Stat154_Lab2

Jiyoon Clover Jeong

9/8/2017

```r
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```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
SVD <- svd(USArrests)


U <- SVD$u
d <- SVD$d
D <- diag(d)
V <- SVD$v

head(U %*% 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
```

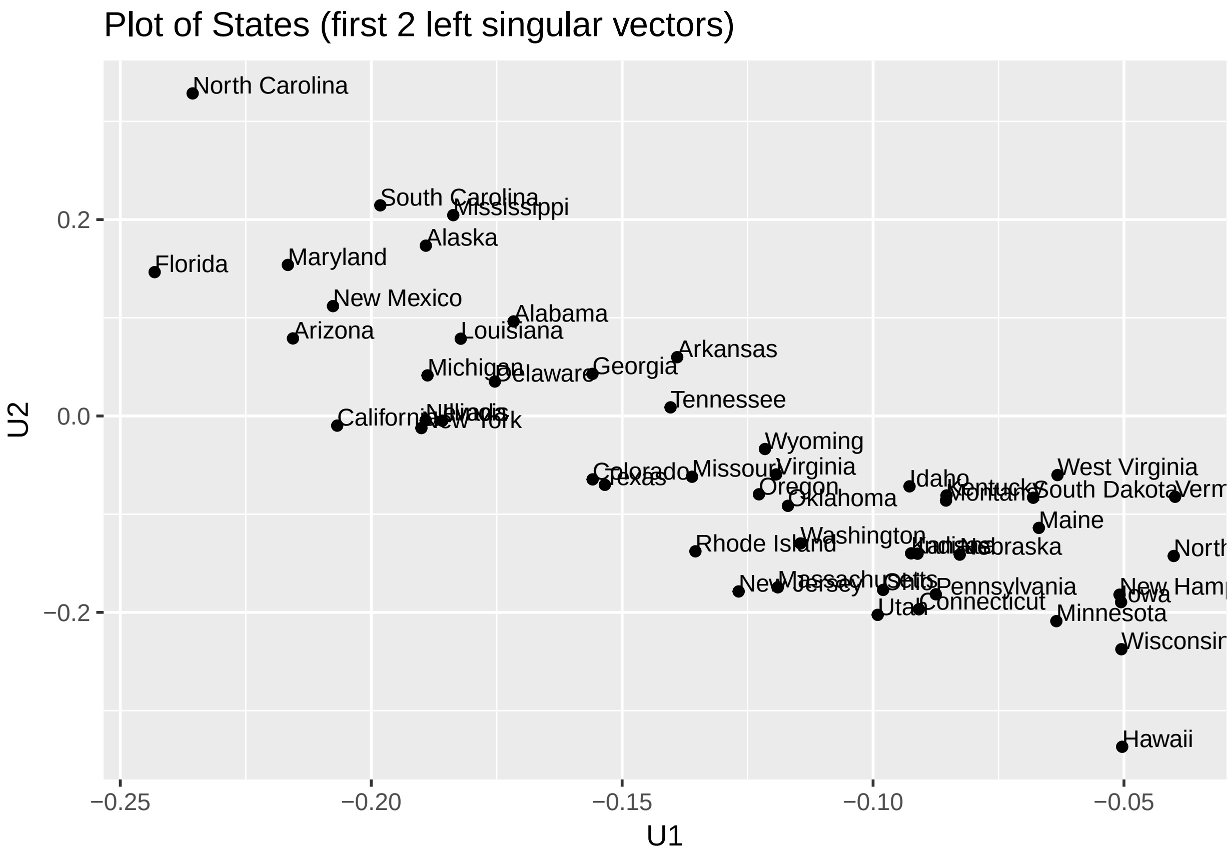
```r
head(USArrests)
```

```
##            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
```

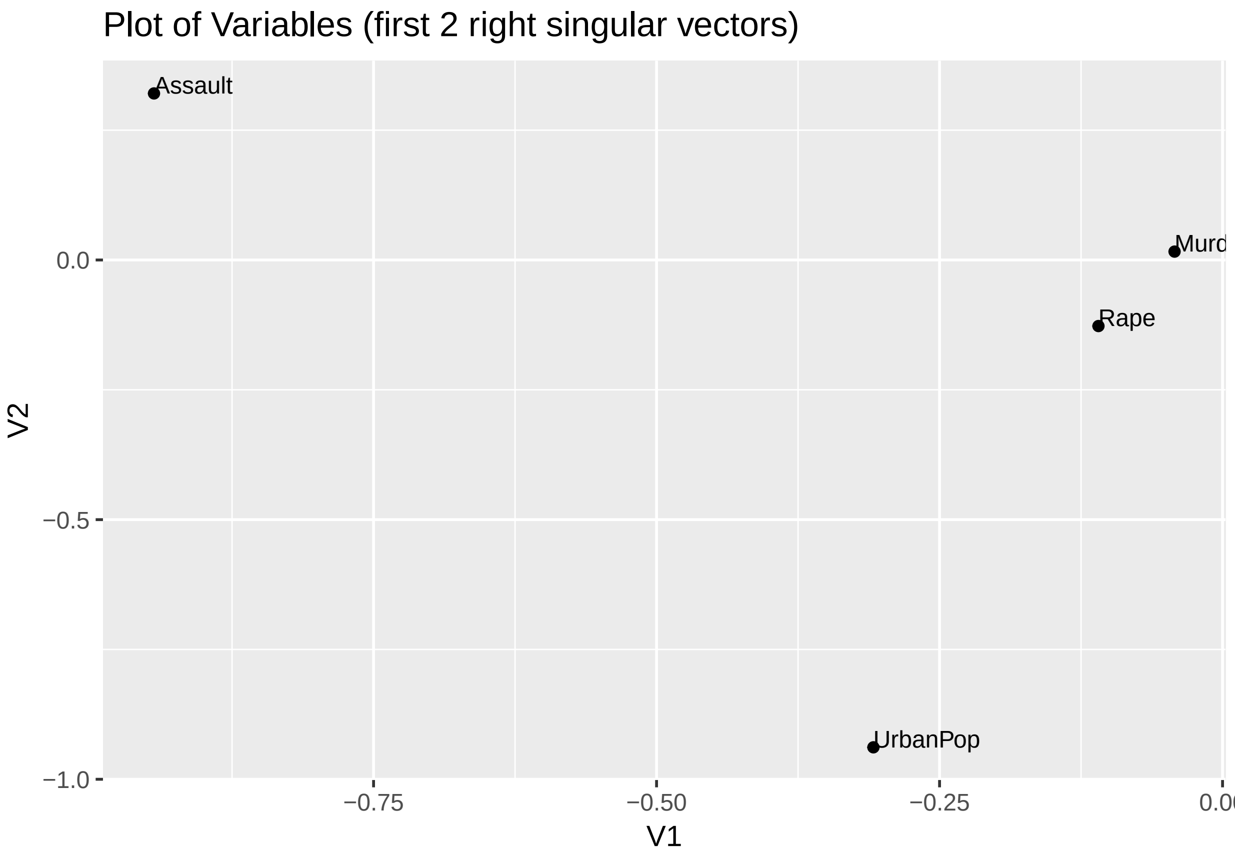
```r
sum <-matrix(0,50,4)


for(i in 1:4){
  sum = sum + d[i] * (U[,i, drop = F] %*% t(V[,i, drop = F]))
}


head(sum)
```

```
##      [,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
sum <-matrix(0,50,4)


for(i in 1:2){
  sum = sum + d[i] * (U[,i, drop = F] %*% t(V[,i, drop = F]))
}

U <- as.data.frame(U)


ggplot(U,aes(x =V1, y = V2)) + geom_point() + labs(x = "U1", y = "U2", title = "Plot of States (first 2
```




```r
V <- as.data.frame(V)

ggplot(V,aes(x =V1, y = V2)) + geom_point() + labs(x = "V1", y = "V2", title = "Plot of Variables (firs
```




```r
R <- cor(USArrests)
evd <- eigen(R, symmetric = T)

eigenvalues <- eigen(R, symmetric = TRUE, only.values = TRUE)
eigenvalues
```

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

```r
X <- as.matrix(scale(USArrests, scale = FALSE))
S <- t(X) %*% X
S
```

```
##               Murder   Assault  UrbanPop      Rape
## Murder      929.5528  14262.06   214.924  1126.579
## Assault   14262.0560 340313.12 15301.480 25444.184
## UrbanPop    214.9240  15301.48 10266.420  2732.636
## Rape       1126.5792  25444.18  2732.636  4298.729
```

```r
inverseS <- solve(S,diag(4))  # or solve(S)

eigens <- eigen(S)

anotherS <- eigens$vectors %*% diag(eigens$values) %*% t(eigens$vectors)

anotherS
```

```
##              [,1]      [,2]      [,3]      [,4]
## [1,]    929.5528  14262.06   214.924  1126.579
## [2,] 14262.0560 340313.12 15301.480 25444.184
## [3,]   214.9240  15301.48 10266.420  2732.636
## [4,]  1126.5792  25444.18  2732.636  4298.729
```

```r
inverseS
```

```
##                   [,1]          [,2]          [,3]          [,4]
## 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
```

```r
anotherinverseS <- eigens$vectors %*% solve(diag(eigens$values)) %*% t(eigens$vectors)

anotherinverseS
```

```
##                [,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
```

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

power1 <- function(n, initial, A){
  old <- initial
  for(i in 1:n){
    new <- A %*% old

    new <- new / new[which.max(abs(new))]
    old <- new
  }

  output <- list("eigenvector" = new,
                 "eigenvalue" = (A %*% new)[which.max(abs(A %*% new))])

  # "eigenvalue" = (t(new) %*% t(A) %*% new) / lpnorm(new,2)
  return(output)

}

eigenoutput <- power1(55,initial,A)
eigenoutput$eigenvector
```

```
##                [,1]
## [1,]  1.000000e+00
## [2,] -5.000000e-01
## [3,] -8.963641e-17
```

```r
eigenoutput$eigenvalue
```

```
## [1] 12
```

```r
eigen(A)
```

```
## $values
## [1]  1.200000e+01 -6.000000e+00  4.930713e-16
##
## $vectors
##                [,1]          [,2]       [,3]
## [1,] -8.944272e-01  7.071068e-01 -0.2672612
## [2,]  4.472136e-01  1.040834e-16  0.5345225
## [3,] -5.945103e-17 -7.071068e-01  0.8017837
```

```
lpnorm <- function(v,p) {
  sum = 0
  for(i in 1:length(v)){
    sum = sum + abs(v[i])^p
  }
  return(sum^(1/p))
}

power2 <- function(n, initial, A){
  old <- initial
  for(i in 1:n){
    new <- A %*% old

    new <- new / lpnorm(new,2)
    old <- new
  }

  output <- list("eigenvector" = new, "eigenvalue" = (t(new) %*% t(A) %*% new) / lpnorm(new,2))

  # "eigenvalue" = (t(new) %*% t(A) %*% new) / lpnorm(new,2)
  # "eigenvalue" = t(new) %*% (A %*% new)
  return(output)

}

eigenoutput <- power2(55,initial,A)
firsteigenvector<- eigenoutput$eigenvector
firsteigenvector
```

```
##                [,1]
## [1,]  8.944272e-01
## [2,] -4.472136e-01
## [3,] -5.960422e-17
```

```
firsteigenvalue<-eigenoutput$eigenvalue
firsteigenvalue
```

```
##      [,1]
## [1,]   12
```

```
eigen(A)
```

```
## $values
## [1]  1.200000e+01 -6.000000e+00  4.930713e-16
##
## $vectors
##                [,1]          [,2]       [,3]
```

```
## [1,] -8.944272e-01  7.071068e-01 -0.2672612
## [2,]  4.472136e-01  1.040834e-16  0.5345225
## [3,] -5.945103e-17 -7.071068e-01  0.8017837
```

```r
deflate <- A - firsteigenvalue[1] * (firsteigenvector) %*% t(firsteigenvector)
eigenoutput <- power2(100,initial,deflate)
secondeigenvector<- eigenoutput$eigenvector
secondeigenvector
```

```
##            [,1]
## [1,] -0.8970852
## [2,]  0.2760262
## [3,]  0.3450328
```

```r
secondeigenvalue<-eigenoutput$eigenvalue
secondeigenvalue
```

```
##      [,1]
## [1,]   -6
```

```r
eigen(deflate)
```

```
## $values
## [1] -6e+00+0.000000e+00i  0e+00+9.589313e-08i  0e+00-9.589313e-08i
##
## $vectors
##               [,1]                      [,2]                      [,3]
## [1,]  0.8970852+0i  8.944272e-01+0.000000e+00i  8.944272e-01+0.000000e+00i
## [2,] -0.2760262+0i -4.472136e-01+0.000000e+00i -4.472136e-01-0.000000e+00i
## [3,] -0.3450328+0i  0.000000e+00+1.340147e-08i  0.000000e+00-1.340147e-08i
```

2 by 2

```r
A <- matrix(c(2,1,-12,-5), 2, 2)
initial <- c(1,1)

eigenoutput <- power2(55,initial,A)
firsteigenvector<- eigenoutput$eigenvector
firsteigenvector
```

```
##            [,1]
## [1,] -0.9486833
## [2,] -0.3162278
```

```r
firsteigenvalue<-eigenoutput$eigenvalue
firsteigenvalue
```

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

```r
eigen(A)
```

```
## $values
## [1] -2 -1
##
## $vectors
##           [,1]       [,2]
```

```
## [1,] 0.9486833 0.9701425
## [2,] 0.3162278 0.2425356
```

```
deflate <- A - firsteigenvalue[1] * (firsteigenvector) %*% t(firsteigenvector)
eigenoutput <- power2(100,initial,deflate)
secondeigenvector<- eigenoutput$eigenvector
secondeigenvector
```

```
##           [,1]
## [1,] 0.9216354
## [2,] 0.3880570
```

```
secondeigenvalue<-eigenoutput$eigenvalue
secondeigenvalue
```

```
##      [,1]
## [1,]   -1
```

```
eigen(deflate)
```

```
## $values
## [1] -1  0
##
## $vectors
##           [,1]      [,2]
## [1,] 0.9216354 0.9486833
## [2,] 0.3880570 0.3162278
```