Final Stat 5311

Arkaprava Hajra

2023-05-10

R Markdown

Problem 1 : Consider the mtcars data data from the CCA package

i. Determine appropriate number of significant canonical dimensions or pairs. Interpret first two loadings for the driver feature.

```
library(dbplyr)
```

```
## Warning: package 'dbplyr' was built under R version 4.2.3
```

```
library(CCA)
```

```
## Warning: package 'CCA' was built under R version 4.2.3
```

```
## Loading required package: fda
```

```
## Warning: package 'fda' was built under R version 4.2.3
```

```
## Loading required package: splines
```

```
## Loading required package: fds
```

```
## Warning: package 'fds' was built under R version 4.2.3
```

```
## Loading required package: rainbow
```

```
## Warning: package 'rainbow' was built under R version 4.2.3
```

```
## Loading required package: MASS
```

```
## Loading required package: pcaPP
```

```
## Warning: package 'pcaPP' was built under R version 4.2.3
```

```
## Loading required package: RCurl
```

```
## Warning: package 'RCurl' was built under R version 4.2.3
```

```
## Loading required package: deSolve
```

```
## Warning: package 'deSolve' was built under R version 4.2.3
```

```
##
## Attaching package: 'fda'
```

```
## The following object is masked from 'package:graphics':
##
##     matplot
```

```
## Loading required package: fields
```

```
## Warning: package 'fields' was built under R version 4.2.3
```

```
## Loading required package: spam
```

```
## Warning: package 'spam' was built under R version 4.2.3
```

```
## Spam version 2.9-1 (2022-08-07) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
```

```
##
## Attaching package: 'spam'
```

```
## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve
```

```
## Loading required package: viridis
```

```
## Warning: package 'viridis' was built under R version 4.2.3
```

```
## Loading required package: viridisLite
```

```
##
## Try help(fields) to get started.
```

```
library(CCP)
data("mtcars")
head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
summary(mtcars)
```

```
##       mpg             cyl             disp             hp       
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0  
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5  
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0  
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7  
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0  
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0  
##       drat             wt             qsec             vs        
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000  
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000  
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000  
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375  
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000  
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000  
##        am              gear            carb      
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000  
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000  
##  Median :0.0000   Median :4.000   Median :2.000  
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812  
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000  
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

```
user <- c(2,3,5,8,10,11)
driver <- c(1,4,6,7,9)
mtcars[1,user]
```

```
##           cyl disp drat vs gear carb
## Mazda RX4   6  160  3.9  0    4    4
```

```
mtcars[1, driver]
```

```
##           mpg  hp   wt  qsec am
## Mazda RX4  21 110 2.62 16.46  1
```

From below,The correlation of the first canonical pair is 0.985 and second canonical pair is 0.84715~0.85 which indicates a high correlation between these canonical score pairs. There are a total of 5 pairs.

```
cca = cancor(mtcars[,user],mtcars[,driver])
cca$cor                                  # correlation between canonical pairs
```

```
## [1] 0.9850787 0.8471577 0.5796657 0.4137175 0.2548956
```

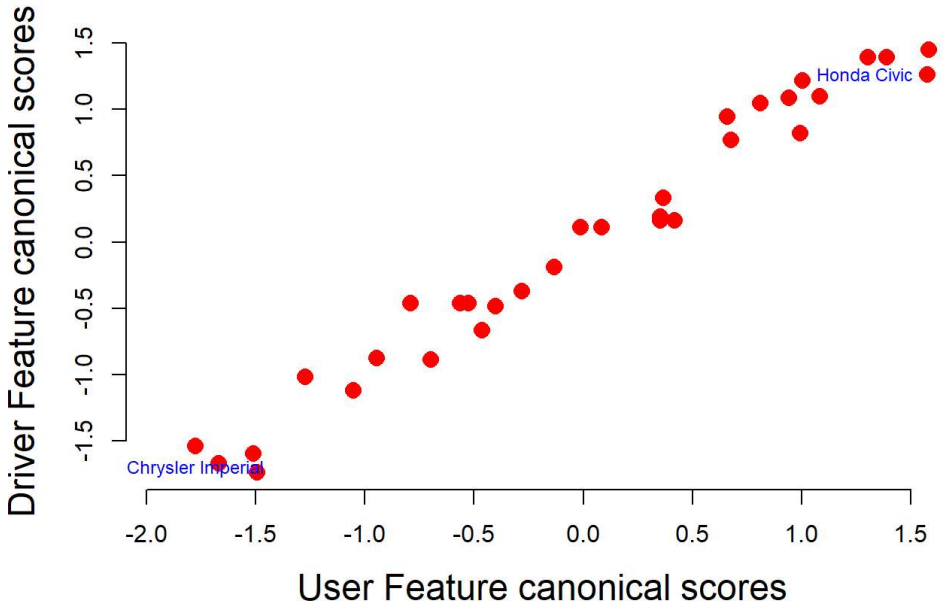
```
cca$cor[2] # correlation = 0.8471577
```

```
## [1] 0.8471577
```

```
library(CCA)
ccs <- cc(mtcars[ , user], mtcars[ , driver])
usercc1  <- ccs$scores$xscores[ , 1]
drivercc1 <- ccs$scores$yscores[ , 1]


sdr <- sort(drivercc1)
sdr <- sdr[c(1, length(sdr) - 1)] # first and next-to-last
ext <- match(sdr, drivercc1)

plot( drivercc1, usercc1, cex.lab = 1.5,
      xlab = "User Feature canonical scores",
      ylab = "Driver Feature canonical scores",
      pch = 16, cex = 1.5, col = "red",
      xlim = sdr * c(1.1, 1), frame.plot=FALSE)
text(drivercc1[ext], usercc1[ext],
     labels = rownames(mtcars)[ext],
     pos = c(1, 2), cex = .75, col = "blue")
```



The first canonical pair/scores plot

shows that the first canonical pair correlation between the user variables and driver variables is very high (0.9850787). The Chrysler Imperial and Honda Civic are on opposite ends indicating that these two cars are very different and belong to two different groups of car with Chrysler Imperial being a luxury car whereas Honda Civic an economy car.

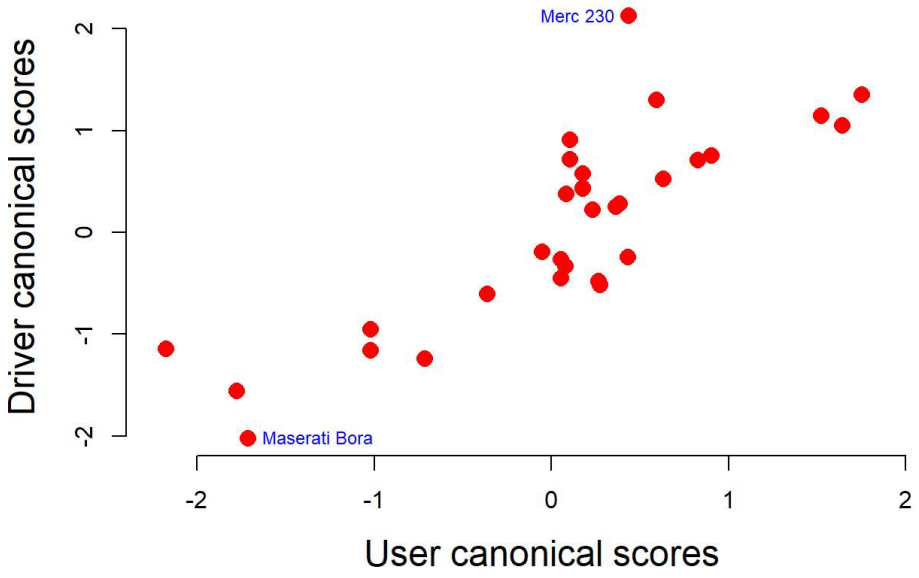
```
usercc2  <- ccs$scores$xscores[ , 2]
drivercc2  <- ccs$scores$yscores[ , 2]
sdr2 <- sort(drivercc2)
sdr2 <- sdr2[c(1, length(sdr2))] # first and next-to-last
sdr2
```

```
## Maserati Bora      Merc 230
##     -2.023857      2.122882
```

```
ext2 <- match(sdr2, drivercc2)
ext2
```

```
## [1] 31  9
```

```
plot( usercc2, drivercc2, cex.lab = 1.5,
      xlab = "User canonical scores",
      ylab = "Driver canonical scores",
      pch = 16, cex = 1.5, col = "red",
      xlim = sdr2 * c(1.1, 1), frame.plot=FALSE)
text(usercc2[ext2], drivercc2[ext2],
     labels = rownames(mtcars)[ext2],
     pos = c(4, 2), cex = .75, col = "blue")
```



The above is for the second canonical pair. We can see a pattern of luxury cars like Merc 230 at top right while sports/exotic cars near the bottom left.

ii. Using comput from CCA package appropriately for mtcars data.

PCA using correlation for driver for the features.

```
pc <- princomp(mtcars,cor=TRUE);pc
```

```
## Call:
## princomp(x = mtcars, cor = TRUE)
##
## Standard deviations:
##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7    Comp.8
## 2.5706809 1.6280258 0.7919579 0.5192277 0.4727061 0.4599958 0.3677798 0.3505730
##    Comp.9   Comp.10   Comp.11
## 0.2775728 0.2281128 0.1484736
##
##  11  variables and  32 observations.
```

```
summary(pc)
```

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3     Comp.4     Comp.5
## Standard deviation     2.5706809 1.6280258 0.79195787 0.51922773 0.47270615
## Proportion of Variance 0.6007637 0.2409516 0.05701793 0.02450886 0.02031374
## Cumulative Proportion  0.6007637 0.8417153 0.89873322 0.92324208 0.94355581
##                           Comp.6     Comp.7     Comp.8      Comp.9     Comp.10
## Standard deviation     0.45999578 0.36777981 0.35057301 0.277572792 0.228112781
## Proportion of Variance 0.01923601 0.01229654 0.01117286 0.007004241 0.004730495
## Cumulative Proportion  0.96279183 0.97508837 0.98626123 0.993265468 0.997995963
##                           Comp.11
## Standard deviation     0.148473587
## Proportion of Variance 0.002004037
## Cumulative Proportion  1.000000000
```

```
pc <- princomp(mtcars,cor=TRUE); pc
```

```
## Call:
## princomp(x = mtcars, cor = TRUE)
##
## Standard deviations:
##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7    Comp.8
## 2.5706809 1.6280258 0.7919579 0.5192277 0.4727061 0.4599958 0.3677798 0.3505730
##    Comp.9   Comp.10   Comp.11
## 0.2775728 0.2281128 0.1484736
##
##  11  variables and  32 observations.
```

```
L=pc$loadings; L
```

```
##
## Loadings:
##      Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10
## mpg   0.363         0.226         0.103  0.109  0.368  0.754  0.236  0.139
## cyl  -0.374         0.175               -0.169         0.231        -0.846
## disp -0.368               -0.257  0.394  0.336  0.214         0.198
## hp   -0.330  0.249 -0.140         0.540                0.222 -0.576  0.248
## drat  0.294  0.275 -0.161 -0.855        -0.244                      -0.101
## wt   -0.346 -0.143 -0.342 -0.246         0.465                0.359
## qsec  0.200 -0.463 -0.403        -0.165  0.330         0.232 -0.528 -0.271
## vs    0.307 -0.232 -0.429  0.215  0.600 -0.194 -0.266         0.359 -0.159
## am    0.235  0.429  0.206                0.571 -0.587               -0.178
## gear  0.207  0.462 -0.290  0.265         0.244  0.605 -0.336        -0.214
## carb -0.214  0.414 -0.529  0.127 -0.361 -0.184 -0.175  0.396  0.171
##      Comp.11
## mpg   0.125
## cyl   0.141
## disp -0.661
## hp    0.256
## drat
## wt    0.567
## qsec -0.181
## vs
## am
## gear
## carb -0.320
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9
## SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000
## Proportion Var  0.091  0.091  0.091  0.091  0.091  0.091  0.091  0.091  0.091
## Cumulative Var  0.091  0.182  0.273  0.364  0.455  0.545  0.636  0.727  0.818
##                Comp.10 Comp.11
## SS loadings      1.000   1.000
## Proportion Var   0.091   0.091
## Cumulative Var   0.909   1.000
```

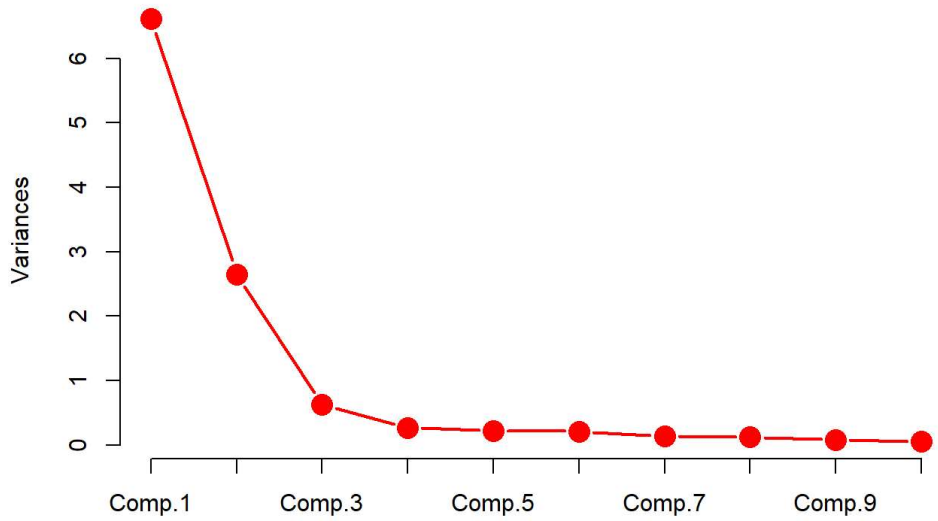
```
apply(L^2, 2, sum)
```

```
##  Comp.1  Comp.2  Comp.3  Comp.4  Comp.5  Comp.6  Comp.7  Comp.8  Comp.9 Comp.10
##       1       1       1       1       1       1       1       1       1       1
## Comp.11
##       1
```

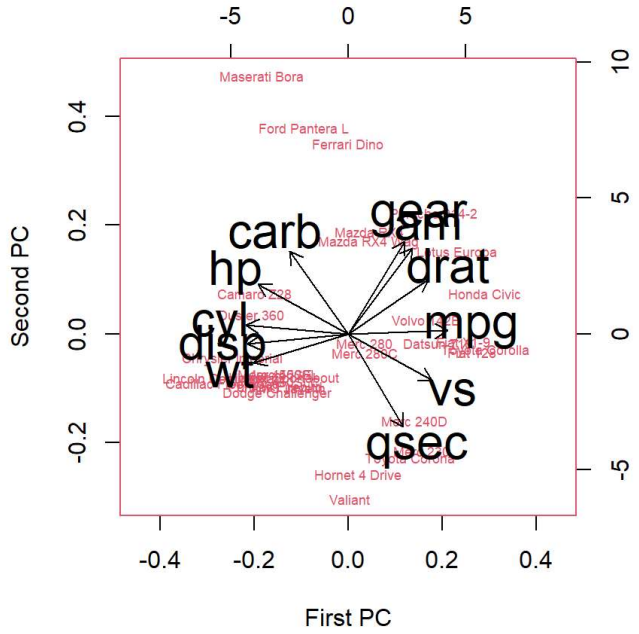
```
apply(L^2, 2, sum)/11
```

```
##     Comp.1     Comp.2     Comp.3     Comp.4     Comp.5     Comp.6     Comp.7
## 0.09090909 0.09090909 0.09090909 0.09090909 0.09090909 0.09090909 0.09090909
##     Comp.8     Comp.9    Comp.10    Comp.11
## 0.09090909 0.09090909 0.09090909 0.09090909
```

```
screeplot(pc, col = "red", pch = 16,
type = "lines", cex = 2, lwd = 2, main = "")
```




```
biplot(pc, col = c(2, 1), cex = c(.55, 2),
xlim = c( -.45, .45), xlab = "First PC", ylab = "Second PC")
```



The scree plot shown starts to form a straight line after the 3rd principal components suggesting that 3 PCAs are sufficient to explain the variability in the data. The first three PCs explain 89% of variance in data which is sufficient to reduce the 11-dimension dataset to 3 principal components.

Looking at bi-plot, gear,am,drat are negatively correlated to qsec,vs. mpg and cyl are negatively correlated. Gear and AM are positively correlated. Hp and Carb are positively correlated

1. What is correlation between disp and first 'driver' canonical values?

There is positive correlation b/w disp and first Driver.

2. What is correlation between hp and 1st design canonical values?

There is negative correlation b/w hp and first design.

3. What is correlation between first driver and second design canonical variate?

here is positive correlation b/w first driver and second design.

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
require(GGally)
```

```
## Loading required package: GGally
```

```
## Warning: package 'GGally' was built under R version 4.2.3
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(CCA)

summary(mtcars)
```

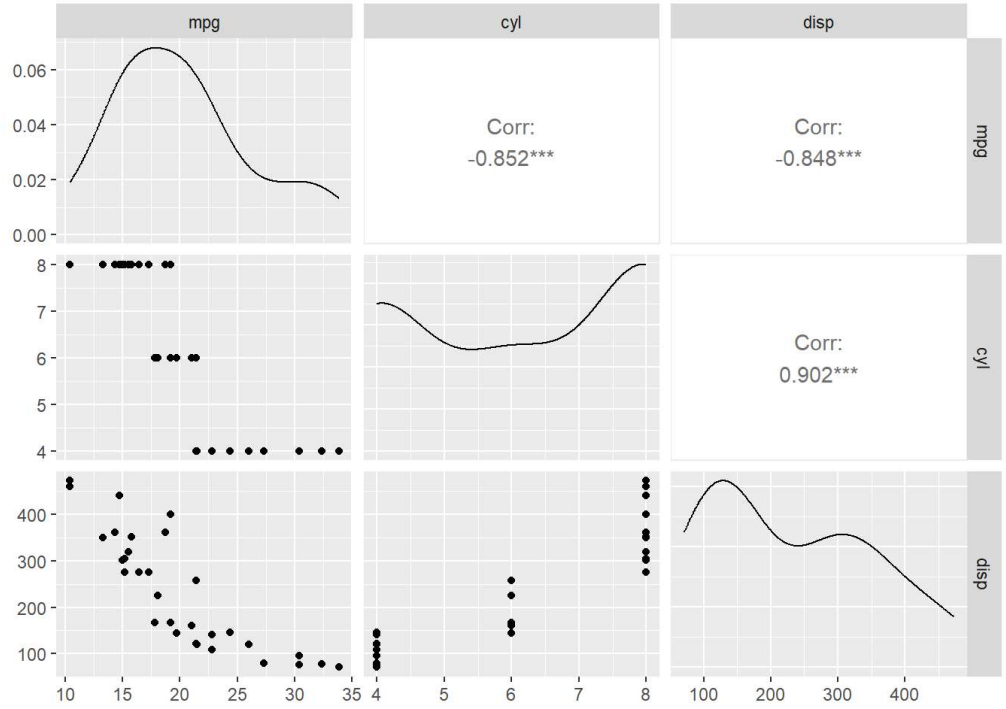
```
##       mpg             cyl             disp             hp       
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0  
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5  
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0  
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7  
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0  
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0  
##       drat             wt             qsec             vs        
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000  
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000  
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000  
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375  
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000  
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000  
##        am              gear            carb      
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000  
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000  
##  Median :0.0000   Median :4.000   Median :2.000  
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812  
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000  
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

```
xtabs(~disp, data = mtcars)
```

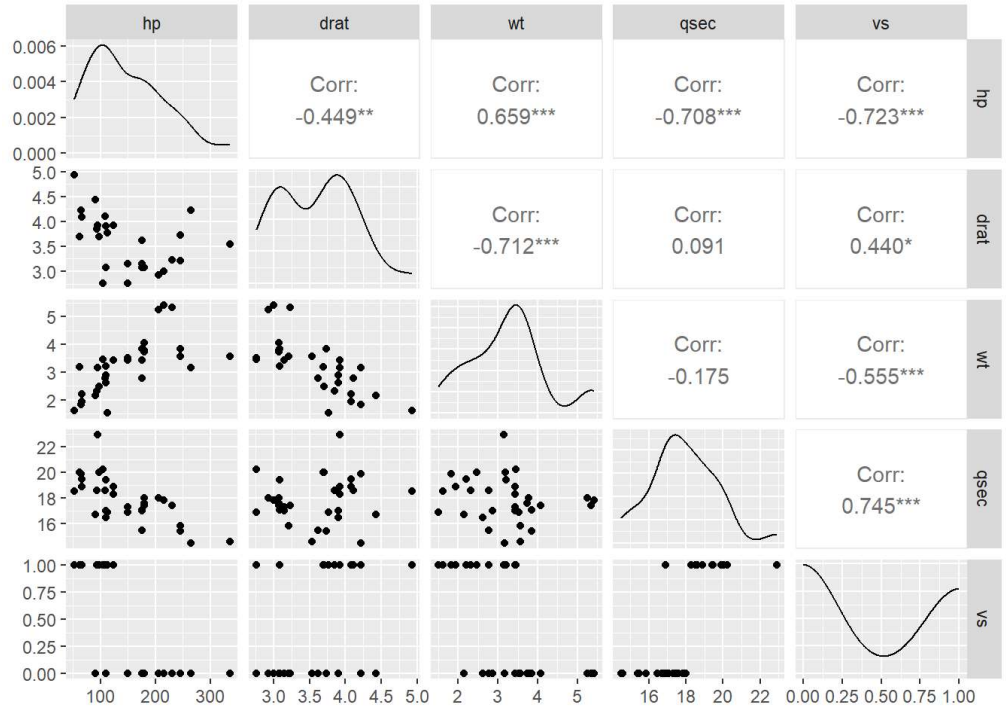
```
## disp
##  71.1  75.7  78.7    79  95.1   108 120.1 120.3   121 140.8   145 146.7   160 
##     1     1     1     1     1     1     1     1     1     1     1     1     2 
## 167.6   225   258 275.8   301   304   318   350   351   360   400   440   460 
##     2     1     1     3     1     1     1     1     1     2     1     1     1 
##   472 
##     1
```

```
ph <- mtcars[, 1:3]
ad <- mtcars[, 4:8]

ggpairs(ph)
```




```
ggpairs(ad)
```




```
matcor(ph, ad)
```

```
## $Xcor
##             mpg        cyl       disp
## mpg   1.0000000 -0.8521620 -0.8475514
## cyl  -0.8521620  1.0000000  0.9020329
## disp -0.8475514  0.9020329  1.0000000
##
## $Ycor
##              hp        drat         wt        qsec         vs
## hp    1.0000000 -0.44875912  0.6587479 -0.70822339 -0.7230967
## drat -0.4487591  1.00000000 -0.7124406  0.09120476  0.4402785
## wt    0.6587479 -0.71244065  1.0000000 -0.17471588 -0.5549157
## qsec -0.7082234  0.09120476 -0.1747159  1.00000000  0.7445354
## vs   -0.7230967  0.44027846 -0.5549157  0.74453544  1.0000000
##
## $XYcor
##             mpg        cyl       disp         hp        drat         wt
## mpg   1.0000000 -0.8521620 -0.8475514 -0.7761684  0.68117191 -0.8676594
## cyl  -0.8521620  1.0000000  0.9020329  0.8324475 -0.69993811  0.7824958
## disp -0.8475514  0.9020329  1.0000000  0.7909486 -0.71021393  0.8879799
## hp   -0.7761684  0.8324475  0.7909486  1.0000000 -0.44875912  0.6587479
## drat  0.6811719 -0.6999381 -0.7102139 -0.4487591  1.00000000 -0.7124406
## wt   -0.8676594  0.7824958  0.8879799  0.6587479 -0.71244065  1.0000000
## qsec  0.4186840 -0.5912421 -0.4336979 -0.7082234  0.09120476 -0.1747159
## vs    0.6640389 -0.8108118 -0.7104159 -0.7230967  0.44027846 -0.5549157
##             qsec         vs
## mpg   0.41868403  0.6640389
## cyl  -0.59124207 -0.8108118
## disp -0.43369788 -0.7104159
## hp   -0.70822339 -0.7230967
## drat  0.09120476  0.4402785
## wt   -0.17471588 -0.5549157
## qsec  1.00000000  0.7445354
## vs    0.74453544  1.0000000
```

```
cca <- cc(ph, ad)
cca$cor
```

```
## [1] 0.97410878 0.61952466 0.08324239
```

```
cca[3:4]
```

```
## $xcoef
##              [,1]       [,2]       [,3]
## mpg   0.056089039  0.1616678 -0.2919219
## cyl  -0.152658863  1.3906056  0.1182135
## disp -0.003517598 -0.0121093 -0.0154556
##
## $ycoef
##              [,1]         [,2]         [,3]
## hp   -0.003285095  0.002285289  0.006082531
## drat  0.381869198 -0.999362168 -0.136973444
## wt   -0.543813472 -1.312007985  0.291085189
## qsec  0.081048983 -0.117592032 -0.720731834
## vs    0.237019360 -1.373618299  3.387166439
```

```
cca2 <- comput(ph, ad, cca)
cca2[3:6]
```

```
## $corr.X.xscores
##            [,1]        [,2]        [,3]
## mpg   0.9398804  0.13002311 -0.31578276
## cyl  -0.9539626  0.29941612 -0.01747349
## disp -0.9684050 -0.08642427 -0.23392874
##
## $corr.Y.xscores
##            [,1]         [,2]         [,3]
## hp   -0.8341633  0.124059157  0.026236929
## drat  0.7307256 -0.008699105  0.014224078
## wt   -0.8937752 -0.234767633 -0.009207775
## qsec  0.4918064 -0.409508100 -0.030686609
## vs    0.7552502 -0.300445952  0.021347044
##
## $corr.X.yscores
##            [,1]        [,2]         [,3]
## mpg   0.9155458  0.08055252 -0.026286511
## cyl  -0.9292634  0.18549567 -0.001454535
## disp -0.9433318 -0.05354196 -0.019472786
##
## $corr.Y.yscores
##            [,1]        [,2]       [,3]
## hp   -0.8563348  0.20024894  0.3151871
## drat  0.7501479 -0.01404158  0.1708754
## wt   -0.9175312 -0.37894801 -0.1106140
## qsec  0.5048784 -0.66100372 -0.3686416
## vs    0.7753243 -0.48496206  0.2564444
```

PROBLEM 2

Given x = (x1,x2,x3)' with

sigma = (7 0 0 0 2 1 0 1 2)

i. Calculate correlation matrix R

```
mu = c(1,2,3);mu
```

```
## [1] 1 2 3
```

```
sigma <- matrix(c(7, 0, 0, 0, 2, 1, 0, 1, 2), nrow = 3, byrow = TRUE)
sigma
```

```
##      [,1] [,2] [,3]
## [1,]    7    0    0
## [2,]    0    2    1
## [3,]    0    1    2
```

```
mvndat=mvrnorm(n = 10, mu, sigma)
cor(sigma)
```

```
##           [,1]       [,2]       [,3]
## [1,]  1.0000000 -0.8660254 -0.8660254
## [2,] -0.8660254  1.0000000  0.5000000
## [3,] -0.8660254  0.5000000  1.0000000
```

```
pc <- princomp(sigma)
summary(pc)
```

```
## Importance of components:
##                           Comp.1     Comp.2       Comp.3
## Standard deviation     3.4480268 0.57735027 9.424322e-08
## Proportion of Variance 0.9727273 0.02727273 7.266914e-16
## Cumulative Proportion  0.9727273 1.00000000 1.000000e+00
```

Another method to calculate correlation matrix

```
mu = c(1,2,3);mu
```

```
## [1] 1 2 3
```

```
sigma <- matrix(c(7, 0, 0, 0, 2, 1, 0, 1, 2), nrow = 3, byrow = TRUE)
sigma
```

```
##      [,1] [,2] [,3]
## [1,]    7    0    0
## [2,]    0    2    1
## [3,]    0    1    2
```

```
# Calculate the correlation matrix
R <- cov2cor(sigma)
R
```

```
##      [,1] [,2] [,3]
## [1,]    1  0.0  0.0
## [2,]    0  1.0  0.5
## [3,]    0  0.5  1.0
```

ii. Determine the variance of the second principal component(PC)

By definition, the proportion of variance explained by the PCs are the eigen value for that PC divided by the sum of all eigen values. The variance using the built-in PCA function and the ratio of the respective eigen values to the total sum of all eigen values give the same results.

```
var <- cor(sigma)
var
```

```
##           [,1]       [,2]       [,3]
## [1,]  1.0000000 -0.8660254 -0.8660254
## [2,] -0.8660254  1.0000000  0.5000000
## [3,] -0.8660254  0.5000000  1.0000000
```

```
ei <- eigen(var)
ei
```

```
## eigen() decomposition
## $values
## [1] 2.500000e+00 5.000000e-01 1.332268e-15
##
## $vectors
##            [,1]       [,2]      [,3]
## [1,]  0.6324555  0.0000000 0.7745967
## [2,] -0.5477226 -0.7071068 0.4472136
## [3,] -0.5477226  0.7071068 0.4472136
```

```
#variance explained by each PC
((ei$values))/sum(ei$values)
```

```
## [1] 8.333333e-01 1.666667e-01 4.440892e-16
```

```
# Variance through PCA function
summary(pc)
```

```
## Importance of components:
##                           Comp.1     Comp.2       Comp.3
## Standard deviation     3.4480268 0.57735027 9.424322e-08
## Proportion of Variance 0.9727273 0.02727273 7.266914e-16
## Cumulative Proportion  0.9727273 1.00000000 1.000000e+00
```

iii. Write down the formula for calculating the second PC

check note.

```
# Create the covariance matrix
sigma <- matrix(c(7, 0, 0, 0, 2, 1, 0, 1, 2), nrow = 3, byrow = TRUE)

# Calculate the eigenvalues and eigenvectors of the covariance matrix
eis <- eigen(sigma)
eigen_values <- eis$values
eigen_vectors <- eis$vectors

# Calculate the coefficients or loadings of the second principal component
l2 <- eigen_vectors[, 2] / sqrt(eigen_values[2])

# Calculate the second principal component
pc2 <- eigen_values[2] / sum(eigen_values)
pc2
```

```
## [1] 0.2727273
```

iv. Find the proportion of total variance explained by the first PC. Are two PCs enough. Explain in one sentence.

```
summary(pc)
```

```
## Importance of components:
##                           Comp.1     Comp.2       Comp.3
## Standard deviation     3.4480268 0.57735027 9.424322e-08
## Proportion of Variance 0.9727273 0.02727273 7.266914e-16
## Cumulative Proportion  0.9727273 1.00000000 1.000000e+00
```

The first component alone explains 97% of the total variance. So, 1 component is enough.