Multivariate Analysis Lecture 9: Use Simulations to Assess Statistical Methods

Zhaoxia Yu
Professor, Department of Statistics

2023-05-02

Section 1

Midterm Project

Too many choices?

- Due to the nature of multivariate analysis, we have see many choices for conducting one-way MANOVA
- Do they work equally well?
- Does their performance depend on the true distribution?
- You will be asked to compare the methods in your midterm project

Midterm Project

- Goal: conduct a simulation study to compare four MANOVA tests

Type I Error Rate

- The type I error rate of a test is the probability of rejecting the null hypothesis when it is actually true.
- In most research contexts, the acceptable type I error rate is set at 0.05, meaning that there is a 5% chance of falsely rejecting the null hypothesis.
- In many test, critical values are chosen to aim to control the type I error rate of a test
- The true type I error rate of a test depends on whether the underlying assumption of a test is met
- A simulation study can help assess whether a statistical test maintains the desired type I error rate under various conditions.

Power

- The power of a statistical test is the probability of correctly rejecting a null hypothesis when it is false.
- In other words, power is the ability of a test to detect a true effect or relationship between variables.
- Power is affected by several factors, including sample size, effect size, and significance level.
- For simple tests, analytical formulas for calculation power might exist. For complicated tests, simulations can be used to evaluate power

Rationale for Simulation Studies

- Simulation studies allow researchers to evaluate and compare the performance of statistical tests under controlled conditions.
- They can help determine which test is most appropriate for a specific situation.
- By simulating data with known properties, researchers can assess the type I error rate, power, and other performance metrics of each test.

Designing a Simulation Study

- Choose the statistical tests to compare: Select the tests you want to evaluate and ensure they are applicable to your research question.
- Define the data-generating process: Create a model to simulate data with known properties that reflect the characteristics of your target population.
- Determine performance metrics: Decide on the metrics you will use to evaluate the performance of each test (e.g., type I error, power).
- Set up the simulation: Determine the number of repetitions and the sample sizes to use in the simulation.
- Analyze and interpret the results: Assess the performance of each test based on the simulation results and draw conclusions.

Section 2

Simulation Studies in Stat Research

A Practical Guide Based on the Paper by Morris et al. (2021)

- Paper reference: Morris, T. P., White, I. R., & Crowther, M. J. (2021). Using simulation studies to evaluate statistical methods. Statistics in Medicine, 40(8), 2073-2102.

Objectives of Simulation Studies

- Evaluate performance of statistical methods
- Compare competing methods
- Assess sensitivity of methods to different data structures
- Investigate robustness of methods to model assumptions

Key Components of a Simulation Study

- Data Generating Process (DGP)
- Method Implementation
- Performance Measures
- Simulation Parameters
- Analysis and Interpretation

Data Generating Process (DGP)

- Define the true model and relationships
- Consider the distribution of the data
- Incorporate relevant covariates and confounders
- Generate data samples based on the defined model

Method Implementation and Performance Measures

- Implement the statistical method(s) being evaluated
- Apply the method(s) to the generated data samples
- Record results and performance metrics for each simulation run
- Select appropriate performance measures to evaluate the method(s)
- Examples: bias, mean squared error (MSE), coverage, power, type I error rate, etc.

Simulation Parameters

- Determine sample size(s) and number of simulation runs
- Consider variations in the DGP or model assumptions
- Ensure adequate coverage of the parameter space
- "Trial and Error" method to choose parameters/effect sizes to obtain reasonable power
    - when the effect size / treatment effect is too small, all methods have low power
    - when the effect size / treatment effect is large, all methods have power close to 100%

Analysis and Interpretation

- Summarize results across simulation runs
- Use graphical and numerical summaries
- Compare methods and assess performance
- Draw conclusions and provide recommendations

Reporting Simulation Studies

- Clearly describe the DGP, method implementation, and performance measures
- Present results in a transparent and reproducible manner
- Discuss limitations and potential biases
- Provide guidance for future research and applications

Analyzing and Interpreting the Results

- Compare the type I error rates and power of the tests.
- Determine which test performs best for your specific situation.
- Consider other factors such as assumptions, sample size, and computational efficiency.
- Use the results to inform your choice of statistical test in your research.

Section 3

A Case Study

Compare two-sample t-test and Mann-Whitney-Wilcoxon test in the presence of outliers

- The two-sample t-test is a widely used test
- Mann-Whitney-Wilcoxon is often used as a nonparametric alternative when distributional assumptions are suspected
- It has been frequently reported that the two-sample t-test is still a good choice as long as the sample size is large enough, even for data from a skewed distribution
- The purpose of this simulation study is to compare their performances in the presence of outliers

Design of Simulation Study

- Verify that indeed that t-test is more powerful when normality assumptions are met
  - Equal variance: t_E
  - Unequal variance: t
- Compare the performance of the two-tests when there are outliers

Functions 1

```
generate_data <- function(n, mu, sd) {
  rnorm(n, mean = mu, sd = sd)}

generate_data_out <- function(n, mu, sd, sd_out, p) {
  n1=round(n*p); n2=n-n1
  c(rnorm(n1, mean = mu, sd = sd), rnorm(n2, mean=mu, sd=sd_out))
  }

run_tests <- function(data1, data2) {
  t_test <- t.test(data1, data2)$p.value
  t_e_test <- t.test(data1, data2, var.equal=TRUE)$p.value
  wilcox_test <- wilcox.test(data1, data2)$p.value
  c(t = t_test, t_e_test=t_e_test, wilcox_test = wilcox_test)}
```

Functions 2

```r
#without outliers
pvalues <- function(n, mu1, mu2,sd, nrep=1000){
  results=matrix(0, n_rep, 3)
  for(b in 1:n_rep){
    data1 <- generate_data(n, mu1, sd)
    data2 <- generate_data(n, mu2, sd)
    results[b, ]=run_tests(data1, data2)
  }
  return(colMeans(results<0.05))}

#with outliers
pvalues_out <- function(n, mu1, mu2, sd, sd_out, p, nrep=1000){
  results=matrix(0, n_rep, 3)
  for(b in 1:n_rep){
    data1 <- generate_data_out(n, mu1, sd, sd_out, p)
    data2 <- generate_data_out(n, mu2, sd, sd_out, p)
    results[b, ]=run_tests(data1, data2)
  }
  return(colMeans(results<0.05))}
```

Power: No Outlier

```
power_n5=rbind(pvalues(n=5, mu1=0, mu2=0.2, sd=1, nrep=1000),
  pvalues(n=5, mu1=0, mu2=0.5, sd=1,nrep=1000),
  pvalues(n=5, mu1=0, mu2=1, sd=1, nrep=1000))

power_n10=rbind(pvalues(n=10, mu1=0, mu2=0.2, sd=1, nrep=1000),
  pvalues(n=10, mu1=0, mu2=0.5, sd=1, nrep=1000),
  pvalues(n=10, mu1=0, mu2=1, sd=1, nrep=1000))

power_n50=rbind(pvalues(n=50, mu1=0, mu2=0.2, sd=1, nrep=1000),
  pvalues(n=50, mu1=0, mu2=0.5, sd=1, nrep=1000),
  pvalues(n=50, mu1=0, mu2=1, sd=1, nrep=1000))

colnames(power_n5)=c("t", "t_e", "r")
colnames(power_n10)=c("t", "t_e", "r")
colnames(power_n50)=c("t", "t_e", "r")
power_normal_wide=data.frame(cbind(rbind(power_n5, power_n10, power_n50),
                                   diff=rep(c(0.2, 0.5, 1.0),3),
                                   n=rep(c(5, 10, 50), each=3)))
power_normal=gather(power_normal_wide, test, power, 1:3, factor_key=TRUE)

ggplot(power_normal, aes(x=diff, y=power,
                         group=interaction(test, n), color=factor(n), shape=test)) +
  geom_point(size=3) +geom_line() +  ggtitle("Power: no outlier")
```

Type I Error Rate: No Outlier

```
type1=rbind(
  pvalues(n=5, mu1=0, mu2=0, sd=1, nrep=1000),
  pvalues(n=10, mu1=0, mu2=0, sd=1, nrep=1000),
  pvalues(n=50, mu1=0, mu2=0, sd=1, nrep=1000))
colnames(type1)=c("t", "t_e", "r")
type1_wide=data.frame(type1, n=c(5,10,50))
type1=gather(type1_wide, test, power, 1:3, factor_key=TRUE)
ggplot(type1, aes(x=n, y=power, group=test, shape=test, color=test))+
  geom_point(size=3) + geom_line()+
  geom_hline(yintercept=0.05, linetype="dashed", color = "red") +
  ggtitle("Type I Error: no outlier")
```

Type I Error Rate: 1% Outliers

```
#1% outliers
type1_out_mat=rbind(
  pvalues_out(n=5, mu1=0, mu2=0, sd=1, sd_out=5, p=0.01, nrep=1000),
  pvalues_out(n=10, mu1=0, mu2=0, sd=1, sd_out=5, p=0.01, nrep=1000),
  pvalues_out(n=20, mu1=0, mu2=0, sd=1, sd_out=5, p=0.01, nrep=1000),
  pvalues_out(n=50, mu1=0, mu2=0, sd=1, sd_out=5, p=0.01, nrep=1000))
colnames(type1_out_mat)=c("t", "t_e", "r")
type1_wide_out=data.frame(type1_out_mat, n=c(5,10,20,50))
type1_out=gather(type1_wide_out, test, power, 1:3, factor_key=TRUE)
ggplot(type1_out, aes(x=n, y=power, group=test, shape=test, color=test))+
  geom_point(size=3) +
  geom_line()+
  geom_hline(yintercept=0.05, linetype="dashed", color = "red")+
  ggtitle("Type I Error: 1% outliers")
```

Type I error rate: 5% Outliers

```r
#5% outliers
type1_out_mat=rbind(
  pvalues_out(n=5, mu1=0, mu2=0, sd=1, sd_out=5, p=0.05, nrep=1000),
  pvalues_out(n=10, mu1=0, mu2=0, sd=1, sd_out=5, p=0.05, nrep=1000),
  pvalues_out(n=20, mu1=0, mu2=0, sd=1, sd_out=5, p=0.05, nrep=1000),
  pvalues_out(n=50, mu1=0, mu2=0, sd=1, sd_out=5, p=0.05, nrep=1000))
colnames(type1_out_mat)=c("t", "t_e", "r")
type1_wide_out=data.frame(type1_out_mat, n=c(5,10,20,50))
type1_out=gather(type1_wide_out, test, power, 1:3, factor_key=TRUE)
ggplot(type1_out, aes(x=n, y=power, group=test, shape=test, color=test))+
  geom_point(size=3) +
  geom_line()+
  geom_hline(yintercept=0.05, linetype="dashed", color = "red")+
  ggtitle("Type I Error: 5% outliers")
```

Conclusions and Limitations

- Conclusions:
  - As has been reported by many others, t-test tends to be more powerful, especially for small sample sizes
  - We examined an outlier model. For this model we observed, the type I error rates of both tests are reasonable
- Limitations:
  - Only examined two distributions: normal and normal mixture
  - The effect of unequal Variances was not examined