Project 4: Visualization with Matplotlib

John Wesley Mathis

Dr. Anthony Choi

June 14, 2024

ECE/SSE 591, Summer 2024

Table of Contents:

[Deliverable Table 5](#__RefHeading___Toc2569_331549091)

[1. Introduction 6](#__RefHeading___Toc2573_331549091)

[2. Adapting SIR Model for Visualization 7](#__RefHeading___Toc2577_331549091)

[3. Adapting Movie Analysis Model for Visualization 7](#__RefHeading___Toc2969_2801450533)

[4. Conclusion 7](#__RefHeading___Toc2609_331549091)

[References 8](#__RefHeading___Toc3692_331549091)

Table of Figures

[Figure 1: Using Pandas objects and importing CSV files 7](#Figure!0|sequence)

[Figure 2: Collecting information about the data 7](#Figure!1|sequence)

[Figure 3: Collecting information about the dataset 8](#Figure!2|sequence)

[Figure 4: Changing the index of the dataset 9](#Figure!3|sequence)

[Figure 5: Adding a Rank column 10](#Figure!4|sequence)

[Figure 6: Moving 'Rank' column to the beginning of the dataset 11](#Figure!5|sequence)

[Figure 7: Sorting the data by IMDB rating 12](#Figure!6|sequence)

[Figure 8: Finding what data is missing 13](#Figure!7|sequence)

[Figure 9: Removing unused columns from the dataset 14](#Figure!8|sequence)

[Figure 10: Dealing with missing data 15](#Figure!9|sequence)

[Figure 11: Checking for duplicated data 16](#Figure!10|sequence)

[Figure 12: Converting 'Gross' column from String to Integer 16](#Figure!11|sequence)

[Figure 13: Cleaning up the 'Runtime' column 17](#Figure!12|sequence)

[Figure 14: Converting release year to datatime format 18](#Figure!13|sequence)

[Figure 15: Cleaning up the genre column 18](#Figure!14|sequence)

[Figure 16: Determining top movie genres 19](#Figure!15|sequence)

[Figure 17: Determining top 5 most voted movies 19](#Figure!16|sequence)

[Figure 18: Determining bottom 5 least voted movies 20](#Figure!17|sequence)

[Figure 19: Determining which movie has the longest runtime 20](#Figure!18|sequence)

[Figure 20: Determining which movie has the shortest runtime 21](#Figure!33|sequence)

[Figure 21: Using `query()` to filter data 21](#Figure!19|sequence)

[Figure 22: Using `eval()` to calculate new data 22](#Figure!20|sequence)

[Figure 23: Determining which year had the most top movies 22](#Figure!21|sequence)

[Figure 24: Determining highest grossing movie 23](#Figure!22|sequence)

[Figure 25: Determining which genre has the IMDB rating and which has the highest gross 24](#Figure!23|sequence)

[Figure 26: Discovering which directors consistently produce top movies using pivot tables 25](#Figure!24|sequence)

[Figure 27: Demonstrating hierarchical indexing 26](#Figure!25|sequence)

[Figure 28: Importing two new datasets and showing information about dataset 1 27](#Figure!26|sequence)

[Figure 29: Dataset 2 information 27](#Figure!27|sequence)

[Figure 30: Merging the two datasets on 'id' 28](#Figure!28|sequence)

[Figure 31: Removing unused columns 29](#Figure!29|sequence)

[Figure 32: Setting the index to 'genres' and 'release\_date' 30](#Figure!30|sequence)

[Figure 33: Code for vectorized string operations 30](#Figure!31|sequence)

[Figure 34: Output to Figure 32 code 31](#Figure!32|sequence)

[Figure 35: Demonstrating `concat()` 32](#Figure!34|sequence)

[Figure 36: Using `concat()` to append new data 33](#Figure!35|sequence)

# Deliverable Table

The purpose of this table is to provide a complete view of the concepts covered in chapter 4 of *"Python Data Science Handbook"* (VanderPlas, 2016) and provide a general page location for where the topic was demonstrated.

|  |  |
| --- | --- |
| Deliverables | Location |
| Simple Line Plots |  |
| Simple Scatter Plots |  |
| Visualizing Errors |  |
| Density and Contour Plots |  |
| Histograms, Binnings, and Density |  |
| Customizing Plot Legends |  |
| Customizing Colorbars |  |
| Multiple Subplots |  |
| Text and Annotation |  |
| Customizing Ticks |  |
| Customizing Matplotlib: Configurations and Stylesheets |  |
| Three-Dimensional Plotting in Matplotlib |  |
| Geographic Data with Basemap |  |
| Visualization with Seaborn |  |

Additionally, here is a link to my GitHub were the datasets and the Jupyter Notebook for the project can be downloaded: https://github.com/jwmathis/SSE591\_Project3.git. In order to run the file, Python and Pandas package must be installed.

# 1. Introduction

Python has a rich repository of libraries that aid scientists and researchers in data analysis and manipulation. One of the most common libraries in use is Pandas, which is built on top of NumPy and provides a higher-level, and more flexible interface for data handling. While NumPy excels at efficient numerical computations with arrays, Pandas introduces data structures like Series and DataFrame that offer a more intuitive means to work with structured data.

Because of Pandas’ Series and DataFrame objects, data scientists have an indispensable tool to handle, clean and manipulate data in tabular form. These objects support a wide range of operations, from simple data aggregation and filtering to complex time-series analysis. The library’s ability to handle missing data, merge datasets, and perform group-by operations adds significant value to Python’s data manipulation kit.

This report aims to demonstrate my proficiency in Python data manipulation techniques as covered in Chapter 3 of the “Python Data Science Handbook” by Jake VanderPlas (2016). This report attempts to illustrate the core concepts and functionalities of the Pandas library by implementing the concepts into a single project. The code presented in this report was developed using Visual Studio Code with Jupyter Notebook extensions. I will provide detailed explanations, highlighting key features and operations that make Pandas an essential tool for data analysis.

# 2. Adapting SIR Model for Visualization

# 3. Adapting Movie Analysis Model for Visualization

# 4. Conclusion

This report documents my journey in learning Pandas, a powerful data manipulation library in Python. Key concepts I explored include data structures essential for handling and analyzing structured data. I learned to perform various data operations, including data indexing, merging datasets, grouping data and more.

By using real data, I was able to explore how to go about cleaning the data up properly before beginning to analyze it. Many errors I encountered were related to missing values and data types, which are significantly different from syntax errors I encountered in previous projects. However, through practice and persistence, I was able to clean the data up and obtain datasets that could be analyzed properly using Pandas.

# References

1. Harshit Shankhdhar. (2021). IMDB Dataset of Top 1000 Movies and TV Shows. Retrieved from https://www.kaggle.com/datasets/harshitshankhdhar/IMDB-dataset-of-top-1000-movies-and-tv-shows
2. The Movie Database (TMDB). (2018). TMDB 5000 Movie Dataset. Retrieved from https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata
3. VanderPlas, J. (*2016*).  *Python Data Science Handbook*. O’Reilly Media. Retrieved from https://jakevdp.github.io/PythonDataScienceHandbook/index.html