20311298_MATH3057 Coursework 1

Lee Yen May

Question 1

Data Processing

Overview of Auto MPG Dataset The Auto MPG dataset consists of 6 numerical variables (2 of data type 'float' and 4 of data type 'integer') and 2 qualitative variables. The objective our analysis is to explore the relationships of fuel efficiency with other attributes of the cars included within the dataset. The description of each attribute is extracted from this source and Wikipedia definitions.

```
ds <- read.csv("auto-mpg.csv")
str(ds) # display the structure of the dataset
```

```
## 'data.frame':    398 obs. of  8 variables:
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinders   : int  8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : chr  "130" "165" "150" "150" ...
##  $ weight      : int  3504 3693 3436 3433 3449 4341 4354 4312 4425 3850 ...
##  $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ model.year  : int  70 70 70 70 70 70 70 70 70 70 ...
##  $ car.name    : chr  "chevrolet chevelle malibu" "buick skylark 320" "plymouth satellite" "amc rebel
```

Data Cleaning The variable 'horsepower' is observed to have character values (ie."?"), those rows are removed.

```
ds <- ds[!(ds$horsepower=="?"),]
ds$horsepower <- as.integer(ds$horsepower)
```

Table 1: Auto-MPG Dataset Variables

Variable	Variable Type	Data Type	Description
mpg	quantitative	float	Miles per gallon (fuel efficiency)
cylinders	qualitative	integer	Number of cylinders in the engine
displacement	quantitative	integer	Engine displacement or measure of the cylinder volume
horsepower	quantitative	integer	Engine horsepower
weight	quantitative	integer	Vehicle weight (in pounds)
acceleration	quantitative	float	Time to accelerate from 0 to 60 mph (in seconds)
model.year	quantitative	integer	Model year
car.name	qualitative	string	Car name

The original dataset has 398 rows, a total of 6 rows are removed, leaving 392 rows, n = 392.

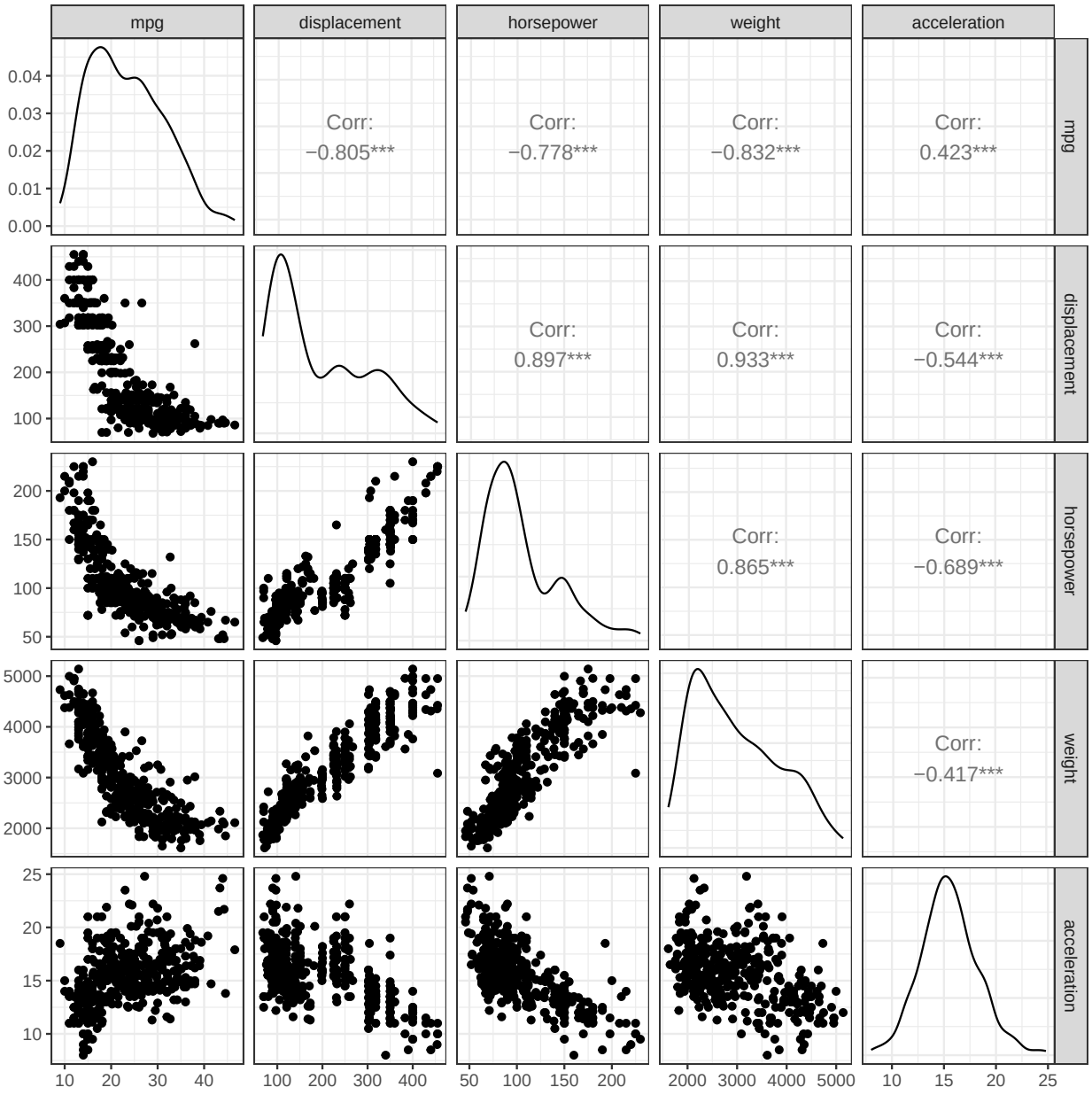
```
n = nrow(ds)
```

Exploratory Analysis

We attempt to visualise the correlation between the numerical variables, therefore, a subset of the dataset which excludes "car.name" is created. Here, we adopt 2 methods of data visualisation.

1. Scatterplot Since "cylinders" and "model.year" are categorical variables, they are excluded from the figure below.

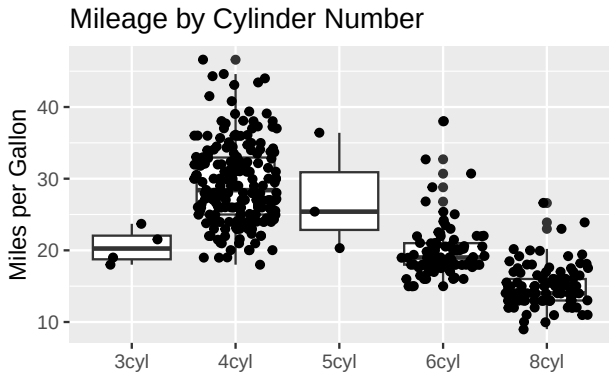
```
library(dplyr)
library("ggplot2")
library("GGally")
table = ds %>% dplyr::select(mpg, cylinders, displacement,
                             horsepower, weight, acceleration, model.year)
ggpairs(select(table, -c(cylinders, model.year)))+theme_bw()
```

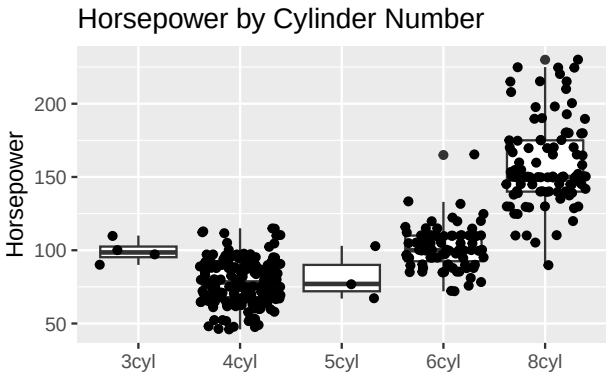


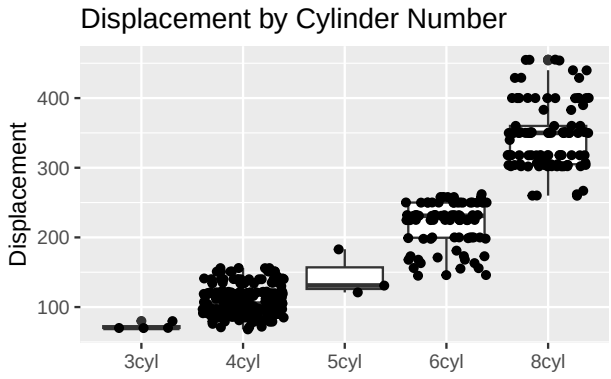
All 5 variables seem to have a linear relationship with each other, though more weakly when compared against acceleration, indicating that acceleration has little impact on fuel efficiency. There is a strong positive correlation between horsepower and displacement, horsepower and weight, and weight and displacement. This implies that a car with more engine power tends to have higher volume of cylinder and is heavier. There is a strong negative correlation when comparing mpg against displacement, weight, and horsepower. This indicates that as displacement, horsepower and weight increase, the car's mpg decreases instead. Hence, a high displacement, high horsepower and heavy vehicle is adverse to fuel efficiency.

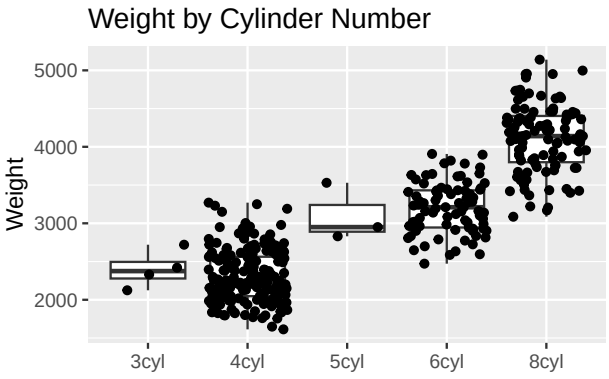
2. Boxplots We introduce "cylinders" and "model.year" back into our analysis. We observe some obvious patterns in each boxplot when comparing the variables against the number of cylinders in the vehicle. We also acknowledge the data deficit for vehicles with 3 and 5 cylinders.

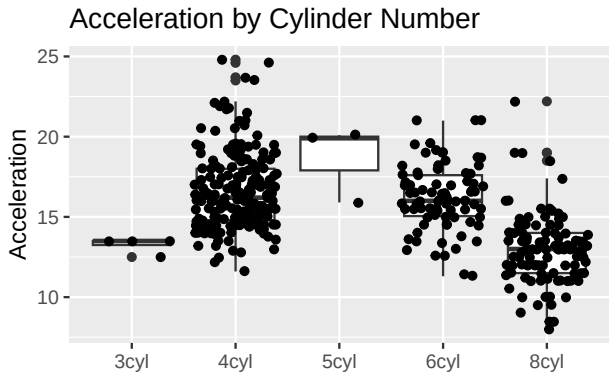
```
library(gridExtra)

table$cylinders = factor(table$cylinders, levels=c(3,4,5,6,8),
                         labels=c("3cyl","4cyl","5cyl","6cyl","8cyl"))

p1 <- qplot(cylinders, mpg, data=table, geom=c("boxplot","jitter"),
            main="Mileage by Cylinder Number", xlab="",ylab="Miles per Gallon")
p2 <- qplot(cylinders, horsepower, data=table, geom=c("boxplot","jitter"),
            main="Horsepower by Cylinder Number", xlab="",ylab="Horsepower")
p3 <- qplot(cylinders, displacement, data=table, geom=c("boxplot","jitter"),
            main="Displacement by Cylinder Number", xlab="",ylab="Displacement")
p4 <- qplot(cylinders, weight, data=table, geom=c("boxplot","jitter"),
            main="Weight by Cylinder Number", xlab="",ylab="Weight")
p5 <- qplot(cylinders, acceleration, data=table, geom=c("boxplot","jitter"),
            main="Acceleration by Cylinder Number", xlab="",ylab="Acceleration")
p6 <- qplot(cylinders, model.year, data=table, geom=c("boxplot","jitter"),
            main="Model Year by Cylinder Number", xlab="",ylab="Model Year")

grid.arrange(p1, p2, p3, p4, p5, p6, ncol=2)
```

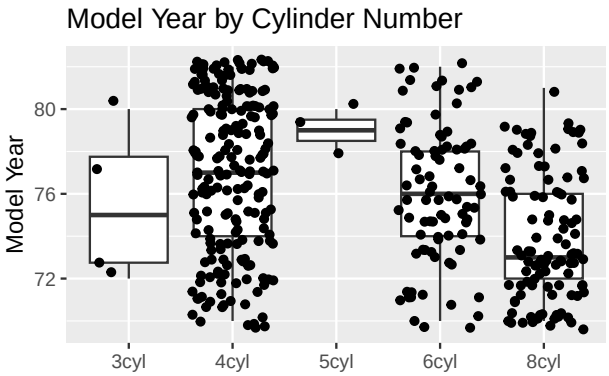












We observe that as the number of cylinders increases, the horsepower, displacement and weight also increase. On the contrary, mpg decreases. This is consistent with our inference on the positive correlation between the three variables and their negative correlation to mpg. The spread for 4 cylinders is greater in mpg whereas the spread for 8 cylinders is greater in horsepower. This indicates that data points for these variables at their respective number of cylinders have broader range of observation and are less consistent compared to other number of cylinders. The spreads in displacement and weight are relatively small and consistent throughout. The boxplot for 8 cylinders does not overlap with other number of cylinders when plotted against acceleration, hence it may suggest that an attribute of 8 cylinders has a positive relationship with lower acceleration in vehicle. The model year does not seem to have a significant correlation with the number of cylinders.

Principal Component Analysis (PCA)

From the summary, we observe that the quantities each variable represents differ widely from each other. Therefore, PCA based on correlation matrix is carried out.

```
table$cylinders = ds$cylinders
summary(table)
```

```
##       mpg          cylinders      displacement     horsepower        weight
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613
##  1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225
##  Median :22.75   Median :4.000   Median :151.0   Median : 93.5   Median :2804
##  Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5   Mean   :2978
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140
##   acceleration     model.year
##  Min.   : 8.00   Min.   :70.00
##  1st Qu.:13.78   1st Qu.:73.00
##  Median :15.50   Median :76.00
##  Mean   :15.54   Mean   :75.98
##  3rd Qu.:17.02   3rd Qu.:79.00
##  Max.   :24.80   Max.   :82.00
```

```
# Center the data
tablebar = colMeans(table)
table <- as.matrix(sweep(table, 2, tablebar))

# Correlation matrix
R = cor(table)
eigen(R)
```

```
## eigen() decomposition
## $values
## [1] 5.01063582 0.86559140 0.72839377 0.18391509 0.12191632 0.05425716 0.03529043
##
## $vectors
##            [,1]         [,2]        [,3]        [,4]        [,5]        [,6]
## [1,]  0.3981348 -0.206758641 -0.25721494  0.75096624  0.34077556 -0.2097589
## [2,] -0.4161242 -0.198541133  0.13915928  0.47730649 -0.49322226  0.3325483
## [3,] -0.4292827 -0.180362422  0.10031610  0.29784705 -0.05658084 -0.1429671
## [4,] -0.4228129 -0.085241832 -0.16968441 -0.04207625  0.71128893  0.5228025
## [5,] -0.4140457 -0.224674565  0.27610337 -0.10773508  0.26515768 -0.6965178
## [6,]  0.2848971  0.006971629  0.89330772  0.12112398  0.23075501  0.2237849
## [7,]  0.2295100 -0.909674802 -0.03724635 -0.30243525 -0.08896075  0.1281955
##             [,7]
## [1,] -0.09221162
## [2,] -0.43171605
## [3,]  0.81287676
## [4,] -0.06438539
## [5,] -0.36715386
## [6,]  0.05279944
## [7,]  0.05113155
```

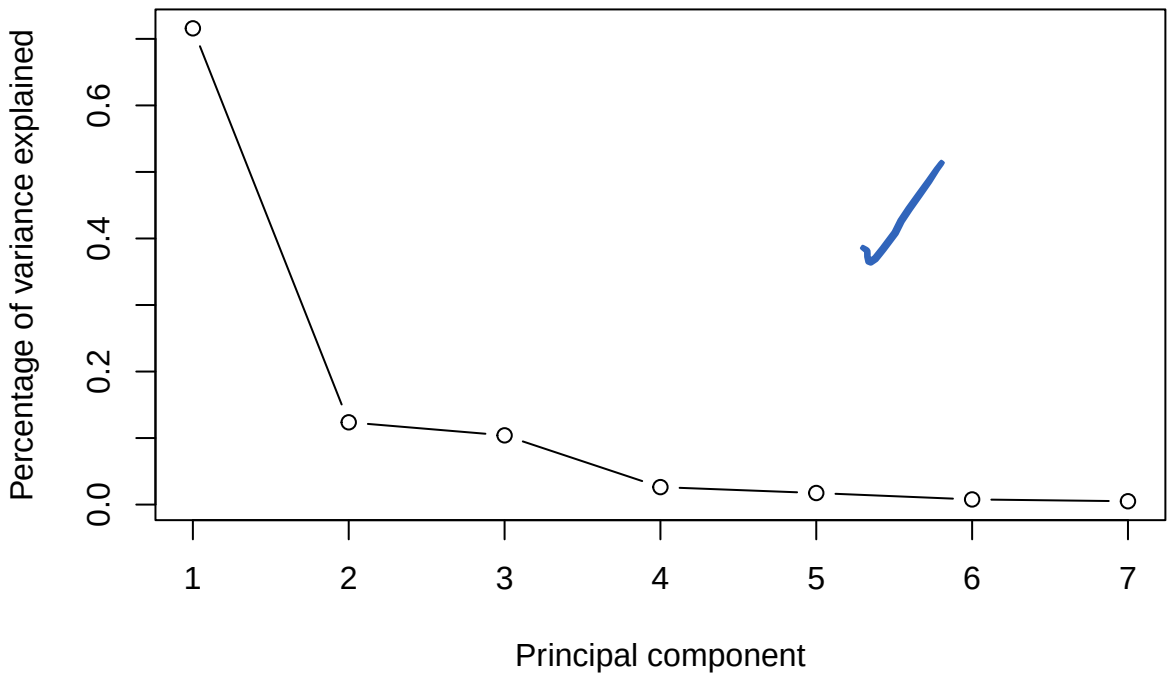
```r
# PCA
table.pca = prcomp(table, scale = TRUE)
head(table.pca$x) # transformed variable
```

```
##         PC1        PC2       PC3        PC4         PC5          PC6
## 1 -2.631685 -0.9278532 0.5339964 -0.7446385  0.54531082 -0.094852576
## 2 -3.489341 -0.8044447 0.6486656 -0.4941973  0.03559007  0.206869564
## 3 -2.966623 -0.8800618 0.9575186 -0.7188062  0.28654174  0.136412813
## 4 -2.906483 -0.9604935 0.5822090 -0.5308103  0.28358945  0.292864658
## 5 -2.900120 -0.9515733 1.0534897 -0.5643803  0.54410697 -0.001894291
## 6 -4.646688 -0.4200083 0.9934670 -0.5349352 -0.60833496 -0.105880911
##           PC7
## 1  0.12192173
## 2 -0.09773657
## 3  0.05967329
## 4  0.12436188
## 5  0.17060817
## 6 -0.34740517
```

```r
summary(table.pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.2384 0.9304 0.8535 0.42885 0.34917 0.23293 0.18786
## Proportion of Variance 0.7158 0.1237 0.1041 0.02627 0.01742 0.00775 0.00504
## Cumulative Proportion  0.7158 0.8395 0.9435 0.96979 0.98721 0.99496 1.00000
```

```r
plot(table.pca$sdev^2/sum(table.pca$sdev^2), type="b",
     xlab="Principal component", ylab="Percentage of variance explained")
```

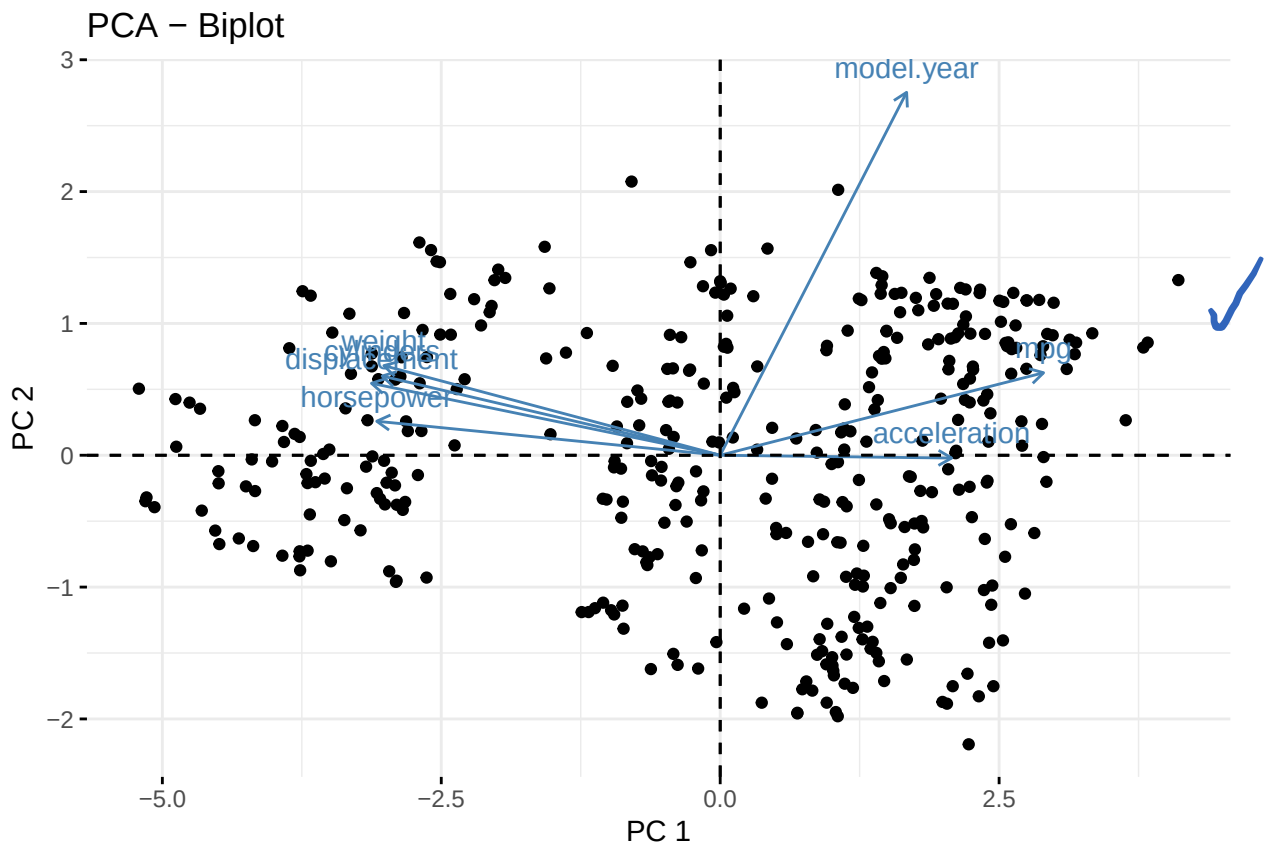


Scree Plot

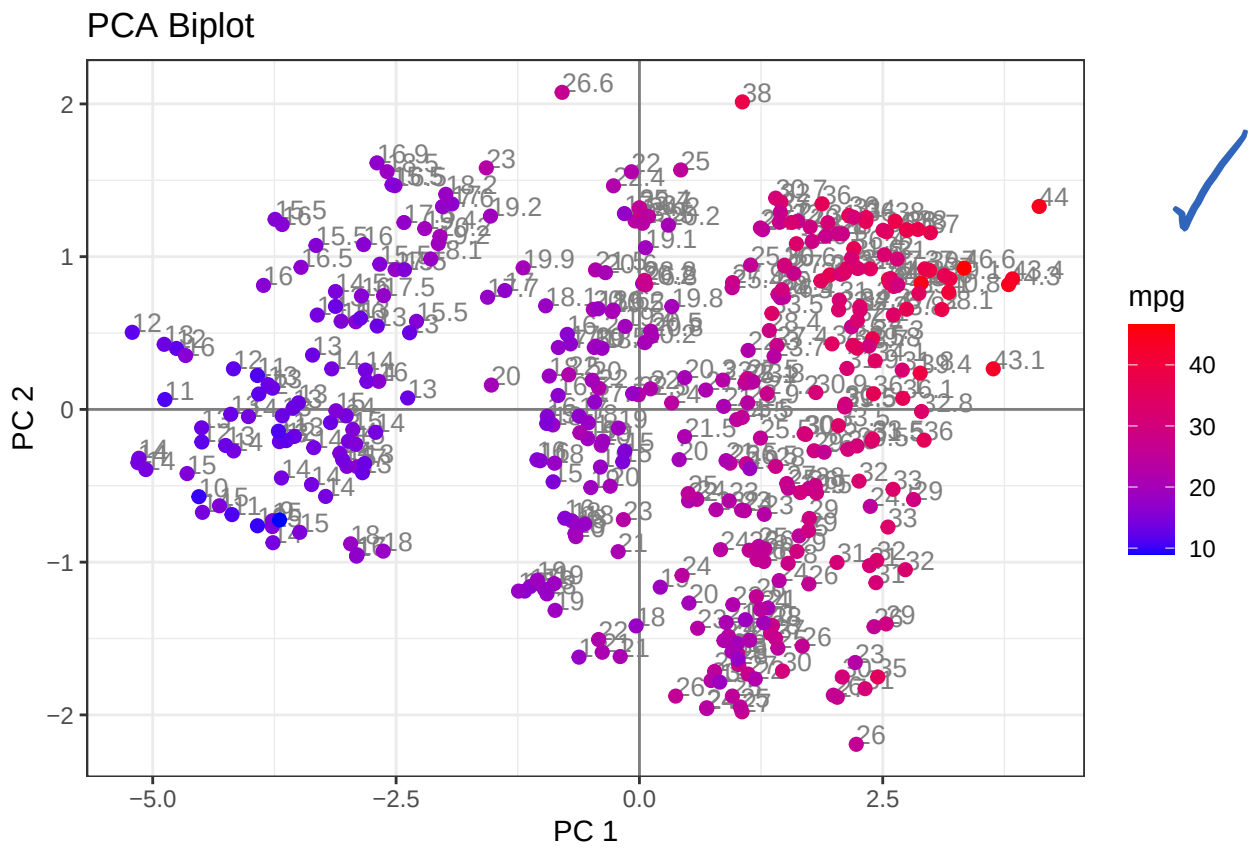
We decide that PC1, PC2 and PC3 best represent our data because (1) the last 4 eigenvalues are close to 0, (2) a 94.35% cumulative proportion of variance is achieved with the first 3 eigenvalues and eigenvectors and (3) percentage of variance explained has achieved a good coverage by PC3 based on the scree plot. We will ignore PC4 until PC7 for the subsequent operations.

Interpretation of PC Scores We only visualise PC1 against PC2. We could include PC3 with a 3D plot but it is not demonstrated here.

```
library(factoextra)
par(pty="s")

fviz_pca_biplot(table.pca, scale = TRUE, cex = 0.7,
                label = "var", xlab = "PC 1", ylab = "PC 2")
```




```
pca_scores <- data.frame(PC1 = table.pca$x[,1],
                         PC2 = table.pca$x[,2],
                         mpg = ds[,1])
# Indicate points by colour and mpg value
library(ggplot2)
ggplot(pca_scores, aes(x = PC1, y = PC2, label = mpg)) +
  geom_hline(yintercept = 0, color = "grey50") +
  geom_vline(xintercept = 0, color = "grey50") +
  geom_text(hjust = 0, vjust = 0, color = "grey50", size = 3.5) +
  geom_point(aes(color = mpg), size = 2) +
  scale_color_gradient(low = "blue", high = "red") +
  labs(title = "PCA Biplot", x = "PC 1", y = "PC 2", color = "mpg") +
  theme_bw()
```



PC1 as a linear combination of the original features:

$$\begin{aligned}\mathbf{y}_1 = &\ 0.40 \times \text{mpg}\\ &- 0.42 \times \text{cylinders}\\ &- 0.43 \times \text{displacement}\\ &- 0.42 \times \text{horsepower}\\ &- 0.41 \times \text{weight}\\ &+ 0.28 \times \text{acceleration}\\ &+ 0.23 \times \text{model.year}\end{aligned}$$

PC2 as a linear combination of the original features:

$$\begin{aligned}\mathbf{y}_2 = &\ - 0.21 \times \text{mpg}\\ &- 0.20 \times \text{cylinders}\\ &- 0.18 \times \text{displacement}\\ &- 0.09 \times \text{horsepower}\\ &- 0.22 \times \text{weight}\\ &+ 0.01 \times \text{acceleration}\\ &- 0.91 \times \text{model.year}\end{aligned}$$

From $\mathbf{y}_1$, we observe that cylinders, displacement, horsepower and weight have high negative coefficients. Therefore, a vehicle with a low PC1 score implies that it has a low fuel efficiency, and likely to have attributes

such as many cylinders, high volume of cylinders, high engine power and heavy. From $\mathbf{y}_2$, we observe that acceleration is the only positive coefficient and model year has an exceptionally high negative coefficient. Therefore, a vehicle with a positive PC2 score has the attribute of a high acceleration car and a high negative PC2 score is likely to represent a very old vehicle. This is in line with the coloured biplot above, where vehicles with low fuel efficiency (ie. low mpg) is observed to have high negative PC1 and PC2 score. The order of fuel efficiency from the lowest to the highest is vehicles in quadrant 3, followed by quadrant 2 and quadrant 4. Vehicles with high positive PC1 and PC2 score (ie. quadrant 1) have the highest fuel efficiency.

Question 2

Suggestion: 10×3

We construct a 10x3 data matrix with the third column as a linear combination of the first and second columns. Random numbers in the matrix are generated from the normal distribution.

```
# for reproducibility
set.seed(123)
# create a 10x3 matrix of random normal values
X <- matrix(rnorm(30), nrow = 10)
# create the third column as a linear combination of the first and second columns
X[,3] <- X[,1] + X[,2]

# centering
xbar = colMeans(X)
X <- as.matrix(sweep(X, 2, xbar))
X
```

```
##               [,1]        [,2]       [,3]
##  [1,] -0.635101291  1.01545984  0.3803585
##  [2,] -0.304803134  0.15119187 -0.1536113
##  [3,]  1.484082670  0.19214949  1.6762322
##  [4,] -0.004117253 -0.09793924 -0.1020565
##  [5,]  0.054662091 -0.76446310 -0.7098010
##  [6,]  1.640439343  1.57829118  3.2187305
##  [7,]  0.386290562  0.28922852  0.6755191
##  [8,] -1.339686879 -2.17523912 -3.5149260
##  [9,] -0.761478496  0.49273394 -0.2687446
## [10,] -0.520287614 -0.68141337 -1.2017010
```

(a) Sample covariance matrix with n as the denominator.

```
# sample covariance matrix
cov_mat = 1/10 * t(X) %*% X
# cov(X)*9/10 # alternative method
```

(b) Eigenvalues and eigenvectors of the covariance matrix. We observe that $\lambda_3 = 0$, which is within expectation because the third column is a linear combination of the first and second column. We will ignore PC 3 in the subsequent workings.

```
eigen(cov_mat)
```

```
## eigen() decomposition
```

```
## $values
## [1] 4.231419e+00 3.751335e-01 1.974696e-16
##
## $vectors
##            [,1]        [,2]       [,3]
## [1,] -0.3840041  0.72056056  0.5773503
## [2,] -0.4320217 -0.69283759  0.5773503
## [3,] -0.8160258  0.02772297 -0.5773503
```

(c) Produce 2 plots side-by-side.

- The plot of the first two columns of the centered data along with the first two PCs.
- The plot of the transformed variables for the first two PCs.

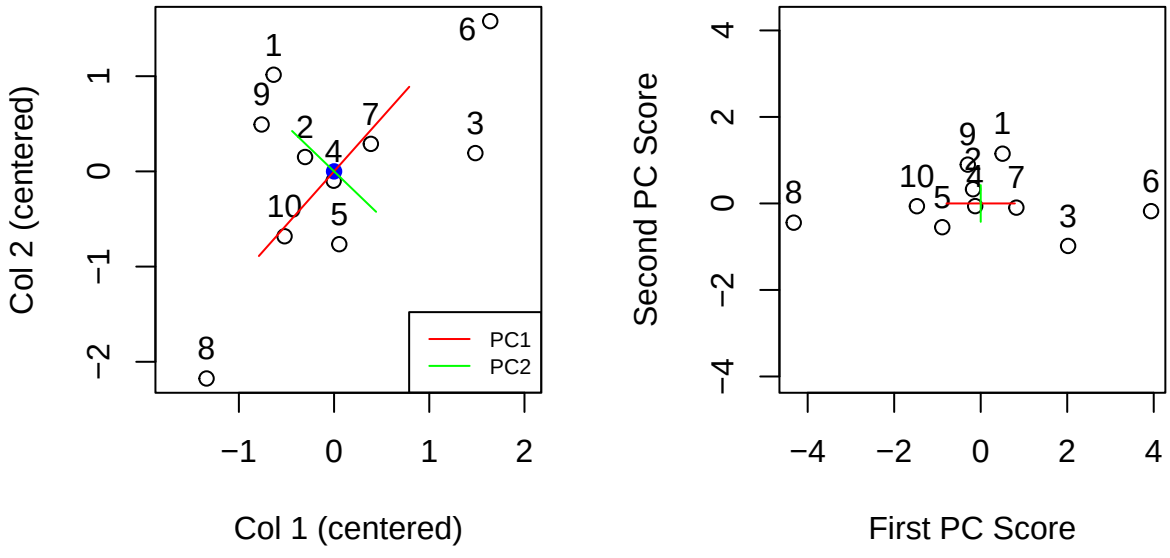
We observe that the variation is indeed in line with the new coordinate axes, it is more obvious when we observe the points that fall into the four respective quadrants separated by segments formed from PC scores. All 10 randomly generated observations are numbered accordingly.

```
par(pty="s")
pca <- prcomp(X) # compute principal components
(X_svd = svd(X))
```

```
## $d
## [1] 6.504936e+00 1.936836e+00 8.702159e-16
##
## $u
##               [,1]        [,2]        [,3]
##  [1,]  0.07766434  0.59407856 -0.33757320
##  [2,] -0.02722214  0.16967828  0.85320527
##  [3,]  0.31064953 -0.50738057 -0.08571863
##  [4,] -0.01955033 -0.03204192  0.04225401
##  [5,] -0.13658707 -0.28363692  0.12533034
##  [6,]  0.60544158 -0.09178328 -0.19124106
##  [7,]  0.12675469 -0.04991890 -0.02682895
##  [8,] -0.66449019 -0.22940384 -0.28667826
##  [9,] -0.04594075  0.46339812 -0.08248843
## [10,] -0.22671968 -0.03298954 -0.08344234
##
## $v
##           [,1]        [,2]       [,3]
## [1,] 0.3840041 -0.72056056  0.5773503
## [2,] 0.4320217  0.69283759  0.5773503
## [3,] 0.8160258 -0.02772297 -0.5773503
```

```
x1 = X_svd$v[1,1]
y1 = X_svd$v[2,1]

x2 = X_svd$v[1,2]
y2 = X_svd$v[2,2]

par(mfrow = c(1,2)) # create two subplots side-by-side
```

```r
# plot (i)
plot(X[,1], X[,2], xlab="Col 1 (centered)", ylab="Col 2 (centered)", asp=1)
text(X[,1], X[,2], labels = 1:nrow(X), pos = 3)
points(0,0, pch = 19, col = "blue") # the centered mean

lambda1 = X_svd$d[1]^2/10
segments(0,0, sqrt(lambda1)*x1, sqrt(lambda1)*y1, col = "red")
segments(0,0, -sqrt(lambda1)*x1, -sqrt(lambda1)*y1, col = "red")

lambda2 = X_svd$d[2]^2/10
segments(0,0, sqrt(lambda2)*x2, sqrt(lambda2)*y2, col = "green")
segments(0,0, -sqrt(lambda2)*x2, -sqrt(lambda2)*y2, col = "green")

legend("bottomright", legend = c("PC1", "PC2"),
       col = c("red", "green"), lty = 1, cex=0.70)
text(1.4, 1.5, "6", col="black", cex=1)

# plot (ii)
plot(pca$x[,1], pca$x[,2], xlab="First PC Score", ylab="Second PC Score", asp=1)
text(pca$x[,1], pca$x[,2], labels = 1:nrow(pca$x), pos = 3)
segments(0,0, sqrt(lambda1)*x1, 0, col = "red")
segments(0,0, -sqrt(lambda1)*x1, 0, col = "red")
segments(0,0, 0, sqrt(lambda2)*y2, col = "green")
segments(0,0, 0, -sqrt(lambda2)*y2, col = "green")
```



(d) Singular value decomposition of $\mathbf{HX}$ and $\frac{1}{\sqrt{n}}\mathbf{HX}$.

```
n = 10
# Calculate the centering matrix H
H = diag(rep(1,n))-rep(1,n)%*%t(rep(1,n))/n

# Calculate SVD of HX
svd(H%*%X)
```

```
## $d
## [1] 6.504936e+00 1.936836e+00 1.105007e-15
##
## $u
##               [,1]        [,2]        [,3]
##  [1,]  0.07766434  0.59407856 -0.26235209
##  [2,] -0.02722214  0.16967828  0.82362384
##  [3,]  0.31064953 -0.50738057  0.05383436
##  [4,] -0.01955033 -0.03204192  0.04246316
##  [5,] -0.13658707 -0.28363692  0.17207913
##  [6,]  0.60544158 -0.09178328 -0.31789823
##  [7,]  0.12675469 -0.04991890  0.01860110
##  [8,] -0.66449019 -0.22940384 -0.32722475
##  [9,] -0.04594075  0.46339812 -0.02783412
## [10,] -0.22671968 -0.03298954 -0.09615931
##
## $v
##           [,1]        [,2]       [,3]
## [1,] 0.3840041 -0.72056056  0.5773503
## [2,] 0.4320217  0.69283759  0.5773503
## [3,] 0.8160258 -0.02772297 -0.5773503
```

```
# Calculate SVD of 1/sqrt(n)*HX
svd(1/sqrt(n)*H%*%X)
```

```
## $d
## [1] 2.057041e+00 6.124815e-01 2.265171e-16
##
## $u
##               [,1]        [,2]        [,3]
##  [1,]  0.07766434  0.59407856 -0.41481186
##  [2,] -0.02722214  0.16967828 -0.20418275
##  [3,]  0.31064953 -0.50738057 -0.18679766
##  [4,] -0.01955033 -0.03204192  0.04078606
##  [5,] -0.13658707 -0.28363692  0.05374083
##  [6,]  0.60544158 -0.09178328 -0.58648411
##  [7,]  0.12675469 -0.04991890  0.03349730
##  [8,] -0.66449019 -0.22940384 -0.61080598
##  [9,] -0.04594075  0.46339812  0.01078926
## [10,] -0.22671968 -0.03298954 -0.16884771
##
## $v
##           [,1]        [,2]       [,3]
## [1,] 0.3840041 -0.72056056  0.5773503
```

```
## [2,] 0.4320217  0.69283759  0.5773503
## [3,] 0.8160258 -0.02772297 -0.5773503
```

- How are the two sets of singular values related? The two sets of singular values are related by scaling: The singular values of $c\mathbf{X}$ is equivalent to $c\times$ singular values of $\mathbf{X}$. Therefore, the singular values of $\frac{1}{\sqrt{10}}\mathbf{X}$ is equivalent to $\frac{1}{\sqrt{10}}\times$ singular values of $\mathbf{X}$.

```
# 1/sqrt(n) * singular values of HX
svd_X = svd(H%*%X)$d
(c_svd_X = 1/sqrt(n)*svd_X)
```

```
## [1] 2.057041e+00 6.124815e-01 3.494338e-16
```

```
# is equivalent to singular values of 1/sqrt(n)*HX
(svd_cX = svd(1/sqrt(n)*H%*%X)$d)
```

```
## [1] 2.057041e+00 6.124815e-01 2.265171e-16
```

- How do the singular values relate to the eigenvalues computed previously? There are two ways to derive eigenvalues when conducting PCA: Method 1 - Derive the eigenvalues of covariance matrix $\mathbf{S} = \frac{1}{n}\mathbf{X}^\mathbf{T}\mathbf{X}$. Method 2 - The squared value of singular values computed in previous section.

Eigenvalues computed by Method 1:

$$\mathbf{\Lambda} = \begin{pmatrix} 4.231 \\ 0.375 \\ 0 \end{pmatrix}$$

Eigenvalues computed by Method 2:

```
c_svd_X^2
```

```
## [1] 4.231419e+00 3.751335e-01 1.221040e-31
```

```
svd_cX^2
```

```
## [1] 4.231419e+00 3.751335e-01 5.130998e-32
```

Observe that eigenvalues computed using Method 2 is the same as Method 1.