**Python Assignment**

**Experience**

Although the experience was quite demanding and chaotic, it also provided me with a fantastic opportunity to become familiar with a wide variety of Python modules. In contrast to the second assignment, the first one was rather simple because I could do the duties whenever I saw fit. The second portion of the homework wasn't particularly challenging, but it was more challenging because I had to use several libraries and comprehend how each method should work. I frequently visited stackoverflow and other websites to view the original implementation and documentation.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Programming Assignment#1**

The first programming task focuses on the student's ability to calculate and analyze data from CSV files using programming. The first section has five functional criteria, each of which carries out a certain duty. I've used the code reusability design pattern to combine various functions into one.

**Functional Requirement#1**

**Thought Process:**

By reading a single column at first, I was able to get a rudimentary understanding of the CSV library and gradually worked my way up. I wrote a function that accepts two parameters, file\_path and column\_index. It then uses the csv.reader() method to perform the reading and returns file\_path and column\_index.

**Strengths:**

improved handling of errors

**Weakness:**

Requires the provision of a correct column index

**Enhancements:**

Using a particular column name in place of the index

**Alternative approach**

A different strategy would be to use different Python libraries to retrieve larger datasets more quickly.

**Functional Requirement#2**

**Thought Process:**

Extended from FR1. using the extract\_single\_column() function from FR1 and putting the information in a dictionary, respectively.

**Strength:**

FR1-based structure with well-organized storage

**Weakness:**

Presumes the uniqueness of a column name

**Improvements:**

Include a uniqueness check

**Alternative Approach:**

All of the data may be loaded using pandas without having to be stored in a dictionary.

**Functional Requirement#3**

**Thought Process:**

Studied and applied the Kendal Tau Formula step-by-step (concordant & discordant pairs).

**Strength:**

Direct formula implementation

**Weakness:**

Assumes lists are of equal length and adds complexity for large datasets

**Improvements:**

Add a check for lists length check

**Alternative Approach:**

SciPy can be used for pre-built correlation coefficient functions

**Functional Requirement#4**

**Thought Process:**

Loading data in dictionary using FR2 and integrated with FR3 to calculate Kendal Tau coefficients.

**Strength:**

Reusing Functional Requirement 2 and Functional Requirement 3.

**Weakness:**

Assumes column name uniqueness

**Improvements:**

Add a check column name uniqueness

**Alternative Approach:**

Pandas and SciPy can be used for pre-built correlation coefficient functions.

**Functional Requirement#5**

**Thought Process:**

Developed a unique logic to display the data in a in a tabular format. Function display\_custom\_table takes three arguments (list\_of\_pairs, border\_char='\*', columns=None). border\_char are optional and columns arguments, if provide would only display the data that is required with specified number of columns.

**Strength:**

Flexibility with customization (change border char and columns)

**Weakness:**

Assumes input list format

**Improvements:**

More customization can be added

**Alternative Approach:**

Import additional libraries for better customization

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Programming Assignment#2**

Individuals learning skills can be analyzed in the second assignment as it includes calculation and analysis of data with the help of multiple python libraries such as SciPy, Pandas, SciPy.stats along with performing data visualization using Matplotlib and Seaborn. Similar to the first part, second assignment also contains 5 FR’s with each requirement using a separate library for a particular task. Screenshots from data visualization have also been added.

­­

**Functional Requirement#6**

**Thought Process:**

Use of Python CSV library, creating a merge function to merge both files (task2a.csv and task2b.csv).

**Strength:**

Pandas efficiency and easy integration.

**Weakness:**

Assumes common columns for merging.

**Improvements:**

Add a column check mechanism

**Alternative Approach:**

Use of NumPy for larger dataset to use hardware acceleration.

**Functional Requirement#7**

**Thought Process:**

Utilized matplotlib and seaborn libraries for data visualization and iterative exploration for difference types of graphical plots (line plot, heatmap and boxplot). The output in the form of screenshots have been added down below.

**Strength:**

Visual insight for checking of outliers

**Weakness:**

Subjective identification for “interesting” pattern recognition.

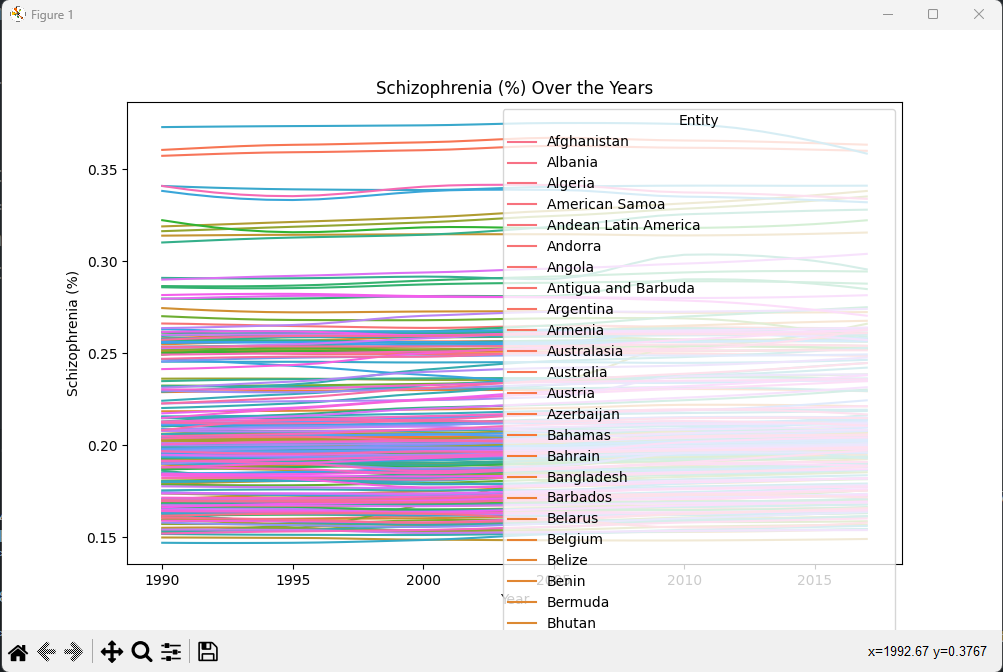
**Improvements:**

Combining visual exploration with statistical measurement as well.

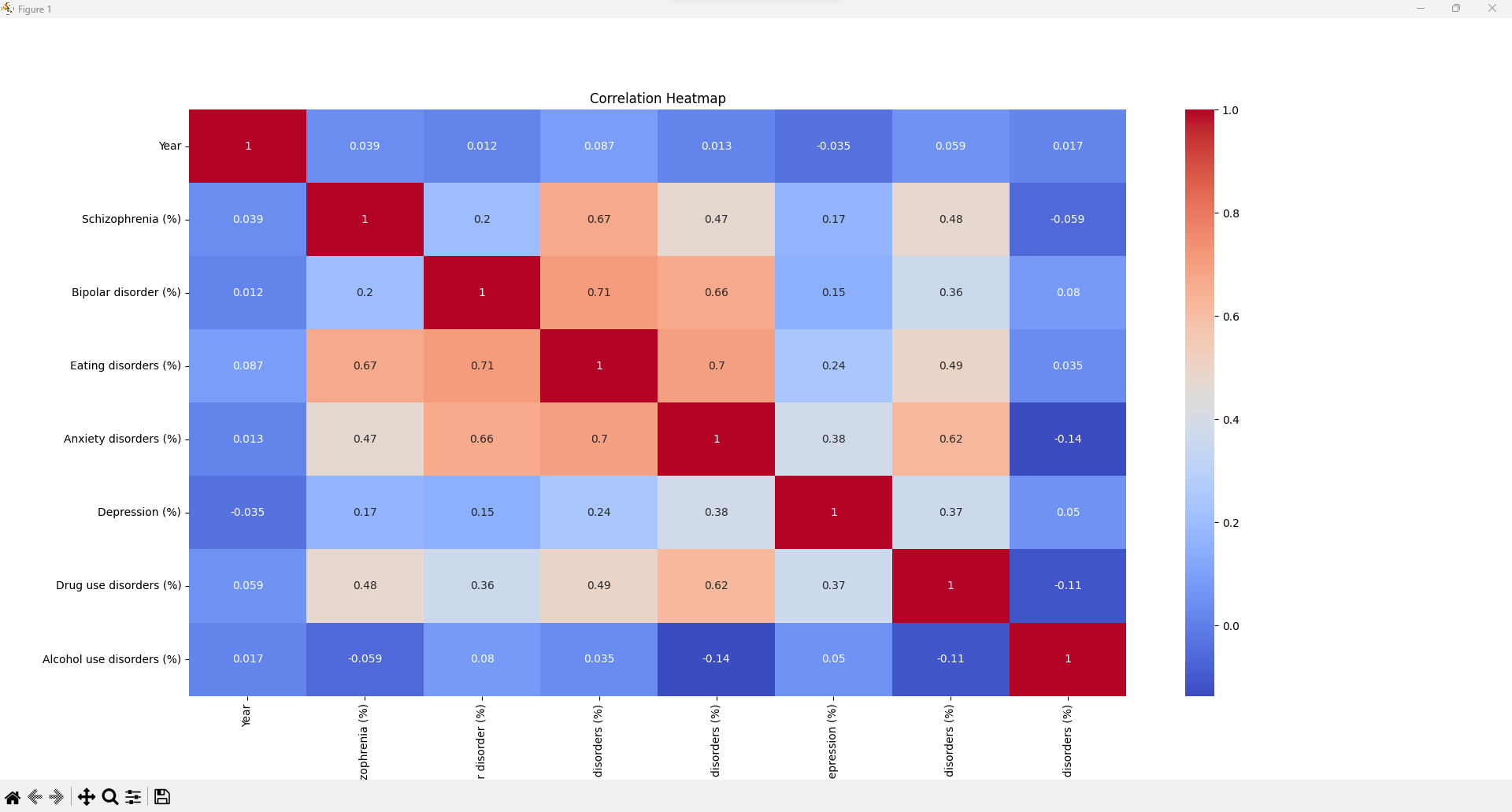
**Alternative Approach**

Use of simple machine learning algorithms (such as regression) for automated and better pattern recognition

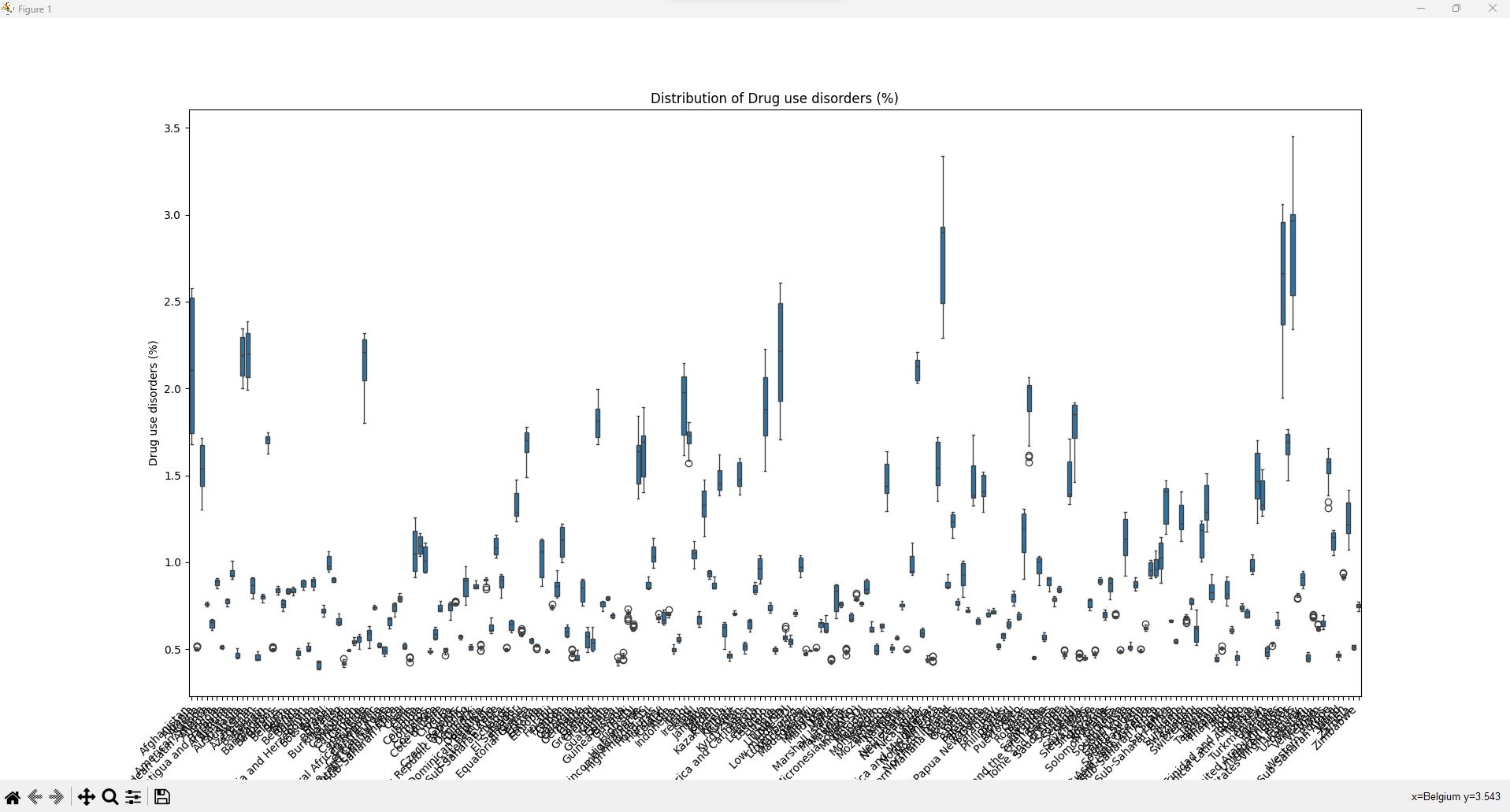
**Line Plot**

****

**Heatmap**

****

**Boxplot**

****

**Functional Requirement#8**

**Thought Process:**

Utilizing SciPy for statistical analysis of data and detection of outliers using z-score calculation.

**Strength:**

Z-Score Robustness is a widely used technique for outliers’ detection used with the help of SciPy library

**Weakness:**

Z-score may not be effective for certain types of distributions.

**Improvements:**

Some other techniques could also be used for outliers’ detection and visualization tools such as matplotlib.

**Alternative Approach:**

Training Machine learning models for outliers’ detection.

**Functional Requirement#9**

**Thought Process:**

Developed a clear and simple hypothesis that can certainly be tested in the next method.

**Strength:**

Specific and testable hypothesis.

**Weakness:**

Formulation would add subjective hypothesis.

**Improvements:**

Incorporation of statistical measures in hypothesis formulation for better, measurable and accurate results.

**Alternative Approach:**

Using difference tools for automatic hypothesis generation.

**Functional Requirement#10**

**Thought Process:**

In order to check the significance level for hypothesis testing we would utilize statistical testing setting significance level to 0.05.

**Strength:**

Statistical Analysis for better and accurate results.

**Weakness:**

Sensitivity with the significance level. Results may change upon different significance level.

**Improvements:**

Adding multiple significance levels and multiple hypothesis tests.

**Alternative Approach:**

Difference machine learning approaches can be used along with difference Bayesian methods as well.

­