

Wishart Distribution & Hotelling's T2 test

Tarek El-Hajjaoui

2023-04-29

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

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

- **Problem 1:** Choose a 3×3 covariance matrix with non-zero covariances (the off-diagonal elements should not be 0). Also 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  0.7000000 -0.1000000
## [2,]  0.7000000  2.015412  0.1000000
## [3,] -0.1000000  0.1000000  0.998936
```

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

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

```
cor(Sigma)
```

```
##           [,1]      [,2]      [,3]
## [1,]  1.0000000  0.8255654 -0.9529181
## [2,]  0.8255654  1.0000000 -0.6155827
## [3,] -0.9529181 -0.6155827  1.0000000
```

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)
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

```

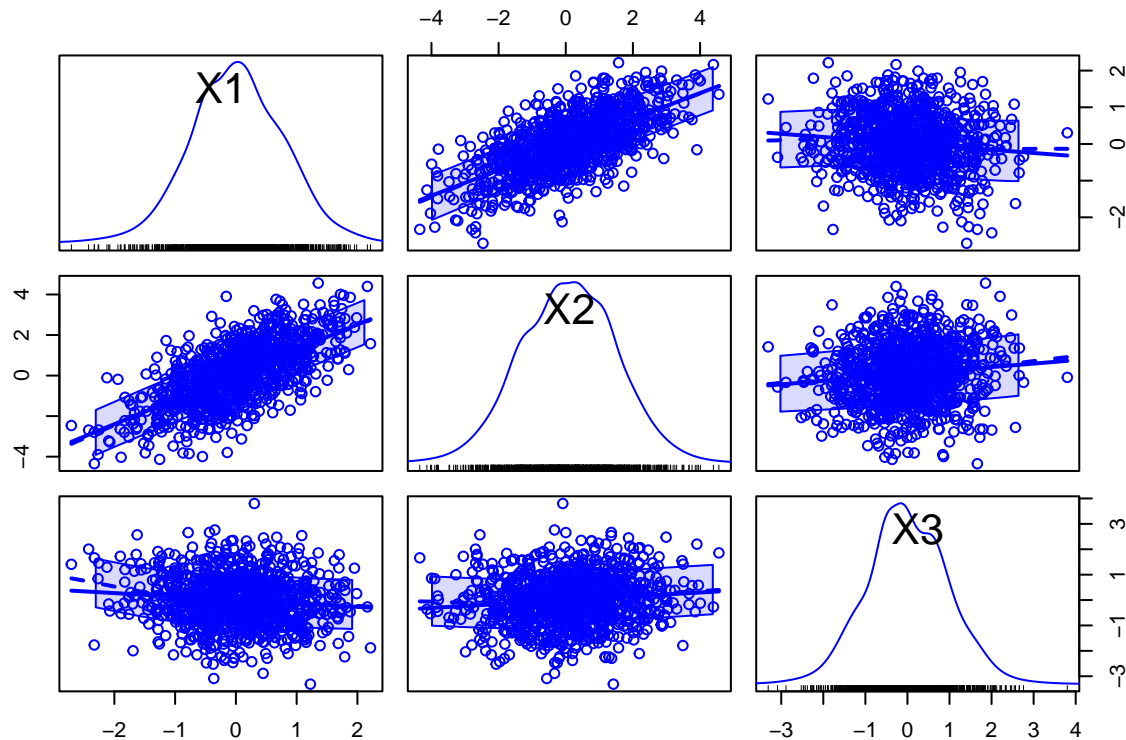
```

##           [,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 diagonal 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-diagonal 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 can 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))
```

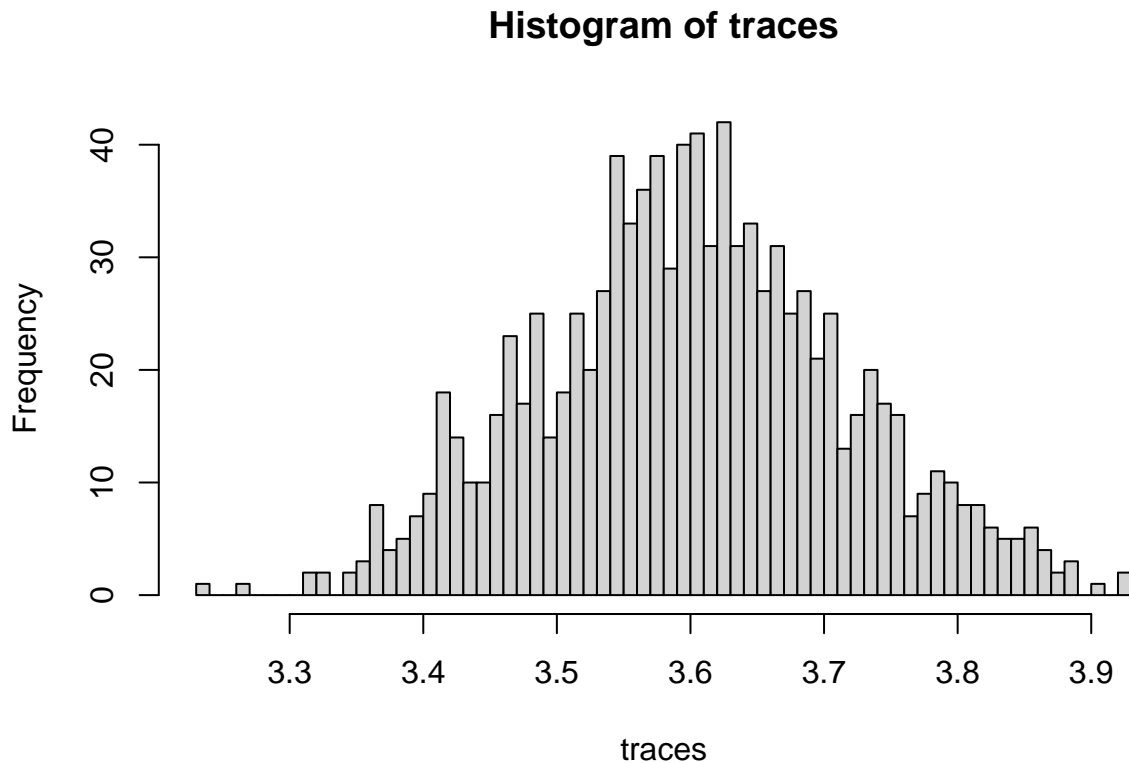
Compute and store the traces.

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

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

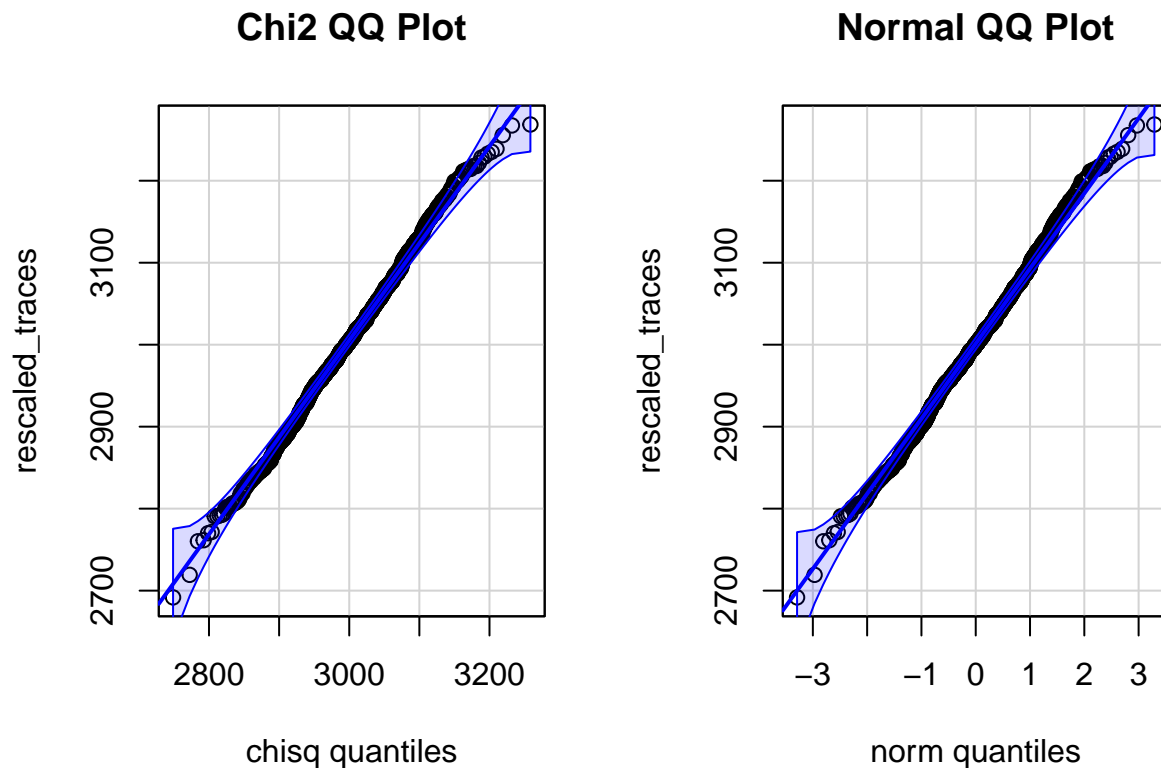
Distribution of the traces

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



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

```
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")
```



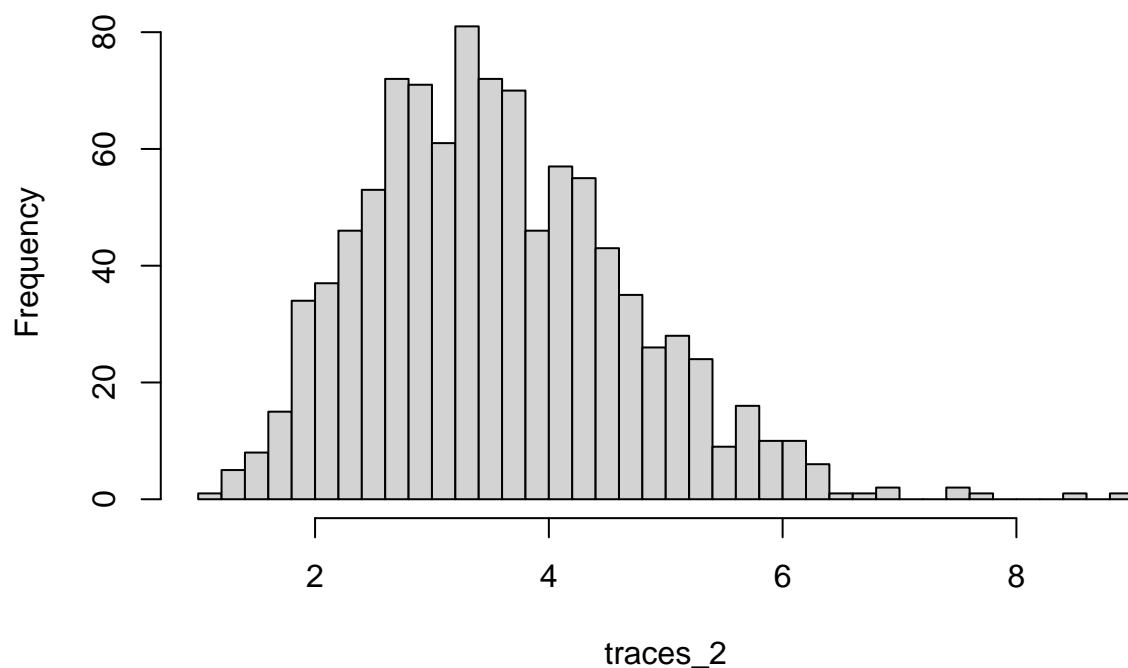
1.2 During the simulation, you will generate 1,000 Wishart distributed random matrices. Calculate the trace for each of them. Explain what distribution the traces should follow and examine their histogram.

- 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.

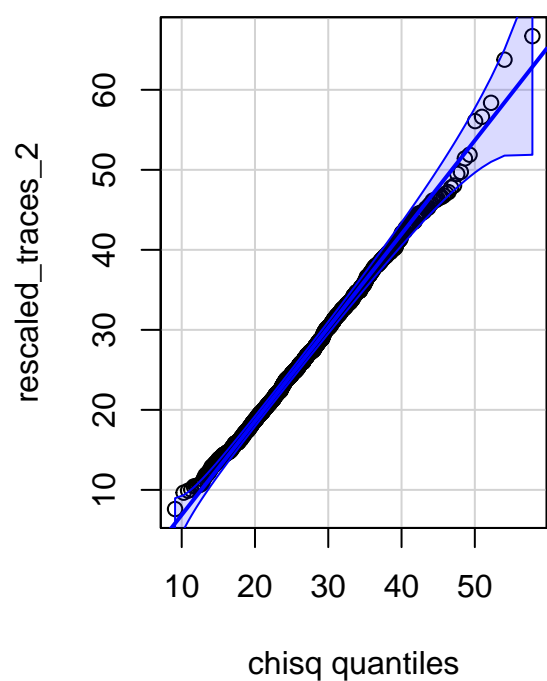
```
p <- 3
T <- 1000
n <- 10 # 100 instead of 10 this time
simulation_2 <- create_trivariate_norm(T=T, n=n, p=p, Sigma=Sigma)
S_2.array <- array(apply(simulation_2, p, cov), c(p, p, T))
traces_2 <- apply(S_2.array, p, function(x) sum(diag(x)))
hist(traces_2, breaks = 50)
```

Histogram of traces_2

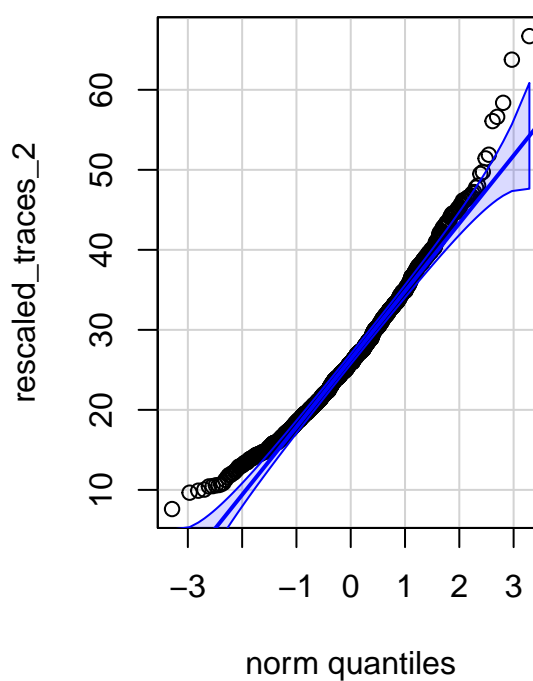


```
rescaled_traces_2 <- (n-1)*traces_2/mean(diag(Sigma))
par(mfrow = c(1, 2))
qqPlot(rescaled_traces_2, distribution = "chisq", df=3*(n-1), id = F, main = "Chi2 QQ Plot")
qqPlot(rescaled_traces_2, id = F, main = "Normal QQ Plot")
```

Chi2 QQ Plot



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.