

## Discretionary Note

Anish Krishna Lakkapragada

**IF YOU USE THIS FILE TO CHEAT, YOU ARE NOT ONLY STUPID BUT YOU ARE CHEATING YOURSELF OUT OF THE ABILITY TO FALL IN LOVE WITH MATH.** Furthermore, I am not smarter than you and my solutions did not always get a perfect score.

**CONTENT STARTS ON NEXT PAGE.**

To access the general instructions for this repository head [here](#).

## S&DS 242/542: Homework 2

Due Wednesday, January 29, at 1PM

1. **Binomial MGF.** Let  $X \sim \text{Binomial}(n, p)$ . Derive a simple expression for the moment generating function of  $X$  in terms of  $n$  and  $p$ .

2. **Construction of the bivariate normal.** Let  $Z_1$  and  $Z_2$  be two independent  $\mathcal{N}(0, 1)$  variables, and consider the bivariate normal vector  $(X_1, X_2)$  given by

$$X_1 = c_1 Z_1 + d_1 Z_2 + e_1$$

$$X_2 = c_2 Z_1 + d_2 Z_2 + e_2$$

where  $c_1, c_2, d_1, d_2, e_1, e_2 \in \mathbb{R}$  are fixed constants.

Given any values  $\mu_1, \mu_2 \in \mathbb{R}$ ,  $\sigma_1^2, \sigma_2^2 > 0$ , and  $\rho \in [-1, 1]$ , show how one may choose  $c_1, c_2, d_1, d_2, e_1, e_2$  so that  $X_1$  and  $X_2$  have means  $\mu_1$  and  $\mu_2$ , variances  $\sigma_1^2$  and  $\sigma_2^2$ , and correlation  $\rho$ .

3. **Monte Carlo integration (based on Rice 5.21 and 5.22).** For a given function  $f : [a, b] \rightarrow \mathbb{R}$ , suppose we wish to numerically approximate

$$I(f) = \int_a^b f(x) dx.$$

A Monte Carlo approach to doing this is the following: Let  $g$  be the PDF of any continuous random variable taking values in  $[a, b]$ , where  $g(x) > 0$  for all  $x \in [a, b]$ . Generate independent random draws  $X_1, \dots, X_n$  from  $g$ . Then approximate  $I(f)$  by

$$\hat{I}_n(f) = \frac{1}{n} \sum_{i=1}^n \frac{f(X_i)}{g(X_i)}.$$

(a) Show that  $\mathbb{E}[\hat{I}_n(f)] = I(f)$ . Assuming that  $\text{Var}[f(X_i)/g(X_i)] < \infty$ , explain why  $\hat{I}_n(f) \rightarrow I(f)$  in probability as  $n \rightarrow \infty$ .

(b) Derive a formula for  $\text{Var}[\hat{I}_n(f)]$ . Show that for some  $c_n \in \mathbb{R}$ ,  $c_n(\hat{I}_n(f) - I(f)) \rightarrow \mathcal{N}(0, 1)$  in distribution as  $n \rightarrow \infty$ .

(c) Consider concretely the task of approximating

$$I(f) = \int_0^1 \cos(2\pi x) dx.$$

If  $g$  is the uniform distribution on  $[0, 1]$ , provide an explicit form for the above value  $c_n$ .

4. **Simulating a sample median.** Let  $X_1, \dots, X_{99} \stackrel{IID}{\sim} \mathcal{N}(0, 1)$ . The sample median  $X_{\text{median}}$  is the 50th largest value among  $X_1, \dots, X_{99}$ .

Compute  $X_{\text{median}}$  for 5000 simulations of  $X_1, \dots, X_{99}$ . What is the mean of  $X_{\text{median}}$  across your simulations? What is the standard deviation? Plot a histogram of the 5000 values of  $X_{\text{median}}$  — what does the sampling distribution of  $X_{\text{median}}$  look like?

Derive (analytically) the standard deviation of the sample mean  $\bar{X} = \frac{X_1 + \dots + X_{99}}{99}$ , and compare this with your simulated standard deviation of  $X_{\text{median}}$ . According to your simulation, is  $X_{\text{median}}$  more or less variable than  $\bar{X}$ ? Include both your code and your histogram in your homework submission.

If you are new to programming, the following will walk you through how to do this in R:

- Install R from <https://www.r-project.org/>. Launch R and select “New Document” from the “File” drop-down menu.

- We will write our code in this document. First, let’s create a numeric vector of length 5000 that will save the results from the 5000 simulations. Call it `X.median`:

```
X.median = numeric(5000)
```

- To repeat a series of commands 5000 times, we’ll use a `for` loop:

```
for (i in 1:5000) {  
  ...  
}
```

We can fill in any commands in the “...” section above, and these will be executed once for each value of `i` from 1 to 5000.

- Let’s fill in the ... section. We can simulate 99 independent samples from  $\mathcal{N}(0, 1)$  using the `rnorm` function in R, and save it to a temporary vector variable `X`:

```
X = rnorm(99, mean=0, sd=1)
```

(The `mean` and `sd` arguments indicate the mean and standard deviation of the normal distribution.) We can then use the `median` function in R to compute the sample median of the values in `X`. We will save this as `X.median[i]`, referring to element `i` of the numeric vector we created above:

```
X.median[i] = median(X)
```

(Hence, in the first loop iteration the sample median is saved as `X.median[1]`, in the second iteration it is saved as `X.median[2]`, etc.) Let's put the above two lines of code inside the `for` loop.

- After the `for` loop, we can now write some commands that compute and print the mean and standard deviation of the values in `X.median`, and plot a histogram of these values:

```
print(mean(X.median))  
print(sd(X.median))  
hist(X.median)
```

- Let's save our file as `sample_median.R`. Then go back to the main R Console, and select "Source File..." under the "File" drop-down menu. Select our file `sample_median.R`, and voila! You should see the the mean and standard deviation of the 5000 sample medians printed in the R Console, and a separate plot window displaying the histogram.

We'll use more built-in functions in R as we go. To see what a function does and how to use it, type `?` followed by the function name in the R console to pull up the help page. For example, entering

```
?median
```

into the R console pulls up a page about the `median` function used above.