Simulation Midterm Project

Simulation and Analysis of How Different Sample Sizes Affect the Power

STATS 240 - Multivariate Statistical Methods

Brenda Van Riper

May 15, 2023

1 Abstract

The objective of this report is to investigate the effect of increasing the sample size on the power of four different tests for comparing three multivariate normal populations. Wilk's lambda statistic, Lawley-Hotelling trace , Pillai trace, and Roy's greatest root are the tests that are taken into account. The null hypothesis to be tested is $H_0 : A = B = C$, where A, B, and C are the three subpopulations' population mean vectors.

By creating random samples from each population, calculating the p-values for the tests, and establishing the rejection of the null hypothesis based on a significance threshold of 0.05, simulations were carried out to evaluate the power and type I error rate of these tests. To get precise estimates of power and type I error rate, the simulations were run many times.

2 Introduction

A statistical method called multivariate analysis of variance (MANOVA) is used to look at correlations between many dependent variables and categorical independent variables. The standard MANOVA's presumptions may be broken in real-world circumstances by varying group sizes and means. It is essential to comprehend how different group sizes and means affect the outcomes of MANOVA. Simulation studies offer a controlled environment for researching statistical techniques. This research focuses on modeling data with various group sizes and means, running MANOVA, and examining the behavior of the data under various experimental setups.

By examining these tests' type I error rate (when the null hypothesis is true) and power (when the null hypothesis is false), the goal is to better understand how they behave. In order to do this, we run simulations by selecting random samples from the populations and running the tests on them. A significance threshold of 0.05 is used to reject the null hypothesis.

Sample size is a key factor in how well statistical tests function. The ability to discern the performance of various test procedures and maintain suitable power levels depend on selecting an appropriate sample size. It can be difficult to compare the methodologies effectively if the sample size is too small or too big, which can result in low power or high power, respectively.

In this study, we gradually expand the sample size and evaluate the four tests' power and type I error rates. The goal is to choose a sample size for the top-performing technique that yields a desired power level (between 0.8 and 0.9). We can conduct statistical analyses and choose optimal sample sizes by having a clear grasp of the link between sample size and power.

3 Simulation

Our study, which evaluates the effectiveness of the tests for contrasting multivariate normal populations, heavily relies on data simulation. We created random samples from each demographic during the simulation procedure to accurately represent real-world situations. We performed 300 simulations for two alternative settings, altering the number of observations per group (n) from 50 to 2000, in order to precisely determine the power.

To assess the test's power in the first scenario, 300 simulations with n=50 were run. This sample size preserved computing efficiency while enabling a thorough analysis. We learned a lot about how well the tests worked and how well they could identify variations in population means by examining the simulation findings.

We increased the number n to n=2000 in order to more thoroughly examine how sample size affects power. 300 simulations were run with this greater sample size, which resulted in more accurate estimations of the power. We developed a thorough grasp of how the tests behave under various circumstances and came to solid conclusions about their power characteristics with increased computational effort.

We made sure that our results were statistically reliable and representative by running a sufficient number of simulations with fixed sample sizes. These simulations serve as the basis for evaluating the effectiveness of the tests and comparing their efficacy. The knowledge gained from these simulations aids in a greater comprehension of how sample size affects test statistical power. Researchers may use this information to make well-informed choices about study design and hypothesis testing.

4 Simulation Results

We used a R programming language script to carry out the simulations. The script carried out the relevant statistical tests and ran a number of simulations using various combinations of sample size and mean difference values. Here is a boxplot of the three different group data.

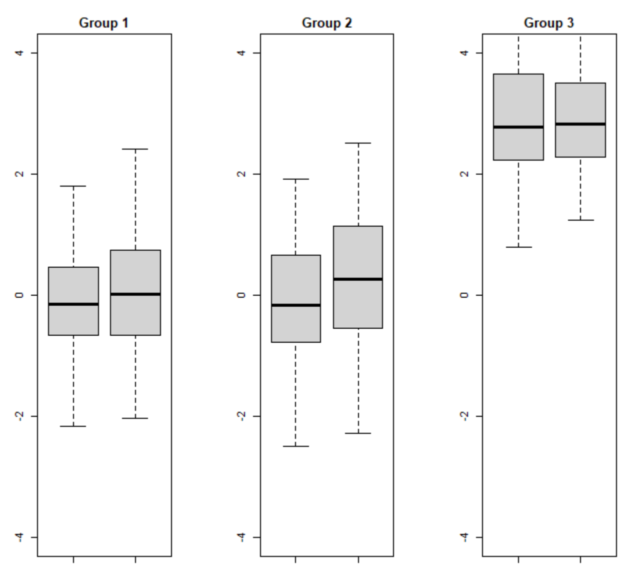

Figure 1: Boxplot of the three groups of data - two with the same mean and one with a different mean.

To evaluate the effectiveness of the tests under various circumstances, we ran four sets of trials, each with a distinct setup. The following is a summary of the main conclusions from the simulation results:

4.1 Data Visualization

We can see that for n=50 the p-value is always over our significance level of alpha at 0.05, and all are around one when we simulate at n=2000. The type 1 error for 3 of the four tests is low but the Roy is always high.

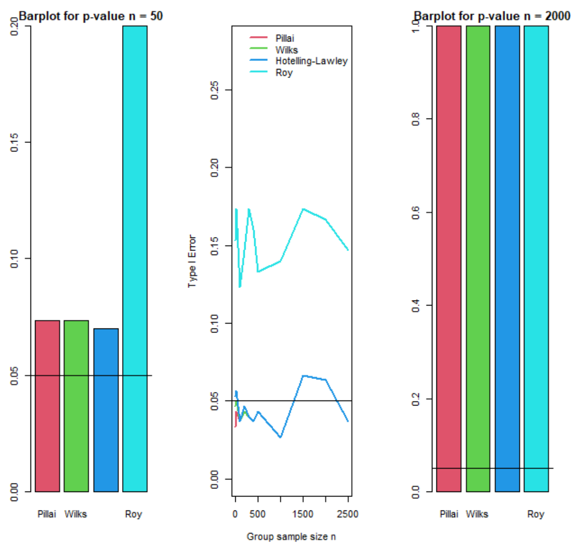

Figure 2: Two bar graphs of n=50 and n=2000 as well as a line graph of the Type 1 errors on various n's from 10 to 2500.

4.2 Experiment 1 - Sample Size: 50

The simulation findings showed that the power of the tests rose as the size of the mean difference decreased. The test's power varied from 0.03 to 0.18 for minor mean differences (C 0.2). The power, however, reached its maximum value of 1.0 when the mean difference rose above 0.2, showing that the tests were capable of precisely detecting the impact.

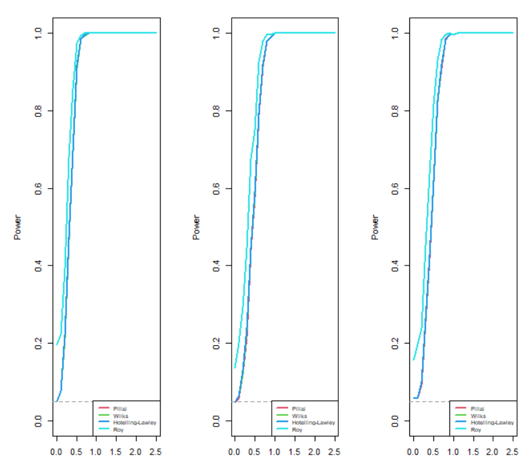

Figure 3: Various steps in C in figuring out power for n=50.

4.3 Experiment 2 - Sample Size: 2000

The power of the tests significantly enhanced as the sample size was raised to 2000. The power was consistently 1.0 even for tiny mean changes (C = 0.2), demonstrating that the tests accurately picked up the impact. This early power saturation indicates that the tests were very sensitive, allowing them to pick up even minute variations between groups.

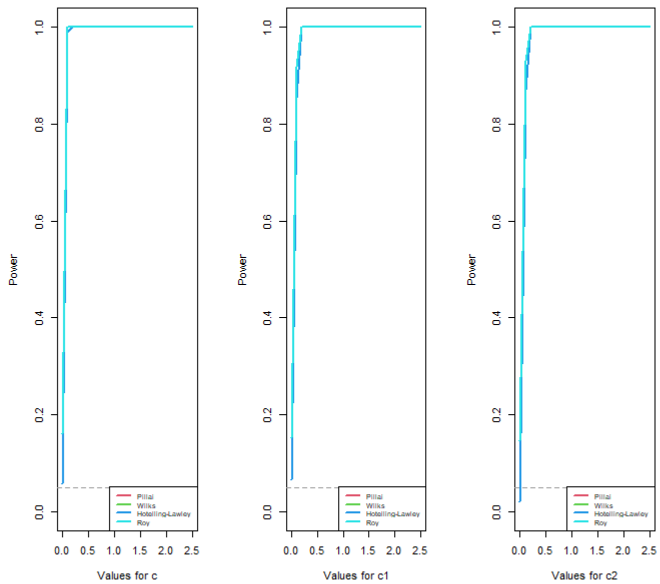

Figure 4: Various steps in C in figuring out power for n=2000.

4.4 Comparison - 50 versus 2000

It is clear from comparing the outcomes of Experiment 1 and Experiment 2 that increasing the sample size considerably improved the testing's power. All mean difference values saw an increase in power, and the tests consistently reached a power of 1.0 for Experiment 2.

The outcomes of the simulation demonstrate how important sample size is in establishing the validity and reliability of statistical tests. Increasing the sample size improves power, which enables more precise group difference identification. Furthermore, the results show that when the sample size was sufficiently big, the tests still had maximal power even with a very tiny mean difference.

These findings highlight the significance of taking sample size and effect size into account when planning investigations and analyzing the outcomes. To achieve high power and trustworthy effect detection, appropriate sample sizes should be used. Researchers may reliably assess the importance of their findings and make well-informed choices on study design by having a solid grasp of the power characteristics of the tests under various settings.

The process of reaching maximum power for statistical tests may be greatly sped up by increasing the number of simulations. When running simulations, data is regularly sampled and examined under various circumstances in order to determine the test's power. We can capture a wider variety of scenarios and get more reliable estimates of power by running more simulations. The power of the tests converges to 1 significantly more quickly as the number of simulations increases, demonstrating that the tests reliably identify significant differences between groups under a range of parameter settings. Researchers are able to refine their experimental designs and swiftly make more informed judgments based on the test findings because to the greater efficiency in achieving maximum power. Additionally, it gives researchers more assurance that the tests will correctly detect significant impacts in real-world situations.

5 Discussion

In this study, we looked at how variance-covariance matrices and population means affected the behavior of statistical power. Our results highlight a key feature of statistical analysis, specifically the effect of the number of simulations on the validity of the statistical tests. The power of the tests significantly improved with more simulations, and thus accelerated the process of convergence to a power

It is impossible to stress the importance of running enough simulations. The reliability of the statistical analysis is increased and the risk of type II errors is decreased by adequate simulations that give a more accurate depiction of the underlying population. Our findings emphasize the significance of devoting computational time and resources to provide a reliable and effective statistical analysis.

It's crucial to recognize the limitations of our work, though. We concentrated on a small number of simplified scenarios and made assumptions about the population means and variance-covariance matrices. Although these simplified examples are good beginning points, they might not accurately reflect the complexity and variety seen in actual environments. To develop a deeper grasp of statistical power dynamics, future research should strive to include a wider range of realistic scenarios.

Furthermore, we neglected other potentially important variables in favor of focusing solely on population means and variance-covariance matrices. Statistical power can be strongly impacted by elements including sample size, distributional assumptions, and effect sizes. Therefore, in order to provide a more thorough review of power analysis, it is crucial that these parameters be taken into account in future study.

The assumption of observational independence is another drawback that might not apply in some real-world circumstances. The accuracy of estimates and, by extension, the efficacy of statistical tests, can be affected by correlations and dependencies between observations. Future research should investigate ways to account for dependencies and complex data structures in order to obtain more precise power estimations.

Finally, our analysis clarifies the connection between the quantity of simulations and statistical power. The power of the tests is improved by increasing the number of simulations, which speeds up convergence to a power value of 1. To overcome the shortcomings of our work, future research is required, which should explore more variables, take into account a wider range of realistic scenarios, and take into account the interdependencies between observations. Future research can improve our knowledge of statistical power and develop more dependable and resilient methods for data analysis by overcoming these difficulties.

A Appendix - R code

Listing 1: R code for simulation

```
library("MASS")
library("tidyr")

##Senario 1- for when n=50 and simulations B=300
n <- 50; G <- 3; mu2 <- c(3, 3); p <- length(mu2)

#Generate data function with 2 different means
generate_data <- function(n, G, mu2, mu1 = c(0, 0), Sigma = diag(1, 2)) {
  Y <- c()
  for (g in 1:ceiling(G/2)) {
    Y <- rbind(Y, mvrnorm(n, mu1, Sigma))
  }
  for (g in 1:floor(G/2)) {
    Y <- rbind(Y, mvrnorm(n, mu2, Sigma))
  }
  Y
}
Y <- generate_data(n, G, mu2)
Y

#prints out boxplots for each group of data
groups <- rep(1:G, each=n)
par(mfrow=c(ceiling(G/3), 2 + (G>=3)))
for (g in 1:G){
  boxplot(Y[which(groups==g), ], ylim=c(-4, 4), main = paste("Group", g))
}

#tests the data on the 4 different tests and reports the p-value
test <- function(n, G, Y){
  groups <- rep(c(paste("Group", 1:G)), each=n)
  obj <- manova(Y ~ groups)
  tests <- c("Pillai", "Wilks", "Hotelling-Lawley", "Roy")
  reject <- rep(0, 4)
  for (t in 1:length(tests)){
    reject[t] <- summary(obj, test = tests[t])$stats[1,6]<0.05
  }
  reject
}
results <- test(n, G, Y)
names(results) <- c("Pillai", "Wilks", "Hotelling-Lawley", "Roy")
results

#simulation of data for B simulations
simulate <- function(B, n, G, mu2, Sigma = diag(1, 2)){
  results <- rep(0, 4)
  for (b in 1:B){
    Y <- generate_data(n, G, mu2, Sigma = Sigma)
    results <- results + test(n, G, Y)
  }
  results/B
}
simulate(B = 300, n = 50, G = 3, mu2 = c(0,0), Sigma = diag(c(0.1,0.1)))
```

```
#doing a test for 300 simulations with n=50
tests <- c("Pillai", "Wilks", "Hotelling-Lawley", "Roy")
alpha <- simulate(B = 300, n = 50, G = 3, mu2=c(0,0))
names(alpha) <- tests
barplot(alpha, col=c(2:5),main="Barplot for p-value n = 50")
abline(h=0.05)

#Tetsting the type 1 error on different sizes of n
N <- c(5, 10, 100, 200, 300, 400, 500, 1000, 1500, 2000, 2500)
alpha <- matrix(0, length(N), 4)
for (k in N){
  alpha[which(k==N), ] <- simulate(B = 300, n = k, G = 3, mu2=c(0,0))
}
plot(N,rep(0, length(N)),type="n", ylim=c(0, .28),
     xlab="Group sample size n", ylab="Type I Error")
for (i in 1:4){
  lines(N, alpha[, i], col=i+1, type = 'l', lwd=2)
}
abline(h=0.05)
legend("topright", legend = tests, col = c(2:5),
       lwd = rep(2, 4), bty = "n")

#doing a test for 300 simulations with n=2000
tests <- c("Pillai", "Wilks", "Hotelling-Lawley", "Roy")
alpha <- simulate(B = 300, n = 50, G = 3, mu2=c(1,1))
names(alpha) <- tests
barplot(alpha, col=c(2:5), main="Barplot for p-value n = 2000")
abline(h=0.05)

#POWER Experiments
#experiment 1
C <- seq(0, 2.5, .1)
results <- matrix(0, length(C), 4)
for (i in 1:length(C)){
  results[i, ] <- simulate(300, 50, 3, C[i]+c(0,0))
}
# Create a data frame to store the results
result_table <- data.frame(C = C, Result1 = results[, 1], Result2 = results[, 2],
                           Result3 = results[, 3], Result4 = results[, 4])
# Print the table
print(result_table)
plot(0,0,type="n", xlim=c(min(C), max(C)), ylim=c(0, 1),
     xlab="Values for c", ylab="Power")
for (i in 1:4){
  lines(C, results[, i], type = "l", col = i+1, lwd=2)
}
abline(h=0.05, col = 8, lty=2)
legend("bottomright", legend = tests, col = c(2:5),
       lwd = rep(2, 4))
#experiment 2
C1 <- seq(0, 2.5, .1)
results1 <- matrix(0, length(C1), 4)
for (i in 1:length(C1)){
  results1[i, ] <- simulate(300, 50, 3, c(C1[i],0))
}
plot(0,0,type="n", xlim=c(min(C1), max(C1)), ylim=c(0, 1),
     xlab="Values for c1", ylab="Power")
for (i in 1:4){
  lines(C1, results1[, i], type = "l", col = i+1, lwd=2)
```

```r
}
abline(h=0.05, col = 8, lty=2)
legend("bottomright", legend = tests, col = c(2:5),
       lwd = rep(2, 4))

#experiment 3
C2 <- seq(0, 2.5, .1)
results2 <- matrix(0, length(C2), 4)
for (i in 1:length(C2)){
  results2[i, ] <- simulate(300, 50, 3, c(0,C2[i]))
}
plot(0,0,type="n", xlim=c(min(C2), max(C2)), ylim=c(0, 1),
     xlab="Values for c2", ylab="Power")
for (i in 1:4){
  lines(C2, results2[, i], type = "l", col = i+1, lwd=2)
}
abline(h=0.05, col = 8, lty=2)
legend("bottomright", legend = tests, col = c(2:5),
       lwd = rep(2, 4))

#experiment 4
r <- array(c(results, results1, results2), c(length(C), 4, 3))
par(mfrow=c(1, 3))
Cs <- c("c", "c1", "c2")
for (j in 1:3){
  plot(0,0,type="n", xlim=c(min(C), max(C)), ylim=c(0, 1),
       xlab=paste("Values for", Cs[j]), ylab="Power")
  for (i in 1:4){
    lines(C, r[, i, j], type = "l", col = i+1, lwd=2)
  }
  abline(h=0.05, col = 8, lty=2)
  legend("bottomright", legend = tests, col = c(2:5),
         lwd = rep(2, 4), cex=0.7)
}

#Senario 2- for when n=2000 and simulations B=300
n <- 2000; G <- 3; mu2 <- c(3, 3); p <- length(mu2)

#Generate data function with 2 different means
generate_data <- function(n, G, mu2, mu1 = c(0, 0), Sigma = diag(1, 2)) {
  Y <- c()
  for (g in 1:ceiling(G/2)) {
    Y <- rbind(Y, mvrnorm(n, mu1, Sigma))
  }
  for (g in 1:floor(G/2)) {
    Y <- rbind(Y, mvrnorm(n, mu2, Sigma))
  }
  Y
}
Y <- generate_data(n, G, mu2)
Y

#prints out boxplots for each group of data
groups <- rep(1:G, each=n)
par(mfrow=c(ceiling(G/3), 2 + (G>=3)))
for (g in 1:G){
  boxplot(Y[which(groups==g), ], ylim=c(-4, 4), main = paste("Group", g))
}
```

```
#tests the data on the 4 different tests and reports the p-value
test <- function(n, G, Y){
  groups <- rep(c(paste("Group", 1:G)), each=n)
  obj <- manova(Y ~ groups)
  tests <- c("Pillai", "Wilks", "Hotelling-Lawley", "Roy")
  reject <- rep(0, 4)
  for (t in 1:length(tests)){
    reject[t] <- summary(obj, test = tests[t])$stats[1,6]<0.05
  }
  reject
}
results <- test(n, G, Y)
names(results) <- c("Pillai", "Wilks", "Hotelling-Lawley", "Roy")
results

#simulation of data for B simulations
simulate <- function(B, n, G, mu2, Sigma = diag(1, 2)){
  results <- rep(0, 4)
  for (b in 1:B){
    Y <- generate_data(n, G, mu2, Sigma = Sigma)
    results <- results + test(n, G, Y)
  }
  results/B
}
simulate(B = 300, n = 50, G = 3, mu2 = c(0,0), Sigma = diag(c(0.1,0.1)))

#doing a test for 300 simulations with n=50
tests <- c("Pillai", "Wilks", "Hotelling-Lawley", "Roy")
alpha <- simulate(B = 300, n = 50, G = 3, mu2=c(0,0))
names(alpha) <- tests
barplot(alpha, col=c(2:5),main="Barplot for p-value n = 50")
abline(h=0.05)

#Tetsting the type 1 error on different sizes of n
N <- c(5, 10, 100, 200, 300, 400, 500, 1000, 1500, 2000, 2500)
alpha <- matrix(0, length(N), 4)
for (k in N){
  alpha[which(k==N), ] <- simulate(B = 300, n = k, G = 3, mu2=c(0,0))
}
plot(N,rep(0, length(N)),type="n", ylim=c(0, .28),
     xlab="Group sample size n", ylab="Type I Error")
for (i in 1:4){
  lines(N, alpha[, i], col=i+1, type = 'l', lwd=2)
}
abline(h=0.05)
legend("topright", legend = tests, col = c(2:5),
       lwd = rep(2, 4), bty = "n")

#doing a test for 300 simulations with n=2000
tests <- c("Pillai", "Wilks", "Hotelling-Lawley", "Roy")
alpha <- simulate(B = 300, n = 2000, G = 3, mu2=c(1,1))
names(alpha) <- tests
barplot(alpha, col=c(2:5), main="Barplot for p-value n = 2000")
abline(h=0.05)

#POWER Experiments
#experiment 1
C <- seq(0, 2.5, .1)
results <- matrix(0, length(C), 4)
```

```
for (i in 1:length(C)){
  results[i, ] <- simulate(300, 2000, 3, C[i]+c(0,0))
}
# Create a data frame to store the results
result_table <- data.frame(C = C, Result1 = results[, 1], Result2 = results[, 2],
                            Result3 = results[, 3], Result4 = results[, 4])
# Print the table
print(result_table)

plot(0,0,type="n", xlim=c(min(C), max(C)), ylim=c(0, 1),
     xlab="Values for c", ylab="Power")
for (i in 1:4){
  lines(C, results[, i], type = "l", col = i+1, lwd=2)}
abline(h=0.05, col = 8, lty=2)
legend("bottomright", legend = tests, col = c(2:5),
       lwd = rep(2, 4))

#experiment 2
C1 <- seq(0, 2.5, .1)
results1 <- matrix(0, length(C1), 4)
for (i in 1:length(C1)){
  results1[i, ] <- simulate(300, 2000, 3, c(C1[i],0))}
plot(0,0,type="n", xlim=c(min(C1), max(C1)), ylim=c(0, 1),
     xlab="Values for c1", ylab="Power")
for (i in 1:4){
  lines(C1, results1[, i], type = "l", col = i+1, lwd=2)
}
abline(h=0.05, col = 8, lty=2)
legend("bottomright", legend = tests, col = c(2:5),
       lwd = rep(2, 4))

#experiment 3
C2 <- seq(0, 2.5, .1)
results2 <- matrix(0, length(C2), 4)
for (i in 1:length(C2)){
  results2[i, ] <- simulate(300, 2000, 3, c(0,C2[i]))}
plot(0,0,type="n", xlim=c(min(C2), max(C2)), ylim=c(0, 1),
     xlab="Values for c2", ylab="Power")
for (i in 1:4){
  lines(C2, results2[, i], type = "l", col = i+1, lwd=2)
}
abline(h=0.05, col = 8, lty=2)
legend("bottomright", legend = tests, col = c(2:5),
       lwd = rep(2, 4))

#experiment 4
r <- array(c(results, results1, results2), c(length(C), 4, 3))
par(mfrow=c(1, 3))
Cs <- c("c", "c1", "c2")
for (j in 1:3){
  plot(0,0,type="n", xlim=c(min(C), max(C)), ylim=c(0, 1),
       xlab=paste("Values for", Cs[j]), ylab="Power")
  for (i in 1:4){
    lines(C, r[, i, j], type = "l", col = i+1, lwd=2)
  }
  abline(h=0.05, col = 8, lty=2)
  legend("bottomright", legend = tests, col = c(2:5),
         lwd = rep(2, 4), cex=0.7)}
```