Lab 2 - Jin Kweon

Jin Kweon

8/28/2017

Use Latex to describe the problem by using latex.

$$M = UDV^T$$

```
?USArrests
head(USArrests)
```

```
##            Murder Assault UrbanPop Rape
## Alabama      13.2     236       58 21.2
## Alaska       10.0     263       48 44.5
## Arizona       8.1     294       80 31.0
## Arkansas      8.8     190       50 19.5
## California    9.0     276       91 40.6
## Colorado      7.9     204       78 38.7
```

```
class(USArrests)
```

```
## [1] "data.frame"
```

```
dim(USArrests)
```

```
## [1] 50  4
```

```
SVD <- svd(USArrests) # use nu and nv to control over the number of left/right singular vectors to be c
U <- SVD$u
D <- SVD$d
V <- SVD$v
U
```

```
##              [,1]         [,2]         [,3]         [,4]
##  [1,] -0.17162510  0.096325710  0.065154797  0.153695511
##  [2,] -0.18911657  0.173452566 -0.426657848 -0.178014378
##  [3,] -0.21559302  0.078998111  0.020637399 -0.280707843
##  [4,] -0.13902443  0.059889811  0.013922695  0.016104178
##  [5,] -0.20677884 -0.009812026 -0.176332443 -0.218674250
##  [6,] -0.15587942 -0.064555293 -0.282882796 -0.117974190
##  [7,] -0.09086363 -0.196817368  0.177814176 -0.056150268
##  [8,] -0.17536307  0.035102548  0.242423936 -0.223770615
##  [9,] -0.24315375  0.146502368  0.050477542  0.025718639
## [10,] -0.15591071  0.042885364 -0.069631843  0.426192214
## [11,] -0.05035785 -0.336841681 -0.093988180  0.169255594
## [12,] -0.09273525 -0.071651205  0.048571905 -0.144733647
## [13,] -0.18583902 -0.004760115  0.112681109 -0.023621705
## [14,] -0.09113246 -0.140219345 -0.077396606  0.106957520
## [15,] -0.05057860 -0.189585706  0.028511452 -0.008876337
## [16,] -0.09241257 -0.139884238 -0.004741157  0.048135122
## [17,] -0.08535772 -0.080906191 -0.029723458  0.262636519
## [18,] -0.18215443  0.078717908  0.086540399  0.247322269
## [19,] -0.06696147 -0.113964054  0.123029673 -0.066578837
## [20,] -0.21660706  0.153849690  0.049568029 -0.114685718
```

```
## [21,] -0.11897975 -0.174463458  0.148220955 -0.097833662
## [22,] -0.18878247  0.041358881 -0.150647316  0.008886179
## [23,] -0.06347096 -0.208911684  0.004116555 -0.024871190
## [24,] -0.18365168  0.204539391  0.135286600  0.267330340
## [25,] -0.13606712 -0.061866664 -0.123362957  0.035441385
## [26,] -0.08547226 -0.086095370 -0.023144761  0.058837358
## [27,] -0.08272894 -0.141176014 -0.002503642 -0.013306588
## [28,] -0.18915343 -0.004408923 -0.366175508 -0.014797638
## [29,] -0.05088440 -0.181955350  0.065488479 -0.010607576
## [30,] -0.12678389 -0.178655055  0.118196789  0.038378458
## [31,] -0.20761634  0.111908381 -0.054872342 -0.089217448
## [32,] -0.19000021 -0.012294397  0.083909562 -0.002144131
## [33,] -0.23558461  0.328507472  0.278377295 -0.076003102
## [34,] -0.04008505 -0.142575408  0.056532460 -0.055095159
## [35,] -0.09799620 -0.177119351 -0.042015264  0.101946039
## [36,] -0.11696684 -0.091518195  0.010574286 -0.006458784
## [37,] -0.12273882 -0.079748802 -0.179013479 -0.153842949
## [38,] -0.08749932 -0.181550072  0.068840439  0.100774718
## [39,] -0.13539644 -0.137804473  0.365196197 -0.178351930
## [40,] -0.19819145  0.214523785  0.064157794  0.110760337
## [41,] -0.06809008 -0.083249423 -0.003105654 -0.002402099
## [42,] -0.14035332  0.008777056 -0.124180416  0.241821582
## [43,] -0.15344207 -0.070106952 -0.003086949  0.204363402
## [44,] -0.09907634 -0.202523816 -0.051437763 -0.126424375
## [45,] -0.03981563 -0.082265694 -0.065840848 -0.007450710
## [46,] -0.11931695 -0.059503177 -0.016925750  0.080989279
## [47,] -0.11446373 -0.129744250 -0.110987949 -0.156070555
## [48,] -0.06324636 -0.060134257  0.041886574  0.122736214
## [49,] -0.05051244 -0.237516175  0.064875555  0.028005591
## [50,] -0.12154608 -0.033632183  0.092478877 -0.007088912
```

```r
D
```

```
## [1] 1419.06140  194.82585   45.66134   18.06956
```

```r
V
```

```
##             [,1]        [,2]        [,3]        [,4]
## [1,] -0.04239181  0.01616262 -0.06588426  0.99679535
## [2,] -0.94395706  0.32068580  0.06655170 -0.04094568
## [3,] -0.30842767 -0.93845891  0.15496743  0.01234261
## [4,] -0.10963744 -0.12725666 -0.98347101 -0.06760284
```

```r
identical(U %*% diag(D) %*% t(V), USArrests) # It is false but they are really similar.
```

```
## [1] FALSE
```

```r
U %*% diag(D) %*% t(V)
```

```
##       [,1] [,2] [,3] [,4]
##  [1,] 13.2  236   58 21.2
##  [2,] 10.0  263   48 44.5
##  [3,]  8.1  294   80 31.0
##  [4,]  8.8  190   50 19.5
##  [5,]  9.0  276   91 40.6
##  [6,]  7.9  204   78 38.7
##  [7,]  3.3  110   77 11.1
```

```
##  [8,]  5.9  238   72 15.8
##  [9,] 15.4  335   80 31.9
## [10,] 17.4  211   60 25.8
## [11,]  5.3   46   83 20.2
## [12,]  2.6  120   54 14.2
## [13,] 10.4  249   83 24.0
## [14,]  7.2  113   65 21.0
## [15,]  2.2   56   57 11.3
## [16,]  6.0  115   66 18.0
## [17,]  9.7  109   52 16.3
## [18,] 15.4  249   66 22.2
## [19,]  2.1   83   51  7.8
## [20,] 11.3  300   67 27.8
## [21,]  4.4  149   85 16.3
## [22,] 12.1  255   74 35.1
## [23,]  2.7   72   66 14.9
## [24,] 16.1  259   44 17.1
## [25,]  9.0  178   70 28.2
## [26,]  6.0  109   53 16.4
## [27,]  4.3  102   62 16.5
## [28,] 12.2  252   81 46.0
## [29,]  2.1   57   56  9.5
## [30,]  7.4  159   89 18.8
## [31,] 11.4  285   70 32.1
## [32,] 11.1  254   86 26.1
## [33,] 13.0  337   45 16.1
## [34,]  0.8   45   44  7.3
## [35,]  7.3  120   75 21.4
## [36,]  6.6  151   68 20.0
## [37,]  4.9  159   67 29.3
## [38,]  6.3  106   72 14.9
## [39,]  3.4  174   87  8.3
## [40,] 14.4  279   48 22.5
## [41,]  3.8   86   45 12.8
## [42,] 13.2  188   59 26.9
## [43,] 12.7  201   80 25.5
## [44,]  3.2  120   80 22.9
## [45,]  2.2   48   32 11.2
## [46,]  8.5  156   63 20.7
## [47,]  4.0  145   73 26.2
## [48,]  5.7   81   39  9.3
## [49,]  2.6   53   66 10.8
## [50,]  6.8  161   60 15.6
```

```r
as.matrix(USArrests)
```

```
##                Murder Assault UrbanPop Rape
## Alabama          13.2     236       58 21.2
## Alaska           10.0     263       48 44.5
## Arizona           8.1     294       80 31.0
## Arkansas          8.8     190       50 19.5
## California        9.0     276       91 40.6
## Colorado          7.9     204       78 38.7
## Connecticut       3.3     110       77 11.1
## Delaware          5.9     238       72 15.8
```

```
## Florida          15.4     335        80 31.9
## Georgia          17.4     211        60 25.8
## Hawaii            5.3      46        83 20.2
## Idaho             2.6     120        54 14.2
## Illinois         10.4     249        83 24.0
## Indiana           7.2     113        65 21.0
## Iowa              2.2      56        57 11.3
## Kansas            6.0     115        66 18.0
## Kentucky          9.7     109        52 16.3
## Louisiana        15.4     249        66 22.2
## Maine             2.1      83        51  7.8
## Maryland         11.3     300        67 27.8
## Massachusetts     4.4     149        85 16.3
## Michigan         12.1     255        74 35.1
## Minnesota         2.7      72        66 14.9
## Mississippi      16.1     259        44 17.1
## Missouri          9.0     178        70 28.2
## Montana           6.0     109        53 16.4
## Nebraska          4.3     102        62 16.5
## Nevada           12.2     252        81 46.0
## New Hampshire     2.1      57        56  9.5
## New Jersey        7.4     159        89 18.8
## New Mexico       11.4     285        70 32.1
## New York         11.1     254        86 26.1
## North Carolina   13.0     337        45 16.1
## North Dakota      0.8      45        44  7.3
## Ohio              7.3     120        75 21.4
## Oklahoma          6.6     151        68 20.0
## Oregon            4.9     159        67 29.3
## Pennsylvania      6.3     106        72 14.9
## Rhode Island      3.4     174        87  8.3
## South Carolina   14.4     279        48 22.5
## South Dakota      3.8      86        45 12.8
## Tennessee        13.2     188        59 26.9
## Texas            12.7     201        80 25.5
## Utah              3.2     120        80 22.9
## Vermont           2.2      48        32 11.2
## Virginia          8.5     156        63 20.7
## Washington        4.0     145        73 26.2
## West Virginia     5.7      81        39  9.3
## Wisconsin         2.6      53        66 10.8
## Wyoming           6.8     161        60 15.6
```

```r
#SVD rank reduction theorem
a <- matrix(0, nrow(USArrests) ,length(USArrests))
for (i in 1:4){
  a <- a + D[i] * U[,i] %*% t(V[,i])
}

head(a)
```

```
##      [,1] [,2] [,3] [,4]
## [1,] 13.2  236   58 21.2
## [2,] 10.0  263   48 44.5
## [3,]  8.1  294   80 31.0
```

```
## [4,]  8.8  190   50 19.5
## [5,]  9.0  276   91 40.6
## [6,]  7.9  204   78 38.7
```

```r
head(USArrests)
```

```
##            Murder Assault UrbanPop Rape
## Alabama      13.2     236       58 21.2
## Alaska       10.0     263       48 44.5
## Arizona       8.1     294       80 31.0
## Arkansas      8.8     190       50 19.5
## California    9.0     276       91 40.6
## Colorado      7.9     204       78 38.7
```

```r
#Define MA
MA <- USArrests$Murder + USArrests$Assault
head(MA)
```

```
## [1] 249.2 273.0 302.1 198.8 285.0 211.9
```

```r
#Define Arrests2
Arrests2 <- as.data.frame(cbind(USArrests, MA))
head(Arrests2)
```

```
##            Murder Assault UrbanPop Rape    MA
## Alabama      13.2     236       58 21.2 249.2
## Alaska       10.0     263       48 44.5 273.0
## Arizona       8.1     294       80 31.0 302.1
## Arkansas      8.8     190       50 19.5 198.8
## California    9.0     276       91 40.6 285.0
## Colorado      7.9     204       78 38.7 211.9
```

```r
#Compute SVD of it.
SVD2 <- svd(Arrests2)

SVD2$d
```

```
## [1] 1.993780e+03 2.003163e+02 4.566134e+01 2.166518e+01 2.140724e-13
```

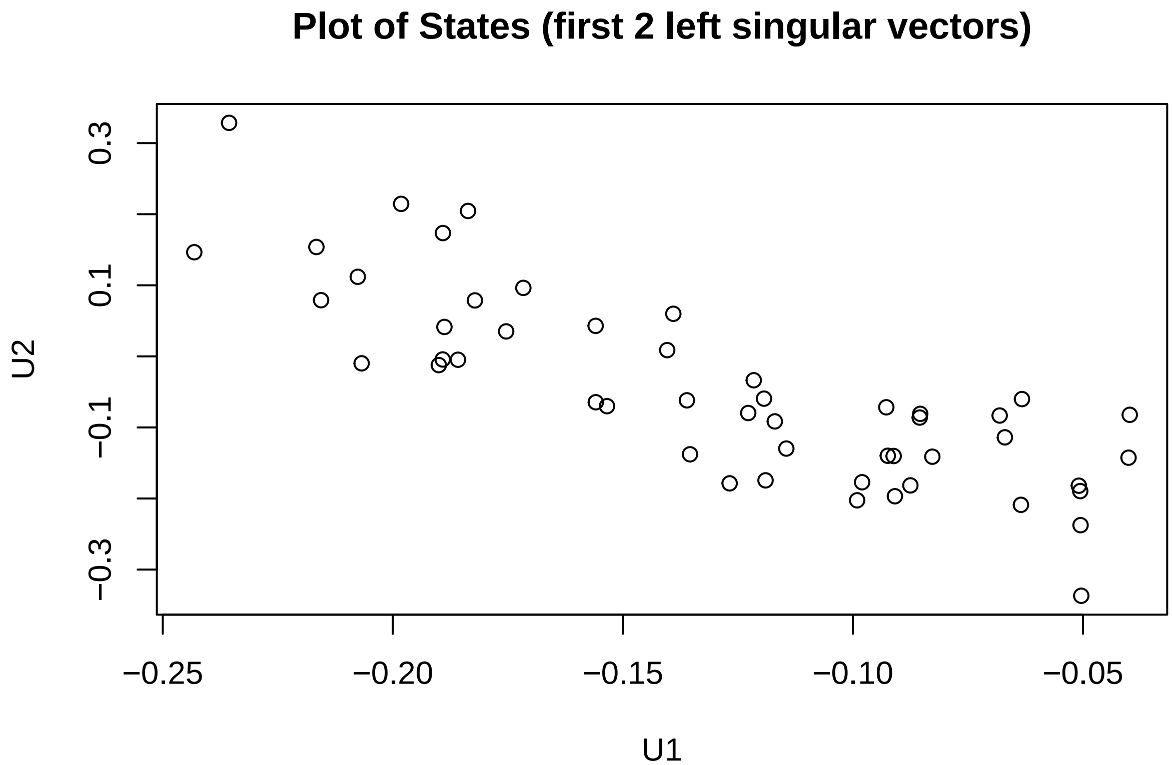
```r
SVD$d
```

```
## [1] 1419.06140  194.82585   45.66134   18.06956
```

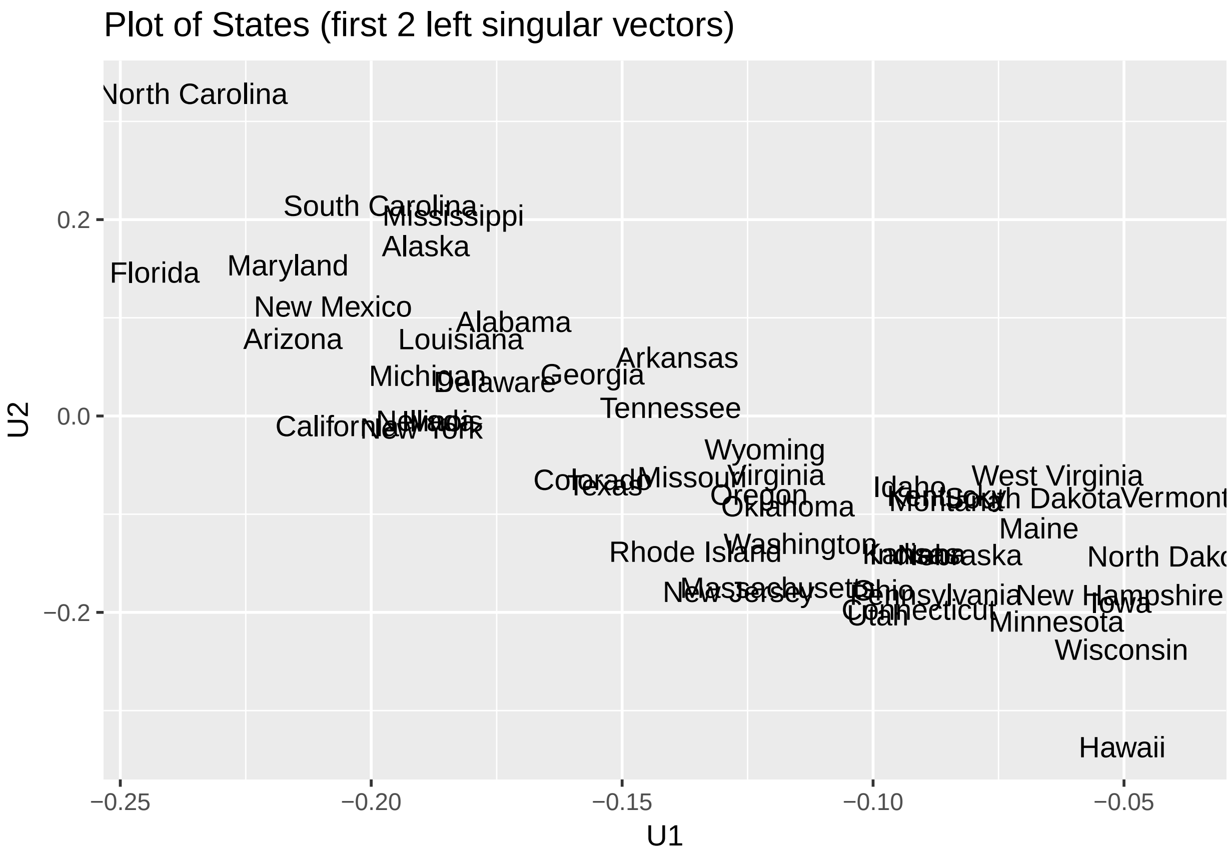
```r
#Q. signular values of Arrests2 are way smaller than USArrests'....???? ==> They are close!! Since the





library(Matrix)
rankMatrix(Arrests2)[1] #dimension of Arrests2
```

```
## [1] 4
```

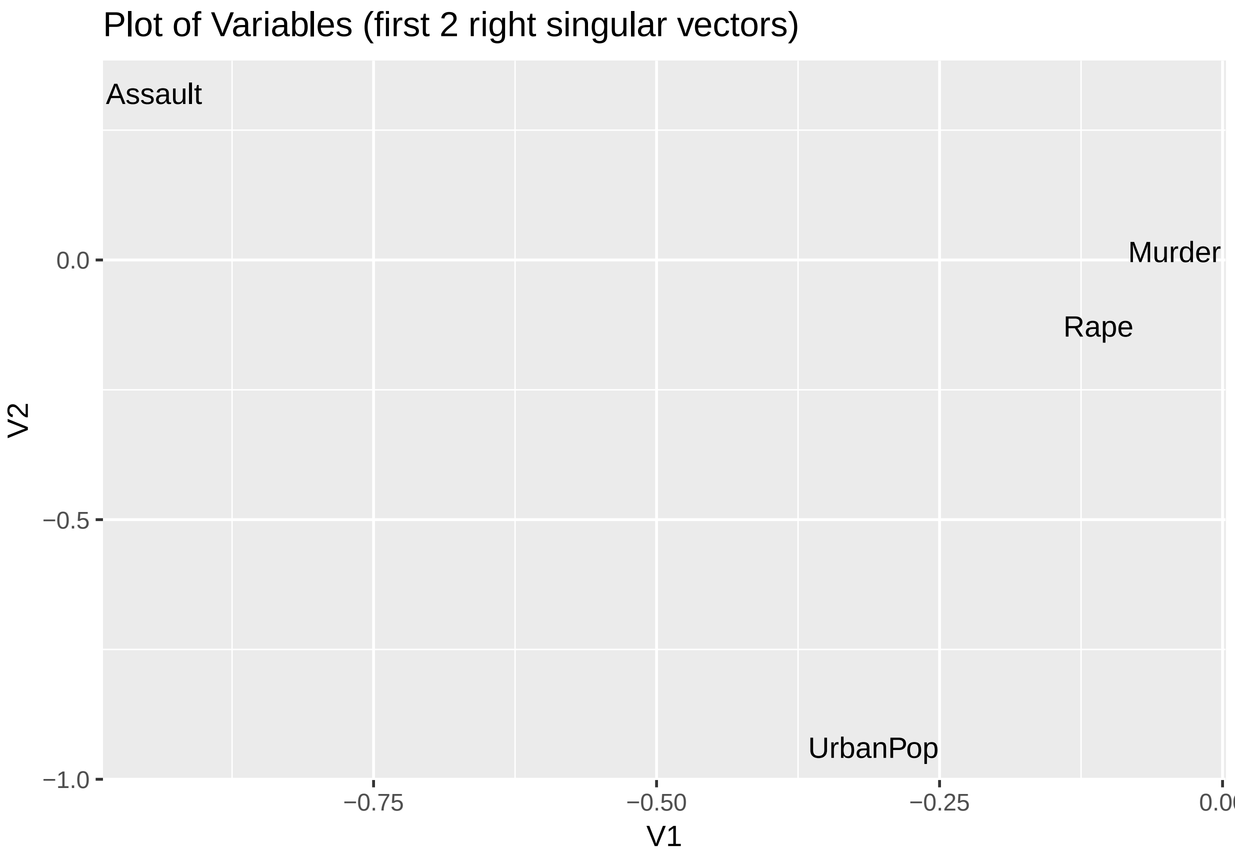
```r
plot(U[,1], U[,2], xlab = "U1", ylab = "U2", main = "Plot of States (first 2 left singular vectors)")
```




```r
library(ggplot2)
datas <- as.data.frame(cbind(U[,1], U[,2]))#ggplot does not read matrix. They can read data.frame.
rownames(datas) <- rownames(USArrests)
two_left <- ggplot(datas, aes(x = V1, y = V2, label = rownames(datas))) + geom_text() + labs(title = "Pl
two_left
```




```
datas2 <- as.data.frame(cbind(V[,1], V[,2]))#ggplot does not read matrix. They can read data.frame.
rownames(datas2) <- colnames(USArrests)
two_right <- ggplot(datas2, aes(x = V1, y = V2, label = rownames(datas2))) + geom_text() + labs(title =
two_right
```




```r
#Q. how to interpret these two left and right signular vectors? ==> need prior knowledge
```

```r
#Correlation and covariance matrix will give min(n,p) square matrix.
R <- cor(USArrests)
R
```

```
##               Murder  Assault   UrbanPop      Rape
## Murder    1.00000000 0.8018733 0.06957262 0.5635788
## Assault   0.80187331 1.0000000 0.25887170 0.6652412
## UrbanPop  0.06957262 0.2588717 1.00000000 0.4113412
## Rape      0.56357883 0.6652412 0.41134124 1.0000000
```

```r
evd <- eigen(R)
evd
```

```
## $values
## [1] 2.4802416 0.9897652 0.3565632 0.1734301
##
## $vectors
##            [,1]       [,2]       [,3]        [,4]
## [1,] -0.5358995  0.4181809 -0.3412327  0.64922780
## [2,] -0.5831836  0.1879856 -0.2681484 -0.74340748
## [3,] -0.2781909 -0.8728062 -0.3780158  0.13387773
## [4,] -0.5434321 -0.1673186  0.8177779  0.08902432
```

```r
x<- scale(R, T, T)
(1/(ncol(x)-1)) * t(x) %*% x
```

```
##               Murder    Assault   UrbanPop       Rape
```

```
## Murder    1.0000000  0.8935130 -0.9934269  0.2729914
## Assault   0.8935130  1.0000000 -0.8957584  0.3819624
## UrbanPop -0.9934269 -0.8957584  1.0000000 -0.3797916
## Rape      0.2729914  0.3819624 -0.3797916  1.0000000
```

```r
evd2 <- eigen(R, symmetric = T) #Q. What does symmetric really do....? ==> if TRUE, the matrix is assum
eigenvalues <- eigen(R, T, T)
eigenvalues
```

```
## $values
## [1] 2.4802416 0.9897652 0.3565632 0.1734301
##
## $vectors
## NULL
```

```r
X <- scale(USArrests, T, F)
X1 <- sweep(USArrests, 2, apply(USArrests, 2, mean), "-")




#Q. Sum of square matrix = cross-product matrix ??? What is sum of square matrix? ==> They are the same
S <- crossprod(X)




cov(X)
```

```
##              Murder   Assault   UrbanPop      Rape
## Murder    18.970465  291.0624   4.386204  22.99141
## Assault  291.062367 6945.1657 312.275102 519.26906
## UrbanPop   4.386204  312.2751 209.518776  55.76808
## Rape      22.991412  519.2691  55.768082  87.72916
```

```r
(1/(nrow(X) - 1)) * S
```

```
##              Murder   Assault   UrbanPop      Rape
## Murder    18.970465  291.0624   4.386204  22.99141
## Assault  291.062367 6945.1657 312.275102 519.26906
## UrbanPop   4.386204  312.2751 209.518776  55.76808
## Rape      22.991412  519.2691  55.768082  87.72916
```

```r
#Double check
scaler1 <- round((cov(X) /S), 3)
scaler2 <- round(1/(nrow(X) - 1), 3)
scaler1
```

```
##          Murder Assault UrbanPop Rape
## Murder     0.02    0.02     0.02 0.02
## Assault    0.02    0.02     0.02 0.02
## UrbanPop   0.02    0.02     0.02 0.02
## Rape       0.02    0.02     0.02 0.02
```

```r
scaler2
```

```
## [1] 0.02
```

```r
solve(S)
```

```
##                  Murder      Assault      UrbanPop          Rape
## Murder    0.0032804923 -1.304887e-04  1.794220e-04 -2.014203e-04
## Assault  -0.0001304887  1.046419e-05 -6.597122e-06 -2.354634e-05
## UrbanPop  0.0001794220 -6.597122e-06  1.271111e-04 -8.877578e-05
## Rape     -0.0002014203 -2.354634e-05 -8.877578e-05  4.812179e-04
```

```
eigen(S)$vectors %*% solve(diag(eigen(S)$values)) %*% t(eigen(S)$vectors)
```

```
##                [,1]          [,2]          [,3]          [,4]
## [1,]  0.0032804923 -1.304887e-04  1.794220e-04 -2.014203e-04
## [2,] -0.0001304887  1.046419e-05 -6.597122e-06 -2.354634e-05
## [3,]  0.0001794220 -6.597122e-06  1.271111e-04 -8.877578e-05
## [4,] -0.0002014203 -2.354634e-05 -8.877578e-05  4.812179e-04
```

```
eigen(S)
```

```
## $values
## [1] 343544.6277   9897.6259   2063.5199    302.0481
##
## $vectors
##             [,1]        [,2]        [,3]        [,4]
## [1,] -0.04170432  0.04482166  0.07989066  0.99492173
## [2,] -0.99522128  0.05876003 -0.06756974 -0.03893830
## [3,] -0.04633575 -0.97685748 -0.20054629  0.05816914
## [4,] -0.07515550 -0.20071807  0.97408059 -0.07232502
```

```
eigen(solve(S))
```

```
## $values
## [1] 3.310731e-03 4.846089e-04 1.010343e-04 2.910830e-06
##
## $vectors
##             [,1]        [,2]        [,3]        [,4]
## [1,]  0.99492173  0.07989066 -0.04482166 -0.04170432
## [2,] -0.03893830 -0.06756974 -0.05876003 -0.99522128
## [3,]  0.05816914 -0.20054629  0.97685748 -0.04633575
## [4,] -0.07232502  0.97408059  0.20071807 -0.07515550
```

```
#Q. what is the relationship between these two? ==> Eigen vectors are the same and eigen values are inv

svd(S) #compare with eigen(S)!!!
```

```
## $d
## [1] 343544.6277   9897.6259   2063.5199    302.0481
##
## $u
##             [,1]        [,2]        [,3]        [,4]
## [1,] -0.04170432  0.04482166 -0.07989066 -0.99492173
## [2,] -0.99522128  0.05876003  0.06756974  0.03893830
## [3,] -0.04633575 -0.97685748  0.20054629 -0.05816914
## [4,] -0.07515550 -0.20071807 -0.97408059  0.07232502
##
## $v
##             [,1]        [,2]        [,3]        [,4]
## [1,] -0.04170432  0.04482166 -0.07989066 -0.99492173
## [2,] -0.99522128  0.05876003  0.06756974  0.03893830
## [3,] -0.04633575 -0.97685748  0.20054629 -0.05816914
```

```
## [4,] -0.07515550 -0.20071807 -0.97408059  0.07232502
```

```
vnorm <- function(x){
  as.numeric(sqrt(t(x) %*% x))
}

#Using L_2 norm
power_method2 <- function(mat, init) {
  for (i in 1:150){
    init <- mat %*% init
    init <- init / as.numeric(vnorm(init))
  }
  return(init)
}


A <- matrix(c(2,1,-12,-5), 2, 2)
v0 <- c(1,1)
first_evector <- power_method2(A, v0)






Reyleigh <- function(mat, vec){
  lam <- crossprod((mat %*% vec), vec)
  lam <- lam / (vnorm(vec))^2
  return(lam)
}

first_evalue <- Reyleigh(A, first_evector)






#https://math.stackexchange.com/questions/768882/power-method-for-finding-all-eigenvectors
# Q. Why this is not working....???? Deflating function not working for some reasons.... ?  ==> Now it
deflating <- function(mat, my_vec, e_value){
  new_mat <- mat -  as.numeric(e_value / vnorm(my_vec)^2) * (tcrossprod(my_vec))
  return(new_mat)
}

A2 <- deflating(A, first_evector, first_evalue)

second_evector <- power_method2(A2, v0)
#eigen(A2)

second_evalue <- Reyleigh(A2, second_evector)

#Q. So, why can I not get the right second dominant eigen vector...??????
```

Check iteration

v_old <- v0 for (k in 1:4){ v_new <- A %*% v_old print(paste(“iteration =”, k)) print(v_new) v_old <-

v_new }

```
#scale it with L_inf norm.

A <- matrix(c(5,-4,3,-14,4,6,11,-4,-3),3,3)
v0 <- c(1,0,0)

#Q. Why not work in other norms???? ==> divide by the norms.

#Using L_inf norm
power_method_inf <- function(mat, init) {
  for (i in 1:150){
    init <- mat %*% init
    init <- init / max(abs(x))
  }
  return(init)
}

#Q. I have a wrong e_vector, but got the right e_value.... ??? why???... ==> Because eigen() function g
vec1_inf <- power_method_inf(A,v0)
vec1_inf <- vec1_inf / as.numeric(vnorm(vec1_inf)) #Always normalize with L2 norm...
#or
vec1_inf <- vec1_inf / as.numeric(max(abs(vec1_inf))) #Not work!!


#Q. Why do we always normalize with euclidean, even though we apply power method with other Lp norm????

e <- Reyleigh(A, vec1_inf)
```

```
#Using L_p norm

#Q. Why not work in other norms???? ==> same above.
power_methodp <- function(mat, init, p) {
  for (i in 1:150){
    init <- mat %*% init
    init <- init / ((sum(abs(x)^p))^(1/p))
  }
  return(init)
}
A <- matrix(c(5,-4,3,-14,4,6,11,-4,-3),3,3)
v0 <- c(1,0,0)


#Q. I have a wrong e_vector, but got the right e_value.... ??? why???... ==> No, it is not wrong. You n

value <- power_methodp(A,v0, 1) #First e-vector.
value <- value / as.numeric(vnorm(value))
#value <- value / as.numeric((sum(abs(x)^1))^(1/1)) Not work...

e <- Reyleigh(A, value)
```

```
#If there are more than one dominant eigenvalues, then what willl happen? it still converges to a e-vec
B <- diag(3)
init <- c(1,1,1)
power_method2(B, init)
```

```
##           [,1]
## [1,] 0.5773503
## [2,] 0.5773503
## [3,] 0.5773503
```

```r
C <- deflating(B, power_method2(B, init), Reyleigh(B, power_method2(B, init)))
power_method2(C, power_method2(B, init))
```

```
##           [,1]
## [1,] 0.5773503
## [2,] 0.5773503
## [3,] 0.5773503
```