Project 3: Data Manipulation with Pandas

John Wesley Mathis

Dr. Anthony Choi

June ##, 2024

ECE/SSE 591, Summer 2024

Table of Contents:

[Deliverable Table 4](#__RefHeading___Toc2569_331549091)

[1. Introduction 5](#__RefHeading___Toc2573_331549091)

[2. Top Movie Data Analysis 6](#__RefHeading___Toc2577_331549091)

[3. Conclusion 7](#__RefHeading___Toc2609_331549091)

[References 8](#__RefHeading___Toc3692_331549091)

Table of Figures:

# **Deliverable Table**

The purpose of this table is to provide a complete view of the concepts covered in chapter 2 of *"Python Data Science Handbook"* (VanderPlas, 2016) and provide a general page location and/or project name for where the topic was demonstrated. This is not an exhaustive list. Some areas may be covered by multiple projects. This is meant only as a means to show specific areas where the topic is demonstrated and to ensure that every area was covered.

|  |  |
| --- | --- |
| Deliverables | Location |
| Introducing Pandas Objects |  |
| Data Indexing and Selection |  |
| Operating on Data in Pandas |  |
| Handling Missing Data |  |
| Hierarchical Indexing |  |
| Combining Datasets: Concat and Append |  |
| Combining Datasets: Merge and Join |  |
| Aggregation and Grouping |  |
| Pivot Tables |  |
| Vectorized String Operations |  |
| Working with Time Series |  |
| High-Performance Pandas: eval() and query() |  |

# 1. Introduction

Python is a versatile language with numerous libraries. Because of its easy to understand syntax, it is very popular among the data science community as a useful tool to load, store, and manipulate data. Much of this data is typically managed by converting it into arrays of numbers. However, as useful and easy as it is to use Python programming, managing data efficiently has its many drawbacks. Fortunately, there are specialized tools that have been created to improve Python’s ability to handle such numerical data. One such package that has been created is the Numerical Python package or NumPy as it’s generally known.

NumPy provides a manner to more efficiently handle data arrays. NumPy’s arrays are similar to Python’s built-in arrays but where it shines is how is handles storage as the data grows larger. As a result, NumPy has become the foundation for many scientific tools that are in use.

This report aims to demonstrate my proficiency in Python fundamentals as well as NumPy fundamentals that were covered in chapter 2 of the “*Python Data Science Handbook”* written by Jake VanderPlas (2016). Each exercise and mini-project is used to illustrate the concepts outlined in the deliverable table from the previous section. The code presented throughout this report was written using Visual Studio Code with Jupyter Notebook extensions. In the following sections an analysis is done on the two mini-projects. The remaining sections cover various exercises, and explanations are detailed to explore the structure and functionality of coding using NumPy packages.

# 2. Top Movie Data Analysis

# **3. Conclusion**

This report documents my journey in mastering NumPy which involved grasping fundamental concepts such as arrays, computations using universal functions and broadcasting, incorporating comparisons and boolean logic into arrays, indexing and sorting arrays, and learning how to create structured arrays.

Through the process of coding, many errors were encountered. Due to the simplicity of the projects and exercises, all of the errors were syntax errors. My experience is mainly in C and C++ so I am used to the nuance of adding semicolons to the end of each line. Many times in Python, I found myself accidentally adding semicolons to the ends of blocks and statements, confused why my code block continually showed errors. Other examples of syntax errors I encountered include forget a quotation, forgetting a colon when defining a function or loop, or simply misspelling a variable that I named.

Engaging in mini-projects and working through the exercises has proved to be a valuable source in aiding me to take the theory of Python fundamentals and the NumPy package and put the concepts into practice and think about practical data science applications. This has allowed me to grasp the concepts that make the NumPy package a valuable tool for efficiently managing large sets of data.

# References

1. Lowe, S., Mathis, J., & Wall, N. (2019). *Humans vs. Zombies Lab.* Unpublished paper, Mercer University.
2. VanderPlas, J. (*2016*).  *Python Data Science Handbook*. O’Reilly Media. Retrieved from https://jakevdp.github.io/PythonDataScienceHandbook/index.html