PC3, PC4)
head(PC)
```

```
##        PC1       PC2       PC3        PC4
## 1 1358.467 -5588.636  869.1757  -823.9586
## 2 1069.354 -4796.087  736.3294  -694.8362
## 3 1720.195 -6090.180 1008.1848 -1014.7195
## 4 2048.299 -6686.888 1161.9185 -1224.4735
## 5 2084.451 -7270.519 1235.9419 -1286.8620
## 6 2332.523 -7540.849 1322.1509 -1394.9980
```

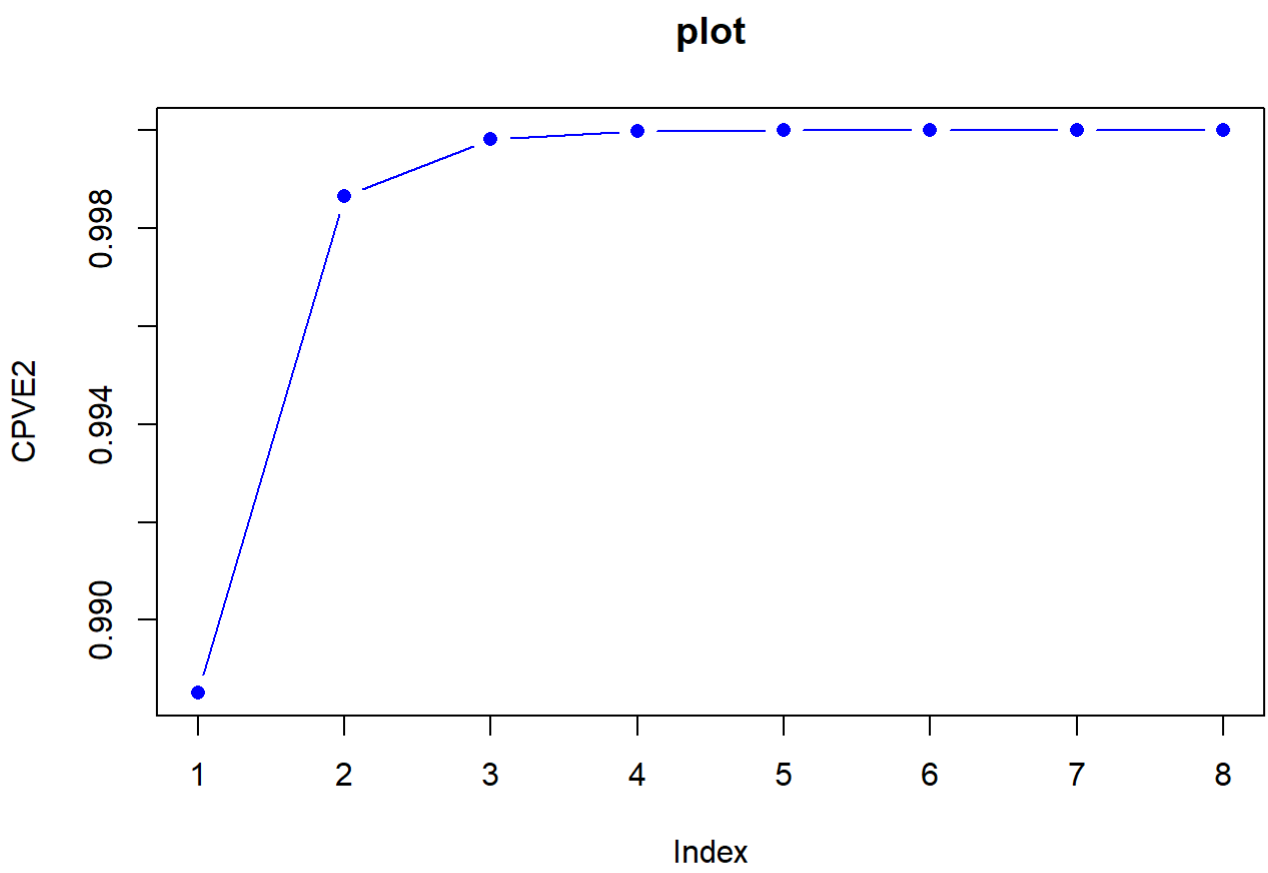
- e1 values represent contrast between the score of Non accidental variables (kms, PetrolPrice and law) to accidentals variables (DriversKilled, drivers front, rear, VanKilled).
- Most of the values in e2 are negative except VanKilled.
- e3 have 5 negative values which are DriversKilled, drivers, PetrolPrice, Vankilled and law.
- e4 has 3 positive value which seems to indicate the relation between DriversKilled, drivers and the laws that were effective during that time.

Task 7: Perform task 5 and 6 using covariance matrix. Compare the results with the ones obtained from correlation matrix. Do the interpretation of the components differ?

```
PVE2 = eigen_covariance$values/sum(eigen_covariance$values)
plot(PVE2,type="b", col="black", pch=16, main = "Scree plot")
```




```
CPVE2 <- cumsum(PVE2)
plot(CPVE2,type="b",col="blue", pch=16,main = "plot")
```



```
evecs = eigen_covariance$vectors[,1]
evecs
```

```
## [1] -2.803988e-03 -4.420087e-02 -2.149609e-02  9.326331e-03  9.987437e-01
## [6]  1.595997e-06 -6.175254e-04  5.445045e-05
```

```
PC1 <- as.matrix(Seatbelts) %*% evecs
PC <- data.frame(PC1)
head(PC)
```

```
##         PC1
## 1  8956.617
## 2  7593.152
## 3  9869.229
## 4 10866.068
## 5 11718.603
## 6 12292.009
```

Yes, the interpretation of the components differ when we use correlation and covariance matrices.