

S&P 500 ANALYSER

Table of Contents

1. Introduction
2. Features
3. System Requirements
4. Setup and Installation
5. How to Use
6. Program Structure
7. Future Enhancements
8. Credits

Introduction

S&P 500 ANALYSER is an interactive Python program that allows users to make graphs to compare different variables relative to the Sector of the enterprises belonging to the S&P500 index. It has functionalities such as viewing the definitions of each variable, selecting the number of variables and the variables themselves, selecting the type of graph and editing it. Also, users are able to select different webpages to invest.

Features

1. **User Authentication:**
 - Login system.
2. **Opening webpages:**
 - Displaying several links to invest.
3. **Variables definition:**
 - Viewing the definition of all the variables the user can use.
4. **Number of variables:**
 - The user can select the number of variables that he wants to use.
5. **Variable selector:**
 - The user can select the desired variables, taking into account the chosen number of variables.
6. **Graph selector:**

- Selecting the type of graph taking into account the chosen number of variables.

7. Graph editor:

- The user can edit the title of the graph, as well as the x-label and y-label.

8. Quitting:

- Exit the current session.

System requirements

1. Install Required Packages:

- matplotlib.pyplot
- pandas
- graphics.py
- numpy
- webbrowser
- wand

2. Install Python 3.10 or 3.12:

- Our program was developed using these two versions of Python, ensuring full compatibility with both.

Setup and Installation

Initial note: This program has been developed using Windows.

1. Download the project repository.

2. Install Python 3.10 or 3.12 if not already installed

3. Install the ImageMagick software

- The .dll file to install it is available in the project repository

4. Install required libraries

How to Use

1. **Run the Program:**
 - Start the program by running the Python script `project_final`.
 2. The program is like a customer journey. This means that the user only needs to follow the instructions.
-

Program Structure

Key Functions

1. **main()**
 - Main function of the program. It is used to call other functions inside it.
2. **clean()**
 - Used to clean the data set and handling missing values.
3. **introduction()**
 - Shows the purpose of the program.
4. **invest_funciton()**
 - Shows several links that the user can follow in order to invest.
5. **var_definition()**
 - Shows to the user the definitions of the variables.
6. **num_of_var()**
 - The user can choose the number of extra variables they want to choose.
 - Note that Sector is a predefined variable.
7. **var_selector()**
 - Based on the chosen number of extra variables, the user can select the variables.
8. **graph_selector()**
 - Based on the chosen number of extra variables, the user can select the type of graph.
9. **graph_editor()**

- Once the graph is created and displayed, the user can edit parameters such as the title, the x-label or the y-label.

10. graph_editor_pie()

- For the pie chart, however, the user can only change the title.

11. lineplot()

- Creates a line plot.

12. boxplot()

- Creates a box plot.

13. histogram()

- Creates a histogram.

14. piechart()

- Creates a pie chart.

15. violinplot()

- Creates a violin plot.

16. errorplot()

- Creates an error plot.

17. barplot()

- Creates a bar plot.

18. scatterplot()

- Creates a scatterplot.

Future improvements

In the future we would like to:

1. Implement a real register and log in system.
 2. Update the data set daily.
-

Credits

Giorgia Merli

Lev Pronozin

Nicolás Lemli Jiménez

Yago Moreno

Yara El Nakadi