

STAT 625 Final Project

Jessica Grant (UCID: 30049554) and Hao Nan Wang (UCID: 30185958)

2022-12-19

Contents

Overview	2
Monte Carlo Repeats	2
Part 1. Monte Carlo Simulations when H_0 is True	3
Results	3
Part 2. Monte Carlo Simulations when H_0 is not True	4
Results	5
Part 3. Asymptotic Distributions of T_1^2 and T_2^2	7
Results for Multivariate t-distribution	8
Results for Multivariate Laplace Distribution	11
Part 4. Performance Measure to Evaluate the Asymptotic Approximation	13
References	16
Appendix	17
Additional Tables	17
Part 1 Code	24
Part 2 Code	30
Part 3 Code	46
Part 4 Code	60

Overview

This project considers 3 independent samples, each from a p-dimensional multivariate normal population. The focus of the project is to test whether the sample means are equivalent. That is, the following hypothesis test is evaluated;

$$H_0 : \mu_1 = \mu_2 \quad H_1 : \mu_1 \neq \mu_2$$

The two Hotelling's T^2 -tests used to test the above null and alternative hypotheses are defined as follows;

$$\begin{aligned} T_1^2 &= \left(\frac{1}{n_1} + \frac{1}{n_2} \right)^{-1} (\bar{X}_1 - \bar{X}_2)^T S_{pooled,12}^{-1} (\bar{X}_1 - \bar{X}_2) \\ S_{pooled,12} &= \frac{(n_1 - 1)S_1 + (n_2 - 1)S_2}{n_1 + n_2 - 2} \\ T_2^2 &= \left(\frac{1}{n_1} + \frac{1}{n_2} + \frac{1}{n_3} \right)^{-1} (\bar{X}_1 - \bar{X}_2)^T S_{pooled,123}^{-1} (\bar{X}_1 - \bar{X}_2) \\ S_{pooled,123} &= \frac{(n_1 - 1)S_1 + (n_2 - 1)S_2 + (n_3 - 1)S_3}{n_1 + n_2 + n_3 - 3} \end{aligned}$$

For both the two and the three sample tests, 2 and 3 dimensional data is considered along with the following sample size configurations;

- (i) $n_1 = n_2 = n_3$;
- (ii) $n_1 \ll n_2, n_3 = n_1$;
- (iii) $n_1 \gg n_2, n_3 = n_1$; and;
- (iv) $n_1 = n_2 = n, n_3 \gg n$.

The following analysis consists of four key parts, in part 1, a Monte Carlo simulation is performed where the sample mean values are fixed such that H_0 is true. In this part, the empirical levels of the two tests, T_1^2 and T_2^2 , are computed so that it can be determined if the tests are able to maintain the nominal 5% level. In the second part, the alternative is considered and a Monte Carlo simulation is performed where the sample mean values are fixed such that H_0 is not true. In this part, the empirical powers of both the T_1^2 and T_2^2 tests are obtained so that the more powerful test can be identified. In the third part of the analysis, the asymptotic distributions of T_1^2 and T_2^2 are explored by evaluating the impact of non-normality and the validity of the large-sample approximation. Finally in the fourth part, performance measures are considered in order to evaluate the asymptotic approximation explored in part 3.

Monte Carlo Repeats

Throughout this analysis, the number of simulation repeats are fixed at 1000. Therefore 1000 repeated samples are generated from the 3 multivariate normal populations $N_p(\mu_1, \Sigma)$, $N_p(\mu_2, \Sigma)$, and $N_p(\mu_3, \Sigma)$, with the respective sizes for (n_1, n_2, n_3) , in parts 1 and 2, and from the non-normal multivariate distributions, multivariate t-distribution and multivariate Laplace distribution, in part 3.

In parts 1 and 2, when both the T^2 tests are carried out, 1000 test results are generated for each of the varying sample sizes, and of the 1000 results, a certain number of them pertain to rejections while the rest pertain to non-rejections, these results are then used to determine the empirical levels of the two tests in part 1 and the empirical powers of the two tests in part 2. In the 3rd part, only the values of T_1^2 and T_2^2 are computed as parts 3 and 4 consider only use the test statistics values to perform their analysis.

Part 1. Monte Carlo Simulations when H_0 is True

The hypothesis test being carried out is below.

$$H_0 : \mu_1 = \mu_2 \quad H_1 : \mu_1 \neq \mu_2$$

The following two Hotelling's T^2 -tests are used to perform the hypothesis test.

$$T_1^2 = \left(\frac{1}{n_1} + \frac{1}{n_2} \right)^{-1} (\bar{X}_1 - \bar{X}_2)^T S_{pooled,12}^{-1} (\bar{X}_1 - \bar{X}_2) \sim \frac{p(n_1 + n_2 - 2)}{n_1 + n_2 - p - 1} F_{p, n_1 + n_2 - p - 1}$$

where $S_{pooled,12} = \frac{(n_1-1)S_1 + (n_2-1)S_2}{n_1 + n_2 - 2}$

$$T_2^2 = \left(\frac{1}{n_1} + \frac{1}{n_2} \right)^{-1} (\bar{X}_1 - \bar{X}_2)^T S_{pooled,123}^{-1} (\bar{X}_1 - \bar{X}_2) \sim \frac{p(n_1 + n_2 - 2)}{n_1 + n_2 - p - 1} F_{p, n_1 + n_2 - p - 1}$$

where $S_{pooled,123} = \frac{(n_1-1)S_1 + (n_2-1)S_2 + (n_3-1)S_3}{n_1 + n_2 + n_3 - 3}$

To begin, 1000 repeats were generated along with their corresponding random samples. The values of μ_1 and μ_2 were fixed such that the null hypothesis is true.

Note that the following configurations of eigenvalues and orientation angels are used to generate the rotation matrix and therefore the variance-covariance matrices.

$$p = 2 : (\lambda_1 = 1, \lambda_2 = 5, \theta = \frac{\pi}{2}) \quad p = 3 : (\lambda_1 = 1, \lambda_2 = 3, \lambda_3 = 10), (\theta_1 = \frac{\pi}{3}, \theta_2 = \frac{\pi}{3}, \theta_3 = -\frac{\pi}{3})$$

Results

$$\text{Empirical Level} : \hat{\alpha}_k(n_1, n_2, n_3 | p) = \frac{\text{number of repeats } \ni \text{ Test k rejected } H_0}{R}$$

The following tables summarize the tests results for when $p = 2$.

Table 1: Test Results for T_1^2 Tests where $p = 2$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T12 Test Rejects	51.000	55.000	45.000	42.000
T12 Test Not Rejects	949.000	945.000	955.000	958.000
T12 Test Empirical Level	0.051	0.055	0.045	0.042

Table 2: Test Results for T_2^2 Tests where $p = 2$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T22 Test Rejects	52.000	54.000	48.000	34.000
T22 Test Not Rejects	948.000	946.000	952.000	966.000
T22 Test Empirical Level	0.052	0.054	0.048	0.034

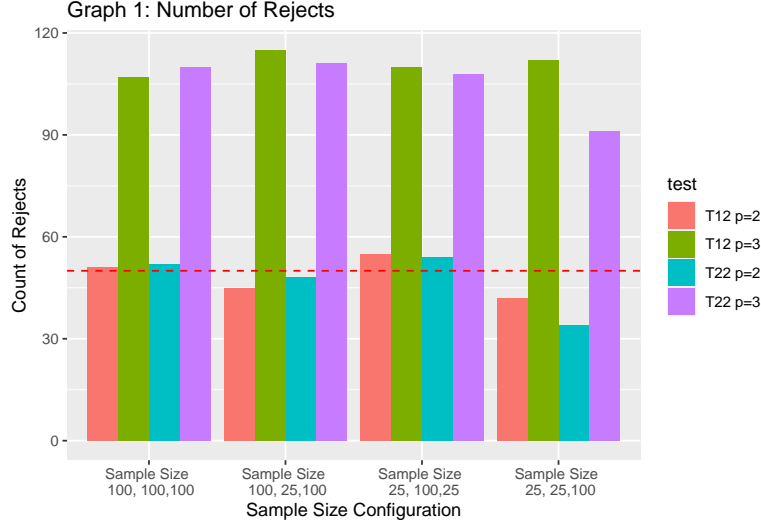
The following tables summarize the tests results for when $p = 3$.

Table 3: Test Results for T_1^2 Tests where $p = 3$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T12 Test Rejects	107.000	110.00	127.000	112.000
T12 Test Not Rejects	893.000	890.00	873.000	888.000
T12 Test Empirical Level	0.107	0.11	0.127	0.112

Table 4: Test Results for T_2^2 Tests where $p = 3$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T22 Test Rejects	110.00	108.000	121.000	91.000
T22 Test Not Rejects	890.00	892.000	879.000	909.000
T22 Test Empirical Level	0.11	0.108	0.121	0.091



Results Summary: In statistical hypothesis, $\alpha = 0.05$ is the nominal level and it is also the tests' probability of a Type I error. It is worth noting when data and sample size are given, the upper bound on probability of one of the errors can be chosen and the probability of the other one is fixed. As the empirical levels shown above, when $p=2$, the empirical levels are close to the nominal level 0.05. The difference among case iv and other cases is that case iv has a smaller sample size for x_1 and x_2 . However, one should remember that a small or large sample size does not affect Type I error as the sample size will not increase the occurrence of α . Additionally, in multivariate data, the pooled covariance matrix, which computes the average of the sample covariance matrices of the groups. If one assume they are equal, then the pooled covariance matrix is an estimate of the common covariance. One can observe that when $p=3$, the empirical level is larger than the nominal level. This is because the pooled sample variance-covariance matrix being used in T_2^2 is not multiplied by the sum of the inverses of the three sample sizes, instead, it uses two sample sizes. So this difference causes a larger pooled variance-covariance matrix. In conclusion, both tests perform better when $p=2$ in comparison with the case when $p=3$. Although the sample will not increase the α value, it affects the pooled sample variance-covariance matrix.

Part 2. Monte Carlo Simulations when H_0 is not True

The hypothesis test being carried out is below.

$$H_0 : \mu_1 = \mu_2 \quad H_1 : \mu_1 \neq \mu_2$$

The following two Hotelling's T^2 -tests are used to perform the hypothesis test.

$$T_{1,adj}^2 = \left(\frac{1}{n_1} + \frac{1}{n_2} \right)^{-1} (\bar{X}_1 - \bar{X}_2 - \delta_0)^T S_{pooled,12}^{-1} (\bar{X}_1 - \bar{X}_2 - \delta_0) \sim \frac{p(n_1 + n_2 - 2)}{n_1 + n_2 - p - 1} F_{p, n_1 + n_2 - p - 1}$$

where $S_{pooled,12} = \frac{(n_1-1)S_1 + (n_2-1)S_2}{n_1 + n_2 - 2}$

$$T_{2,adj}^2 = \left(\frac{1}{n_1} + \frac{1}{n_2} \right)^{-1} (\bar{X}_1 - \bar{X}_2 - \delta_0)^T S_{pooled,123}^{-1} (\bar{X}_1 - \bar{X}_2 - \delta_0) \sim \frac{p(n_1 + n_2 - 2)}{n_1 + n_2 - p - 1} F_{p, n_1 + n_2 - p - 1}$$

where $S_{pooled,123} = \frac{(n_1-1)S_1 + (n_2-1)S_2 + (n_3-1)S_3}{n_1 + n_2 + n_3 - 3}$

Note that the following configurations of eigenvalues and orientation angels are used to generate the rotation matrix and therefore the variance-covariance matrices.

$$p = 2 : (\lambda_1 = 1, \lambda_2 = 5, \theta = \frac{\pi}{2}) \quad p = 3 : (\lambda_1 = 1, \lambda_2 = 3, \lambda_3 = 10), (\theta_1 = \frac{\pi}{3}, \theta_2 = \frac{\pi}{3}, \theta_3 = -\frac{\pi}{3})$$

To begin, 1000 repeats were generated along with their corresponding random samples. The values of μ_1 and μ_2 were fixed such that the null hypothesis is not true and the following condition is satisfied.

$$\mu_1 - \mu_2 = \sqrt{\frac{\delta}{p} I_p} \quad \delta = 2, 4, 8, 16$$

Recall:

1. *Power* is the probability of rejecting the null hypothesis when, in fact, it is false, denoted as $1 - \beta$
2. *Type I Error* incur: reject the null hypothesis when the null hypothesis is indeed true, denoted as α
3. *Type II Error* incur: fail to reject the null hypothesis when it is false, denoted as β

The following equation will be used to compute the empirical power in this analysis.

$$\text{Empirical Power} : \hat{\gamma}_k(n_1, n_2, n_3 | \delta, p) = \frac{\text{number of repeats } \ni \text{ Test k rejected } H_0}{R}$$

We will also use the results that $F = \frac{\chi_{m_1}^2/m_1}{\chi_{m_2}^2/m_2}$ and when the numerator χ^2 is non-central, then F is called as a singly non-central F-distribution with degrees of freedom (p,n-p) and **non-centrality parameter** δ^2 , given by $\delta^2 = (\mu_1 - \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2)$.

It is also important to note that $\mu \neq \mu_0$, the **power function** $\beta_{T^2}(\mu) = P_{\mu, \Sigma}(\text{reject } H_0 | \mu)$ of a level- α Hotelling's T^2 -test with the $(1 - \alpha)^{th}$ quantile c_α of $T_{p, n_1 + n_2 - p - 1}^2$ is

$$\beta_{T^2}(\mu) = P_{\mu, \Sigma}(T^2 > c_\alpha^2 | \mu) = P_{\mu, \Sigma}\left(F > \frac{n_1 + n_2 - p - 1}{p(n_1 + n_2 - 1)} \cdot c_\alpha^2 | \mu\right)$$

Finally, note that: $ncp = \delta^2 = (\mu_1 - \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2) = \delta_0^T \Sigma^{-1} \delta_0 = f(\delta)$ and $\delta = 2, 4, 8, 16$.

Results

The following summarizes the tests results for when p = 2.

Table 5: Test Results for T12 Test where p = 2 and delta = 2

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T12 Test Rejects	51.000	55.000	45.000	42.000
T12 Test Not Rejects	949.000	945.000	955.000	958.000
T12 Test Empirical Power	0.051	0.055	0.045	0.042

Table 6: Test Results for T22 Test where $p = 2$ and $\delta = 2$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T22 Test Rejects	52.000	54.000	48.000	34.000
T22 Test Not Rejects	948.000	946.000	952.000	966.000
T22 Test Empirical Power	0.052	0.054	0.048	0.034

The following summarizes the tests results for when $p = 3$.

Table 7: Test Results for T12 Test where $p = 3$ and $\delta = 2$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T12 Test Rejects	107.000	110.00	115.000	112.000
T12 Test Not Rejects	893.000	890.00	885.000	888.000
T12 Test Empirical Power	0.107	0.11	0.115	0.112

Table 8: Test Results for T22 Test where $p = 3$ and $\delta = 2$

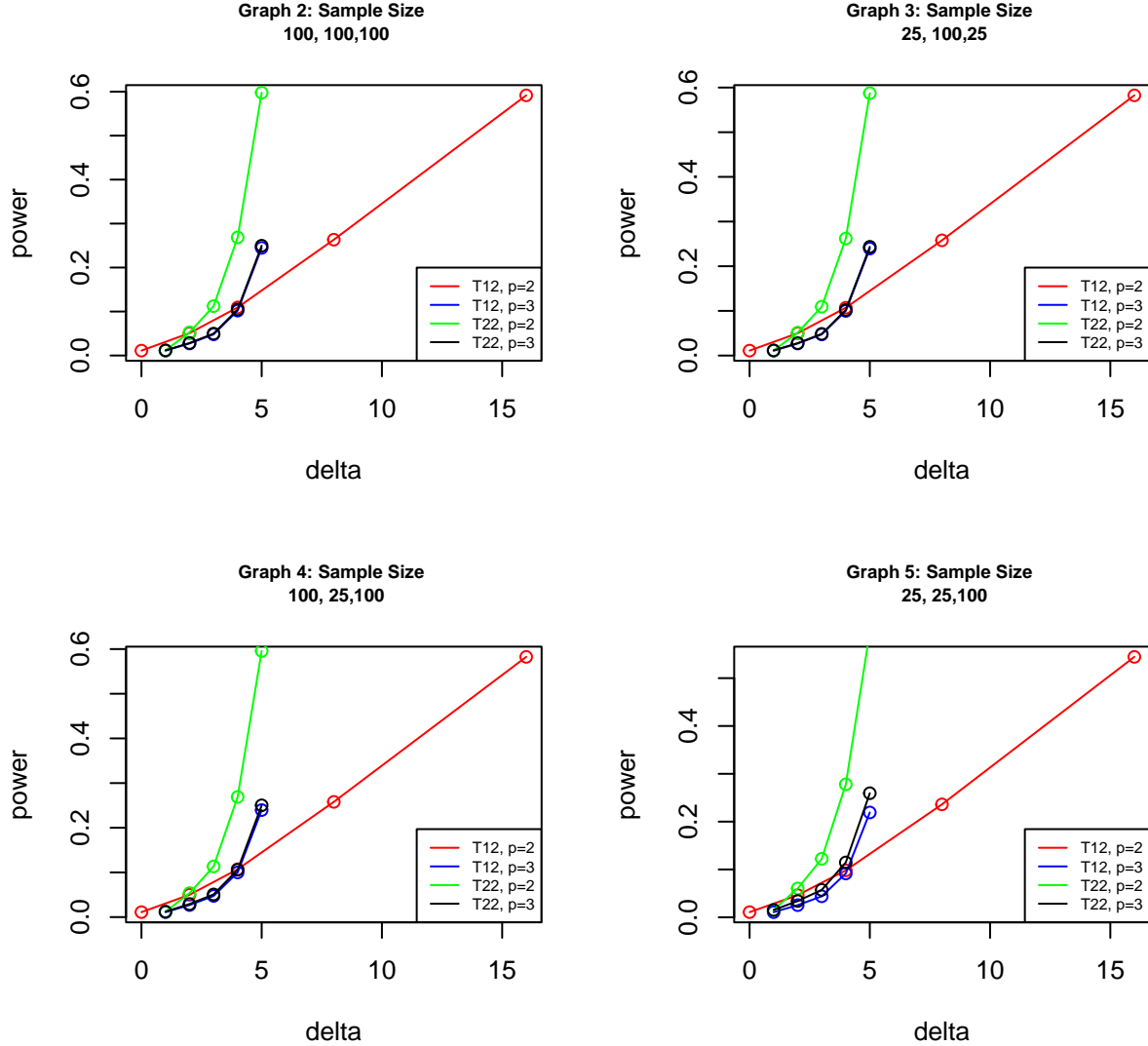
	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T22 Test Rejects	110.00	108.000	111.000	91.000
T22 Test Not Rejects	890.00	892.000	889.000	909.000
T22 Test Empirical Power	0.11	0.108	0.111	0.091

The table below summarizes the empirical powers computed with both the T_1^2 and T_2^2 tests and with 2 and 3 parameters. As expected, the results show that as δ increases, power increases as well.

Table 9: Empirical Power Results

	T12 $p = 2$	T22 $p = 2$	T12 $p = 3$	T22 $p = 3$
Delta = 0	0.01	0.01	0.01	0.01
Delta = 2	0.05	0.05	0.03	0.03
Delta = 4	0.11	0.11	0.05	0.05
Delta = 8	0.26	0.27	0.10	0.10
Delta = 16	0.59	0.60	0.24	0.25

The following displays graphically how the the empirical power computed by the power function changes as delta increases for the various sample size configurations.



Results Summary: The singly non central F distribution is stochastically increasing in its non-centrality parameter δ^2 . This highlights that as μ moves farther and farther away from μ_0 (i.e., as δ^2 increases and H_0 is rendered more and more false), Hotelling's T^2 test becomes more and more powerful. One can observe that when $\delta = 0$ (i.e., no ncp), the power of both tests are pretty small; however, the power increases with δ . In fact, the NCP parameter tells something about the statistical power of a test. When ncp=0, it means that the F distribution is a central F distribution. As the NCP increases, the shape of the F-distribution shifts to the right. This also means that a larger percentage of the curve moves to the right of the critical value for α . The result is an increase in statistical power. When $p=3$, one can observe that the power of T_1^2 increases significantly and it is the fastest among all tests. So it is fair to conclude that T_1^2 is the more powerful than T_2^2 .

Part 3. Asymptotic Distributions of T_1^2 and T_2^2

In this part, the impact of non-normality on the distributions of T_1^2 and T_2^2 is explored as well as the validity of the large-sample approximation of their distributions. The non-normal distributions considered are the multivariate t-distribution and the multivariate Laplace distribution, both of which belong to the family of elliptical distributions.¹ The multivariate t-distribution is symmetric about its mean similarly to the MVN, but it has thicker tails than the MVN distribution for small degrees of freedom, and accommodates outliers

and extreme observations much better.¹ Similarly, the multivariate Laplace distribution is also symmetric and has thicker tails than the MVN distribution but it is similar to the MVN distribution in that it is specified by its location vector and its covariance matrix.¹

As shown by the results to follow, the same sample size configurations, means and variance-covariance matrices that were used in previous parts are used again here. 1000 Monte Carlo repeats were also considered in this part to calculate the T_1^2 and T_2^2 statistics and their empirical distributions were examined.

Large-sample approximation claims that if n is sufficiently large, then asymptotic theory can be used to obtain an asymptotic null distribution for T^2 to approximate its exact null distribution to be $T^2 \sim^a \chi_p^2$. This can be shown as follows.

We know from part 2 that $T^2 \sim \frac{p(n-1)}{n-p} F_{p,n-p}$ and it follows that when we let $F \sim F_{p,n-p}$ we can write,

$$\frac{p(n-1)}{n-p} F \stackrel{D}{=} \left(\frac{p(n-1)}{n-p} \right) \frac{\chi_1^2/p}{\chi_2^2/(n-p)} = \frac{(n-1)}{(n-p)} \frac{\chi_1^2}{\chi_2^2/(n-p)}$$

and by the Weak Law of Large Numbers we get,

$$\frac{1}{n-p} \chi_2^2 \stackrel{D}{=} \frac{1}{n-p} \sum_{i=1}^{n-p} U_i = \bar{U} \rightarrow^p 1$$

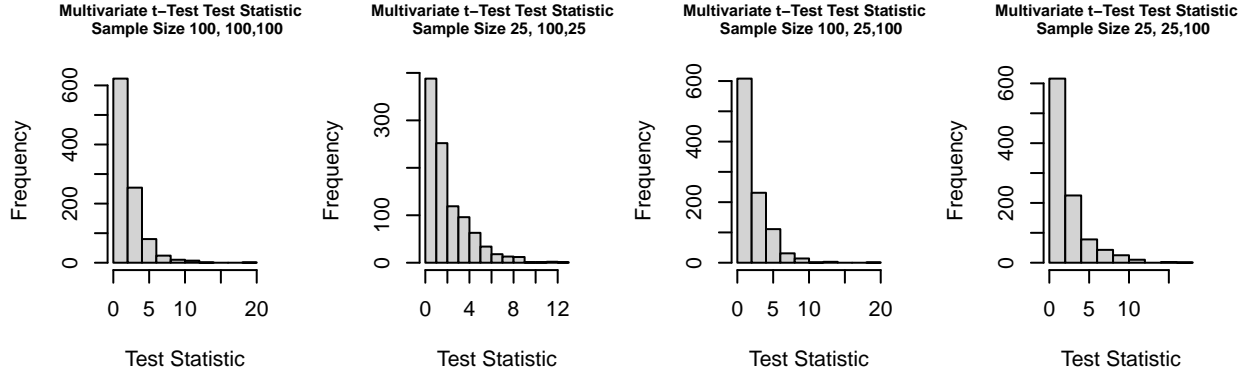
where $U_1, \dots, U_{n-p} \sim^{iid} \chi_1^2$ which by Slutsky's Theorem shows that $T^2 \sim^a \chi_p^2$ for large n as desired.² As explained in Lecture 2, slide 7, this result holds true even without the assumption of multivariate normality.²

It is because of this that when performing the following analysis, the expectation was to arrive at a distribution of the test statistics for T_1^2 and T_2^2 that is chi-squared with degrees of freedom equal to the number of parameters of the data which can be verified by checking the mean of the distributions as for chi-squared the mean is equal to the degrees of freedom.

Results for Multivariate t-distribution

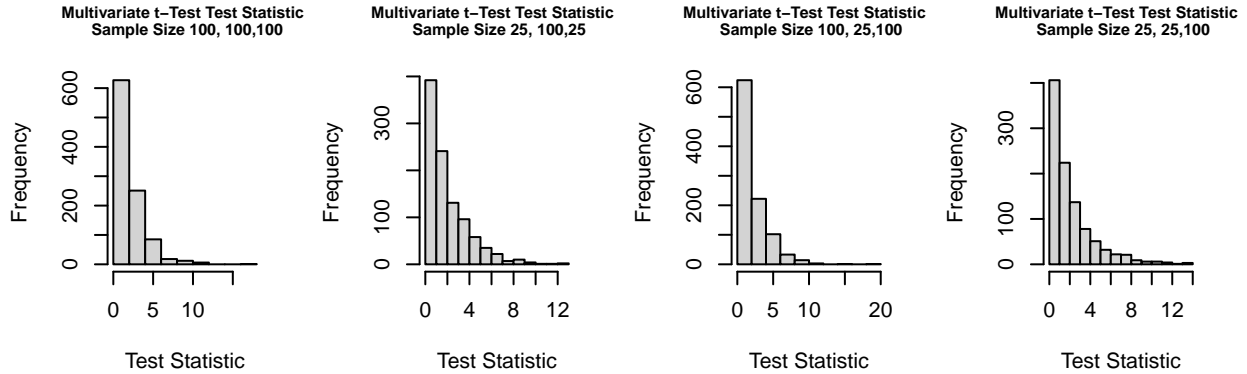
Below are the results for the T_1^2 test where $p = 2$ and the data was generated from the multivariate t-distribution. As shown, the histograms for each sample size configuration roughly follows that expected from a chi-squared distribution as expected. The numeric summary which summarizes the empirical distributions further supports this by displaying that the means of the various sample sizes further converges to the value 2 as the sample sizes increase. This is inline with our expectations as the mean of the chi-squared distribution is equal to its degrees of freedom which in this case is 2.

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
Min.	0.00	0.01	0.00	0.00
1st Qu.	0.62	0.59	0.64	0.61
Median	1.45	1.39	1.48	1.49
Mean	1.99	2.01	2.12	2.20
3rd Qu.	2.72	2.87	3.01	2.98
Max.	18.92	12.12	18.70	17.01



Below are the results for the T_2^2 test where $p = 2$ and the data was generated from the multivariate t-distribution. Similar to the results for the T_1^2 test, the histograms indicate that the expectation that the test statistics follow a chi-squared distribution is reasonable and again the means of the empirical distributions as expected converge to 2 as the sample sizes increase which is the desired result as it indicates that the large-sample approximation holds true and that the expectation that the chi-squared distribution would have degrees of freedom equal to the number of parameters is also accurate.

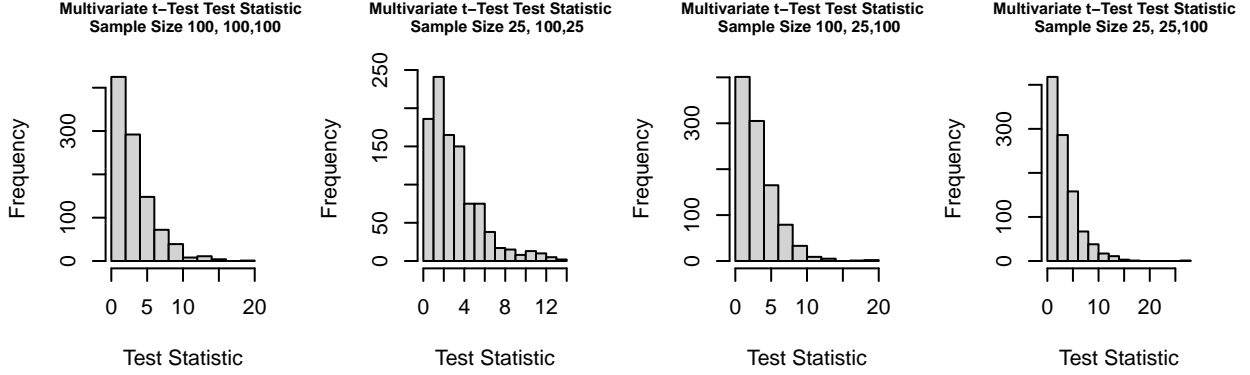
	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
Min.	0.0	0.01	0.00	0.00
1st Qu.	0.6	0.59	0.64	0.61
Median	1.4	1.39	1.47	1.41
Mean	2.0	2.01	2.11	2.13
3rd Qu.	2.8	2.84	3.02	2.81
Max.	17.8	12.46	18.37	13.70



The summary statistics tables above display the first and the third quantiles of the test statistics generated however a more detailed comparison of the distribution's upper tails for both the T_1^2 and T_2^2 tests was also considered by generating and evaluating their 0.85th, 0.90th, 0.95th, 0.975th, 0.99th and 0.995th quantiles for each test with each sample size configuration. These results were stored in tables 54 and 55 in the appendix.

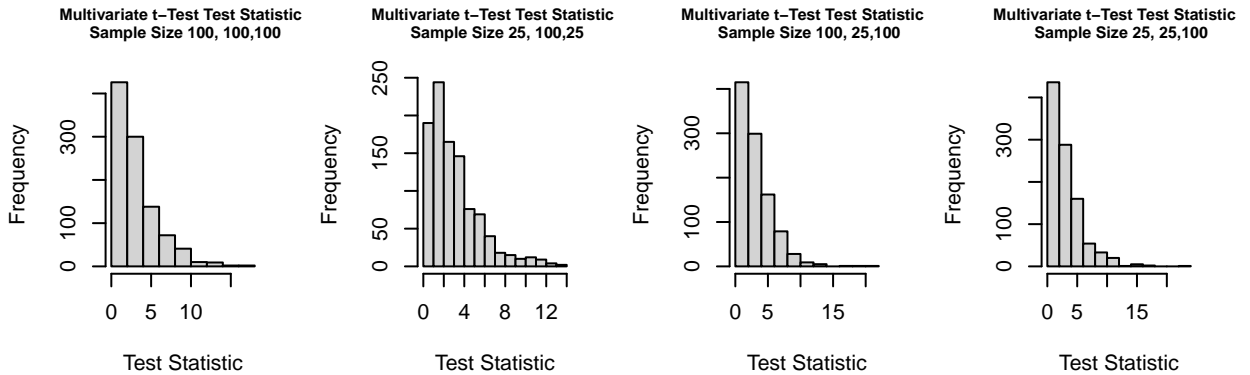
Below are the results for the T_1^2 test where $p = 3$ and the data was generated from the multivariate t-distribution. Again, the histograms indicate the desired chi-squared distribution with the means in the empirical distributions summary table confirming again the expectation that the degrees of freedom of the distribution is equal to the number of parameters, in this case 3.

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
Min.	0.04	0.01	0.02	0.01
1st Qu.	1.24	1.27	1.27	1.25
Median	2.41	2.44	2.58	2.42
Mean	3.15	3.04	3.16	3.25
3rd Qu.	4.27	4.08	4.38	4.42
Max.	18.46	13.65	19.92	26.83



Below are the results for the T_2^2 Test where $p = 3$ and the data was generated from the multivariate t -distribution. The results here are very similar to those generated with the T_1^2 test. When we consider the 0.85th, 0.90th, 0.95th, 0.975th, 0.99th and 0.995th quantiles for for both the T_1^2 and T_2^2 tests with each sample size configuration as shown in tables 56 and 57 in the appendix, we can see that their similarities are equally apparent there with differences in quantile values only differing by a few decimal points and the test that generates the higher quantile values is different depending on which quantile being considered. As a generalization it appears that the quantiles for the T_1^2 test are higher for the 0.85th, 0.90th, 0.95th and 0.99th quantiles while the T_2^2 test has higher quantile values for the 0.975th and 0.995th quantiles.

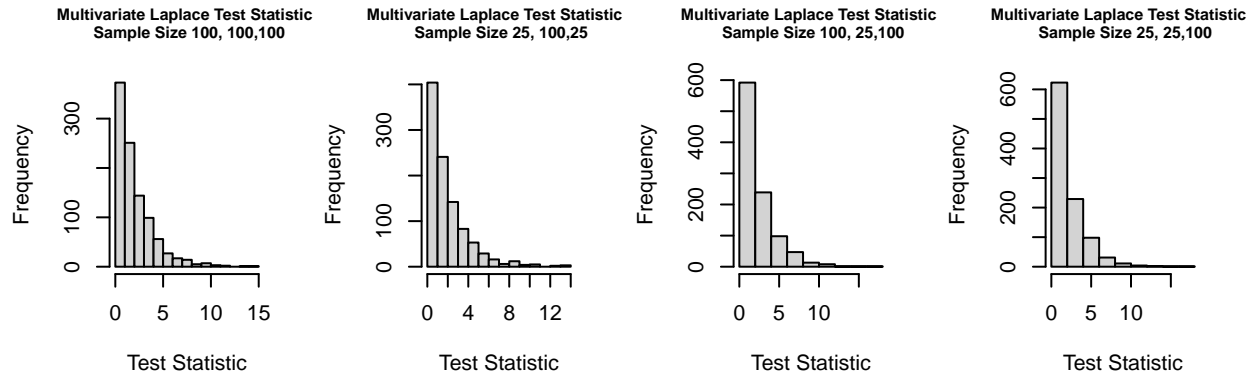
	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
Min.	0.04	0.01	0.02	0.01
1st Qu.	1.24	1.31	1.29	1.23
Median	2.38	2.41	2.56	2.32
Mean	3.14	3.03	3.11	3.08
3rd Qu.	4.32	4.05	4.33	4.22
Max.	17.92	13.61	21.53	22.10



Results for Multivariate Laplace Distribution

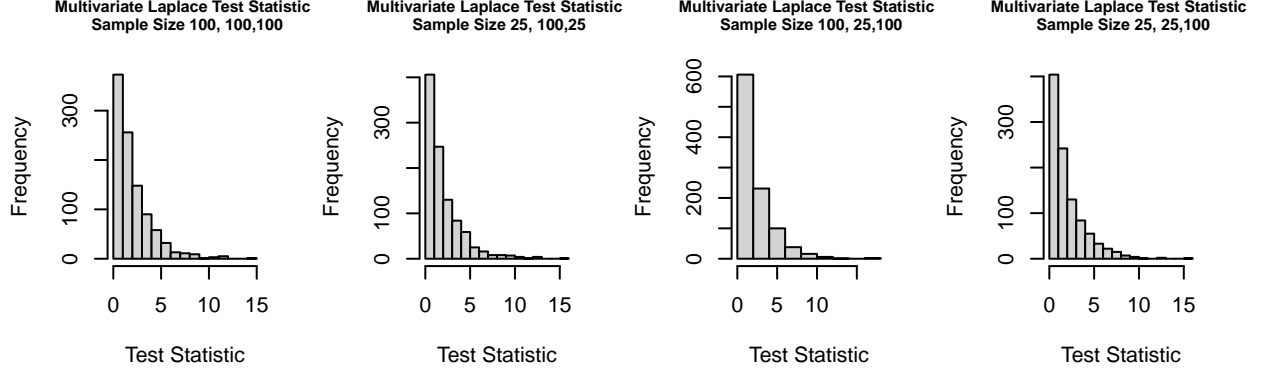
Below are the results for the T_1^2 test where $p = 2$ and the data was generated from the multivariate Laplace distribution. As shown, much like all the results for the multivariate t-distribution generated data's test statistics, the results here with the multivariate Laplace distribution also demonstrate a chi-squared distribution in the histograms as well as the corresponding degrees of freedom of 2 as shown by the mean in the empirical distribution summary statistics table. Again, 2 is the desired degrees of freedom as it is equal to the number of parameters.

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
Min.	0.00	0.00	0.01	0.00
1st Qu.	0.62	0.55	0.74	0.59
Median	1.47	1.32	1.55	1.41
Mean	2.03	1.98	2.25	2.06
3rd Qu.	2.83	2.72	3.01	2.81
Max.	14.25	13.85	17.48	17.66



Below are the results for the T_2^2 Test where $p = 2$ and the data was generated from the multivariate Laplace distribution. Similar again to the results and patterns observed with the multivariate t-distribution, the histograms display the expected chi-squared distribution and the empirical distributions means converges to 2 as the sample size increases which is again the desired result.

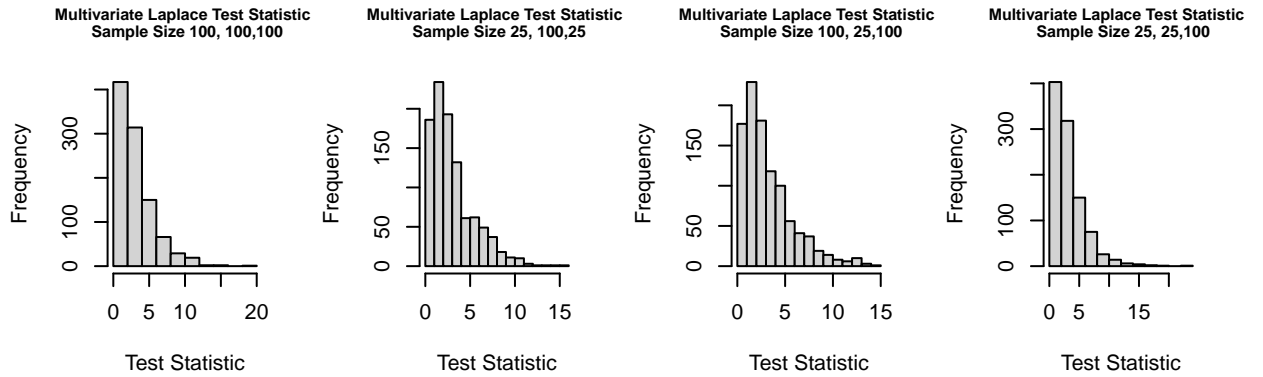
	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
Min.	0.00	0.00	0.01	0.00
1st Qu.	0.63	0.55	0.73	0.57
Median	1.43	1.33	1.58	1.36
Mean	2.00	1.97	2.20	1.97
3rd Qu.	2.83	2.78	3.02	2.74
Max.	14.92	15.14	17.44	15.02



As before, a more detailed comparison of the distribution's upper tails for both the T_1^2 and T_2^2 tests was also considered by generating and evaluating their 0.85th, 0.90th, 0.95th, 0.975th, 0.99th and 0.995th quantiles for each test with each sample size configuration. These results are stored in tables 58 and 59 in the appendix.

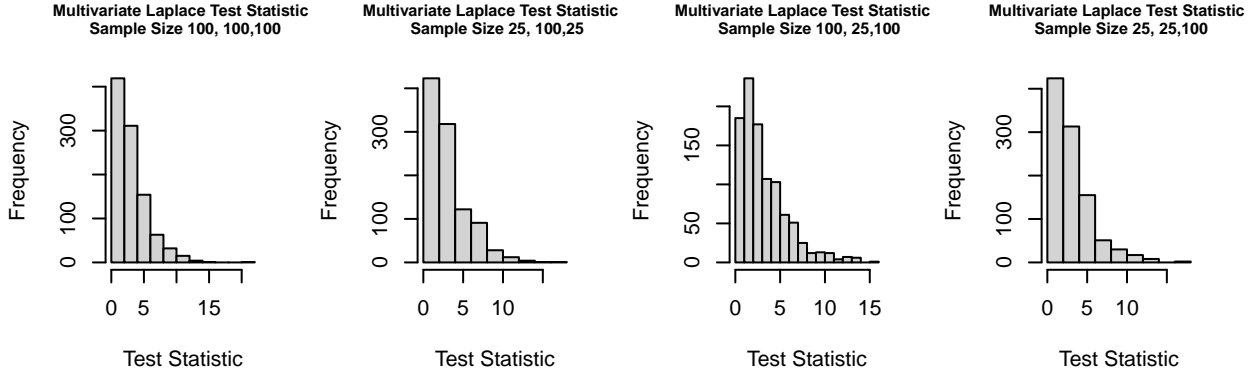
Below are the results for the T_1^2 test where $p = 3$ and the data was generated from the multivariate Laplace distribution. The results follow from all the past results with histograms depicting a chi-squared distribution and the empirical distributions mean converging to the number of parameters as the sample sizes increase which is consistent with our expectations.

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
Min.	0.01	0.0	0.01	0.01
1st Qu.	1.23	1.2	1.27	1.33
Median	2.39	2.4	2.55	2.51
Mean	3.07	3.1	3.23	3.21
3rd Qu.	4.18	4.0	4.45	4.27
Max.	19.79	15.1	14.10	22.46



Below are the results for the T_2^2 Test where $p = 3$ and the data was generated from the multivariate Laplace distribution. These results follow very closely to those found with the T_1^2 test. A further comparison of the two sets of results was performed by considering the distribution's 0.85th, 0.90th, 0.95th, 0.975th, 0.99th and 0.995th quantiles for each test with each sample size configuration. These results are stored in tables 60 and 61 that can be found in the appendix. These results further support that the T_1^2 and T_2^2 tests are very similar for the multivariate Laplace distribution with $p=3$.

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
Min.	0.01	0.0	0.01	0.01
1st Qu.	1.23	1.3	1.26	1.28
Median	2.37	2.4	2.46	2.38
Mean	3.05	3.0	3.17	3.07
3rd Qu.	4.18	4.1	4.36	4.10
Max.	20.47	16.2	15.26	17.27



Results Summary: In summary, all of the generated test statistic results were in line with our expectations, each showing a chi-squared distribution with degrees of freedom that converges to the number of parameters as sample size increases. The performance of each of these multivariate non-normal datasets along with the performance multivariate normal datasets are further evaluated and compared in Part 4 below.

Part 4. Performance Measure to Evaluate the Asymptotic Approximation

$$bias(\hat{q}_h) = \hat{q}_h - q_h \text{ and } RB(\hat{q}_h) = \frac{bias(\hat{q}_h)}{q_h} \times 100$$

Below is a summary of the bias and relative bias results for the T_1^2 test where $p = 2$ and the data was generated from the multivariate normal distribution. The equivalent table for where $p = 3$ can be found in the appendix as Table 57.

Table 18: Bias and RB for T_1^2 Test where $p = 2$ for Multivariate Normal Distribution

	Sample Size 100, 100,100		Sample Size 25, 100, 25		Sample Size 100, 25, 100		Sample Size 25, 25, 100	
	Bias	RB	Bias	RB	Bias	RB	Bias	RB
85%	0.10	2.5	0.15	4.0	-0.11	-3.02	0.29	7.53
90%	0.11	2.5	0.09	1.9	-0.08	-1.65	0.22	4.67
95%	0.28	4.7	0.32	5.3	-0.03	-0.42	0.02	0.41
97.5%	0.56	7.5	0.76	10.3	-0.03	-0.42	1.34	18.18
99%	1.15	12.4	1.22	13.2	-0.70	-7.60	1.63	17.72
99.5%	0.59	5.6	1.39	13.1	-1.06	-9.96	2.15	20.29

Below is a summary of the bias and relative bias results for the T_2^2 test where $p = 2$ and the data was generated from the multivariate normal distribution. The equivalent table for where $p = 3$ can be found in the appendix as Table 58.

Table 19: Bias and RB for T_2^2 Test where $p = 2$ for Multivariate Normal Distribution

	Sample Size 100, 100,100		Sample Size 25, 100, 25		Sample Size 100, 25, 100		Sample Size 25, 25, 100	
	Bias	RB	Bias	RB	Bias	RB	Bias	RB
85%	0.01	0.15	0.07	1.86	-0.08	-2.12	0.05	1.23
90%	0.10	2.21	-0.03	-0.64	-0.04	-0.97	0.06	1.39
95%	0.25	4.12	0.39	6.43	0.13	2.11	-0.04	-0.68
97.5%	0.36	4.94	0.84	11.33	-0.31	-4.16	0.54	7.27
99%	0.67	7.27	1.16	12.62	-0.64	-6.94	0.74	8.00
99.5%	0.95	8.97	1.06	10.03	-1.26	-11.93	1.06	9.98

Below is a summary of the bias and relative bias results for the T_1^2 test where $p = 2$ and the data was generated from the multivariate t-distribution. The equivalent table for where $p = 3$ can be found in the appendix as Table 59.

Table 20: Bias and RB for T_1^2 Test where $p = 2$ for Multivariate T Distribution

	Sample Size 100, 100,100		Sample Size 25, 100, 25		Sample Size 100, 25, 100		Sample Size 25, 25, 100	
	Bias	RB	Bias	RB	Bias	RB	Bias	RB
85%	-1.4	-38	-1.3	-34	-1.3	-35	-1.4	-36
90%	-1.5	-32	-1.4	-31	-1.4	-31	-1.4	-31
95%	-1.6	-26	-1.6	-27	-1.5	-25	-1.7	-29
97.5%	-1.8	-24	-2.0	-28	-1.4	-19	-1.8	-25
99%	-2.5	-28	-2.4	-26	-1.2	-13	-2.4	-26
99.5%	-1.8	-17	-2.0	-19	-1.3	-13	-2.8	-26

Below is a summary of the bias and relative bias results for the T_2^2 test where $p = 2$ and the data was generated from the multivariate t-distribution. The equivalent table for where $p = 3$ can be found in the appendix as Table 60.

Table 21: Bias and RB for T_2^2 Test where $p = 2$ for Multivariate T Distribution

	Sample Size 100, 100,100		Sample Size 25, 100, 25		Sample Size 100, 25, 100		Sample Size 25, 25, 100	
	Bias	RB	Bias	RB	Bias	RB	Bias	RB
85%	-1.4	-36	-1.3	-34	-1.3	-35	-1.6	-42
90%	-1.5	-32	-1.5	-33	-1.5	-33	-1.7	-37
95%	-1.6	-27	-1.7	-29	-1.4	-23	-1.9	-32
97.5%	-1.9	-26	-1.9	-26	-1.8	-25	-2.2	-30
99%	-2.7	-29	-2.5	-27	-1.3	-14	-2.0	-22
99.5%	-1.8	-17	-1.9	-18	-1.9	-18	-2.6	-25

Below is a summary of the bias and relative bias results for the T_1^2 test where $p = 2$ and the data was generated from the multivariate Laplace distribution. The equivalent table for where $p = 3$ can be found in the appendix as Table 61.

Table 22: Bias and RB for T_1^2 Test where $p = 2$ for Multivariate Laplace Distribution

	Sample Size 100, 100,100		Sample Size 25, 100, 25		Sample Size 100, 25, 100		Sample Size 25, 25, 100	
	Bias	RB	Bias	RB	Bias	RB	Bias	RB
85%	-0.11	-2.78	-0.02	-0.43	0.46	12.2	0.15	3.95
90%	0.00	-0.06	-0.15	-3.24	0.58	12.6	0.09	2.06
95%	-0.09	-1.57	-0.21	-3.43	0.58	9.7	0.00	0.06
97.5%	0.08	1.15	0.71	9.66	0.41	5.6	0.11	1.44
99%	0.42	4.54	0.66	7.21	1.09	11.8	0.40	4.29
99.5%	-0.37	-3.51	0.17	1.64	1.16	10.9	0.29	2.69

Below is a summary of the bias and relative bias results for the T_2^2 test where $p = 2$ and the data was generated from the multivariate Laplace distribution. The equivalent table for where $p = 3$ can be found in the appendix as Table 62.

Table 23: Bias and RB for T_2^2 Test where $p = 2$ for Multivariate Laplace Distribution

	Sample Size 100, 100,100		Sample Size 25, 100, 25		Sample Size 100, 25, 100		Sample Size 25, 25, 100	
	Bias	RB	Bias	RB	Bias	RB	Bias	RB
85%	-0.20	-5.4	-0.10	-2.55	0.37	9.7	0.04	0.95
90%	-0.11	-2.4	-0.21	-4.48	0.47	10.3	0.01	0.30
95%	-0.27	-4.5	-0.06	-0.99	0.41	6.8	0.08	1.28
97.5%	-0.13	-1.8	0.38	5.19	0.59	7.9	-0.10	-1.37
99%	-0.22	-2.4	0.60	6.53	0.37	4.0	-0.56	-6.12
99.5%	0.41	3.9	0.91	8.61	0.68	6.4	-1.09	-10.26

Results Summary: As the asymptotic theory states, if the sample size increases, one can observe that the asymptotic null distribution of T_1^2 and the distribution of T_2^2 approximate its exact null distribution, this result, $T^2 \sim \chi_p^2$, was confirmed in Part 3. In terms of the tables shown above, it is observed that the multivariate Laplace distribution generated the smallest quantile bias', the multivariate normal distribution generated the second smallest quantile bias' and the multivariate T-distribution generated the largest quantile bias'. That said, each of these distributions still generated reasonably favorable results by achieving a distribution that roughly followed a chi-squared distribution with degrees of freedom equal to the number of parameters. It is worth nothing that achieving any unbiased results was not an objective of this analysis as only relatively small sizes with only 1000 Monte Carlo Simulation Repeats ($R=1000$) were considered and performed. Ideally, increasing the number of repeats and increasing the sample sizes should help get more accurate results for each of the distributions considered to generate the data.

References

1. De Leon, A. (n.d.). Project Description and Due Date. Lecture. <https://d2l.ucalgary.ca/d2l/le/content/468188/viewContent/5620262/View>
2. De Leon, A. (n.d.). Lecture 2. Lecture. <https://d2l.ucalgary.ca/d2l/le/content/468188/viewContent/5566203/View>
3. Stephanie. (2021, May 28). Non centrality parameter (NCP). Statistics How To. Retrieved December 21, 2022, from <https://www.statisticshowto.com/non-centrality-parameter-ncp/>

Appendix

Additional Tables

Tables from Part 1

Table 24: Part 1: Upper Quantiles for T_1^2 Test where $p = 2$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
85%	3.9	4.0	3.7	4.1
90%	4.7	4.7	4.5	4.8
95%	6.3	6.3	6.0	6.0
97.5%	7.9	8.1	7.3	8.7
99%	10.4	10.4	8.5	10.8
99.5%	11.2	12.0	9.5	12.8

Table 25: Part 1: Upper Quantiles for T_2^2 Test where $p = 2$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
85%	3.8	3.9	3.7	3.8
90%	4.7	4.6	4.6	4.7
95%	6.2	6.4	6.1	6.0
97.5%	7.7	8.2	7.1	7.9
99%	9.9	10.4	8.6	9.9
99.5%	11.6	11.7	9.3	11.7

Table 26: Part 1: Upper Quantiles for T_1^2 Test where $p = 3$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
85%	5.2	5.3	5.6	5.8
90%	6.4	6.4	6.5	6.8
95%	8.1	7.8	8.2	8.4
97.5%	10.1	9.3	9.7	9.7
99%	11.8	11.3	11.3	12.1
99.5%	13.7	11.8	12.5	13.8

Table 27: Part 1: Upper Quantiles for T_2^2 Test where $p = 3$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
85%	5.1	5.4	5.4	5.4
90%	6.4	6.3	6.4	6.3
95%	8.3	7.9	8.2	7.6
97.5%	10.0	9.5	9.5	8.9
99%	11.8	11.1	11.1	10.5
99.5%	12.9	11.8	12.4	12.4

Tables from Part 2

Table 28: Test Results for T12 Test where $p = 2$ and $\delta = 2$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T12 Test Rejects	51.00	55.00	45.00	42.00
T12 Test Not Rejects	949.00	945.00	955.00	958.00
T12 Test Empirical Power	0.05	0.06	0.04	0.04

Table 29: Test Results for T22 Test where $p = 2$ and $\delta = 2$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T22 Test Rejects	52.00	54.00	48.00	34.00
T22 Test Not Rejects	948.00	946.00	952.00	966.00
T22 Test Empirical Power	0.05	0.05	0.05	0.03

Table 30: Test Results for T12 Test where $p = 2$ and $\delta = 4$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T12 Test Rejects	51.00	55.00	45.00	42.00
T12 Test Not Rejects	949.00	945.00	955.00	958.00
T12 Test Empirical Power	0.05	0.06	0.04	0.04

Table 31: Test Results for T22 Test where $p = 2$ and $\delta = 4$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T22 Test Rejects	52.00	54.00	48.00	34.00
T22 Test Not Rejects	948.00	946.00	952.00	966.00
T22 Test Empirical Power	0.05	0.05	0.05	0.03

Table 32: Test Results for T12 Test where $p = 2$ and $\delta = 8$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T12 Test Rejects	51.00	55.00	45.00	42.00
T12 Test Not Rejects	949.00	945.00	955.00	958.00
T12 Test Empirical Power	0.05	0.06	0.04	0.04

Table 33: Test Results for T22 Test where $p = 2$ and $\delta = 8$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T22 Test Rejects	52.00	54.00	48.00	34.00
T22 Test Not Rejects	948.00	946.00	952.00	966.00
T22 Test Empirical Power	0.05	0.05	0.05	0.03

Table 34: Test Results for T12 Test where $p = 2$ and $\delta = 16$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T12 Test Rejects	51.00	55.00	45.00	42.00
T12 Test Not Rejects	949.00	945.00	955.00	958.00
T12 Test Empirical Power	0.05	0.06	0.04	0.04

Table 35: Test Results for T22 Test where $p = 2$ and $\delta = 16$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T22 Test Rejects	52.00	54.00	48.00	34.00
T22 Test Not Rejects	948.00	946.00	952.00	966.00
T22 Test Empirical Power	0.05	0.05	0.05	0.03

Table 36: Upper Quantiles for T122 Test where $p = 2$ and $\delta = 2$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
85%	3.9	4.0	3.7	4.1
90%	4.7	4.7	4.5	4.8
95%	6.3	6.3	6.0	6.0
97.5%	7.9	8.1	7.3	8.7
99%	10.4	10.4	8.5	10.8
99.5%	11.2	12.0	9.5	12.8

Table 37: Upper Quantiles for T22 Test where $p = 2$ and $\delta = 2$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
85%	3.8	3.9	3.7	3.8
90%	4.7	4.6	4.6	4.7
95%	6.2	6.4	6.1	6.0
97.5%	7.7	8.2	7.1	7.9
99%	9.9	10.4	8.6	9.9
99.5%	11.6	11.7	9.3	11.7

Table 38: Test Results for T12 Test where $p = 3$ and $\delta = 2$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T12 Test Rejects	107.00	110.00	115.00	112.00
T12 Test Not Rejects	893.00	890.00	885.00	888.00
T12 Test Empirical Power	0.11	0.11	0.12	0.11

Table 39: Test Results for T22 Test where $p = 3$ and $\delta = 2$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T22 Test Rejects	110.00	108.00	111.00	91.00
T22 Test Not Rejects	890.00	892.00	889.00	909.00
T22 Test Empirical Power	0.11	0.11	0.11	0.09

Table 40: Test Results for T12 Test where $p = 3$ and $\delta = 4$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T12 Test Rejects	107.00	110.00	115.00	112.00
T12 Test Not Rejects	893.00	890.00	885.00	888.00
T12 Test Empirical Power	0.11	0.11	0.12	0.11

Table 41: Test Results for T22 Test where $p = 3$ and $\delta = 4$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T22 Test Rejects	110.00	108.00	111.00	91.00
T22 Test Not Rejects	890.00	892.00	889.00	909.00
T22 Test Empirical Power	0.11	0.11	0.11	0.09

Table 42: Test Results for T12 Test where $p = 3$ and $\delta = 8$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T12 Test Rejects	107.00	110.00	115.00	112.00
T12 Test Not Rejects	893.00	890.00	885.00	888.00
T12 Test Empirical Power	0.11	0.11	0.12	0.11

Table 43: Test Results for T22 Test where $p = 3$ and $\delta = 8$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T22 Test Rejects	110.00	108.00	111.00	91.00
T22 Test Not Rejects	890.00	892.00	889.00	909.00
T22 Test Empirical Power	0.11	0.11	0.11	0.09

Table 44: Test Results for T12 Test where $p = 3$ and $\delta = 2$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T12 Test Rejects	107.00	110.00	115.00	112.00
T12 Test Not Rejects	893.00	890.00	885.00	888.00
T12 Test Empirical Power	0.11	0.11	0.12	0.11

Table 45: Test Results for T22 Test where $p = 3$ and $\delta = 2$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
T22 Test Rejects	110.00	108.00	111.00	91.00
T22 Test Not Rejects	890.00	892.00	889.00	909.00
T22 Test Empirical Power	0.11	0.11	0.11	0.09

Table 46: Upper Quantiles for T122 Test where $p = 3$ and $\delta = 2$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
85%	5.2	5.3	5.6	5.8
90%	6.4	6.4	6.5	6.8
95%	8.1	7.8	8.2	8.4
97.5%	10.1	9.3	9.7	9.7
99%	11.8	11.3	11.3	12.1
99.5%	13.7	11.8	12.5	13.8

Table 47: Upper Quantiles for T22 Test where $p = 3$ and $\delta = 2$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
85%	5.1	5.4	5.4	5.4
90%	6.4	6.3	6.4	6.3
95%	8.3	7.9	8.2	7.6
97.5%	10.0	9.5	9.5	8.9
99%	11.8	11.1	11.1	10.5
99.5%	12.9	11.8	12.4	12.4

Tables from Part 3

Table 48: Multivariate T Distribution Upper Quantiles for Test T12 with $p = 2$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
85%	2.4	2.5	2.5	2.4
90%	3.1	3.2	3.2	3.2
95%	4.4	4.4	4.5	4.3
97.5%	5.6	5.3	6.0	5.5
99%	6.7	6.8	8.0	6.8
99.5%	8.8	8.6	9.3	7.8

Table 49: Multivariate T Distribution Upper Quantiles for Test T22 with $p = 2$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
85%	2.4	2.5	2.5	2.2
90%	3.1	3.1	3.1	2.9
95%	4.4	4.3	4.6	4.0
97.5%	5.4	5.4	5.6	5.2
99%	6.5	6.7	7.9	7.2
99.5%	8.8	8.7	8.6	8.0

Table 50: Multivariate T Distribution Upper Quantiles for Test T12 with $p = 3$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
85%	3.7	3.8	3.7	3.9
90%	4.5	4.4	4.5	4.8
95%	6.0	5.3	5.7	6.0
97.5%	7.7	6.7	6.6	7.7
99%	9.8	9.1	9.0	9.3
99.5%	11.7	10.9	9.9	9.9

Table 51: Multivariate T Distribution Upper Quantiles for Test T32 with $p = 3$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
85%	3.7	3.8	3.6	3.7
90%	4.3	4.4	4.5	4.4
95%	5.9	5.3	5.7	5.8
97.5%	7.8	6.8	6.7	6.7
99%	9.6	8.8	8.7	8.3
99.5%	11.8	10.2	9.3	9.3

Table 52: Multivariate Laplace Distribution Upper Quantiles for Test T12 with $p = 2$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
85%	3.7	3.8	4.3	3.9
90%	4.6	4.5	5.2	4.7
95%	5.9	5.8	6.6	6.0
97.5%	7.5	8.1	7.8	7.5
99%	9.6	9.9	10.3	9.6
99.5%	10.2	10.8	11.8	10.9

Table 53: Multivariate Laplace Distribution Upper Quantiles for Test T22 with $p = 2$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
85%	3.6	3.7	4.2	3.8
90%	4.5	4.4	5.1	4.6
95%	5.7	5.9	6.4	6.1
97.5%	7.2	7.8	8.0	7.3
99%	9.0	9.8	9.6	8.7
99.5%	11.0	11.5	11.3	9.5

Table 54: Multivariate Laplace Distribution Upper Quantiles for Test T12 with $p = 3$

	Sample Size 100, 100,100	Sample Size 25, 100,25	Sample Size 100, 25,100	Sample Size 25, 25,100
85%	5.4	5.7	5.8	5.6
90%	6.3	6.5	7.0	6.5
95%	8.2	7.8	8.5	8.3
97.5%	9.4	9.2	10.2	10.2
99%	11.0	10.7	12.3	12.6
99.5%	11.7	11.7	12.7	15.7

Table 55: Multivariate Laplace Distribution Upper Quantiles for Test T22 with $p = 3$

	Sample Size 100, 100,100		Sample Size 25, 100,25		Sample Size 100, 25,100		Sample Size 25, 25,100	
85%	5.5		5.7		5.7		5.4	
90%	6.2		6.6		6.6		6.2	
95%	8.2		7.8		8.8		8.3	
97.5%	9.4		9.4		10.5		10.3	
99%	11.2		10.8		12.4		11.9	
99.5%	12.0		12.1		13.6		12.4	

Tables from Part 4

Table 56: Bias and RB for T_1^2 Test where $p = 3$ for Multivariate Normal Distribution

	Sample Size 100, 100,100		Sample Size 25, 100, 25		Sample Size 100, 25, 100		Sample Size 25, 25, 100	
	Bias	RB	Bias	RB	Bias	RB	Bias	RB
85%	-0.16	-3.1	0.04	0.70	0.29	5.4	0.44	8.3
90%	0.17	2.7	0.11	1.82	0.26	4.2	0.54	8.7
95%	0.31	4.0	0.03	0.34	0.41	5.3	0.62	7.9
97.5%	0.70	7.5	0.00	0.01	0.31	3.3	0.37	4.0
99%	0.49	4.3	0.01	0.07	-0.06	-0.5	0.76	6.7
99.5%	0.84	6.6	-1.05	-8.19	-0.38	-3.0	0.92	7.1

Table 57: Bias and RB for T_2^2 Test where $p = 3$ for Multivariate Normal Distribution

	Sample Size 100, 100,100		Sample Size 25, 100, 25		Sample Size 100, 25, 100		Sample Size 25, 25, 100	
	Bias	RB	Bias	RB	Bias	RB	Bias	RB
85%	-0.21	-3.92	0.05	0.88	0.08	1.5	0.10	1.8
90%	0.18	2.89	0.09	1.43	0.17	2.8	0.07	1.1
95%	0.52	6.60	0.12	1.59	0.36	4.5	-0.21	-2.7
97.5%	0.61	6.54	0.14	1.47	0.15	1.6	-0.40	-4.3
99%	0.42	3.74	-0.23	-2.01	-0.28	-2.4	-0.84	-7.4
99.5%	0.07	0.55	-1.01	-7.84	-0.40	-3.1	-0.42	-3.3

Table 58: Bias and RB for T_1^2 Test where $p = 3$ for Multivariate T Distribution

	Sample Size 100, 100,100		Sample Size 25, 100, 25		Sample Size 100, 25, 100		Sample Size 25, 25, 100	
	Bias	RB	Bias	RB	Bias	RB	Bias	RB
85%	-1.6	-29.7	-1.5	-28	-1.6	-30	-1.4	-26
90%	-1.8	-28.7	-1.9	-30	-1.8	-28	-1.4	-23
95%	-1.8	-22.8	-2.5	-32	-2.1	-26	-1.8	-23
97.5%	-1.7	-17.8	-2.6	-28	-2.8	-30	-1.7	-18
99%	-1.5	-13.4	-2.2	-20	-2.3	-20	-2.0	-18
99.5%	-1.1	-8.5	-1.9	-15	-2.9	-23	-2.9	-23

Table 59: Bias and RB for T_2^2 Test where $p = 3$ for Multivariate T Distribution

	Sample Size 100, 100,100		Sample Size 25, 100, 25		Sample Size 100, 25, 100		Sample Size 25, 25, 100	
	Bias	RB	Bias	RB	Bias	RB	Bias	RB
85%	-1.6	-30.0	-1.5	-28	-1.7	-32	-1.6	-30
90%	-1.9	-30.4	-1.9	-30	-1.8	-29	-1.8	-29
95%	-1.9	-24.8	-2.5	-33	-2.1	-27	-2.0	-26
97.5%	-1.6	-16.8	-2.5	-27	-2.7	-29	-2.6	-28
99%	-1.8	-15.8	-2.6	-23	-2.7	-24	-3.0	-27
99.5%	-1.0	-7.8	-2.6	-21	-3.5	-28	-3.5	-27

Table 60: Bias and RB for T_1^2 Test where $p = 3$ for Multivariate Laplace Distribution

	Sample Size 100, 100,100		Sample Size 25, 100, 25		Sample Size 100, 25, 100		Sample Size 25, 25, 100	
	Bias	RB	Bias	RB	Bias	RB	Bias	RB
85%	0.10	0.02	0.41	7.67	0.44	8.27	0.27	5.1
90%	0.08	0.01	0.25	4.02	0.71	11.40	0.27	4.3
95%	0.37	0.05	0.04	0.47	0.66	8.41	0.51	6.6
97.5%	0.07	0.01	-0.20	-2.11	0.85	9.06	0.87	9.3
99%	-0.36	-0.03	-0.69	-6.07	0.95	8.42	1.30	11.4
99.5%	-1.18	-0.09	-1.17	-9.11	-0.12	-0.95	2.84	22.1

Table 61: Bias and RB for T_2^2 Test where $p = 3$ for Multivariate Laplace Distribution

	Sample Size 100, 100,100		Sample Size 25, 100, 25		Sample Size 100, 25, 100		Sample Size 25, 25, 100	
	Bias	RB	Bias	RB	Bias	RB	Bias	RB
85%	0.17	3.18	0.35	6.67	0.34	6.5	0.10	1.9
90%	-0.01	-0.12	0.33	5.29	0.36	5.8	-0.07	-1.2
95%	0.40	5.12	0.03	0.42	1.00	12.8	0.49	6.3
97.5%	0.08	0.86	0.07	0.73	1.15	12.3	0.98	10.5
99%	-0.15	-1.29	-0.56	-4.95	1.02	9.0	0.56	4.9
99.5%	-0.80	-6.22	-0.74	-5.77	0.74	5.8	-0.41	-3.2

Part 1 Code

```
## theta
theta = pi / 2
## rotation matrix
rotation.matrix.2.2 = matrix(c(cos(theta),-sin(theta), sin(theta), cos(theta)), 2, 2)
## lambda
lambda1 = 1
lambda2 = 5
## sigma matrix
sigma.2.2 = rotation.matrix.2.2 %*% matrix(c(lambda1, 0, 0, lambda2),
                                             2, 2) %*% t(rotation.matrix.2.2)

## theta
theta1 = pi / 3
theta2 = pi / 3
```



```

theta3 = -pi / 3
## rotation matrix
rotation.matrix.3.3 = matrix(
  c(
    cos(theta1) * cos(theta3) - cos(theta2) * sin(theta1) * sin(theta3), -cos(theta2) *
      cos(theta3) * sin(theta1) - cos(theta1) * sin(theta3),
    sin(theta1) * sin(theta2),
    cos(theta3) * sin(theta1) + cos(theta1) * cos(theta2) * sin(theta3),
    cos(theta1) * cos(theta2) * cos(theta3) - sin(theta1) * sin(theta3), -cos(theta1) *
      sin(theta2),
    sin(theta2) * sin(theta3),
    cos(theta3) * sin(theta2),
    cos(theta2)
  ),
  3,
  3,
  byrow = TRUE
)
## lambda
lambda1 = 1
lambda2 = 3
lambda3 = 10
## sigma matrix
sigma.3.3 = rotation.matrix.3.3 %*% matrix(c(lambda1, 0, 0, 0, lambda2, 0, 0, 0, lambda3),
                                             3, 3) %*% t(rotation.matrix.3.3)

library(matlib)
library(mvtnorm)
# R updated version after optimization
T1_square.all = c()
T2_square.all = c()
F.value.t1 = c()
P.value.t1 = c()
F.value.t2 = c()
P.value.t2 = c()
generator.t2 = function(mu1, mu2, mu3, n1, n2, n3) {
  ## mu where mu1 = mu2 = mu0 = 0 and we set mu3 = 10
  set.seed(625)
  for (i in 1:1000) {
    if (length(mu1) == 2) {
      x1 = rmvnorm(n = n1,
                  mean = mu1,
                  sigma = sigma.2.2)
      x2 = rmvnorm(n = n2,
                  mean = mu2,
                  sigma = sigma.2.2)
      x3 = rmvnorm(n = n3,
                  mean = mu3,
                  sigma = sigma.2.2)
    }
    else {
      x1 = rmvnorm(n = n1,
                  mean = mu1,
                  sigma = sigma.3.3)
    }
  }
}

```

```

    x2 = rmvnorm(n = n2,
                mean = mu2,
                sigma = sigma.3.3)
    x3 = rmvnorm(n = n3,
                mean = mu3,
                sigma = sigma.3.3)
}

S_pool.2.2 = ((n1 - 1) * cov(x1) + (n2 - 1) * cov(x2)) / (n1 + n2 -
2)

T1_square = (1 / n1 + 1 / n2) ^ (-1) * t(colMeans(x1) - colMeans(x2)) %*%
inv(S_pool.2.2) %*%
(colMeans(x1) - colMeans(x2))
T1_square.all[i] = T1_square
F.value.t1 = (n1 + n2 - 2 - 1) / (2 * (n1 + n2 - 2)) * T1_square
P.value.t1 = cbind(P.value.t1, 1 - pf(F.value.t1, 2, n1 + n2 - 2 - 1))

S_pool.3.3 = ((n1 - 1) * cov(x1) + (n2 - 1) * cov(x2) + (n3 - 1) * cov(x3)) /
(n1 + n2 + n3 - 3)
T2_square = (1 / n1 + 1 / n2) ^ (-1) * t(colMeans(x1) - colMeans(x2)) %*%
inv(S_pool.3.3) %*%
(colMeans(x1) - colMeans(x2))
T2_square.all[i] = T2_square
F.value.t2 = (n1 + n2 - 2 - 1) / (2 * (n1 + n2 - 2)) * T2_square
P.value.t2 = cbind(P.value.t2, 1 - pf(F.value.t2, 2, n1 + n2 - 2 - 1))
}
Reject.t1 = sum(P.value.t1 < 0.05)
Not.Reject.t1 = sum(P.value.t1 >= 0.05)
Empirical.Level.t1 = Reject.t1 / (Reject.t1 + Not.Reject.t1)
Reject.t2 = sum(P.value.t2 < 0.05)
Not.Reject.t2 = sum(P.value.t2 >= 0.05)
Empirical.Level.t2 = Reject.t2 / (Reject.t2 + Not.Reject.t2)
upper.quantile.t1 = quantile(T1_square.all[order(T1_square.all)], c(.85, .90, .95, .975, .99, .995))
upper.quantile.t2 = quantile(T2_square.all[order(T2_square.all)], c(.85, .90, .95, .975, .99, .995))
return(
  c(
    upper.quantile.t1,
    upper.quantile.t2,
    Reject.t1,
    Not.Reject.t1,
    Empirical.Level.t1,
    Reject.t2,
    Not.Reject.t2,
    Empirical.Level.t2
  )
)
}
#case i: $n_1=n_2=n_3$:
#p=2
upper.quantile_t12_1.1 <-
  generator.t2(c(0, 0), c(0, 0), c(10, 10), 100, 100, 100)[1:6]
upper.quantile_t22_1.1 <-
  generator.t2(c(0, 0), c(0, 0), c(10, 10), 100, 100, 100)[7:12]

```

```

rejects_t2_1.1 <-
  generator.t2(c(0, 0), c(0, 0), c(10, 10), 100, 100, 100)[13:18]
#p=3
upper.quantile_t12_1.2 <-
  generator.t2(c(0, 0, 0), c(0, 0, 0), c(10, 10, 10), 100, 100, 100)[1:6]
upper.quantile_t22_1.2 <-
  generator.t2(c(0, 0, 0), c(0, 0, 0), c(10, 10, 10), 100, 100, 100)[7:12]
rejects_t2_1.2 <-
  generator.t2(c(0, 0, 0), c(0, 0, 0), c(10, 10, 10), 100, 100, 100)[13:18]

#case ii $n_1<<n_2, n_3=n_1$:
#p=2
upper.quantile_t12_2.1 <-
  generator.t2(c(0, 0), c(0, 0), c(10, 10), 25, 100, 25)[1:6]
upper.quantile_t22_2.1 <-
  generator.t2(c(0, 0), c(0, 0), c(10, 10), 25, 100, 25)[7:12]
rejects_t2_2.1 <-
  generator.t2(c(0, 0), c(0, 0), c(10, 10), 25, 100, 25)[13:18]
#p=3
upper.quantile_t12_2.2 <-
  generator.t2(c(0, 0, 0), c(0, 0, 0), c(10, 10, 10), 25, 100, 25)[1:6]
upper.quantile_t22_2.2 <-
  generator.t2(c(0, 0, 0), c(0, 0, 0), c(10, 10, 10), 25, 100, 25)[7:12]
rejects_t2_2.2 <-
  generator.t2(c(0, 0, 0), c(0, 0, 0), c(10, 10, 10), 25, 100, 25)[13:18]

#case iii $n_1>>n_2, n_3=n_1$:
#p=2
upper.quantile_t12_3.1 <-
  generator.t2(c(0, 0), c(0, 0), c(10, 10), 100, 25, 100)[1:6]
upper.quantile_t22_3.1 <-
  generator.t2(c(0, 0), c(0, 0), c(10, 10), 100, 25, 100)[7:12]
rejects_t2_3.1 <-
  generator.t2(c(0, 0), c(0, 0), c(10, 10), 100, 25, 100)[13:18]
#p=3
upper.quantile_t12_3.2 <-
  generator.t2(c(0, 0, 0), c(0, 0, 0), c(10, 10, 10), 100, 25, 100)[1:6]
upper.quantile_t22_3.2 <-
  generator.t2(c(0, 0, 0), c(0, 0, 0), c(10, 10, 10), 100, 25, 100)[7:12]
rejects_t2_3.2 <-
  generator.t2(c(0, 0, 0), c(0, 0, 0), c(10, 10, 10), 25, 100, 100)[13:18]

#case iv $n_1=n_2=n, n_3>>n$:
#p=2
upper.quantile_t12_4.1 <-
  generator.t2(c(0, 0), c(0, 0), c(10, 10), 25, 25, 100)[1:6]
upper.quantile_t22_4.1 <-
  generator.t2(c(0, 0), c(0, 0), c(10, 10), 25, 25, 100)[7:12]
rejects_t2_4.1 <-
  generator.t2(c(0, 0), c(0, 0), c(10, 10), 25, 25, 100)[13:18]
#p=3
upper.quantile_t12_4.2 <-
  generator.t2(c(0, 0, 0), c(0, 0, 0), c(10, 10, 10), 25, 25, 100)[1:6]

```

```

upper.quantile_t22_4.2 <-
  generator.t2(c(0, 0, 0), c(0, 0, 0), c(10, 10, 10), 25, 25, 100)[7:12]
rejects_t2_4.2 <-
  generator.t2(c(0, 0, 0), c(0, 0, 0), c(10, 10, 10), 25, 25, 100)[13:18]

library(knitr)
library(kableExtra)
#summary table for p =2
table <-
  cbind(rejects_t2_1.1[1:3],
        rejects_t2_2.1[1:3],
        rejects_t2_3.1[1:3],
        rejects_t2_4.1[1:3])
rownames(table) <-
  c("T12 Test Rejects",
    "T12 Test Not Rejects",
    "T12 Test Empirical Level")
knitr::kable(
  table,
  col.names = c(
    "Sample Size 100, 100,100",
    "Sample Size 25, 100,25",
    "Sample Size 100, 25,100",
    "Sample Size 25, 25,100"
  ),
  caption = "Test Results for  $T_1^2$  Tests where  $p = 2$ "
) %>%
  kable_styling(latex_options = c("scale_down", "HOLD_position"))
table <-
  cbind(rejects_t2_1.1[4:6],
        rejects_t2_2.1[4:6],
        rejects_t2_3.1[4:6],
        rejects_t2_4.1[4:6])
rownames(table) <-
  c("T22 Test Rejects",
    "T22 Test Not Rejects",
    "T22 Test Empirical Level")
knitr::kable(
  table,
  col.names = c(
    "Sample Size 100, 100,100",
    "Sample Size 25, 100,25",
    "Sample Size 100, 25,100",
    "Sample Size 25, 25,100"
  ),
  caption = "Test Results for  $T_2^2$  Tests where  $p = 2$ "
) %>%
  kable_styling(latex_options = c("scale_down", "HOLD_position"))

#summary table for p =3
table <-
  cbind(rejects_t2_1.2[1:3],
        rejects_t2_2.2[1:3],

```

```

      rejects_t2_3.2[1:3],
      rejects_t2_4.2[1:3])
rownames(table) <-
  c("T12 Test Rejects",
    "T12 Test Not Rejects",
    "T12 Test Empirical Level")
knitr::kable(
  table,
  col.names = c(
    "Sample Size 100, 100,100",
    "Sample Size 25, 100,25",
    "Sample Size 100, 25,100",
    "Sample Size 25, 25,100"
  ),
  caption = "Test Results for  $T_1^2$  Tests where  $p = 3$ "
) %>%
  kable_styling(latex_options = c("scale_down", "HOLD_position"))
table <-
  cbind(rejects_t2_1.2[4:6],
        rejects_t2_2.2[4:6],
        rejects_t2_3.2[4:6],
        rejects_t2_4.2[4:6])
rownames(table) <-
  c("T22 Test Rejects",
    "T22 Test Not Rejects",
    "T22 Test Empirical Level")
knitr::kable(
  table,
  col.names = c(
    "Sample Size 100, 100,100",
    "Sample Size 25, 100,25",
    "Sample Size 100, 25,100",
    "Sample Size 25, 25,100"
  ),
  caption = "Test Results for  $T_2^2$  Tests where  $p = 3$ "
) %>%
  kable_styling(latex_options = c("scale_down", "HOLD_position"))

# Grouped Barchart
## case i
a = generator.t2(c(0, 0), c(0, 0), c(10, 10), 100, 100, 100)
b = generator.t2(c(0, 0, 0), c(0, 0, 0), c(10, 10, 10), 100, 100, 100)
## case ii
c = generator.t2(c(0, 0), c(0, 0), c(10, 10), 25, 100, 25)
d = generator.t2(c(0, 0, 0), c(0, 0, 0), c(10, 10, 10), 25, 100, 25)
## case iii
e = generator.t2(c(0, 0), c(0, 0), c(10, 10), 100, 25, 100)
f = generator.t2(c(0, 0, 0), c(0, 0, 0), c(10, 10, 10), 100, 25, 100)
## case iv
g = generator.t2(c(0, 0), c(0, 0), c(10, 10), 25, 25, 100)
h = generator.t2(c(0, 0, 0), c(0, 0, 0), c(10, 10, 10), 25, 25, 100)

library(ggplot2)

```

```

case = c(
  rep("Sample Size \n 100, 100,100", 4),
  rep("Sample Size \n 25, 100,25", 4),
  rep("Sample Size \n 100, 25,100", 4),
  rep("Sample Size \n 25, 25,100", 4)
)
test = rep(c("T12 p=2", "T22 p=2", "T12 p=3", "T22 p=3"), 4)
value = c(a[13], a[16], b[13], b[16], c[13], c[16], d[13], d[16],
          e[13], e[16], f[13], f[16], g[13], g[16], h[13], h[16])
bardata = data.frame(case, test, value)
ggplot(bardata, aes(fill = test, y = value, x = case, )) +
  geom_bar(position = "dodge", stat = "identity") +
  geom_hline(yintercept = 50,
             col = "red",
             lty = 2) +
  ggtitle("Graph 1: Number of Rejects") + xlab("Sample Size Configuration") + ylab("Count of Rejects")

```

Part 2 Code

```

library(mvtnorm)
# R updated version after optimization
T1_square.all = c()
T2_square.all = c()
F.value.t1 = c()
P.value.t1 = c()
F.value.t2 = c()
P.value.t2 = c()
generator.delta.t2 = function(mu1, mu2, mu3, n1, n2, n3) {
  ## mu where mu1 = mu2 = mu0 = 0 and we set mu3 = 10
  set.seed(625)
  d0 = mu1
  for (i in 1:1000) {
    if (length(mu1) == 2) {
      x1 = rmvnorm(n = n1,
                  mean = mu1,
                  sigma = sigma.2.2)
      x2 = rmvnorm(n = n2,
                  mean = mu2,
                  sigma = sigma.2.2)
      x3 = rmvnorm(n = n3,
                  mean = mu3,
                  sigma = sigma.2.2)
    }
    else {
      x1 = rmvnorm(n = n1,
                  mean = mu1,
                  sigma = sigma.3.3)
      x2 = rmvnorm(n = n2,
                  mean = mu2,
                  sigma = sigma.3.3)
      x3 = rmvnorm(n = n3,

```

```

        mean = mu3,
        sigma = sigma.3.3)
    }

    S_pool.2.2 = ((n1 - 1) * cov(x1) + (n2 - 1) * cov(x2)) / (n1 + n2 -
        2)
    T1_square = (1 / n1 + 1 / n2) ^ (-1) * t(colMeans(x1) - colMeans(x2) -
        d0) %*% inv(S_pool.2.2) %*%
        (colMeans(x1) - colMeans(x2) - d0)
    T1_square.all[i] = T1_square
    F.value.t1 = (n1 + n2 - 2 - 1) / (2 * (n1 + n2 - 2)) * T1_square
    P.value.t1 = cbind(P.value.t1, 1 - pf(F.value.t1, 2, n1 + n2 - 2 - 1))

    S_pool.3.3 = ((n1 - 1) * cov(x1) + (n2 - 1) * cov(x2) + (n3 - 1) * cov(x3)) /
        (n1 + n2 + n3 - 3)
    T2_square = (1 / n1 + 1 / n2) ^ (-1) * t(colMeans(x1) - colMeans(x2) -
        d0) %*% inv(S_pool.3.3) %*%
        (colMeans(x1) - colMeans(x2) - d0)
    T2_square.all[i] = T2_square
    F.value.t2 = (n1 + n2 - 2 - 1) / (2 * (n1 + n2 - 2)) * T2_square
    P.value.t2 = cbind(P.value.t2, 1 - pf(F.value.t2, 2, n1 + n2 - 2 - 1))
}
Reject.t1 = sum(P.value.t1 < 0.05)
Not.Reject.t1 = sum(P.value.t1 >= 0.05)
Empirical.Level.t1 = Reject.t1 / (Reject.t1 + Not.Reject.t1)
Reject.t2 = sum(P.value.t2 < 0.05)
Not.Reject.t2 = sum(P.value.t2 >= 0.05)
Empirical.Level.t2 = Reject.t2 / (Reject.t2 + Not.Reject.t2)
upper.quantile.t1 = quantile(T1_square.all[order(T1_square.all)], c(.85, .90, .95, .975, .99, .995))
upper.quantile.t2 = quantile(T2_square.all[order(T2_square.all)], c(.85, .90, .95, .975, .99, .995))
return(
    c(
        upper.quantile.t1,
        upper.quantile.t2,
        Reject.t1,
        Not.Reject.t1,
        Empirical.Level.t1,
        Reject.t2,
        Not.Reject.t2,
        Empirical.Level.t2
    )
)
}

#case i: $n_1=n_2=n_3$:
#delta = 2
# p = 2
delta2.upper.quantile_t12_1.1 <-
    generator.delta.t2(c(sqrt(2 / 2), sqrt(2 / 2)), c(0, 0), c(10, 10), 100, 100, 100)[1:6]
delta2.upper.quantile_t22_1.1 <-
    generator.delta.t2(c(sqrt(2 / 2), sqrt(2 / 2)), c(0, 0), c(10, 10), 100, 100, 100)[7:12]
delta2.rejects_t2_1.1 <-
    generator.delta.t2(c(sqrt(2 / 2), sqrt(2 / 2)), c(0, 0), c(10, 10), 100, 100, 100)[13:18]

```

```

# p = 3
delta2.upper.quantile_t12_1.2 <-
  generator.delta.t2(c(sqrt(2 / 3), sqrt(2 / 3), sqrt(2 / 3)), c(0, 0, 0), c(10, 10, 10), 100, 100, 100)
delta2.upper.quantile_t22_1.2 <-
  generator.delta.t2(c(sqrt(2 / 3), sqrt(2 / 3), sqrt(2 / 3)), c(0, 0, 0), c(10, 10, 10), 100, 100, 100)
delta2.rejects_t2_1.2 <-
  generator.delta.t2(c(sqrt(2 / 3), sqrt(2 / 3), sqrt(2 / 3)), c(0, 0, 0), c(10, 10, 10), 100, 100, 100)

#delta = 4
# p = 2
delta4.upper.quantile_t12_1.1 <-
  generator.delta.t2(c(sqrt(4 / 2), sqrt(4 / 2)), c(0, 0), c(10, 10), 100, 100, 100)[1:6]
delta4.upper.quantile_t22_1.1 <-
  generator.delta.t2(c(sqrt(4 / 2), sqrt(4 / 2)), c(0, 0), c(10, 10), 100, 100, 100)[7:12]
delta4.rejects_t2_1.1 <-
  generator.delta.t2(c(sqrt(4 / 2), sqrt(4 / 2)), c(0, 0), c(10, 10), 100, 100, 100)[13:18]
# p= 3
delta4.upper.quantile_t12_1.2 <-
  generator.delta.t2(c(sqrt(4 / 3), sqrt(4 / 3), sqrt(4 / 3)), c(0, 0, 0), c(10, 10, 10), 100, 100, 100)
delta4.upper.quantile_t22_1.2 <-
  generator.delta.t2(c(sqrt(4 / 3), sqrt(4 / 3), sqrt(4 / 3)), c(0, 0, 0), c(10, 10, 10), 100, 100, 100)
delta4.rejects_t2_1.2 <-
  generator.delta.t2(c(sqrt(4 / 3), sqrt(4 / 3), sqrt(4 / 3)), c(0, 0, 0), c(10, 10, 10), 100, 100, 100)

#delta = 8
# p = 2
delta8.upper.quantile_t12_1.1 <-
  generator.delta.t2(c(sqrt(8 / 2), sqrt(8 / 2)), c(0, 0), c(10, 10), 100, 100, 100)[1:6]
delta8.upper.quantile_t22_1.1 <-
  generator.delta.t2(c(sqrt(8 / 2), sqrt(8 / 2)), c(0, 0), c(10, 10), 100, 100, 100)[7:12]
delta8.rejects_t2_1.1 <-
  generator.delta.t2(c(sqrt(8 / 2), sqrt(8 / 2)), c(0, 0), c(10, 10), 100, 100, 100)[13:18]
# p= 3
delta8.upper.quantile_t12_1.2 <-
  generator.delta.t2(c(sqrt(8 / 3), sqrt(8 / 3), sqrt(8 / 3)), c(0, 0, 0), c(10, 10, 10), 100, 100, 100)
delta8.upper.quantile_t22_1.2 <-
  generator.delta.t2(c(sqrt(8 / 3), sqrt(8 / 3), sqrt(8 / 3)), c(0, 0, 0), c(10, 10, 10), 100, 100, 100)
delta8.rejects_t2_1.2 <-
  generator.delta.t2(c(sqrt(8 / 3), sqrt(8 / 3), sqrt(8 / 3)), c(0, 0, 0), c(10, 10, 10), 100, 100, 100)

#delta = 16
# p = 2
delta16.upper.quantile_t12_1.1 <-
  generator.delta.t2(c(sqrt(16 / 2), sqrt(16 / 2)), c(0, 0), c(10, 10), 100, 100, 100)[1:6]
delta16.upper.quantile_t22_1.1 <-
  generator.delta.t2(c(sqrt(16 / 2), sqrt(16 / 2)), c(0, 0), c(10, 10), 100, 100, 100)[7:12]
delta16.rejects_t2_1.1 <-
  generator.delta.t2(c(sqrt(16 / 2), sqrt(16 / 2)), c(0, 0), c(10, 10), 100, 100, 100)[13:18]
# p= 3
delta16.upper.quantile_t12_1.2 <-
  generator.delta.t2(c(sqrt(16 / 3), sqrt(16 / 3), sqrt(16 / 3)), c(0, 0, 0), c(10, 10, 10), 100, 100, 100)
delta16.upper.quantile_t22_1.2 <-

```



```

    generator.delta.t2(c(sqrt(16 / 3), sqrt(16 / 3), sqrt(16 / 3)), c(0, 0, 0), c(10, 10, 10), 100, 100, 100)
delta16.rejects_t2_1.2 <-
    generator.delta.t2(c(sqrt(16 / 3), sqrt(16 / 3), sqrt(16 / 3)), c(0, 0, 0), c(10, 10, 10), 100, 100, 100)

#case ii $n_1 < n_2, n_3 = n_1$:
#delta = 2
# p = 2
delta2.upper.quantile_t12_2.1 <-
    generator.delta.t2(c(sqrt(2 / 2), sqrt(2 / 2)), c(0, 0), c(10, 10), 25, 100, 25)[1:6]
delta2.upper.quantile_t22_2.1 <-
    generator.delta.t2(c(sqrt(2 / 2), sqrt(2 / 2)), c(0, 0), c(10, 10), 25, 100, 25)[7:12]
delta2.rejects_t2_2.1 <-
    generator.delta.t2(c(sqrt(2 / 2), sqrt(2 / 2)), c(0, 0), c(10, 10), 25, 100, 25)[13:18]
# p = 3
delta2.upper.quantile_t12_2.2 <-
    generator.delta.t2(c(sqrt(2 / 3), sqrt(2 / 3), sqrt(2 / 3)), c(0, 0, 0), c(10, 10, 10), 25, 100, 25)[1:6]
delta2.upper.quantile_t22_2.2 <-
    generator.delta.t2(c(sqrt(2 / 3), sqrt(2 / 3), sqrt(2 / 3)), c(0, 0, 0), c(10, 10, 10), 25, 100, 25)[7:12]
delta2.rejects_t2_2.2 <-
    generator.delta.t2(c(sqrt(2 / 3), sqrt(2 / 3), sqrt(2 / 3)), c(0, 0, 0), c(10, 10, 10), 25, 100, 25)[13:18]

#delta = 4
# p = 2
delta4.upper.quantile_t12_2.1 <-
    generator.delta.t2(c(sqrt(4 / 2), sqrt(4 / 2)), c(0, 0), c(10, 10), 25, 100, 25)[1:6]
delta4.upper.quantile_t22_2.1 <-
    generator.delta.t2(c(sqrt(4 / 2), sqrt(4 / 2)), c(0, 0), c(10, 10), 25, 100, 25)[7:12]
delta4.rejects_t2_2.1 <-
    generator.delta.t2(c(sqrt(4 / 2), sqrt(4 / 2)), c(0, 0), c(10, 10), 25, 100, 25)[13:18]
# p = 3
delta4.upper.quantile_t12_2.2 <-
    generator.delta.t2(c(sqrt(4 / 3), sqrt(4 / 3), sqrt(4 / 3)), c(0, 0, 0), c(10, 10, 10), 25, 100, 25)[1:6]
delta4.upper.quantile_t22_2.2 <-
    generator.delta.t2(c(sqrt(4 / 3), sqrt(4 / 3), sqrt(4 / 3)), c(0, 0, 0), c(10, 10, 10), 25, 100, 25)[7:12]
delta4.rejects_t2_2.2 <-
    generator.delta.t2(c(sqrt(4 / 3), sqrt(4 / 3), sqrt(4 / 3)), c(0, 0, 0), c(10, 10, 10), 25, 100, 25)[13:18]

#delta = 8
# p = 2
delta8.upper.quantile_t12_2.1 <-
    generator.delta.t2(c(sqrt(8 / 2), sqrt(8 / 2)), c(0, 0), c(10, 10), 25, 100, 25)[1:6]
delta8.upper.quantile_t22_2.1 <-
    generator.delta.t2(c(sqrt(8 / 2), sqrt(8 / 2)), c(0, 0), c(10, 10), 25, 100, 25)[7:12]
delta8.rejects_t2_2.1 <-
    generator.delta.t2(c(sqrt(8 / 2), sqrt(8 / 2)), c(0, 0), c(10, 10), 25, 100, 25)[13:18]
# p = 3
delta8.upper.quantile_t12_2.2 <-
    generator.delta.t2(c(sqrt(8 / 3), sqrt(8 / 3), sqrt(8 / 3)), c(0, 0, 0), c(10, 10, 10), 25, 100, 25)[1:6]
delta8.upper.quantile_t22_2.2 <-
    generator.delta.t2(c(sqrt(8 / 3), sqrt(8 / 3), sqrt(8 / 3)), c(0, 0, 0), c(10, 10, 10), 25, 100, 25)[7:12]
delta8.rejects_t2_2.2 <-
    generator.delta.t2(c(sqrt(8 / 3), sqrt(8 / 3), sqrt(8 / 3)), c(0, 0, 0), c(10, 10, 10), 25, 100, 25)[13:18]

```

```

#delta = 16
# p = 2
delta16.upper.quantile_t12_2.1 <-
  generator.delta.t2(c(sqrt(16 / 2), sqrt(16 / 2)), c(0, 0), c(10, 10), 25, 100, 25)[1:6]
delta16.upper.quantile_t22_2.1 <-
  generator.delta.t2(c(sqrt(16 / 2), sqrt(16 / 2)), c(0, 0), c(10, 10), 25, 100, 25)[7:12]
delta16.rejects_t2_2.1 <-
  generator.delta.t2(c(sqrt(16 / 2), sqrt(16 / 2)), c(0, 0), c(10, 10), 25, 100, 25)[13:18]
# p= 3
delta16.upper.quantile_t12_2.2 <-
  generator.delta.t2(c(sqrt(16 / 3), sqrt(16 / 3), sqrt(16 / 3)), c(0, 0, 0), c(10, 10, 10), 25, 100, 25)
delta16.upper.quantile_t22_2.2 <-
  generator.delta.t2(c(sqrt(16 / 3), sqrt(16 / 3), sqrt(16 / 3)), c(0, 0, 0), c(10, 10, 10), 25, 100, 25)
delta16.rejects_t2_2.2 <-
  generator.delta.t2(c(sqrt(16 / 3), sqrt(16 / 3), sqrt(16 / 3)), c(0, 0, 0), c(10, 10, 10), 25, 100, 25)

#case iii $n_1>n_2, n_3=n_1$
#delta = 2
# p = 2
gen3.2.2 <-
  generator.delta.t2(c(sqrt(2 / 2), sqrt(2 / 2)), c(0, 0), c(10, 10), 100, 25, 100)
delta2.upper.quantile_t12_3.1 <- gen3.2.2[1:6]
delta2.upper.quantile_t22_3.1 <- gen3.2.2[7:12]
delta2.rejects_t2_3.1 <- gen3.2.2[13:18]
# p = 3
gen3.2.3 <-
  generator.delta.t2(c(sqrt(2 / 3), sqrt(2 / 3), sqrt(2 / 3)), c(0, 0, 0), c(10, 10, 10), 100, 25, 100)
delta2.upper.quantile_t12_3.2 <- gen3.2.3[1:6]
delta2.upper.quantile_t22_3.2 <- gen3.2.3[7:12]
delta2.rejects_t2_3.2 <- gen3.2.3[13:18]

#delta = 4
# p = 2
gen3.4.2 <-
  generator.delta.t2(c(sqrt(4 / 2), sqrt(4 / 2)), c(0, 0), c(10, 10), 100, 25, 100)
delta4.upper.quantile_t12_3.1 <- gen3.4.2[1:6]
delta4.upper.quantile_t22_3.1 <- gen3.4.2[7:12]
delta4.rejects_t2_3.1 <- gen3.4.2[13:18]
# p = 3
gen3.4.3 <-
  generator.delta.t2(c(sqrt(4 / 3), sqrt(4 / 3), sqrt(4 / 3)), c(0, 0, 0), c(10, 10, 10), 100, 25, 100)
delta4.upper.quantile_t12_3.2 <- gen3.4.3[1:6]
delta4.upper.quantile_t22_3.2 <- gen3.4.3[7:12]
delta4.rejects_t2_3.2 <- gen3.4.3[13:18]

#delta = 8
# p = 2
gen3.8.2 <-
  generator.delta.t2(c(sqrt(8 / 2), sqrt(8 / 2)), c(0, 0), c(10, 10), 100, 25, 100)
delta8.upper.quantile_t12_3.1 <- gen3.8.2[1:6]
delta8.upper.quantile_t22_3.1 <- gen3.8.2[7:12]
delta8.rejects_t2_3.1 <- gen3.8.2[13:18]
# p = 3

```

```

gen3.8.3 <-
  generator.delta.t2(c(sqrt(8 / 3), sqrt(8 / 3), sqrt(8 / 3)), c(0, 0, 0), c(10, 10, 10), 100, 25, 100)
delta8.upper.quantile_t12_3.2 <- gen3.8.3[1:6]
delta8.upper.quantile_t22_3.2 <- gen3.8.3[7:12]
delta8.rejects_t2_3.2 <- gen3.8.3[13:18]

#delta = 16
# p = 2
gen3.16.2 <-
  generator.delta.t2(c(sqrt(16 / 2), sqrt(16 / 2)), c(0, 0), c(10, 10), 100, 25, 100)
delta16.upper.quantile_t12_3.1 <- gen3.16.2[1:6]
delta16.upper.quantile_t22_3.1 <- gen3.16.2[7:12]
delta16.rejects_t2_3.1 <- gen3.16.2[13:18]

# p = 3
gen3.16.3 <-
  generator.delta.t2(c(sqrt(16 / 3), sqrt(16 / 3), sqrt(16 / 3)), c(0, 0, 0), c(10, 10, 10), 100, 25, 100)
delta16.upper.quantile_t12_3.2 <- gen3.16.3[1:6]
delta16.upper.quantile_t22_3.2 <- gen3.16.3[7:12]
delta16.rejects_t2_3.2 <- gen3.16.3[13:18]

#case iv $n_1=n_2=n, n_3>>n$:
#delta = 2
# p = 2
gen4.2.2 <-
  generator.delta.t2(c(sqrt(2 / 2), sqrt(2 / 2)), c(0, 0), c(10, 10), 25, 25, 100)
delta2.upper.quantile_t12_4.1 <- gen4.2.2[1:6]
delta2.upper.quantile_t22_4.1 <- gen4.2.2[7:12]
delta2.rejects_t2_4.1 <- gen4.2.2[13:18]

# p = 3
gen4.2.3 <-
  generator.delta.t2(c(sqrt(2 / 3), sqrt(2 / 3), sqrt(2 / 3)), c(0, 0, 0), c(10, 10, 10), 25, 25, 100)
delta2.upper.quantile_t12_4.2 <- gen4.2.3[1:6]
delta2.upper.quantile_t22_4.2 <- gen4.2.3[7:12]
delta2.rejects_t2_4.2 <- gen4.2.3[13:18]

#delta = 4
# p = 2
gen4.4.2 <-
  generator.delta.t2(c(sqrt(4 / 2), sqrt(4 / 2)), c(0, 0), c(10, 10), 25, 25, 100)
delta4.upper.quantile_t12_4.1 <- gen4.4.2[1:6]
delta4.upper.quantile_t22_4.1 <- gen4.4.2[7:12]
delta4.rejects_t2_4.1 <- gen4.4.2[13:18]

# p = 3
gen4.4.3 <-
  generator.delta.t2(c(sqrt(4 / 3), sqrt(4 / 3), sqrt(4 / 3)), c(0, 0, 0), c(10, 10, 10), 25, 25, 100)
delta4.upper.quantile_t12_4.2 <- gen4.4.3[1:6]
delta4.upper.quantile_t22_4.2 <- gen4.4.3[7:12]
delta4.rejects_t2_4.2 <- gen4.4.3[13:18]

#delta = 8
# p = 2
gen4.8.2 <-
  generator.delta.t2(c(sqrt(8 / 2), sqrt(8 / 2)), c(0, 0), c(10, 10), 25, 25, 100)

```

```

delta8.upper.quantile_t12_4.1 <- gen4.8.2[1:6]
delta8.upper.quantile_t22_4.1 <- gen4.8.2[7:12]
delta8.rejects_t2_4.1 <- gen4.8.2[13:18]
# p = 3
gen4.8.3 <-
  generator.delta.t2(c(sqrt(8 / 3), sqrt(8 / 3), sqrt(8 / 3)), c(0, 0, 0), c(10, 10, 10), 25, 25, 100)
delta8.upper.quantile_t12_4.2 <- gen4.8.3[1:6]
delta8.upper.quantile_t22_4.2 <- gen4.8.3[7:12]
delta8.rejects_t2_4.2 <- gen4.8.3[13:18]

#delta = 16
# p = 2
gen4.16.2 <-
  generator.delta.t2(c(sqrt(16 / 2), sqrt(16 / 2)), c(0, 0), c(10, 10), 25, 25, 100)
delta16.upper.quantile_t12_4.1 <- gen4.16.2[1:6]
delta16.upper.quantile_t22_4.1 <- gen4.16.2[7:12]
delta16.rejects_t2_4.1 <- gen4.16.2[13:18]
# p = 3
gen4.16.3 <-
  generator.delta.t2(c(sqrt(16 / 3), sqrt(16 / 3), sqrt(16 / 3)), c(0, 0, 0), c(10, 10, 10), 25, 25, 100)
delta16.upper.quantile_t12_4.2 <- gen4.16.3[1:6]
delta16.upper.quantile_t22_4.2 <- gen4.16.3[7:12]
delta16.rejects_t2_4.2 <- gen4.16.3[13:18]

##### TEST SUMMARY TABLES FROM PART 2 FOR P=2 (only delta = 2) #####
#summary table for p =2 and delta = 2 t12
table <-
  cbind(
    delta2.rejects_t2_1.1[1:3],
    delta2.rejects_t2_2.1[1:3],
    delta2.rejects_t2_3.1[1:3],
    delta2.rejects_t2_4.1[1:3]
  )
rownames(table) <-
  c("T12 Test Rejects",
    "T12 Test Not Rejects",
    "T12 Test Empirical Power")
knitr::kable(
  table,
  col.names = c(
    "Sample Size 100, 100,100",
    "Sample Size 25, 100,25",
    "Sample Size 100, 25,100",
    "Sample Size 25, 25,100"
  ),
  caption = "Test Results for T12 Test where p = 2 and delta = 2"
) %>%
  kable_styling(latex_options = c("scale_down", "HOLD_position"))

#summary table for p =2 and delta = 2 t22
table <-
  cbind(
    delta2.rejects_t2_1.1[4:6],

```

```

    delta2.rejects_t2_2.1[4:6],
    delta2.rejects_t2_3.1[4:6],
    delta2.rejects_t2_4.1[4:6]
  )
rownames(table) <-
  c("T22 Test Rejects",
    "T22 Test Not Rejects",
    "T22 Test Empirical Power")
knitr::kable(
  table,
  col.names = c(
    "Sample Size 100, 100,100",
    "Sample Size 25, 100,25",
    "Sample Size 100, 25,100",
    "Sample Size 25, 25,100"
  ),
  caption = "Test Results for T22 Test where p = 2 and delta = 2"
) %>%
  kable_styling(latex_options = c("scale_down", "HOLD_position"))

##### TEST SUMMARY TABLES FROM PART 2 FOR P=3 (delta = 2 only)#####
#summary table for p =2 and delta = 2 for t12
table <-
  cbind(
    delta2.rejects_t2_1.2[1:3],
    delta2.rejects_t2_2.2[1:3],
    delta2.rejects_t2_3.2[1:3],
    delta2.rejects_t2_4.2[1:3]
  )
rownames(table) <-
  c("T12 Test Rejects",
    "T12 Test Not Rejects",
    "T12 Test Empirical Power")
knitr::kable(
  table,
  col.names = c(
    "Sample Size 100, 100,100",
    "Sample Size 25, 100,25",
    "Sample Size 100, 25,100",
    "Sample Size 25, 25,100"
  ),
  caption = "Test Results for T12 Test where p = 3 and delta = 2"
) %>%
  kable_styling(latex_options = c("scale_down", "HOLD_position"))

#summary table for p =2 and delta = 2 for t22
table <-
  cbind(
    delta2.rejects_t2_1.2[4:6],
    delta2.rejects_t2_2.2[4:6],
    delta2.rejects_t2_3.2[4:6],
    delta2.rejects_t2_4.2[4:6]
  )

```

```

rownames(table) <-
  c("T22 Test Rejects",
    "T22 Test Not Rejects",
    "T22 Test Empirical Power")
knitr::kable(
  table,
  col.names = c(
    "Sample Size 100, 100,100",
    "Sample Size 25, 100,25",
    "Sample Size 100, 25,100",
    "Sample Size 25, 25,100"
  ),
  caption = "Test Results for T22 Test where p = 3 and delta = 2"
) %>%
  kable_styling(latex_options = c("scale_down", "HOLD_position"))

# case i
options(digits = 2)
# T=1
n1 = 100
n2 = 100
n3 = 100
critical.t1 = qf(.95, 2, n1 + n2 - 2 - 1) ^ 2 * (n1 + n2 - 2 - 1) / (2 *
                                                                    (n1 + n2 - 2))

#critical.t1
power.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1)
#power.t1
# T=2
n1 = 100
n2 = 100
n3 = 100
critical.t2 = qf(.95, 2, n1 + n2 + n3 - 3 - 1) ^ 2 * (n1 + n2 - 2 - 1) /
(2 * (n1 + n2 - 2))
#critical.t2
power.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1)
#power.t2

# case i, p=2
# delta = 2
ncp = t(c(sqrt(2 / 2), sqrt(2 / 2))) %*% solve(sigma.2.2) %*% c(sqrt(2 /
                                                                    2), sqrt(2 / 2))
power.new_2.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_2.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
# delta = 4
ncp = t(c(sqrt(4 / 2), sqrt(4 / 2))) %*% solve(sigma.2.2) %*% c(sqrt(4 /
                                                                    2), sqrt(4 / 2))
power.new_4.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_4.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
# delta = 8
ncp = t(c(sqrt(8 / 2), sqrt(8 / 2))) %*% solve(sigma.2.2) %*% c(sqrt(8 /
                                                                    2), sqrt(8 / 2))
power.new_8.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_8.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)

```

```

# delta = 16
ncp = t(c(sqrt(16 / 2), sqrt(16 / 2))) %%% solve(sigma.2.2) %%% c(sqrt(16 /
2), sqrt(16 / 2))

power.new_16.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_16.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
t12.2.i = c(power.t1,
            power.new_2.t1,
            power.new_4.t1,
            power.new_8.t1,
            power.new_16.t1)
t22.2.i = c(power.t2,
            power.new_2.t2,
            power.new_4.t2,
            power.new_8.t2,
            power.new_16.t2)

# case i, p= 3
# delta = 2
ncp = t(c(sqrt(2 / 3), sqrt(2 / 3), sqrt(2 / 3))) %%% solve(sigma.3.3) %%%
c(sqrt(2 / 3), sqrt(2 / 3), sqrt(2 / 3))
power.new_2.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_2.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
# delta = 4
ncp = t(c(sqrt(4 / 3), sqrt(4 / 3), sqrt(4 / 3))) %%% solve(sigma.3.3) %%%
c(sqrt(4 / 3), sqrt(4 / 3), sqrt(4 / 3))
power.new_4.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_4.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
# delta = 8
ncp = t(c(sqrt(8 / 3), sqrt(8 / 3), sqrt(8 / 3))) %%% solve(sigma.3.3) %%%
c(sqrt(8 / 3), sqrt(8 / 3), sqrt(8 / 3))
power.new_8.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_8.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
# delta = 16
ncp = t(c(sqrt(16 / 3), sqrt(16 / 3), sqrt(16 / 3))) %%% solve(sigma.3.3) %%%
c(sqrt(16 / 3), sqrt(16 / 3), sqrt(16 / 3))
power.new_16.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_16.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)

t12.3.i = c(power.t1,
            power.new_2.t1,
            power.new_4.t1,
            power.new_8.t1,
            power.new_16.t1)
t22.3.i = c(power.t2,
            power.new_2.t2,
            power.new_4.t2,
            power.new_8.t2,
            power.new_16.t2)

#summary table
table <- cbind(t12.2.i, t22.2.i, t12.3.i, t22.3.i)
rownames(table) <-
c("Delta = 0", "Delta = 2", "Delta = 4", "Delta = 8", "Delta = 16")

```

```

knitr::kable(
  table,
  col.names = c("T12 \n p = 2", "T22 \n p = 2", "T12 \n p = 3", "T22 \n p = 3"),
  caption = "Empirical Power Results"
) %>%
  kable_styling(latex_options = c("HOLD_position"))

# case ii
options(digits = 2)
# compute (1-alpha)th quantile of T_1^2
n1 = 25
n2 = 100
n3 = 25
critical.t1 = qf(.95, 2, n1 + n2 - 2 - 1) ^ 2 * (n1 + n2 - 2 - 1) / (2 *
                                                                    (n1 + n2 - 2))

#critical.t1
power.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1)
#power.t1
# compute (1-alpha)th quantile of T_2^2
n1 = 25
n2 = 100
n3 = 25
critical.t2 = qf(.95, 2, n1 + n2 + n3 - 3 - 1) ^ 2 * (n1 + n2 - 2 - 1) /
  (2 * (n1 + n2 - 2))
#critical.t2
power.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1)
#power.t2

# case ii, p=2
# delta = 2
ncp = t(c(sqrt(2 / 2), sqrt(2 / 2))) %%% solve(sigma.2.2) %%% c(sqrt(2 /
                                                                    2), sqrt(2 / 2))
power.new_2.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_2.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
# delta = 4
ncp = t(c(sqrt(4 / 2), sqrt(4 / 2))) %%% solve(sigma.2.2) %%% c(sqrt(4 /
                                                                    2), sqrt(4 / 2))
power.new_4.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_4.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
# delta = 8
ncp = t(c(sqrt(8 / 2), sqrt(8 / 2))) %%% solve(sigma.2.2) %%% c(sqrt(8 /
                                                                    2), sqrt(8 / 2))
power.new_8.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_8.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
# delta = 16
ncp = t(c(sqrt(16 / 2), sqrt(16 / 2))) %%% solve(sigma.2.2) %%% c(sqrt(16 /
                                                                    2), sqrt(16 / 2))
power.new_16.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_16.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)

t12.2.ii = c(power.t1,
              power.new_2.t1,
              power.new_4.t1,

```



```

        power.new_8.t1,
        power.new_16.t1)
t22.2.ii = c(power.t2,
             power.new_2.t2,
             power.new_4.t2,
             power.new_8.t2,
             power.new_16.t2)

# case ii, p=3
# delta = 2
ncp = t(c(sqrt(2 / 3), sqrt(2 / 3), sqrt(2 / 3))) %%% solve(sigma.3.3) %%%
      c(sqrt(2 / 3), sqrt(2 / 3), sqrt(2 / 3))
power.new_2.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_2.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
# delta = 4
ncp = t(c(sqrt(4 / 3), sqrt(4 / 3), sqrt(4 / 3))) %%% solve(sigma.3.3) %%%
      c(sqrt(4 / 3), sqrt(4 / 3), sqrt(4 / 3))
power.new_4.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_4.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
# delta = 8
ncp = t(c(sqrt(8 / 3), sqrt(8 / 3), sqrt(8 / 3))) %%% solve(sigma.3.3) %%%
      c(sqrt(8 / 3), sqrt(8 / 3), sqrt(8 / 3))
power.new_8.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_8.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
# delta = 16
ncp = t(c(sqrt(16 / 3), sqrt(16 / 3), sqrt(16 / 3))) %%% solve(sigma.3.3) %%%
      c(sqrt(16 / 3), sqrt(16 / 3), sqrt(16 / 3))
power.new_16.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_16.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)

t12.3.ii = c(power.t1,
             power.new_2.t1,
             power.new_4.t1,
             power.new_8.t1,
             power.new_16.t1)
t22.3.ii = c(power.t2,
             power.new_2.t2,
             power.new_4.t2,
             power.new_8.t2,
             power.new_16.t2)

# case iii
options(digits = 2)
# compute (1-alpha)th quantile of  $T_1^2$ 
n1 = 100
n2 = 25
n3 = 100
critical.t1 = qf(.95, 2, n1 + n2 - 2 - 1) ^ 2 * (n1 + n2 - 2 - 1) / (2 *
                                                    (n1 + n2 - 2))

#critical.t1
power.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1)
#power.t1
# compute (1-alpha)th quantile of  $T_2^2$ 

```

```

n1 = 100
n2 = 25
n3 = 100
critical.t2 = qf(.95, 2, n1 + n2 + n3 - 3 - 1) ^ 2 * (n1 + n2 - 2 - 1) /
  (2 * (n1 + n2 - 2))
#critical.t2
power.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1)
#power.t2

# case iii, p=2
# delta = 2
ncp = t(c(sqrt(2 / 2), sqrt(2 / 2))) %% solve(sigma.2.2) %% c(sqrt(2 /
  2), sqrt(2 / 2))
power.new_2.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_2.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
# delta = 4
ncp = t(c(sqrt(4 / 2), sqrt(4 / 2))) %% solve(sigma.2.2) %% c(sqrt(4 /
  2), sqrt(4 / 2))
power.new_4.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_4.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
# delta = 8
ncp = t(c(sqrt(8 / 2), sqrt(8 / 2))) %% solve(sigma.2.2) %% c(sqrt(8 /
  2), sqrt(8 / 2))
power.new_8.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_8.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
# delta = 16
ncp = t(c(sqrt(16 / 2), sqrt(16 / 2))) %% solve(sigma.2.2) %% c(sqrt(16 /
  2), sqrt(16 / 2))
power.new_16.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_16.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)

t12.2.iii = c(power.t1,
  power.new_2.t1,
  power.new_4.t1,
  power.new_8.t1,
  power.new_16.t1)
t22.2.iii = c(power.t2,
  power.new_2.t2,
  power.new_4.t2,
  power.new_8.t2,
  power.new_16.t2)

# case iii, p=3
# delta = 2
ncp = t(c(sqrt(2 / 3), sqrt(2 / 3), sqrt(2 / 3))) %% solve(sigma.3.3) %%
  c(sqrt(2 / 3), sqrt(2 / 3), sqrt(2 / 3))
power.new_2.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_2.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
# delta = 4
ncp = t(c(sqrt(4 / 3), sqrt(4 / 3), sqrt(4 / 3))) %% solve(sigma.3.3) %%
  c(sqrt(4 / 3), sqrt(4 / 3), sqrt(4 / 3))
power.new_4.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_4.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)

```

```

# delta = 8
ncp = t(c(sqrt(8 / 3), sqrt(8 / 3), sqrt(8 / 3))) %%% solve(sigma.3.3) %%%
  c(sqrt(8 / 3), sqrt(8 / 3), sqrt(8 / 3))
power.new_8.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_8.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
# delta = 16
ncp = t(c(sqrt(16 / 3), sqrt(16 / 3), sqrt(16 / 3))) %%% solve(sigma.3.3) %%%
  c(sqrt(16 / 3), sqrt(16 / 3), sqrt(16 / 3))
power.new_16.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_16.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)

t12.3.iii = c(power.t1,
              power.new_2.t1,
              power.new_4.t1,
              power.new_8.t1,
              power.new_16.t1)
t22.3.iii = c(power.t2,
              power.new_2.t2,
              power.new_4.t2,
              power.new_8.t2,
              power.new_16.t2)

# case iv
options(digits = 2)
# compute (1-alpha)th quantile of  $T_1^2$ 
n1 = 25
n2 = 25
n3 = 100
critical.t1 = qf(.95, 2, n1 + n2 - 2 - 1) ^ 2 * (n1 + n2 - 2 - 1) / (2 *
                                                                    (n1 + n2 - 2))
#critical.t1
power.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1)
#power.t1
# compute (1-alpha)th quantile of  $T_2^2$ 
n1 = 25
n2 = 25
n3 = 100
critical.t2 = qf(.95, 2, n1 + n2 + n3 - 3 - 1) ^ 2 * (n1 + n2 - 2 - 1) /
  (2 * (n1 + n2 - 2))
#critical.t2
power.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1)
#power.t2

# case iv, p=2
# delta = 2
ncp = t(c(sqrt(2 / 2), sqrt(2 / 2))) %%% solve(sigma.2.2) %%% c(sqrt(2 /
                                                                    2), sqrt(2 / 2))
power.new_2.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_2.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
# delta = 4
ncp = t(c(sqrt(4 / 2), sqrt(4 / 2))) %%% solve(sigma.2.2) %%% c(sqrt(4 /
                                                                    2), sqrt(4 / 2))
power.new_4.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)

```

```

power.new_4.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
# delta = 8
ncp = t(c(sqrt(8 / 2), sqrt(8 / 2))) %%% solve(sigma.2.2) %%% c(sqrt(8 /
2), sqrt(8 / 2))

power.new_8.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_8.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
# delta = 16
ncp = t(c(sqrt(16 / 2), sqrt(16 / 2))) %%% solve(sigma.2.2) %%% c(sqrt(16 /
2), sqrt(16 / 2))

power.new_16.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_16.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)

t12.2.iv = c(power.t1,
              power.new_2.t1,
              power.new_4.t1,
              power.new_8.t1,
              power.new_16.t1)
t22.2.iv = c(power.t2,
              power.new_2.t2,
              power.new_4.t2,
              power.new_8.t2,
              power.new_16.t2)

# case iv, p=3
# delta = 2
ncp = t(c(sqrt(2 / 3), sqrt(2 / 3), sqrt(2 / 3))) %%% solve(sigma.3.3) %%%
c(sqrt(2 / 3), sqrt(2 / 3), sqrt(2 / 3))
power.new_2.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_2.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
# delta = 4
ncp = t(c(sqrt(4 / 3), sqrt(4 / 3), sqrt(4 / 3))) %%% solve(sigma.3.3) %%%
c(sqrt(4 / 3), sqrt(4 / 3), sqrt(4 / 3))
power.new_4.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_4.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
# delta = 8
ncp = t(c(sqrt(8 / 3), sqrt(8 / 3), sqrt(8 / 3))) %%% solve(sigma.3.3) %%%
c(sqrt(8 / 3), sqrt(8 / 3), sqrt(8 / 3))
power.new_8.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_8.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)
# delta = 16
ncp = t(c(sqrt(16 / 3), sqrt(16 / 3), sqrt(16 / 3))) %%% solve(sigma.3.3) %%%
c(sqrt(16 / 3), sqrt(16 / 3), sqrt(16 / 3))
power.new_16.t1 = 1 - pf(critical.t1, 2, n1 + n2 - 2 - 1, ncp = ncp)
power.new_16.t2 = 1 - pf(critical.t2, 2, n1 + n2 - 2 - 1, ncp = ncp)

t12.3.iv = c(power.t1,
              power.new_2.t1,
              power.new_4.t1,
              power.new_8.t1,
              power.new_16.t1)
t22.3.iv = c(power.t2,
              power.new_2.t2,
              power.new_4.t2,

```

```

        power.new_8.t2,
        power.new_16.t2)

par(mfrow = c(2, 2))
delta <- c(0, 2, 4, 8, 16)
plot(
  delta,
  t12.2.i,
  type = "o",
  col = "red",
  xlab = "delta",
  ylab = "power",
  main = "Graph 2: Sample Size \n 100, 100,100",
  font.main = 0.5
)
lines(t12.3.i, type = "o", col = "blue")
lines(t22.2.i, type = "o", col = "green")
lines(t22.3.i, type = "o", col = "black")
legend(
  'bottomright',
  cex = 0.6,
  c("T12, p=2", "T12, p=3", "T22, p=2", "T22, p=3"),
  lty = 1,
  col = c("red", "blue", "green", "black")
)
plot(
  delta,
  t12.2.ii,
  type = "o",
  col = "red",
  xlab = "delta",
  ylab = "power",
  main = "Graph 3: Sample Size \n 25, 100,25",
  font.main = 0.5
)
lines(t12.3.ii, type = "o", col = "blue")
lines(t22.2.ii, type = "o", col = "green")
lines(t22.3.ii, type = "o", col = "black")
legend(
  'bottomright',
  cex = 0.6,
  c("T12, p=2", "T12, p=3", "T22, p=2", "T22, p=3"),
  lty = 1,
  col = c("red", "blue", "green", "black")
)
plot(
  delta,
  t12.2.iii,
  type = "o",
  col = "red",
  xlab = "delta",
  ylab = "power",
  main = "Graph 4: Sample Size \n 100, 25,100",

```

```

    font.main = 0.5
)
lines(t12.3.iii, type = "o", col = "blue")
lines(t22.2.iii, type = "o", col = "green")
lines(t22.3.iii, type = "o", col = "black")
legend(
  'bottomright',
  cex = 0.6,
  c("T12, p=2", "T12, p=3", "T22, p=2", "T22, p=3"),
  lty = 1,
  col = c("red", "blue", "green", "black")
)
plot(
  delta,
  t12.2.iv,
  type = "o",
  col = "red",
  xlab = "delta",
  ylab = "power",
  main = "Graph 5: Sample Size \n 25, 25,100",
  font.main = 0.5
)
lines(t12.3.iv, type = "o", col = "blue")
lines(t22.2.iv, type = "o", col = "green")
lines(t22.3.iv, type = "o", col = "black")
legend(
  'bottomright',
  cex = 0.6,
  c("T12, p=2", "T12, p=3", "T22, p=2", "T22, p=3"),
  lty = 1,
  col = c("red", "blue", "green", "black")
)

```

Part 3 Code

```

#following part (a) step 1. mu1=mu2=0, mu3 and Sigma.
library("EnvStats")
#Multivariate t-distribution. p=2 - T12 Test
tests.t12.2 <- c()
generator.t12.2 = function(n1, n2, n3) {
  mu1 = c(0, 0)
  mu2 = c(0, 0)
  sigma2 = sigma.2.2 * (10 - 2) / 10
  set.seed(625)
  for (i in 1:1000) {
    x1 = rmvt(n1,
              sigma2,
              df = 10)
    x2 = rmvt(n2,
              sigma2,
              df = 10)
    x3 = rmvt(n3,

```

```

        sigma2,
        df = 10)
S_pool.2.2 = ((n1 - 1) * cov(x1) + (n2 - 1) * cov(x2)) / (n1 + n2 -
2)
T1_square = (1 / n1 + 1 / n2) ^ (-1) * t(colMeans(x1) - colMeans(x2)) %*%
inv(S_pool.2.2) %*%
(colMeans(x1) - colMeans(x2))
tests.t12.2[i] = T1_square
}
return(tests.t12.2)
}

#Multivariate t-distribution. p=2 - T22 Test
tests.t22.2 <- c()
generator.t22.2 = function(n1, n2, n3) {
  mu1 = c(0, 0)
  mu2 = c(0, 0)
  mu3 = c(10, 10)
  sigma2 = sigma.2.2 * (10 - 2) / 10
  set.seed(625)
  for (i in 1:1000) {
    x1 = rmvt(n1,
              sigma2,
              df = 10)
    x2 = rmvt(n2,
              sigma2,
              df = 10)
    x3 = rmvt(n3,
              sigma2,
              df = 10)
    S_pool.3.3 = ((n1 - 1) * cov(x1) + (n2 - 1) * cov(x2) + (n3 - 1) * cov(x3)) /
(n1 + n2 + n3 - 3)
    T2_square = (1 / n1 + 1 / n2) ^ (-1) * t(colMeans(x1) - colMeans(x2)) %*%
inv(S_pool.3.3) %*%
(colMeans(x1) - colMeans(x2))
    tests.t22.2[i] = T2_square
  }
  return(tests.t22.2)
}

#Multivariate t-distribution. p=3 - T12 Test
tests.t12.3 <- c()
generator.t12.3 = function(n1, n2, n3) {
  mu1 = c(0, 0, 0)
  mu2 = c(0, 0, 0)
  sigma3 = sigma.3.3 * (10 - 2) / 10
  set.seed(625)
  for (i in 1:1000) {
    x1 = rmvt(n1,
              sigma3,
              df = 10)
    x2 = rmvt(n2,
              sigma3,
              df = 10)

```

```

    x3 = rmvt(n3,
              sigma3,
              df = 10)
    S_pool.2.2 = ((n1 - 1) * cov(x1) + (n2 - 1) * cov(x2)) / (n1 + n2 -
2)
    T1_square = (1 / n1 + 1 / n2) ^ (-1) * t(colMeans(x1) - colMeans(x2)) %*%
    inv(S_pool.2.2) %*%
    (colMeans(x1) - colMeans(x2))
    tests.t12.3[i] = T1_square
  }
  return(tests.t12.3)
}

#Multivariate t-distribution. p=3 - T22 Test
tests.t22.3 <- c()
generator.t22.3 = function(n1, n2, n3) {
  mu1 = c(0, 0, 0)
  mu2 = c(0, 0, 0)
  mu3 = c(10, 10, 10)
  sigma3 = sigma.3.3 * (10 - 2) / 10
  set.seed(625)
  for (i in 1:1000) {
    x1 = rmvt(n1,
              sigma3,
              df = 10)
    x2 = rmvt(n2,
              sigma3,
              df = 10)
    x3 = rmvt(n3,
              sigma3,
              df = 10)
    S_pool.3.3 = ((n1 - 1) * cov(x1) + (n2 - 1) * cov(x2) + (n3 - 1) * cov(x3)) /
    (n1 + n2 + n3 - 3)
    T2_square = (1 / n1 + 1 / n2) ^ (-1) * t(colMeans(x1) - colMeans(x2)) %*%
    inv(S_pool.3.3) %*%
    (colMeans(x1) - colMeans(x2))
    T2_square.all[i] = T2_square
    tests.t22.3[i] = T2_square
  }
  return(tests.t22.3)
}

#case i: $n_1=n_2=n_3$:
cat("p=2: ")
dist_t12.2_1 <- ecdf(generator.t12.2(100, 100, 100))
summary(dist_t12.2_1)
dist_t22.2_1 <- ecdf(generator.t22.2(100, 100, 100))
summary(dist_t22.2_1)
cat("p=3: ")
dist_t12.3_1 <- ecdf(generator.t12.3(100, 100, 100))
summary(dist_t12.3_1)
dist_t22.3_1 <- ecdf(generator.t22.3(100, 100, 100))
summary(dist_t22.3_1)

```



```

#case ii $n_1<n_2, n_3=n_1$:
cat("p=2: ")
dist_t12.2_2 <- ecdf(generator.t12.2(25, 100, 25))
summary(dist_t12.2_2)
dist_t22.2_2 <- ecdf(generator.t22.2(25, 100, 25))
summary(dist_t22.2_2)
cat("p=3: ")
dist_t12.3_2 <- ecdf(generator.t12.3(25, 100, 25))
summary(dist_t12.3_2)
dist_t22.3_2 <- ecdf(generator.t22.3(25, 100, 25))
summary(dist_t22.3_2)

#case iii $n_1>n_2, n_3=n_1$:
cat("p=2: ")
dist_t12.2_3 <- ecdf(generator.t12.2(100, 25, 100))
summary(dist_t12.2_3)
dist_t22.2_3 <- ecdf(generator.t22.2(100, 25, 100))
summary(dist_t22.2_3)
cat("p=3: ")
dist_t12.3_3 <- ecdf(generator.t12.3(100, 25, 100))
summary(dist_t12.3_3)
dist_t22.3_3 <- ecdf(generator.t22.3(100, 25, 100))
summary(dist_t22.3_3)

#case iv $n_1=n_2=n, n_3>n$:
cat("p=2: ")
dist_t12.2_4 <- ecdf(generator.t12.2(25, 25, 100))
summary(dist_t12.2_4)
dist_t22.2_4 <- ecdf(generator.t22.2(25, 25, 100))
summary(dist_t22.2_4)
cat("p=3: ")
dist_t12.3_4 <- ecdf(generator.t12.3(25, 25, 100))
summary(dist_t12.3_4)
dist_t22.3_4 <- ecdf(generator.t22.3(25, 25, 100))
summary(dist_t22.3_4)

## Summary Results for Multivariate t-distribution

#p = 2 histograms for Test T12
library(knitr)
library(kableExtra)
par(mfrow = c(2, 4))
hist(
  generator.t12.2(100, 100, 100),
  main = "Multivariate t-Test Test Statistic \n Sample Size 100, 100,100" ,
  xlab = "Test Statistic",
  cex.main = 0.7
)
hist(
  generator.t12.2(25, 100, 25),
  main = "Multivariate t-Test Test Statistic \n Sample Size 25, 100,25" ,
  xlab = "Test Statistic",
  cex.main = 0.7
)

```

```

)
hist(
  generator.t12.2(100, 25, 100),
  main = "Multivariate t-Test Test Statistic \n Sample Size 100, 25,100" ,
  xlab = "Test Statistic",
  cex.main = 0.7
)
hist(
  generator.t12.2(25, 25, 100),
  main = "Multivariate t-Test Test Statistic \n Sample Size 25, 25,100" ,
  xlab = "Test Statistic",
  cex.main = 0.7
)

#table
table = as.matrix(cbind(
  summary(dist_t12.2_1),
  summary(dist_t12.2_2),
  summary(dist_t12.2_3),
  summary(dist_t12.2_4)
))
knitr::kable(
  table,
  "pipe",
  col.names = c(
    "Sample Size 100, 100,100",
    "Sample Size 25, 100,25",
    "Sample Size 100, 25,100",
    "Sample Size 25, 25,100"
  )
) %>%
  kable_styling(latex_options = "HOLD_position")

#p = 2 histograms for Test T22
library(knitr)
library(kableExtra)
par(mfrow = c(2, 4))
hist(
  generator.t22.2(100, 100, 100),
  main = "Multivariate t-Test Test Statistic \n Sample Size 100, 100,100" ,
  xlab = "Test Statistic",
  cex.main = 0.7
)
hist(
  generator.t22.2(25, 100, 25),
  main = "Multivariate t-Test Test Statistic \n Sample Size 25, 100,25" ,
  xlab = "Test Statistic",
  cex.main = 0.7
)
hist(
  generator.t22.2(100, 25, 100),
  main = "Multivariate t-Test Test Statistic \n Sample Size 100, 25,100" ,
  xlab = "Test Statistic",

```

```

    cex.main = 0.7
)
hist(
  generator.t22.2(25, 25, 100),
  main = "Multivariate t-Test Test Statistic \n Sample Size 25, 25,100" ,
  xlab = "Test Statistic",
  cex.main = 0.7
)

#table
table = as.matrix(cbind(
  summary(dist_t22.2_1),
  summary(dist_t22.2_2),
  summary(dist_t22.2_3),
  summary(dist_t22.2_4)
))
knitr::kable(
  table,
  "pipe",
  col.names = c(
    "Sample Size 100, 100,100",
    "Sample Size 25, 100,25",
    "Sample Size 100, 25,100",
    "Sample Size 25, 25,100"
  )
) %>%
  kable_styling(latex_options = "HOLD_position")

#p = 3 histograms for Test T12
par(mfrow = c(2, 4))
hist(
  generator.t12.3(100, 100, 100),
  main = "Multivariate t-Test Test Statistic \n Sample Size 100, 100,100" ,
  xlab = "Test Statistic",
  cex.main = 0.7
)
hist(
  generator.t12.3(25, 100, 25),
  main = "Multivariate t-Test Test Statistic \n Sample Size 25, 100,25" ,
  xlab = "Test Statistic",
  cex.main = 0.7
)
hist(
  generator.t12.3(100, 25, 100),
  main = "Multivariate t-Test Test Statistic \n Sample Size 100, 25,100" ,
  xlab = "Test Statistic",
  cex.main = 0.7
)
hist(
  generator.t12.3(25, 25, 100),
  main = "Multivariate t-Test Test Statistic \n Sample Size 25, 25,100" ,
  xlab = "Test Statistic",
  cex.main = 0.7
)

```

```

)

#table
table = as.matrix(cbind(
  summary(dist_t12.3_1),
  summary(dist_t12.3_2),
  summary(dist_t12.3_3),
  summary(dist_t12.3_4)
))
knitr::kable(
  table,
  "pipe",
  col.names = c(
    "Sample Size 100, 100,100",
    "Sample Size 25, 100,25",
    "Sample Size 100, 25,100",
    "Sample Size 25, 25,100"
  )
) %>%
  kable_styling(latex_options = "HOLD_position")

#p = 3 histograms for Test T22
par(mfrow = c(2, 4))
hist(
  generator.t22.3(100, 100, 100),
  main = "Multivariate t-Test Test Statistic \n Sample Size 100, 100,100" ,
  xlab = "Test Statistic",
  cex.main = 0.7
)
hist(
  generator.t22.3(25, 100, 25),
  main = "Multivariate t-Test Test Statistic \n Sample Size 25, 100,25" ,
  xlab = "Test Statistic",
  cex.main = 0.7
)
hist(
  generator.t22.3(100, 25, 100),
  main = "Multivariate t-Test Test Statistic \n Sample Size 100, 25,100" ,
  xlab = "Test Statistic",
  cex.main = 0.7
)
hist(
  generator.t22.3(25, 25, 100),
  main = "Multivariate t-Test Test Statistic \n Sample Size 25, 25,100" ,
  xlab = "Test Statistic",
  cex.main = 0.7
)

#table
table = as.matrix(cbind(
  summary(dist_t22.3_1),
  summary(dist_t22.3_2),
  summary(dist_t22.3_3),

```

```

summary(dist_t22.3_4)
))
knitr::kable(
  table,
  "pipe",
  col.names = c(
    "Sample Size 100, 100,100",
    "Sample Size 25, 100,25",
    "Sample Size 100, 25,100",
    "Sample Size 25, 25,100"
  )
) %>%
kable_styling(latex_options = "HOLD_position")

#following part (a) step 1. mu1=mu2=0, mu3 and Sigma.
#(i) Multivariate laplace distribution. p=2
library("LaplacesDemon")
#Multivariate laplace distribution. p=2 - T12 Test
tests.l12.2 <- c()
generator.l12.2 = function(n1, n2, n3) {
  mu1 = c(0, 0)
  mu2 = c(0, 0)
  mu3 = c(10, 10)
  set.seed(625)
  for (i in 1:1000) {
    x1 = rmvl(n = n1, mu1, sigma.2.2)
    x2 = rmvl(n = n2, mu2, sigma.2.2)
    x3 = rmvl(n = n3, mu3, sigma.2.2)
    S_pool.2.2 = ((n1 - 1) * cov(x1) + (n2 - 1) * cov(x2)) / (n1 + n2 -
2)

    T1_square = (1 / n1 + 1 / n2) ^ (-1) * t(colMeans(x1) - colMeans(x2)) %*%
      inv(S_pool.2.2) %*%
      (colMeans(x1) - colMeans(x2))
    tests.l12.2[i] = T1_square
  }
  return(tests.l12.2)
}

#Multivariate laplace distribution. p=2 - T22 Test
tests.l22.2 <- c()
generator.l22.2 = function(n1, n2, n3) {
  mu1 = c(0, 0)
  mu2 = c(0, 0)
  mu3 = c(10, 10)
  set.seed(625)
  for (i in 1:1000) {
    x1 = rmvl(n = n1, mu1, sigma.2.2)
    x2 = rmvl(n = n2, mu2, sigma.2.2)
    x3 = rmvl(n = n3, mu3, sigma.2.2)
    S_pool.3.3 = ((n1 - 1) * cov(x1) + (n2 - 1) * cov(x2) + (n3 - 1) * cov(x3)) /
      (n1 + n2 + n3 - 3)
    T2_square = (1 / n1 + 1 / n2) ^ (-1) * t(colMeans(x1) - colMeans(x2)) %*%
      inv(S_pool.3.3) %*%
      (colMeans(x1) - colMeans(x2))
  }
}

```

```

    tests.l22.2[i] = T2_square
  }
  return(tests.l22.2)
}

#(i) Multivariate laplace distribution. p=3
library(QRM)
#Multivariate t-distribution. p=3 - T12 Test
tests.l12.3 <- c()
sigma3 <- eigenmeth(sigma.3.3, delta = 0.00000001)
generator.l12.3 = function(n1, n2, n3) {
  mu1 = c(0, 0, 0)
  mu2 = c(0, 0, 0)
  mu3 = c(10, 10, 10)
  set.seed(625)
  for (i in 1:1000) {
    x1 = rmvl(n = n1, mu1, round(sigma3, 3))
    x2 = rmvl(n = n2, mu2, round(sigma3, 3))
    x3 = rmvl(n = n3, mu3, round(sigma3, 3))

    S_pool.2.2 = ((n1 - 1) * cov(x1) + (n2 - 1) * cov(x2)) / (n1 + n2 - 2)

    T1_square = (1 / n1 + 1 / n2) ^ (-1) * t(colMeans(x1) - colMeans(x2)) %*%
      inv(S_pool.2.2) %*%
      (colMeans(x1) - colMeans(x2))
    tests.l12.3[i] = T1_square
  }
  return(tests.l12.3)
}

#Multivariate t-distribution. p=3 - T22 Test
tests.l22.3 <- c()
generator.l22.3 = function(n1, n2, n3) {
  mu1 = c(0, 0, 0)
  mu2 = c(0, 0, 0)
  mu3 = c(10, 10, 10)
  set.seed(625)
  for (i in 1:1000) {
    x1 = rmvl(n = n1, mu1, round(sigma3, 3))
    x2 = rmvl(n = n2, mu2, round(sigma3, 3))
    x3 = rmvl(n = n3, mu3, round(sigma3, 3))

    S_pool.3.3 = ((n1 - 1) * cov(x1) + (n2 - 1) * cov(x2) + (n3 - 1) * cov(x3)) /
      (n1 + n2 + n3 - 3)
    T2_square = (1 / n1 + 1 / n2) ^ (-1) * t(colMeans(x1) - colMeans(x2)) %*%
      inv(S_pool.3.3) %*%
      (colMeans(x1) - colMeans(x2))
    T2_square.all[i] = T2_square
    tests.l22.3[i] = T2_square
  }
  return(tests.l22.3)
}

#case i: $n_1=n_2=n_3$:

```

```

cat("p=2: ")
dist_l12.2_1 <- ecdf(generator.l12.2(100, 100, 100))
summary(dist_l12.2_1)
dist_l22.2_1 <- ecdf(generator.l22.2(100, 100, 100))
generator.l22.2(100, 100, 100)
summary(dist_l22.2_1)
cat("p=3: ")
dist_l12.3_1 <- ecdf(generator.l12.3(100, 100, 100))
summary(dist_l12.3_1)
dist_l22.3_1 <- ecdf(generator.l22.3(100, 100, 100))
summary(dist_l22.3_1)

#case ii $n_1 < n_2, n_3 = n_1$:
cat("p=2: ")
dist_l12.2_2 <- ecdf(generator.l12.2(25, 100, 25))
summary(dist_l12.2_2)
dist_l22.2_2 <- ecdf(generator.l22.2(25, 100, 25))
summary(dist_l22.2_2)
cat("p=3: ")
dist_l12.3_2 <- ecdf(generator.l12.3(25, 100, 25))
summary(dist_l12.3_2)
dist_l22.3_2 <- ecdf(generator.l22.3(25, 100, 25))
summary(dist_l22.3_2)

#case iii $n_1 > n_2, n_3 = n_1$:
cat("p=2: ")
dist_l12.2_3 <- ecdf(generator.l12.2(100, 25, 100))
summary(dist_l12.2_3)
dist_l22.2_3 <- ecdf(generator.l22.2(100, 25, 100))
summary(dist_l22.2_3)
cat("p=3: ")
dist_l12.3_3 <- ecdf(generator.l12.3(100, 25, 100))
summary(dist_l12.3_3)
dist_l22.3_3 <- ecdf(generator.l22.3(100, 25, 100))
summary(dist_l22.3_3)

#case iv $n_1 = n_2 = n, n_3 > n$:
cat("p=2: ")
dist_l12.2_4 <- ecdf(generator.l12.2(25, 25, 100))
summary(dist_l12.2_4)
dist_l22.2_4 <- ecdf(generator.l22.2(25, 25, 100))
summary(dist_l22.2_4)
cat("p=3: ")
dist_l12.3_4 <- ecdf(generator.l12.3(25, 25, 100))
summary(dist_l12.3_4)
dist_l22.3_4 <- ecdf(generator.l22.3(25, 25, 100))
summary(dist_l22.3_4)

## Summary Results for Multivariate Laplace Distribution

#p = 2 histograms for Test T12
par(mfrow = c(2, 4))
hist(

```

```

generator.l12.2(100, 100, 100),
main = "Multivariate Laplace Test Statistic \n Sample Size 100, 100,100" ,
xlab = "Test Statistic",
cex.main = 0.7
)
hist(
generator.l12.2(25, 100, 25),
main = "Multivariate Laplace Test Statistic \n Sample Size 25, 100,25" ,
xlab = "Test Statistic",
cex.main = 0.7
)
hist(
generator.l12.2(100, 25, 100),
main = "Multivariate Laplace Test Statistic \n Sample Size 100, 25,100" ,
xlab = "Test Statistic",
cex.main = 0.7
)
hist(
generator.l12.2(25, 25, 100),
main = "Multivariate Laplace Test Statistic \n Sample Size 25, 25,100",
xlab = "Test Statistic",
cex.main = 0.7
)

#table
table = as.matrix(cbind(
summary(dist_l12.2_1),
summary(dist_l12.2_2),
summary(dist_l12.2_3),
summary(dist_l12.2_4)
))
knitr::kable(
table,
"pipe",
col.names = c(
"Sample Size 100, 100,100",
"Sample Size 25, 100,25",
"Sample Size 100, 25,100",
"Sample Size 25, 25,100"
)
) %>%
kable_styling(latex_options = "HOLD_position")

#p = 2 histograms for Test T22
par(mfrow = c(2, 4))
hist(
generator.l22.2(100, 100, 100),
main = "Multivariate Laplace Test Statistic \n Sample Size 100, 100,100",
xlab = "Test Statistic",
cex.main = 0.7
)
hist(
generator.l22.2(25, 100, 25),

```



```

    main = "Multivariate Laplace Test Statistic \n Sample Size 25, 100,25" ,
    xlab = "Test Statistic",
    cex.main = 0.7
)
hist(
  generator.l22.2(100, 25, 100),
  main = "Multivariate Laplace Test Statistic \n Sample Size 100, 25,100" ,
  xlab = "Test Statistic",
  cex.main = 0.7
)
hist(
  generator.l22.2(25, 25, 100),
  main = "Multivariate Laplace Test Statistic \n Sample Size 25, 25,100" ,
  xlab = "Test Statistic",
  cex.main = 0.7
)

#table
table = as.matrix(cbind(
  summary(dist_l22.2_1),
  summary(dist_l22.2_2),
  summary(dist_l22.2_3),
  summary(dist_l22.2_4)
))
knitr::kable(
  table,
  "pipe",
  col.names = c(
    "Sample Size 100, 100,100",
    "Sample Size 25, 100,25",
    "Sample Size 100, 25,100",
    "Sample Size 25, 25,100"
  )
) %>%
  kable_styling(latex_options = "HOLD_position")

#p = 3 histograms for Test T12
par(mfrow = c(2, 4))
hist(
  generator.l12.3(100, 100, 100),
  main = "Multivariate Laplace Test Statistic \n Sample Size 100, 100,100",
  xlab = "Test Statistic",
  cex.main = 0.7
)
hist(
  generator.l12.3(25, 100, 25),
  main = "Multivariate Laplace Test Statistic \n Sample Size 25, 100,25",
  xlab = "Test Statistic",
  cex.main = 0.7
)
hist(
  generator.l12.3(100, 25, 100),
  main = "Multivariate Laplace Test Statistic \n Sample Size 100, 25,100" ,

```

```

xlab = "Test Statistic",
cex.main = 0.7
)
hist(
  generator.l12.3(25, 25, 100),
  main = "Multivariate Laplace Test Statistic \n Sample Size 25, 25,100",
  xlab = "Test Statistic",
  cex.main = 0.7
)

#table
table = as.matrix(cbind(
  summary(dist_l12.3_1),
  summary(dist_l12.3_2),
  summary(dist_l12.3_3),
  summary(dist_l12.3_4)
))
knitr::kable(
  table,
  "pipe",
  col.names = c(
    "Sample Size 100, 100,100",
    "Sample Size 25, 100,25",
    "Sample Size 100, 25,100",
    "Sample Size 25, 25,100"
  )
) %>%
  kable_styling(latex_options = "HOLD_position")

#p = 3 histograms for Test T22
par(mfrow = c(2, 4))
hist(
  generator.l22.3(100, 100, 100),
  main = "Multivariate Laplace Test Statistic \n Sample Size 100, 100,100",
  xlab = "Test Statistic",
  cex.main = 0.7
)
hist(
  generator.l22.3(25, 100, 25),
  main = "Multivariate Laplace Test Statistic \n Sample Size 25, 100,25",
  xlab = "Test Statistic",
  cex.main = 0.7
)
hist(
  generator.l22.3(100, 25, 100),
  main = "Multivariate Laplace Test Statistic \n Sample Size 100, 25,100" ,
  xlab = "Test Statistic",
  cex.main = 0.7
)
hist(
  generator.l22.3(25, 25, 100),
  main = "Multivariate Laplace Test Statistic \n Sample Size 25, 25,100" ,
  xlab = "Test Statistic",

```

```

    cex.main = 0.7
)

#table
table = as.matrix(cbind(
  summary(dist_122.3_1),
  summary(dist_122.3_2),
  summary(dist_122.3_3),
  summary(dist_122.3_4)
))
knitr::kable(
  table,
  "pipe",
  col.names = c(
    "Sample Size 100, 100,100",
    "Sample Size 25, 100,25",
    "Sample Size 100, 25,100",
    "Sample Size 25, 25,100"
  )
) %>%
  kable_styling(latex_options = "HOLD_position")

##### QUANTILES FROM PART 3 WITH MULTIVARIATE T DISTRIBUTION #####
#Quantiles p = 2 for Test T12 Part 3
upper.quantile_t12.2_1 = quantile(generator.t12.2(100, 100, 100)[order(generator.t12.2(100, 100, 100))]
upper.quantile_t12.2_2 = quantile(generator.t12.2(25, 100, 25)[order(generator.t12.2(25, 100, 25))], c(
upper.quantile_t12.2_3 = quantile(generator.t12.2(100, 25, 100)[order(generator.t12.2(100, 25, 100))], c(
upper.quantile_t12.2_4 = quantile(generator.t12.2(25, 25, 100)[order(generator.t12.2(25, 25, 100))], c(
#Quantiles p = 2 for Test T22 Part 3
upper.quantile_t22.2_1 = quantile(generator.t22.2(100, 100, 100)[order(generator.t22.2(100, 100, 100))]
upper.quantile_t22.2_2 = quantile(generator.t22.2(25, 100, 25)[order(generator.t22.2(25, 100, 25))], c(
upper.quantile_t22.2_3 = quantile(generator.t22.2(100, 25, 100)[order(generator.t22.2(100, 25, 100))], c(
upper.quantile_t22.2_4 = quantile(generator.t22.2(25, 25, 100)[order(generator.t22.2(25, 25, 100))], c(
#Quantiles p = 3 for Test T12 Part 3
upper.quantile_t12.3_1 = quantile(generator.t12.3(100, 100, 100)[order(generator.t12.3(100, 100, 100))]
upper.quantile_t12.3_2 = quantile(generator.t12.3(25, 100, 25)[order(generator.t12.3(25, 100, 25))], c(
upper.quantile_t12.3_3 = quantile(generator.t12.3(100, 25, 100)[order(generator.t12.3(100, 25, 100))], c(
upper.quantile_t12.3_4 = quantile(generator.t12.3(25, 25, 100)[order(generator.t12.3(25, 25, 100))], c(
#Quantiles p = 3 for Test T22 Part 3
upper.quantile_t22.3_1 = quantile(generator.t22.3(100, 100, 100)[order(generator.t22.3(100, 100, 100))]
upper.quantile_t22.3_2 = quantile(generator.t22.3(25, 100, 25)[order(generator.t22.3(25, 100, 25))], c(
upper.quantile_t22.3_3 = quantile(generator.t22.3(100, 25, 100)[order(generator.t22.3(100, 25, 100))], c(
upper.quantile_t22.3_4 = quantile(generator.t22.3(25, 25, 100)[order(generator.t22.3(25, 25, 100))], c(

##### QUANTILES FROM PART 3 WITH LAPLACE DISTRIBUTION #####
#Quantiles p = 2 for Test T12 Part 3
upper.quantile_l12.2_1 = quantile(generator.l12.2(100, 100, 100)[order(generator.l12.2(100, 100, 100))]
upper.quantile_l12.2_2 = quantile(generator.l12.2(25, 100, 25)[order(generator.l12.2(25, 100, 25))], c(
upper.quantile_l12.2_3 = quantile(generator.l12.2(100, 25, 100)[order(generator.l12.2(100, 25, 100))], c(
upper.quantile_l12.2_4 = quantile(generator.l12.2(25, 25, 100)[order(generator.l12.2(25, 25, 100))], c(
#Quantiles p = 2 for Test T22 Part 3
upper.quantile_l22.2_1 = quantile(generator.l22.2(100, 100, 100)[order(generator.l22.2(100, 100, 100))]
upper.quantile_l22.2_2 = quantile(generator.l22.2(25, 100, 25)[order(generator.l22.2(25, 100, 25))], c(

```

```

upper.quantile_l22.2_3 = quantile(generator.l22.2(100, 25, 100)[order(generator.l22.2(100, 25, 100))], c(
upper.quantile_l22.2_4 = quantile(generator.l22.2(25, 25, 100)[order(generator.l22.2(25, 25, 100))], c(
#Quantiles p = 3 for Test T12 Part 3
upper.quantile_l12.3_1 = quantile(generator.l12.3(100, 100, 100)[order(generator.l12.3(100, 100, 100))], c(
upper.quantile_l12.3_2 = quantile(generator.l12.3(25, 100, 25)[order(generator.l12.3(25, 100, 25))], c(
upper.quantile_l12.3_3 = quantile(generator.l12.3(100, 25, 100)[order(generator.l12.3(100, 25, 100))], c(
upper.quantile_l12.3_4 = quantile(generator.l12.3(25, 25, 100)[order(generator.l12.3(25, 25, 100))], c(
#Quantiles p = 3 for Test T22 Part 3
upper.quantile_l22.3_1 = quantile(generator.l22.3(100, 100, 100)[order(generator.l22.3(100, 100, 100))], c(
upper.quantile_l22.3_2 = quantile(generator.l22.3(25, 100, 25)[order(generator.l22.3(25, 100, 25))], c(
upper.quantile_l22.3_3 = quantile(generator.l22.3(100, 25, 100)[order(generator.l22.3(100, 25, 100))], c(
upper.quantile_l22.3_4 = quantile(generator.l22.3(25, 25, 100)[order(generator.l22.3(25, 25, 100))], c(

```

Part 4 Code

```

#MVN
# CASE 1
s = qchisq(c(.85, .90, .95, .975, .99, .995), df = 2)
bias_n12.2_1 = upper.quantile_t12_1.1 - s
RB_n12.2_1 = bias_n12.2_1 / s * 100
# CASE 2
bias_n12.2_2 = upper.quantile_t12_2.1 - s
RB_n12.2_2 = bias_n12.2_2 / s * 100
# CASE 3
bias_n12.2_3 = upper.quantile_t12_3.1 - s
RB_n12.2_3 = bias_n12.2_3 / s * 100
# CASE 4
bias_n12.2_4 = upper.quantile_t12_4.1 - s
RB_n12.2_4 = bias_n12.2_4 / s * 100

table <- cbind(
  bias_n12.2_1,
  RB_n12.2_1,
  bias_n12.2_2,
  RB_n12.2_2,
  bias_n12.2_3,
  RB_n12.2_3,
  bias_n12.2_4,
  RB_n12.2_4
)
knitr::kable(
  table,
  col.names = c("Bias", "RB", "Bias", "RB", "Bias", "RB", "Bias", "RB"),
  caption = "Bias and RB for $T_1^2$ Test where p = 2 for Multivariate Normal Distribution"
) %>%
  add_header_above(
    c(
      " " = 1,
      "Sample Size 100, 100,100" = 2,
      "Sample Size 25, 100, 25" = 2,
      "Sample Size 100, 25, 100" = 2,

```

```

      "Sample Size 25, 25, 100" = 2
    )
  ) %>%
  kable_styling(latex_options = "HOLD_position")

# CASE 1
s = qchisq(c(.85, .90, .95, .975, .99, .995), df = 3)
bias_n12.3_1 = upper.quantile_t12_1.2 - s
RB_n12.3_1 = bias_n12.3_1 / s * 100
# CASE 2
bias_n12.3_2 = upper.quantile_t12_2.2 - s
RB_n12.3_2 = bias_n12.3_2 / s * 100
# CASE 3
bias_n12.3_3 = upper.quantile_t12_3.2 - s
RB_n12.3_3 = bias_n12.3_3 / s * 100
# CASE 4
bias_n12.3_4 = upper.quantile_t12_4.2 - s
RB_n12.3_4 = bias_n12.3_4 / s * 100

table <- cbind(
  bias_n12.3_1,
  RB_n12.3_1,
  bias_n12.3_2,
  RB_n12.3_2,
  bias_n12.3_3,
  RB_n12.3_3,
  bias_n12.3_4,
  RB_n12.3_4
)
knitr::kable(
  table,
  col.names = c("Bias", "RB", "Bias", "RB", "Bias", "RB", "Bias", "RB"),
  caption = "Bias and RB for  $T_1^2$  Test where  $p = 3$  for Multivariate Normal Distribution"
) %>%
  add_header_above(
    c(
      " " = 1,
      "Sample Size 100, 100, 100" = 2,
      "Sample Size 25, 100, 25" = 2,
      "Sample Size 100, 25, 100" = 2,
      "Sample Size 25, 25, 100" = 2
    )
  ) %>%
  kable_styling(latex_options = "HOLD_position")

# CASE 1
s = qchisq(c(.85, .90, .95, .975, .99, .995), df = 2)
bias_n22.2_1 = upper.quantile_t22_1.1 - s
RB_n22.2_1 = bias_n22.2_1 / s * 100
# CASE 2
bias_n22.2_2 = upper.quantile_t22_2.1 - s
RB_n22.2_2 = bias_n22.2_2 / s * 100
# CASE 3

```

```

bias_n22.2_3 = upper.quantile_t22_3.1 - s
RB_n22.2_3 = bias_n22.2_3 / s * 100
# CASE 4
bias_n22.2_4 = upper.quantile_t22_4.1 - s
RB_n22.2_4 = bias_n22.2_4 / s * 100

table <- cbind(
  bias_n22.2_1,
  RB_n22.2_1,
  bias_n22.2_2,
  RB_n22.2_2,
  bias_n22.2_3,
  RB_n22.2_3,
  bias_n22.2_4,
  RB_n22.2_4
)
knitr::kable(
  table,
  col.names = c("Bias", "RB", "Bias", "RB", "Bias", "RB", "Bias", "RB"),
  caption = "Bias and RB for  $T_2^2$  Test where  $p = 2$  for Multivariate Normal Distribution"
) %>%
  add_header_above(
    c(
      " " = 1,
      "Sample Size 100, 100, 100" = 2,
      "Sample Size 25, 100, 25" = 2,
      "Sample Size 100, 25, 100" = 2,
      "Sample Size 25, 25, 100" = 2
    )
  ) %>%
  kable_styling(latex_options = "HOLD_position")

# CASE 1
s = qchisq(c(.85, .90, .95, .975, .99, .995), df = 3)
bias_n22.3_1 = upper.quantile_t22_1.2 - s
RB_n22.3_1 = bias_n22.3_1 / s * 100
# CASE 2
bias_n22.3_2 = upper.quantile_t22_2.2 - s
RB_n22.3_2 = bias_n22.3_2 / s * 100
# CASE 3
bias_n22.3_3 = upper.quantile_t22_3.2 - s
RB_n22.3_3 = bias_n22.3_3 / s * 100
# CASE 4
bias_n22.3_4 = upper.quantile_t22_4.2 - s
RB_n22.3_4 = bias_n22.3_4 / s * 100

# CASE 1
s = qchisq(c(.85, .90, .95, .975, .99, .995), df = 2)
bias_t12.2_1 = upper.quantile_t12.2_1 - s
RB_t12.2_1 = bias_t12.2_1 / s * 100
# CASE 2
bias_t12.2_2 = upper.quantile_t12.2_2 - s
RB_t12.2_2 = bias_t12.2_2 / s * 100

```

```

# CASE 3
bias_t12.2_3 = upper.quantile_t12.2_3 - s
RB_t12.2_3 = bias_t12.2_3 / s * 100
# CASE 4
bias_t12.2_4 = upper.quantile_t12.2_4 - s
RB_t12.2_4 = bias_t12.2_4 / s * 100

table <- cbind(
  bias_t12.2_1,
  RB_t12.2_1,
  bias_t12.2_2,
  RB_t12.2_2,
  bias_t12.2_3,
  RB_t12.2_3,
  bias_t12.2_4,
  RB_t12.2_4
)
knitr::kable(
  table,
  col.names = c("Bias", "RB", "Bias", "RB", "Bias", "RB", "Bias", "RB"),
  caption = "Bias and RB for  $T_1^2$  Test where  $p = 2$  for Multivariate T Distribution"
) %>%
  add_header_above(
    c(
      " " = 1,
      "Sample Size 100, 100,100" = 2,
      "Sample Size 25, 100, 25" = 2,
      "Sample Size 100, 25, 100" = 2,
      "Sample Size 25, 25, 100" = 2
    )
  ) %>%
  kable_styling(latex_options = "HOLD_position")

# CASE 1
s = qchisq(c(.85, .90, .95, .975, .99, .995), df = 3)
bias_t12.3_1 = upper.quantile_t12.3_1 - s
RB_t12.3_1 = bias_t12.3_1 / s * 100
# CASE 2
bias_t12.3_2 = upper.quantile_t12.3_2 - s
RB_t12.3_2 = bias_t12.3_2 / s * 100
# CASE 3
bias_t12.3_3 = upper.quantile_t12.3_3 - s
RB_t12.3_3 = bias_t12.3_3 / s * 100
# CASE 4
bias_t12.3_4 = upper.quantile_t12.3_4 - s
RB_t12.3_4 = bias_t12.3_4 / s * 100

# CASE 1
s = qchisq(c(.85, .90, .95, .975, .99, .995), df = 2)
bias_t22.2_1 = upper.quantile_t22.2_1 - s
RB_t22.2_1 = bias_t22.2_1 / s * 100
# CASE 2
bias_t22.2_2 = upper.quantile_t22.2_2 - s

```

```

RB_t22.2_2 = bias_t22.2_2 / s * 100
# CASE 3
bias_t22.2_3 = upper.quantile_t22.2_3 - s
RB_t22.2_3 = bias_t22.2_3 / s * 100
# CASE 4
bias_t22.2_4 = upper.quantile_t22.2_4 - s
RB_t22.2_4 = bias_t22.2_4 / s * 100

table <- cbind(
  bias_t22.2_1,
  RB_t22.2_1,
  bias_t22.2_2,
  RB_t22.2_2,
  bias_t22.2_3,
  RB_t22.2_3,
  bias_t22.2_4,
  RB_t22.2_4
)
knitr::kable(
  table,
  col.names = c("Bias", "RB", "Bias", "RB", "Bias", "RB", "Bias", "RB"),
  caption = "Bias and RB for  $T_2^2$  Test where  $p = 2$  for Multivariate T Distribution"
) %>%
  add_header_above(
    c(
      " " = 1,
      "Sample Size 100, 100,100" = 2,
      "Sample Size 25, 100, 25" = 2,
      "Sample Size 100, 25, 100" = 2,
      "Sample Size 25, 25, 100" = 2
    )
  ) %>%
  kable_styling(latex_options = "HOLD_position")

# CASE 1
s = qchisq(c(.85, .90, .95, .975, .99, .995), df = 3)
bias_t22.3_1 = upper.quantile_t22.3_1 - s
RB_t22.3_1 = bias_t22.3_1 / s * 100
# CASE 2
bias_t22.3_2 = upper.quantile_t22.3_2 - s
RB_t22.3_2 = bias_t22.3_2 / s * 100
# CASE 3
bias_t22.3_3 = upper.quantile_t22.3_3 - s
RB_t22.3_3 = bias_t22.3_3 / s * 100
# CASE 4
bias_t22.3_4 = upper.quantile_t22.3_4 - s
RB_t22.3_4 = bias_t22.3_4 / s * 100

# CASE 1
s = qchisq(c(.85, .90, .95, .975, .99, .995), df = 2)
bias_l12.2_1 = upper.quantile_l12.2_1 - s
RB_l12.2_1 = bias_l12.2_1 / s * 100
# CASE 2

```



```

bias_l12.2_2 = upper.quantile_l12.2_2 - s
RB_l12.2_2 = bias_l12.2_2 / s * 100
# CASE 3
bias_l12.2_3 = upper.quantile_l12.2_3 - s
RB_l12.2_3 = bias_l12.2_3 / s * 100
# CASE 4
bias_l12.2_4 = upper.quantile_l12.2_4 - s
RB_l12.2_4 = bias_l12.2_4 / s * 100

table <- cbind(
  bias_l12.2_1,
  RB_l12.2_1,
  bias_l12.2_2,
  RB_l12.2_2,
  bias_l12.2_3,
  RB_l12.2_3,
  bias_l12.2_4,
  RB_l12.2_4
)
knitr::kable(
  table,
  col.names = c("Bias", "RB", "Bias", "RB", "Bias", "RB", "Bias", "RB"),
  caption = "Bias and RB for  $T_1^2$  Test where  $p = 2$  for Multivariate Laplace Distribution"
) %>%
  add_header_above(
    c(
      " " = 1,
      "Sample Size 100, 100,100" = 2,
      "Sample Size 25, 100, 25" = 2,
      "Sample Size 100, 25, 100" = 2,
      "Sample Size 25, 25, 100" = 2
    )
  ) %>%
  kable_styling(latex_options = "HOLD_position")

# CASE 1
s = qchisq(c(.85, .90, .95, .975, .99, .995), df = 3)
bias_l12.3_1 = upper.quantile_l12.3_1 - s
RB_l12.3_1 = bias_l12.3_1 / s
# CASE 2
bias_l12.3_2 = upper.quantile_l12.3_2 - s
RB_l12.3_2 = bias_l12.3_2 / s * 100
# CASE 3
bias_l12.3_3 = upper.quantile_l12.3_3 - s
RB_l12.3_3 = bias_l12.3_3 / s * 100
# CASE 4
bias_l12.3_4 = upper.quantile_l12.3_4 - s
RB_l12.3_4 = bias_l12.3_4 / s * 100

# Use Multivariate Laplace distribution for asymptotic approximation. Below are the bias results for  $T_1^2$ 
# CASE 1
s = qchisq(c(.85, .90, .95, .975, .99, .995), df = 2)
bias_l22.2_1 = upper.quantile_l22.2_1 - s

```

```

RB_122.2_1 = bias_122.2_1 / s * 100
# CASE 2
bias_122.2_2 = upper.quantile_122.2_2 - s
RB_122.2_2 = bias_122.2_2 / s * 100
# CASE 3
bias_122.2_3 = upper.quantile_122.2_3 - s
RB_122.2_3 = bias_122.2_3 / s * 100
# CASE 4
bias_122.2_4 = upper.quantile_122.2_4 - s
RB_122.2_4 = bias_122.2_4 / s * 100

table <- cbind(
  bias_122.2_1,
  RB_122.2_1,
  bias_122.2_2,
  RB_122.2_2,
  bias_122.2_3,
  RB_122.2_3,
  bias_122.2_4,
  RB_122.2_4
)
knitr::kable(
  table,
  col.names = c("Bias", "RB", "Bias", "RB", "Bias", "RB", "Bias", "RB"),
  caption = "Bias and RB for  $T_2^2$  Test where  $p = 2$  for Multivariate Laplace Distribution"
) %>%
  add_header_above(
    c(
      " " = 1,
      "Sample Size 100, 100, 100" = 2,
      "Sample Size 25, 100, 25" = 2,
      "Sample Size 100, 25, 100" = 2,
      "Sample Size 25, 25, 100" = 2
    )
  ) %>%
  kable_styling(latex_options = "HOLD_position")

# Use Multivariate Laplace distribution for asymptotic approximation. Below are the bias results for  $T_2^2$ 
# CASE 1
s = qchisq(c(.85, .90, .95, .975, .99, .995), df = 3)
bias_122.3_1 = upper.quantile_122.3_1 - s
RB_122.3_1 = bias_122.3_1 / s * 100
# CASE 2
bias_122.3_2 = upper.quantile_122.3_2 - s
RB_122.3_2 = bias_122.3_2 / s * 100
# CASE 3
bias_122.3_3 = upper.quantile_122.3_3 - s
RB_122.3_3 = bias_122.3_3 / s * 100
# CASE 4
bias_122.3_4 = upper.quantile_122.3_4 - s
RB_122.3_4 = bias_122.3_4 / s * 100

```