MAPÚA UNIVERSITY
SCHOOL OF ELECTRICAL, ELECTRONICS, AND COMPUTER ENGINEERING

Experiment 7:
Web-scrapping and Data Visualization

CPE106L (Software Design Laboratory)

PALOY, EURO GABRIEL
TAMSON, JOMARI ANGELO
ZALAMEDA, ANDREA KAYLA

Group No.: **10**
Section: **E01**

Int

PreLab

Readings, Insights, and Reflection

METIS Book: Lambert, K. A. (2023). Fundamentals Of Python: First Programs (3rd ed.)

(TAMSON)

This book provided me with a strong foundation in Python programming by explaining essential concepts like variables, loops, functions, and data structures. The step-by-step approach made it easy to follow along, and the practical examples helped me see how Python can be used in real-world applications. I also appreciated how the book emphasized problem-solving techniques, which made coding feel more logical and structured.

(PALOY)

Reading this book gave me a clear and structured understanding of Python's fundamental concepts, making it easier to develop my coding skills. The way the book explained programming logic, functions, loops, and data types helped me see how these elements work together to create a functional program. I liked how the examples and exercises encouraged hands-on practice, reinforcing my understanding of key programming techniques. The book also highlighted best practices in coding, which is useful for writing clean, efficient, and error-free programs. This resource has been essential in improving my confidence in Python.

(ZALAMEDA)

This book helped me build a solid foundation in Python programming by explaining key concepts in a clear and organized manner. The lessons covered everything from basic syntax to more advanced topics like object-oriented programming, giving me a well-rounded understanding of how Python works. I appreciated how the book combined theoretical explanations with practical exercises, making it easier to apply what I learned in real coding situations. It also emphasized problem-solving strategies, helping me approach programming challenges more effectively. Overall, this book has been a great resource for developing my Python skills.

Matplotlib 3.8.0 Documentation

(TAMSON)

Learning from the Matplotlib documentation helped me understand how to create meaningful visual representations of data using Python. I explored different types of graphs, such as line plots, bar charts, and scatter plots, and learned how to customize them by adding labels, colors, and styles. This knowledge is especially useful for analyzing trends, presenting research findings, or making complex data more understandable. The documentation provided detailed explanations and examples that made it easier to grasp the concepts, and I now feel more comfortable incorporating visualizations into my Python projects.

(PALOY)

The Matplotlib documentation provided me with a deeper understanding of how to visualize data in Python, which is an important skill for data analysis and presentation. I learned how to create different types of graphs and charts, adjust their appearance, and

add essential elements like titles, legends, and labels. Understanding these techniques allows me to present data more effectively, making complex information easier to interpret. The detailed explanations and examples in the documentation made it easier to experiment with different plotting styles and customization options, helping me gain practical experience with data visualization.

(ZALAMEDA)
Studying the Matplotlib documentation allowed me to explore different ways to visualize data using Python, which is an essential skill for analyzing and presenting information. I learned how to generate various types of plots, customize their appearance, and make them more informative by adding labels, legends, and color schemes. These visualization techniques are particularly useful in fields like research, business, and data science, where clear and effective data representation is crucial. The documentation was detailed yet easy to follow, helping me gain a better understanding of how to create meaningful graphs and charts.

Beautiful Soup: Build a Web Scraper With Python – Real Python
(TAMSON)
This tutorial introduced me to the concept of web scraping, which is a powerful way to extract data from websites. I learned how to use the Beautiful Soup library to navigate and manipulate HTML content, making it possible to collect specific pieces of information from different web pages. The step-by-step instructions helped me understand how to set up a web scraper, handle different website structures, and organize the extracted data for further use. This skill is particularly useful for tasks such as data collection, research, and automation, making Python even more versatile and practical.

(PALOY)
This tutorial gave me a strong introduction to web scraping using the Beautiful Soup library in Python. I learned how to extract and organize information from web pages, which can be useful for tasks such as market research, data collection, and automation. The tutorial walked me through the process of parsing HTML, selecting specific elements, and handling different website structures. I found it fascinating how Python can be used to gather and analyze web data efficiently, making it a valuable tool for projects that require real-time or large-scale data collection.

(ZALAMEDA)
This tutorial introduced me to web scraping, a technique for extracting data from websites using Python. I learned how to use Beautiful Soup to parse HTML, locate specific elements, and collect useful information from different web pages. This skill is valuable for various applications, such as gathering research data, monitoring prices, or automating repetitive online tasks. The tutorial provided clear instructions and examples, making it easier to understand how web scraping works and how to apply it in real-world projects. Learning this skill has expanded my knowledge of Python's capabilities and its practical uses.

InLab

- **Objectives**
  - Learn WebScraping Techniques
  - Visualize data using Matplotlib
- **Tools Used**
  - Visual Studio Code
- **Procedure**

  The InLab experiment focused on utilizing APIs, web scraping, and data visualization with the help of provided handouts and guides. For most of the part,we were able to get an understanding of APIs first before Web Scraping, but were not able to fully complete the InLab due to lack of provided Python files, but here were notable figures that were accomplished, which we had to install a python module called flask.

Figure 1.1: The source code for json file of chronicling America.json

Int

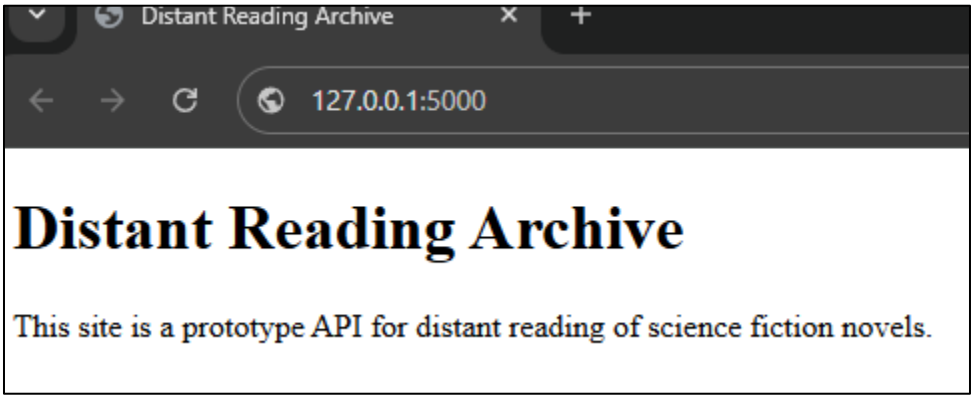
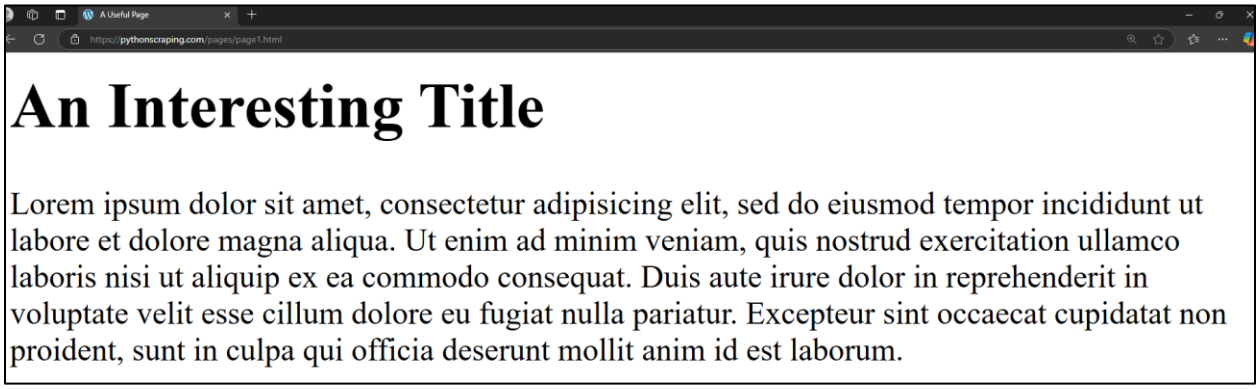


Figure 1.2: The API link

Now for Web Scraping, it was a tedious process due to the instructions, but we were able to highlight information that we learned from following what has to be done.



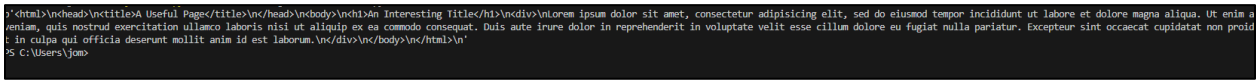


Figure 1.3 and 1.4: The html link what was outputted using scrapetest.py

```python
from urllib.request import urlopen

html = urlopen("http://pythonscraping.com/pages/page1.html")

print(html.read())
```

Figure 1.5: scrapetest.py code

During this experiment, we explored APIs, web scraping, and data visualization by following the given instructions. Our first task involved using the Google Shopping API, where we set up a virtual environment, cloned the repository, and configured the necessary settings. However, we faced challenges due to missing files which prevented us from completing some step and despite this, we adapted and continued with other tasks.

For web scraping, we created an environment and installed libraries like bs4 and requests. Initially, we encountered import errors, but after installing the missing dependencies, the program successfully recognized the modules. We then used scrapetest.py to analyze webpage content and retrieve data through Python.

PostLab

1. **PE #5:** Visit the website of the U.S. Bureau of Labor Statistics at https://www.bls.gov/data/home.htm and download the data for the average price of bread, as shown earlier in this chapter (there will be data for more recent years added since these words were written). You can also use the breadprice.csv file here >> Chapter 11 Data files. Write a program in a file named **breadprice.py** that loads the data set and cleans it as you did earlier in this chapter. Then include code to display a line plot of the average price for each year in the table.

Procedures

The students began by loading the data into a data frame and examining its initial contents and to ensure there were no duplicate rows, we executed a specific code snippet to remove them. Next, we calculated the average price for each entry and added the corresponding values to the dataset. After visualizing the data through a plot, they saved the cleaned dataset for future use, which is figure 1.1

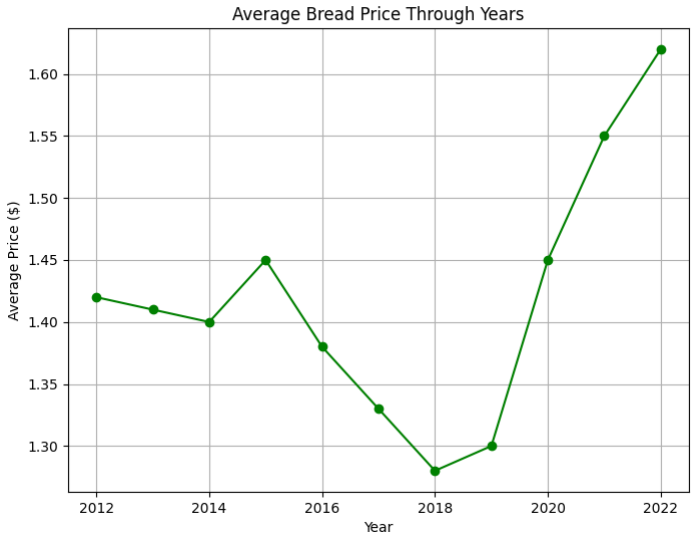


Figure 1.6. Matplotlib chart of Average Price of Bread Over the Years

2. **PE #6:** The columns labeled FG, 3PT, and FT of the data set in the Analyzing Basketball Statistics case study (Download the files here >> CaseStudy2) do not show a single integer value but instead show values with the format <makes-attempts>, which is not suitable for the kind of data analysis performed on the other columns. For example, analysts might like to view the mean of free throws attempted as well as mean of the free throw percentage. You can correct this problem with a cleaning step that, for each such column:
    a. removes it from the data frame
    b. creates two new columns from this series, where the first column includes the numbers of makes and the second column includes the number of attempts
    c. inserts the new pairs of columns into the data frame at the appropriate positions, with the appropriate column headings (for example, FTM and FTA)

Define a function named **cleanStats** in the file hoopstatsapp.py. This function expects a data frame as an argument and returns the frame cleaned according to the steps listed previously. You should call this function after the frame is loaded from the CSV file and before it is passed to the **HoopStatsView** constructor.

Procedure

Here, the process begins with creating a function that splits the data in the FG, 3PT, and FT columns. This function is then executed before the data frame is passed to the application view, to be sure it is properly cleaned beforehand. Finally, within the application view, additional buttons were added to facilitate the analysis of FGM, 3PTM, and FTM.

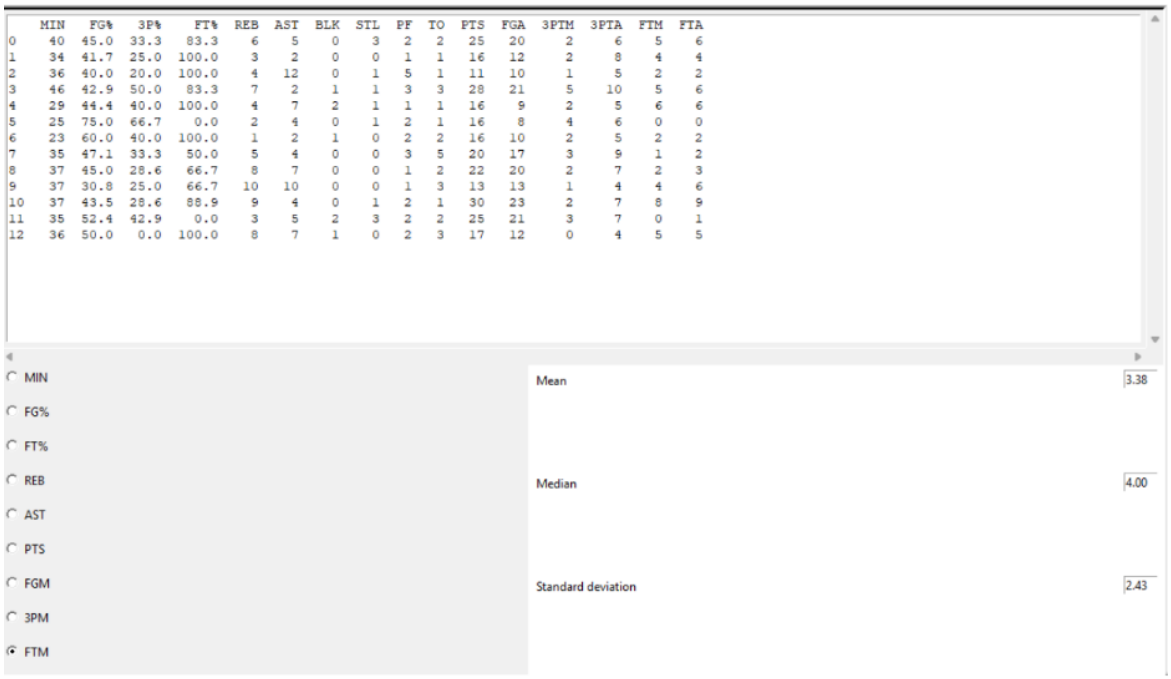


Figure 1.7. The Results of the FGM, 3PTM, and FTM data.