Data Science and Social Inquiry: HW1

Yu-Chang Chen

September 6, 2023

The purpose of this exercise is to help you evaluate your readiness for taking this course. Unlike how it was taught before, we now expect students have a basic understanding of statistics and some experience with programming in Python. You should be at least familiar with the statistical concepts mentioned below (the ones in **bold**) to have an enjoyable learning experience. Some more advanced concepts (they are in ***bold and italicized***), which you probably have never seen before, are also introduced in this exercise. These more advanced concepts are likely to be challenging for most of you, and it is fine if you find it confusing or even intimidating at first glance. On the programming side, we expect you to be capable of finishing the coding part using modules such as NumPy, pandas, and matplotlib.

Remark It is totally fine if this exercise takes you hours to solve or if you can not solve a few questions. While this exercise tests your knowledge in basic statistics and programming, it is by no means easy.

Question 1

The **cumulative distribution function** (CDF) describes the probability of the event that a **random variable**, say X, is less than or equal to a certain value:

$$F_X(x) = P(X \leq x).$$

In statistics 101, we usually estimate $F_X(\cdot)$ by first assuming X belongs a certain family of distribution such as the **normal distribution** and estimate $F_X(\cdot)$ by techniques such as the **maximum likelihood estimation (MLE)**. In statistics, we usually refer to such approaches as "***parametric***" since the procedure assumes that the unknown distribution lies in a class of distribution that can be "parametrized" by one or more parameters, like the mean and variance of a normal distribution.

One obvious drawback of parametric approaches is that the actual distribution may not lie in the specified class of distribution, and we may end up getting a biased estimator. For example, if $F_X(\cdot)$ is not a CDF from the class of normal distribution, then our MLE under the assumption that X is a normal random variable will be biased.

Luckily, we can alternatively estimate the CDF $F_X(\cdot)$ using the "***empirical distribution function***", which is the ***empirical analogue*** of the CDF. Formally, let

$$X_1, X_2, ..., X_n \overset{i.i.d.}{\sim} F_X$$

be a random sample of X.[1] Also, for a set E, we define the corresponding ***indicator function*** $\mathbb{1}_E$ as:

$$\mathbb{1}_E(x) = \begin{cases} 1, & \text{if } x \in E, \\ 0, & \text{if } x \notin E. \end{cases}$$

That is, $\mathbb{1}_E(x)$ is a function that takes values only in either 0 or 1, depending on whether $x \in E$.

Lastly, we define the empirical distribution function. The empirical distribution function $\hat{F}_n$ (the subscript n refers to the sample size) is

$$\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{(-\infty,x]}(X_i).$$

That is, we count how many observations in our sample are below x and use its proportion to estimate $F_X(x)$. In below, we will (1) take a deeper look of the definition of $\hat{F}_n(\cdot)$ through examples, (2) study its statistical properties, and (3) conduct experiments to verify that $\hat{F}_n(\cdot)$ is a "good" estimator.

It is easier to understand the definition of $\hat{F}_n(\cdot)$ through examples. For part (a) - (c), assume that $n = 10$ and

$$(X_1, X_2, ..., X_{10}) = (-2.2,\ 3.46,\ 2.8,\ 5.3,\ 4.84,\ -3.2,\ 6.32,\ 7.7,\ 1.5,\ -1.32).$$

(a) (2pts) What is $\hat{F}_n(4)$? How about $\hat{F}_n(-3)$?

(b) (1pt) In the previous part, we find $\hat{F}_n(x)$ at two points, namely $x = 4$ and $x = -3$. We can of course keep going and try other values of x, but this is rather repetitive and boring. Luckily, we have computers, and we know how to write program. Write a program to find out $\hat{F}_n(x)$ for $x = -10,\ -9.99,\ -9.98, ...,\ 9.99,\ 10$ and plot it below.

[1] $X_1, X_2, ..., X_n \overset{i.i.d.}{\sim} F_X$ means that (1) the distribution of X_i is F_X for $x = 1, 2, ..., n$ and (2) $X_1, X_2, ..., X_n$ are **independent**.

(c) (1pt) Use the result from (b) and plot $\hat{F}_n(\cdot)$. How does it look like? Is it non-decreasing?[2]

Now, let's investigate the statistical property of $\hat{F}_n(\cdot)$. For parts (d) - (f), we no longer assume $n = 10$, and we will treat $X_1, X_2, ..., X_n$ as random. For simplicity, we focus on in $F_n(0)$, the probability that X is less than or equal to 0, for the rest of this exercise.

(d) (1pt) What is the expected value of $\hat{F}_n(0)$? Does it depend on n? **Hint**: $\mathbb{1}_{(-\infty,x]}(X_i)$ takes value only in 0 and 1. Which family of random variable only takes value in 0 and 1? What is its expected value? P.S. Your answer can be is related to $F_X(\cdot)$.

(e) (1pt) What is the variance of $\hat{F}_n(0)$? Does it depend on n?

(f) (1pt) What happens when $n \to \infty$? Do you think $\hat{F}_n(0)$ is a good estimator of $F_X(0)$?

An alternative way to study the statistical property of $\hat{F}_x(0)$ is through conducting simulation experiments, which are commonly known as ***Monte Carlo simulations***. A simulation experiment typically contains many rounds. In each round, we will draw a random sample $(X_1, X_2, ..., X_n)$ from a distribution chosen by the researcher and calculate $\hat{F}_n(x)$ given $(X_1, X_2, ..., X_n)$. For example, we can set $n = 100$, generate

$$X_1, X_2, ..., X_n \overset{i.i.d.}{\sim} N(0, 1),$$

in each round of the simulation, and calculate the resulting $\hat{F}_n(0)$.[3]. If we run $B = 10,000$ rounds, we will get $10,000$ realizations of $\hat{F}_n(0)$. We then evaluate the performance of $\hat{F}_n(0)$ by comparing 1000 realizations of $\hat{F}_n(0)$ to its true value $F_X(0)$.

(g) (1pt) What is $F_X(0)$, the true value of the parameter of interest, given that $X \sim N(0, 1)$?

(h) (1pt) Set the seed with `numpy.random.seed(5516)`, use `numpy.random.normal` to generate $X_1, X_2, ..., X_n \overset{i.i.d.}{\sim} N(0, 1)$ for $n = 100$, and calculate $\hat{F}_{n,1}(0)$, where the subscript 1 means that $\hat{F}_{n,1}(0)$ is obtained in the first round of simulation. Repeat $10,000$ times and collect the estimates $\hat{F}_{n,1}(0)$, $\hat{F}_{n,2}(0)$, ..., and $\hat{F}_{n,10000}(0)$. Calculate the **mean squared error** (MSE)

[2] $\hat{F}_n(\cdot)$ is non-decreasing if $\hat{F}_n(x_1) \leq \hat{F}_n(x_2)$ for $x_1 \leq x_2$.

[3] You might wonder why we choose $N(0, 1)$. In fact, a more realistic, informative simulation experiment would try different possible distribution to see the estimator, which is $\hat{F}_n(0)$ in our case, has good statistical accuracy across different distributions

$$\frac{1}{10000}\sum_{b=1}^{10000}\left[\hat{F}_{n,b}(0)-F_X(0)\right]^2,$$

which is the average squared distance between the estimator $\hat{F}_{n,b}(0)$ and its true value $F_X(0)$.

(i) (1pt) Repeat (h) with $n = 200$ and $n = 500$. Is MSE larger or smaller when n is larger?

(j) **(Bonus, 2pts)** the **Central Limit Theorem** (CLT) implies that

$$\sqrt{n}(\hat{F}_n(0)-F_X(0))$$

will converge to the normal distribution. We can verify that CLT holds in our case by plotting the histogram of

$$\sqrt{n}(\hat{F}_{n,b}(0)-F(0)),\ b=1,2,....,10000$$

for $n = 500$. Does your plot support CLT?

Question 2

The objective of this assignment is help you practice the basic data manipulation skills in Python. You will be working with data scraped from Booking.com and then perform data cleaning and create visualizations using the Plotly library.

Before we start, please download and read the 'hotels.csv' from the course website. It contains information about all the hotels in a certain city sourced from booking.com.

(a) (2 pts) First, we need to clean the data. Examine the downloaded data and identify any inconsistencies, missing values, or formatting issues. Specifically, clean the data by addressing the following:

1. Convert the "price" column to numerical format (float) for further analysis.
2. Handle missing values, if any, in the "rating" column.
3. Convert the 'rating' to float, as 'distance' to integer.
4. Remove the [,] and , from 'comment'.
5. Transform all the distances to meter.
6. Do you spot anything else?

(b) (1pt) Install the plotly library if you haven't already.

1. Use the `plotly.express` module to create a scatter plot. The x-axis represents the price, and the y-axis represents the distance from the center. The color of each point indicates the rating. Ensure that when hovering over a point with the mouse, it displays the corresponding hotel name, rating, and price information.
2. Is there outlier in the data? If so, remove the outlier and plot again.
3. What do you conclude from graph?

Submission Guidelines:

Save the cleaned data as a CSV file. Additionally, the executed program should be submitted as a .py or .ipynb file.

If you have any questions, feel free to raise them in class or write to the course TA.

Question 3

(2 pts) Propose two research questions you are interested in pursuing for the final project. Ensure that your questions align with my expectations, as outlined in the syllabus. Also, please do not propose ideas for which data collection is obviously impossible.