

# ECON 5166: Programming Homework

Yu-Chang Chen

Due: November 24, 2023

Welcome to this hands-on project designed to introduce you to the core programming skills essential for data analysts. In this project, you will be tasked with scraping data from booking.com and conducting various operations. Let's dive into the details. This assignment serves as both a roadmap and a primer. It offers hands-on exposure to key tools like terminal usage, web crawling, Git version control, data application development, and optionally, Docker for containerization.

To encourage thorough engagement with this homework, be aware that the upcoming midterm exam will feature bonus questions directly related to this assignment. These questions are designed to be straightforward—if you've completed the homework yourself, you should find them easy to answer. They serve as a check on your own effort in completing this homework.

**Remark 1:** Direct instructions for these topics won't be provided. However, a set of useful learning resources is available on NTU COOL for your reference. Should you have questions, feel free to consult with your TA, 蔡尚恩.

**Remark 2:** You are allotted two months to complete this project. Please note that partial credit is not an option; the project must be fully completed for credit consideration.

## Instructions

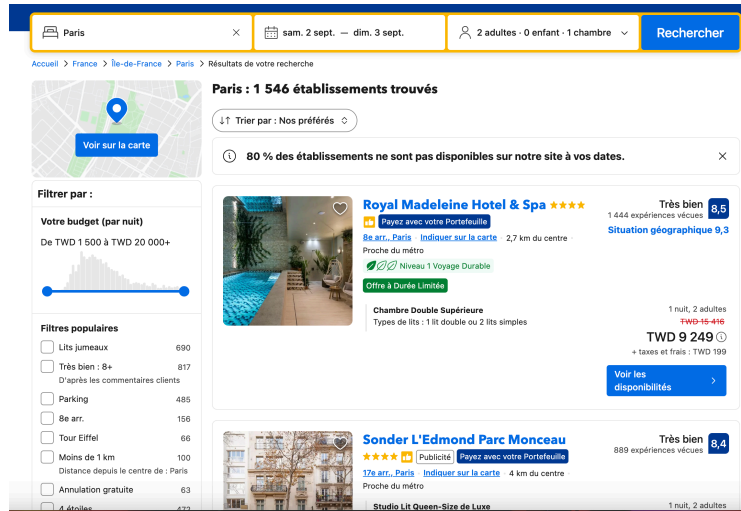
### Step 0: Initialization

Initialize a new folder and a Git repository within it. Name the folder using your student ID.

### Step 1: Web Crawling

In a Jupyter notebook named `web_crawler.ipynb`, write a web crawler to fetch data from booking.com. Implement a function that takes “location”, “check-in date”,

and “check-out date” as inputs and returns a DataFrame containing hotel details like name, location, price, rating, distance, and comments. Commit this notebook to your Git repository with a clear commit message.



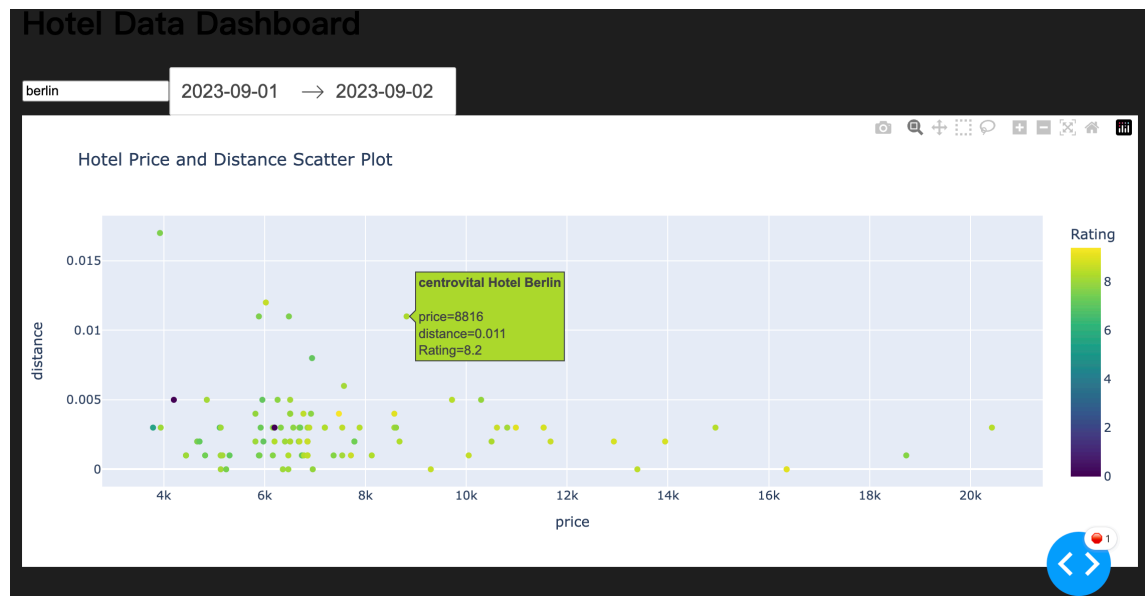
## Step 2: Data Cleaning

After scraping, ensure data types are correctly formatted: “price” as integer, “rating” as float, “comment” as string, and “distance” in kilometers (as a float).

	name	location	price	rating	distance	comment
0	你好咖啡旅館	大安區, 台北	TWD 2,370	[8.5]	距中心 2.8 公里	[非常好, ,]
1	Artinn藝築文旅-台北站前館	中正區, 台北	TWD 1,444	[8.0]	距中心 0.5 公里	[非常好, ,]
2	台北凱達大飯店	萬華區, 台北	TWD 2,800	[8.3]	距中心 2.3 公里	[非常好, ,]
3	明日大飯店	萬華區, 台北	TWD 1,000	[7.3]	距中心 1.2 公里	[好, ,]
4	見潭環旅	士林區, 台北	TWD 2,735	[8.6]	距中心 4.8 公里	[很棒, ,]
5	水滴米窩	中正區, 台北	TWD 1,860	[7.9]	距中心 200 公尺	[好, ,]
6	RF富裕自由商旅 - 忠孝館	大安區, 台北	TWD 1,649	[7.8]	距中心 3.7 公里	[好, ,]
7	和苑三井花園飯店台北忠孝	大安區, 台北	TWD 5,587	[9.0]	距中心 1.7 公里	[好極了, ,]
8	丹迪旅店大安森林公園店	大安區, 台北	TWD 2,760	[8.9]	距中心 2.2 公里	[很棒, ,]
9	城市商旅- 南西館	大同區, 台北	TWD 2,722	[8.3]	距中心 1 公里	[非常好, ,]
10	JR東日本大飯店 台北	中山區, 台北	TWD 4,785	[9.0]	距中心 2.5 公里	[好極了, ,]
11	板橋凱撒大飯店	板橋, 台北	TWD 3,700	[8.8]	距中心 6.7 公里	[很棒, ,]
12	Simple+ 飯店	松山區, 台北	TWD 2,500	[8.5]	距中心 2.8 公里	[非常好, ,]
13	優美飯店-雙連MRT	中山區, 台北	TWD 2,055	[8.1]	距中心 1.4 公里	[非常好, ,]
14	西門日記-六福館	萬華區, 台北	TWD 1,512	[7.0]	距中心 1.2 公里	[好, ,]
15	路徒行旅-中華館	中正區, 台北	TWD 2,299	[8.7]	距中心 0.7 公里	[很棒, ,]
16	苓旅-站前 Lin Inn TMS	中正區, 台北	TWD 1,180	[8.1]	距中心 350 公尺	[非常好, ,]
17	Sunrise Hotel	信義區, 台北	TWD 1,480	[7.8]	距中心 5 公里	[好, ,]
18	沃客汽車旅館	北投區, 台北	TWD 1,584	[7.4]	距中心 7.8 公里	[好, ,]
19	雀客旅館台北站前	中正區, 台北	TWD 2,482	[8.2]	距中心 400 公尺	[非常好, ,]
20	洛基大飯店忠孝館	大安區, 台北	TWD 2,604	[8.5]	距中心 3.5 公里	[非常好, ,]
21	台北北門世民酒店	西門町, 台北	TWD 2,800	[8.7]	距中心 0.7 公里	[很棒, ,]
22	卡夫人背包客棧 台北車站青年館	中正區, 台北	TWD 789	[7.4]	距中心 350 公尺	[好, ,]

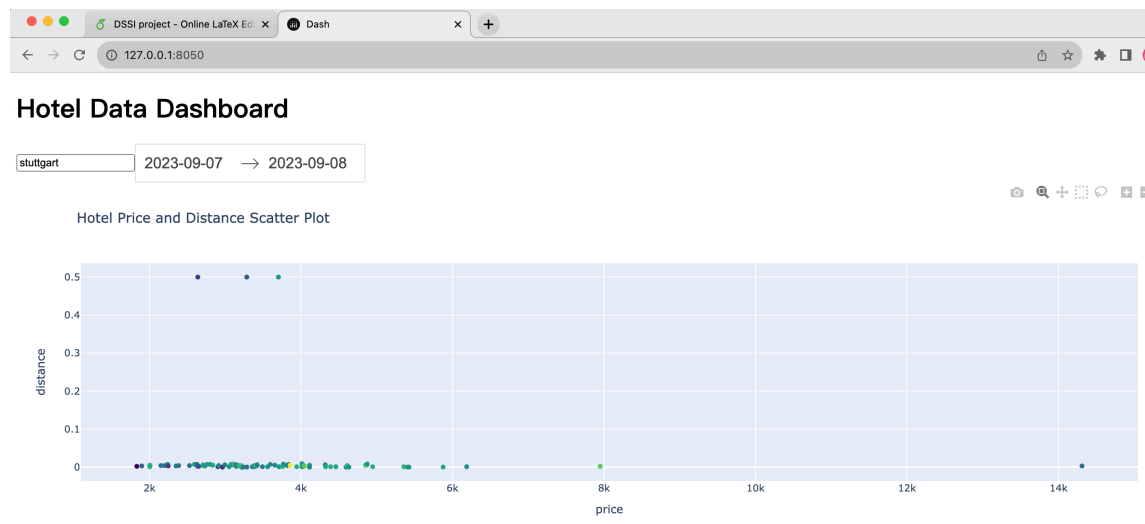
### Step 3: Data Visualization

Use Plotly to visualize the data in `web_crawler.ipynb`. The scatter plot should have the price on the x-axis and distance from the center on the y-axis, color-coded by ratings. Commit the updated notebook to Git (with message).



#### Step 4: Dash Application

Create a Dash web application that integrates everything. In `app.py`, employ Plotly Dash to build an interactive dashboard. Add input fields for location and dates to trigger the web crawler dynamically.



#### Step 5: Containerization (Optional)

Managing different environments can be a hassle. Docker simplifies this by con-

tainerizing your application. Construct a Dockerfile that uses a base image compatible with your Python version and incorporates all required packages at the same versions you're using. Make sure to expose the ports to access the web application.

**Note:** This step is optional and will not affect your grade nor appear in the midterm's bonus questions.

## Submission Guideline

Create a screencast that demonstrates the functionality of your web application. Specifically, input a location and dates to display the corresponding graph in a web browser. In the same video, also showcase your Git log to provide an overview of your project's development history.