Lab/HW 10: Monte Carlo

Your lab/homework must be submitted in with two files: (1) R Markdown format file; (2) a pdf or html file, unless otherwise stated. Other formats will not be accepted. Your responses must be supported by both textual explanations and the code you generate to produce your result.

Part I - Monte Carlo Power Study

Estimate the power of the Student's t-test for difference in means for two normally distributed populations.

The Student's t-test is a standard test for difference in means between two normal populations. In this exercise, we will estimate the power of t-test using Monte Carlo estimators.

Assume two normal populations with a difference in means of `diff_mean`. The first sample with `n1 = 30` observations, `mean = mean1`, `sd = sd1` and the second sample with `n2 = 20` observations, `sd = sd2` and `mean = mean1 + diff_mean`.

a. [7 pt] Write a function that returns a data frame of 50(`= n1 + n2`) rows, where the first column is named `smpl` and contains the normal draws from both populations, and a second column named `pop` indicates which population an observation came from (taking values of 1 or 2). The inputs for this function should be `n1`, `mean1`, `sd1`, `n2`, `sd2` and `true_diff`.

Generate and store one such data frame based on the following parameters:

```
n1 <- 30; mean1 <- 0; sd1 <- 1
n2 <- 20; sd2 <- 1.5
true_diff <- 0.5
```

b. [7 pt] The hypotheses we want to test by this simulation study is:

$$\begin{cases} H_0 : mean_1 - mean_2 = 0 \\ H_a : mean_1 - mean_2 \neq 0 \end{cases}$$

at level α (let us fix `alpha = 0.05` for the rest of the exercise)

We can use a standard t-test for that:

```
alpha <- 0.05
t_test_result <-
  t.test(
    x = df$smpl[df$pop == 1],
    y = df$smpl[df$pop == 2],
    conf.level = 1 - alpha
    )
```

The decision rule is to reject if:

```
t_test_result$p.value < alpha
```

Write a function that performs this t-test for an input data frame that is defined in the same way as ours (a `smpl` column for sample, a `pop` column for population (taking values of 1 or 2). A second input should `alpha`, be the level of the test.

The function's output should be a logical value - whether or not the null is rejected.

Does the t-test reject the null for your data frame?

c. [7 pt] Write a function that estimates the power of the *t*-test for these parameters and this difference in means. That is, the function should take as inputs `S`, the number of tests to perform (which in this case is the number of Monte Carlo samples), and the parameters `n1`, `mean1`, `sd1`, `n2`, `sd2`, `true_diff` and `alpha`.

The function will generate `S` different data frames simulated from the same set of parameters (`S = 500` is a reasonable size, but you may increase it for better accuracy). It will test each of those using the *t*-test and measure the rejection rate for the test. This rate is an estimate of the power for this set of parameters. The function will return a scalar bewteen 0 and 1 which is thee power estimate.

Estimate the power with our above parameters and `S = 2000` (or more, if you are patient enough and want better accuracy).

More power signifies that we have a better test that is better able to identify a difference between the normal means when it is actually there. However, it is also important to verify that our test does not reject the null more than it should (type-I error should be less than or equal to `alpha`). If that happens, the rejections in other cases cannot be considered reliable.

Check that type-I error is properly controlled with the same `mean1`, `sd1` and `sd2`.

d. [49 pt] **Power study**: we will now extend the above to a full scale "power study". That is, we check the power of the test for different levels of true difference in means (take a the following sequence: `seq(-2, 2, 0.2)` to be the domain of true difference values). For each level of true difference in means record the estimated power and conclude by plotting the estimated power vs. the true difference in means. A-priori, we expect several things to happen:

1. as the absolute value of the true difference in means grows, a good test will have more power (rejection rate closer to 1).
2. since the *t*-test and the normal distributions are symmetric, we expect the power curve to be symmetric around 0.
3. Where the difference in means is 0, the power should be below or at `alpha`.

e. [7 pt] How do you expect the parameters `n1`, `mean1`, `sd1`, `n2` and `sd2` to influence the power curve? Demonstrate with one power study per parameter (up to 3 parameters), and add the power curve to the previous plot for comparison.

f. [Extra credit 10 pt] What happens to the power when the normality assumption is violated? Try using gamma distributions with different shape and scale parameters instead of the normal distributions and assess the effect of this misspecification on the power. You should specify `shape1`, `scale1` and `shape2` in an analogous fashion to the normal samples. You should also have a `true_diff` value for difference in means. However, since the mean of the gamma distribution is equal to `shape * scale`, the `scale2` parameter cannot be computed simply by adding `true_diff` to `scale1`. You need to figure out how the difference in means translates to a difference in scales.

Part II - Monte Carlo - Integration

1a. [8 pt] Compute the following integral using Monte Carlo integration (provide estimate and estimated standard error):

$$\int_0^2 \frac{\cos(x(2-x))}{3-x^2+x^{1/3}}dx$$

1b. [7 pt] Compare your estimate to quadrature integration using `integrate()`.

2a. [8 pt] Compute the following integral using Monte Carlo integration (provide estimate and estimated standard error):

$$\int_0^\infty \frac{x^4 \cdot e^{-x/4}}{1+x+\sqrt{x}}dx$$

2b. [7 pt] Compare your estimate to quadrature integration using `integrate()`.