

Statistics 624: Statistical Computing

Homework # 5

Due: Thursday, 20 Sep., 8am

In this assignment, please complete, organize, and turn in **electronically** to `stat624@stat.byu.edu`.

- An easy way to perform power calculations is simulation. Suppose you are testing a drug that reduces blood pressure. You have 50 people in a treatment group and 50 people in a placebo group. Furthermore, you expect the systolic blood pressure (SBP) to have a mean of 150mmHg and a standard deviation of 15mmHg in the placebo group and mean of 140mmHg and a standard deviation of 15mmHg in the treatment group.
 1. Assuming both SBP populations are Normally distributed, simulate one set of data and perform a t -test using the `t.test` function.
 2. Using the `names()` function, find the appropriate inferential summary measure of the significance of the test (p -value).
 3. Write a loop to generate data and perform 1000 t -tests, storing the p -values in a vector `a`. What is the **power** of the study? Compare the results you get with the result from `power.t.test`.
 4. Suppose the treatment group has a smaller standard deviation of 10mmHg . The function `power.t.test` cannot handle this problem ($\sigma_{\text{Placebo}} \neq \sigma_{\text{Treat}}$). However, you can rewrite your simulation to report the power for this alternative.
 5. Suppose instead of a 10mmHg difference between treatment and control, we wish to find the difference that gives 80% power. This can be accomplished in one of three ways:
 - (a) Run several simulations until you arrive at the appropriate power.
 - (b) Write your own optimization code to arrive at an optimal solution.
 - (c) Use the `uniroot` optimizer.Choose one method and find the difference resulting in 80% power.

- Generate a 500×500 matrix of random number and try to find the sums of each column, timing each method below:
 1. Two nested loops, one over columns, one over rows within a column.
 2. One loop, using `sum()` for each column.
 3. `apply()`
 4. matrix multiplication by a vector of 500 ones (`j`).

Record the time required for each computation.

- Suppose you want to estimate the distribution of linear regression estimates when the error distribution is Cauchy rather than Gaussian. A dumb method is

```
beta <- NULL
for(i in 1:5000){
  x <- 1:50
  y <- NULL
  for(j in 1:50){
    y <- c(y, rcauchy(1)+x[i])
  }
  model <- lm(y ~ x)
  coefs <- coef(model)
  beta <- c(beta, coefs[2])
}
```

1. Replace `lm()` with the appropriate form of `lsfit()`.
2. Replace `lsfit()` with the appropriate form of `solve(cbind(1,x),y)`.
3. Allocate all of the space outside of the loop.