Wishart Distribution & Hotelling's T2 test

Tarek El-Hajjaoui

2023-04-29

```
library(car) # scatterplotMatrix(...)
```

```
## Loading required package: carData
```

Problem 1: Choose a $3 - by - 3$ covariance matrix with non-zero covariances (the off-diagonal elements should not be 0). Also a choose a sample size n (e.g., n=100, 500, 1000, etc). Using the covariance matrix you chose, simulate 1,000 data sets from a trivariate normal distribution.

1.1

Creating a covariance matrix:

```
set.seed(2) # Set the seed for reproducibility
m <- matrix(runif(9, -1, 1), nrow = 3, ncol = 3) #  9 random numbers between (-1 and 1)
Sigma <- crossprod(m) # Create a symmetric matrix
# Manipulating the values to give different types of correlations
Sigma[1, 2] <- Sigma[2, 1] <- 0.7
Sigma[1, 3] <- Sigma[3, 1] <- -0.1
Sigma[2, 3] <- Sigma[3, 2] <- 0.1
Sigma
```

```
##            [,1]     [,2]      [,3]
## [1,]  0.5825248 0.700000 -0.100000
## [2,]  0.7000000 2.015412  0.100000
## [3,] -0.1000000 0.100000  0.998936
```

```
round(eigen(Sigma)$values, 2) # Check all of the eigenvalues are positive
```

```
## [1] 2.30 1.02 0.27
```

```
#cor(Sigma)
```

Simulation Using the covariance matrix above, simulate 1,000 data sets from a trivariate normal distribution. The function returns an array simulation of size T where element, X_i, is a (nxp) Trivariate Normal Matrix,

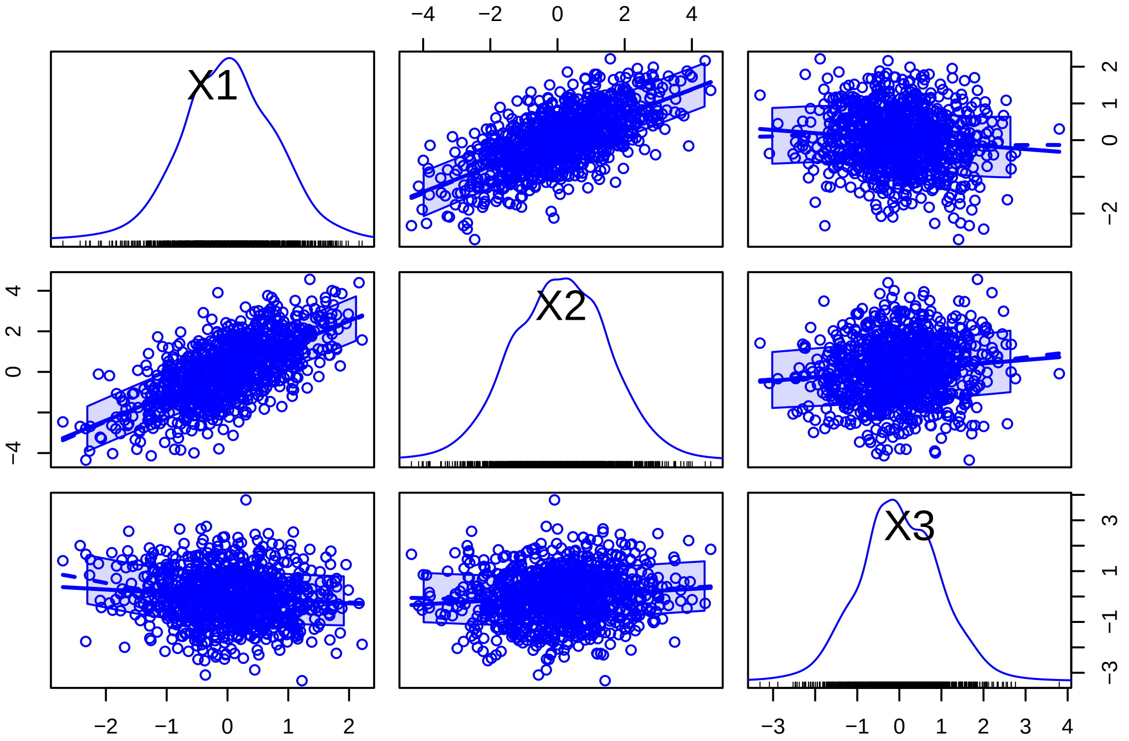
```
library(MASS) # mvrnorm(...)
create_trivariate_norm <- function(T, n, p, Sigma) {
  simulation <- array(0, c(n, 3, T))
   for (t in 1:T) {
    X <- mvrnorm(n, rep(0, p), Sigma)
    simulation[, , t] <- X
  }
  return (simulation)
}
```

```
p <- 3
T <- 1000
n <-1000
simulation <- create_trivariate_norm(T=T, n=n, p=p, Sigma=Sigma)
head(simulation[, , 1:1]) # show head of 1 element
```

```
##              [,1]        [,2]       [,3]
## [1,] -0.53998012  0.05359939 -0.9002465
## [2,]  0.93330577  0.84812834 -1.3662784
## [3,]  0.49569373  1.19344562  2.2044041
## [4,] -0.89115617 -1.35083552 -0.2640324
## [5,]  0.76822026  0.95473478 -0.7041917
## [6,] -0.02500744  2.42210916 -1.1682155
```

Pairwise scatterplot of simulated data (simulation)

```
scatterplotMatrix(simulation[, , 1])
```



```
cor(simulation[, , 1]) # correlation matrix
```

```
##            [,1]      [,2]       [,3]
## [1,]  1.0000000 0.6551686 -0.1087917
## [2,]  0.6551686 1.0000000  0.1137658
## [3,] -0.1087917 0.1137658  1.0000000
```

```
cov(simulation[, , 1]) # covariance matrix
```

```
##              [,1]      [,2]        [,3]
## [1,]  0.62589869 0.7689229 -0.08502275
## [2,]  0.76892291 2.2006727  0.16671582
## [3,] -0.08502275 0.1667158  0.97583089
```

1.1. Try to make sense of the covariance matrix by examining the pairwise scatter plots using the data you simulate.

- The diagnol of the scatter plot matrix above plots the marginal distribution of the first element in sim1 and it can be observed that each marginal distribution of X_i is a normal distribution. The off-diagonol elements of the scatter plot matrix plots the joint distribution between random variables, X_i and X_j. Below the pairwise plots, is the correlation and covariance matrices respectively of the sample. It cna be observed that the plots align with the correlation values. For example, X_1 and X_2 have a positive correlation of 0.655. Their respective scatter plots have a diagonal line that slopes upward from left to right, suggesting a moderately positive linear relationship. Additionally X_1 and X_2 have higher covariance values relative to the other pairwise joint distributions, and so the data points are narrowly spread across the joint distribution. Conversely, X_1 and X_3 have low covariance and a low negative correlation. This is reflected by a high degree of spread in data points and a low negative slope.

1.2

Compute and store S, the sample covariance.

```
S.array <- array(apply(simulation, p, cov), c(p, p, T))
```

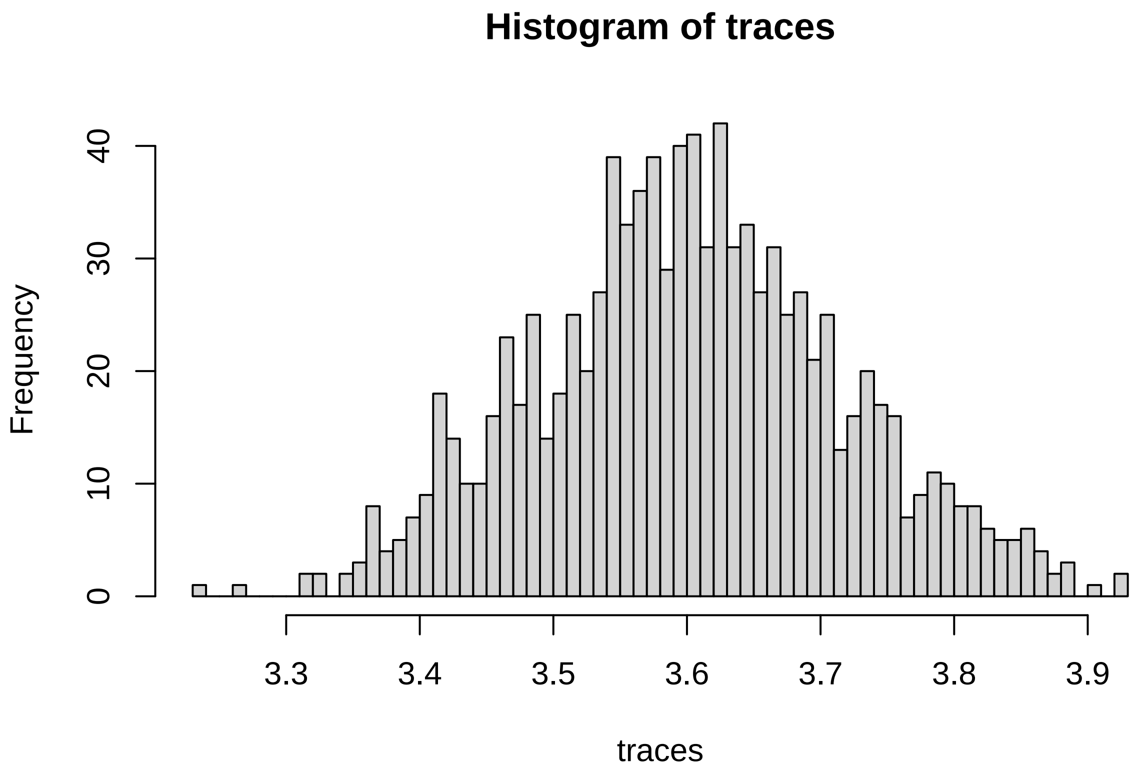
Compute and store the traces.

```
traces <- apply(S.array, p, function(x) sum(diag(x)))
traces[1:3]
```

```
## [1] 3.802402 3.641897 3.725560
```

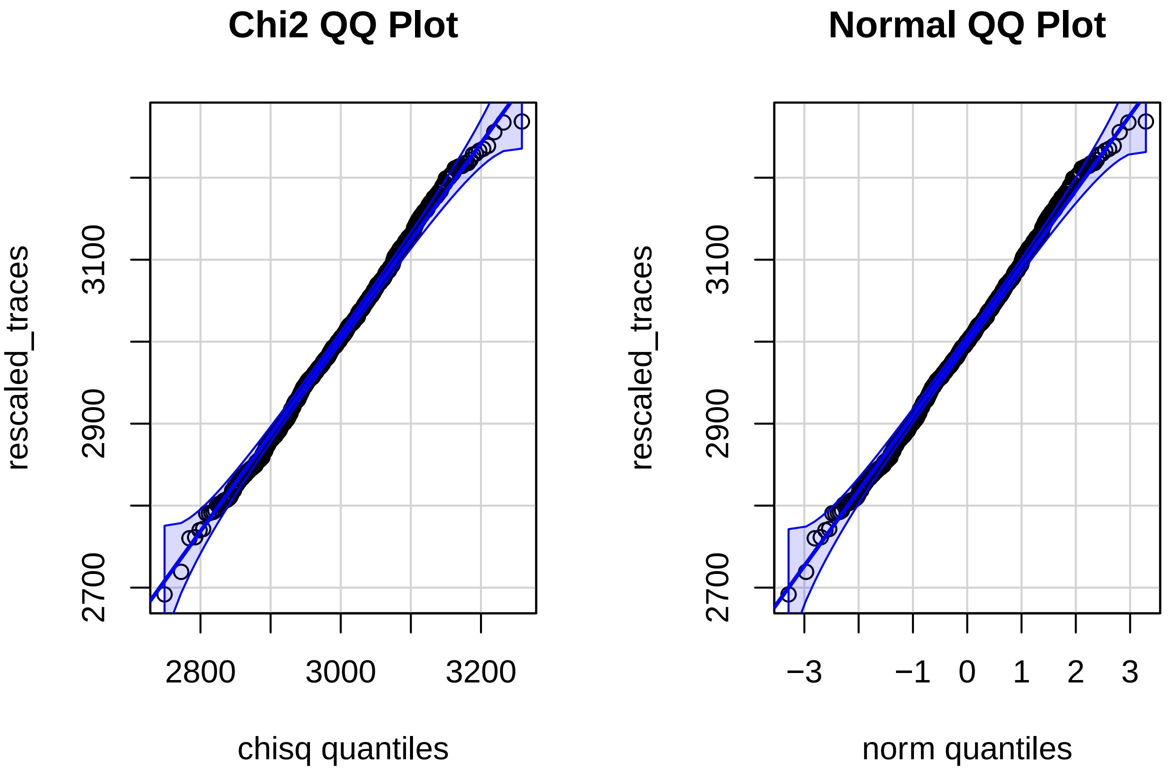
Distribution of the traces

```
hist(traces, breaks = 50)
```



QQ Plots of χ^2 and $\mathcal{N}$ Distributions on re-scaled data.

```r
rescaled_traces <- (n-1)*traces/mean(diag(Sigma))
par(mfrow = c(1, 2))
qqPlot(rescaled_traces, distribution = "chisq", df=3*(n-1), id = F, main = "Chi2 QQ Plot")
qqPlot(rescaled_traces, id = F, main = "Normal QQ Plot")
```

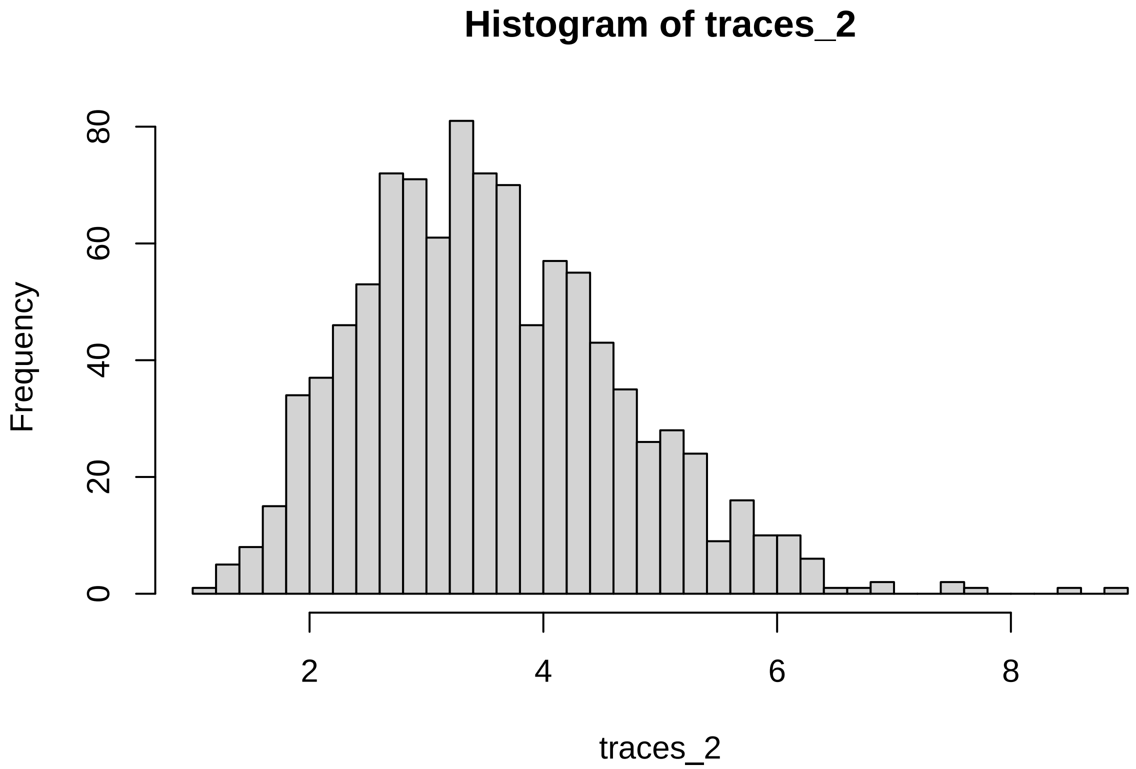


1.2 During the simulation, you will generate 1,000 Wishart distributed random matrices. Calculate the trace for each of them. Explain what distribution the traces should follow and examine their histogram.

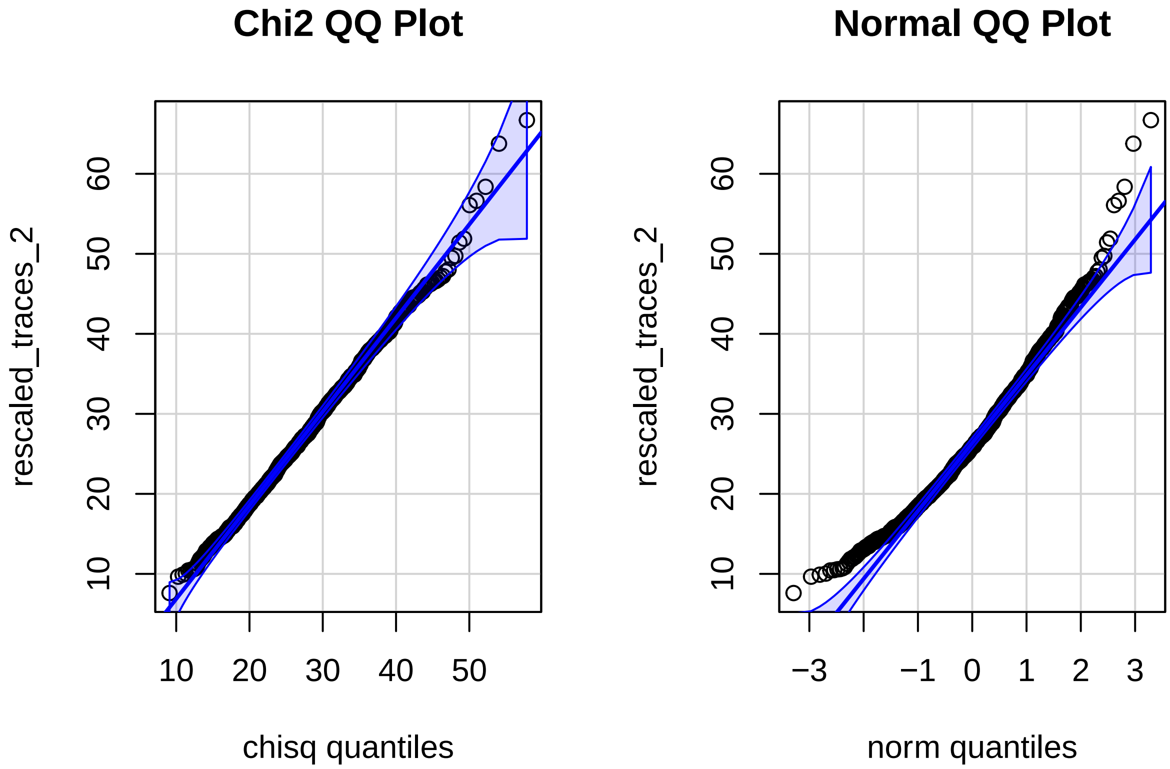
- We would expect that the (rescaled) trace would follow a χ^2 distribution or a linear combination of χ^2, but the histogram looks like a bell-shaped distribution suggesting a $\mathcal{N}$ distribution. This is because the Central Limit Theorem (CLT) effect kicks in when dealing with sums. The CLT tells us that the sum of random variables tend to be normally distributed as the number of elements in the sum increase. Thus, a χ^2 with very high degrees of freedom, (n=1000 above) tends to become a normal distribution.

To further test if CLT effect occurred, a simulation with a much smaller size (n=10) will be conducted to see if the trace plot and QQ-plots suggest a $\mathcal{N}$ distribution.

```r
p <- 3
T <- 1000
n <-10 # 100 instead of 10 this time
simulation_2 <- create_trivariate_norm(T=T, n=n, p=p, Sigma=Sigma)
S_2.array <- array(apply(simulation_2, p, cov), c(p, p, T))
traces_2 <- apply(S_2.array, p, function(x) sum(diag(x)))
hist(traces_2, breaks = 50)
```



```
rescaled_traces_2 <- (n-1)*traces_2/mean(diag(Sigma))
par(mfrow = c(1, 2))
qqPlot(rescaled_traces_2, distribution = "chisq", df=3*(n-1), id = F, main = "Chi2 QQ Plot")
qqPlot(rescaled_traces_2, id = F, main = "Normal QQ Plot")
```



The simulation above used the same covariance matrix as the original simulation however only 10 random variables were used to generate a multivariate normal distribution for each t_1 sample. The histogram of traces is no longer a bell-shape distribution which suggests not a normal distribution. Additionally the rescaled traces follow the χ^2 distribution in QQ-plot but do not follow a $\mathcal{N}$ distribution in the QQ-plot so well. This simulation is evidence that the first simulation results were affected by the Central Limit Theorem.

Problem 2 starts on next page.

Problem 2: Find a good data example to conduct a two-sample Hotelling's T^2 test. Do not use the data example discussed in this course. Please (1) include visualizations as exploratory methods and (2) make conclusion in the context of the data example.

```
# X = random population 1, Y = random population 2
Hotelling.T2.2sample=function(X, Y) {
  n = dim(X)[1]; m=dim(Y)[1]; p=dim(X)[2]
  if(p!= dim(Y)[2])
    return ("Error: the dimensions of X and Y are not the same")

  X.bar=colMeans(X); Y.bar=colMeans(Y)
  X.S=cov(X); Y.S=cov(Y)
  pooled.S=((n-1)*X.S+(m-1)*Y.S)/(m+n-2)
  T2=t(X.bar-Y.bar)%*%solve((1/n+1/m)*pooled.S)%*%(X.bar-Y.bar)
  p.value=1-pf(T2/((n+m-2)*p/(n+m-1-p)),p,n+m-1-p)
  return (list(X.bar=X.bar, Y.bar=Y.bar, T2=T2, p.value=p.value))
}
# dual function Hotelling.T2
Hotelling.T2=function(X, Y=NULL, mu0=NULL) {
  if(is.null(Y) && is.null(mu0))
    return("Error: mu0 is not specified")
  if(!is.null(X) && !is.null(mu0))
    obj=Hotelling.T2.1sample(X, mu0)
  if(!is.null(X) && !is.null(Y))
    obj=Hotelling.T2.2sample(X,Y)
  return (obj)
}
```

Using built-in R dataset, mtcars.

```
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
```

```
# create subset X with samples of cyl = 6, keeping only mpg and hp columns
X <- mtcars[mtcars$cyl == 6, c("mpg", "hp")]
# create subset Y with samples of cyl = 8, keeping only mpg and hp columns
Y <- mtcars[mtcars$cyl == 8, c("mpg", "hp")]
n1 <- nrow(X); n2 <- nrow(Y)
n1;n2
```

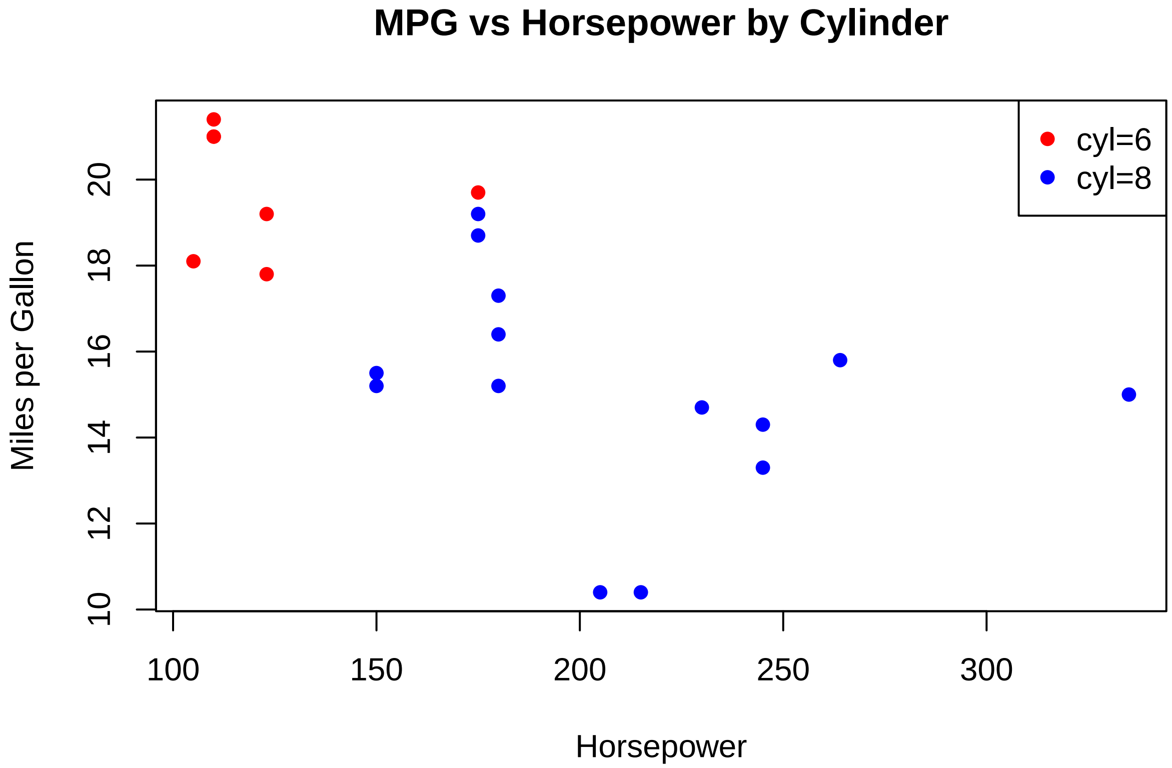
```
## [1] 7
```

```
## [1] 14
```

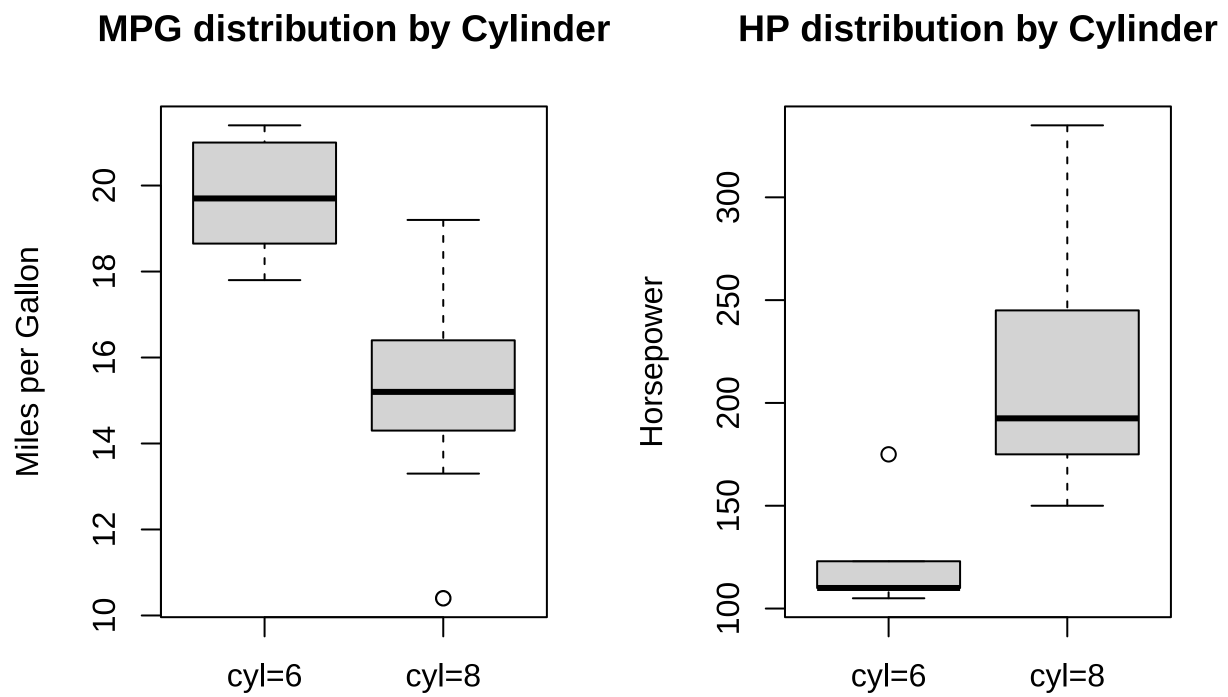
2.1 Explatory Data Analysis on differences between X and Y.

For the 2 groups, mpg (miles per gallon) and hp (horsepower), will be explored below.

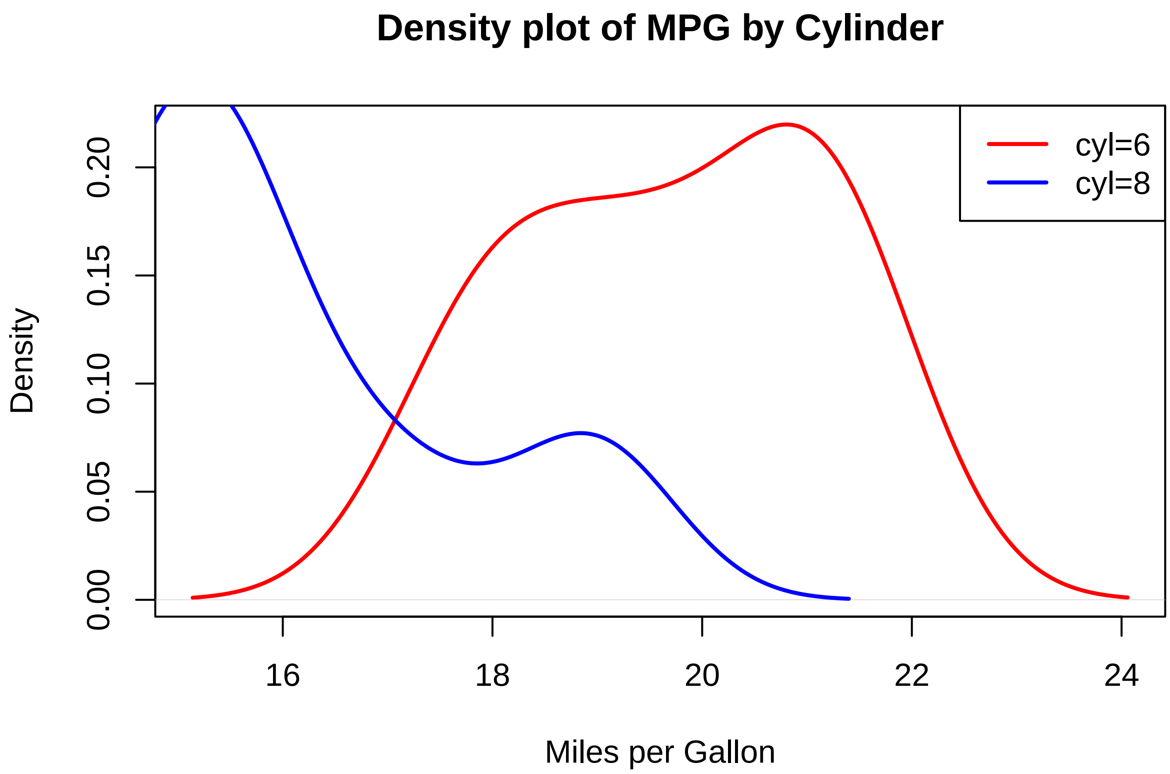
```
# scatterplot of mpg vs. hp for X and Y datasets
plot(X$hp, X$mpg, col = "red", pch = 16,
     main = "MPG vs Horsepower by Cylinder",
     xlab = "Horsepower", ylab = "Miles per Gallon",
     xlim = range(c(X$hp, Y$hp)), ylim = range(c(X$mpg, Y$mpg)))
points(Y$hp, Y$mpg, col = "blue", pch = 16)
legend("topright", legend = c("cyl=6", "cyl=8"),
       col = c("red", "blue"), pch = 16)
```



```
par(mfrow = c(1, 2))
# boxplots of mpg for X and Y datasets
boxplot(X$mpg, Y$mpg, names = c("cyl=6", "cyl=8"),
        main = "MPG distribution by Cylinder",
        ylab = "Miles per Gallon")
boxplot(X$hp, Y$hp, names = c("cyl=6", "cyl=8"),
        main = "HP distribution by Cylinder",
        ylab = "Horsepower")
```



```
par(mfrow = c(1, 1))
# density plot of mpg for X and Y datasets
plot(density(X$mpg), main = "Density plot of MPG by Cylinder",
     xlab = "Miles per Gallon", lwd = 2, col = "red")
lines(density(Y$mpg), col = "blue", lwd = 2)
legend("topright", legend = c("cyl=6", "cyl=8"),
       col = c("red", "blue"), lwd = 2)
```



Calculating mean differences among features between sample groups, sample variances, and pooled variance.

```
p=2
# calculate mean of each sample's features
mean1=matrix(colMeans(X, p, 1))
mean2=matrix(colMeans(Y, p, 1))
mean1; mean2
```

```
##           [,1]
## [1,]  19.74286
## [2,] 122.28571
```

```
##          [,1]
## [1,]  15.1000
## [2,] 209.2143
```

```
# calculate difference in means between samples' features
mean.diff = mean1-mean2
mean.diff
```

```
##            [,1]
## [1,]   4.642857
## [2,] -86.928571
```

```
# calculate each sample variance
S1=cov(X); S2=cov(Y);
# pooled variance
Sp=( (n1-1)*S1+(n2-1)*S2 )/ (n1+n2-2)
Sp
```

```
##            mpg         hp
## mpg   5.151429  -26.74135
## hp  -26.741353 1963.88346
```

Performing the Hotelling's T^2 test on X and Y.

```
Hotelling.T2(X, Y)
```

```
## $X.bar
##       mpg        hp
##  19.74286 122.28571
##
## $Y.bar
##      mpg       hp
##  15.1000 209.2143
##
## $T2
##          [,1]
## [1,] 29.62072
##
## $p.value
##              [,1]
## [1,] 0.0002125151
```

2.2

The p-value is very low, 0.0002, and so there is strong evidence to reject the null hypothesis that the means of the two groups are equal. We can conclude that the difference between the mean vectors (mpg, hp) of the two groups, 6 cylinder vehicles and 8 cylinder vehicles, is statistically significant.