SUPERPY Usage Guide

AUTHOR: Marcin Pietrzak

CONTACT: marcinpiet@gmail.com

VERSION: 2.01

Description:

Superpy is a command-line interface tool designed for supermarkets to meticulously manage and monitor their inventory. This comprehensive management system allows for efficient control over available resources and warehouse stock levels. It provides rapid and accurate updates of product assortments and delivers detailed information regarding product availability. It's an integral solution for businesses seeking precision and effectiveness in inventory tracking and management.

Table of Contents:

1. Introduction
2. Report - highlights three technical elements of implementation.
3. Installation
4. File list
5. Basic Usage
   Buying Products
   Selling Products
   Generating Reports
   Exporting Data
   Advancing Time
6. Advanced Features
   Generating Revenue Reports
   Generating Profit Reports
   Exporting Revenue Data
   Exporting Profit Data
7. Conclusion

1. Introduction

Superpy is a versatile command-line tool designed to help supermarkets manage their inventory efficiently. It offers various features for buying, selling, reporting, and exporting data related to products. This usage guide aims to provide step-by-step instructions and examples for Superpy's usage.

2. Report

- highlights three technical elements of implementation.

Superpy:

Advanced Supermarket Inventory Management Tool

A. Argument Parsing using argparse:

Superpy: employs the `argparse` module for sophisticated command-line argument interpretation. This enhances the tool's command-line interface by allowing users to define various commands and options seamlessly. It supports diverse subcommands such as 'buy,' 'sell,' 'report,' 'export,' 'advance-time,' 'report-revenue,' 'export-revenue,' 'report-profit,' and 'export-profit.' Each of these subcommands comes equipped with a distinct set of mandatory and optional arguments, streamlining user data entry and retrieval.

Advantage: The strategic use of argparse ensures uniformity in command and argument structure, thereby elevating the user experience by offering consistent and user-friendly interactions.

B. Data Management via CSV Files:

Superpy: orchestrates data related to product transactions through CSV files, specifically `bought.csv` and `sold.csv`. The tool reads from and writes to these files, ensuring inventory is continually updated based on transactional activities. Moreover, it is capable of generating detailed reports and exporting data in multiple formats, including but not limited to CSV and XLSX.

Advantage: Utilizing CSV files for data storage offers a streamlined method to preserve inventory records. This ensures effortless data modification and guarantees data continuity across multiple tool operations.

C. Mastery over Date and Time Management:

Date and time management is intrinsic to **Superpy's** functionality. The tool is adept at advancing its internal date by a predetermined number of days, thereby facilitating retroactive data entry and thorough analysis. Additionally, it can generate revenue and profit reports within designated date intervals, producing comprehensive tables and graphical representations for enhanced data visualization.

Advantage: This meticulous date and time management capability empowers users to engage with historical data effectively, monitor sales and profit trajectories over defined durations, and extract valuable insights into overall supermarket operations.

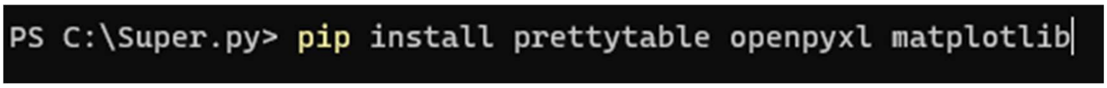
Rationale: The integration of these technical components was meticulously curated to cater to the pivotal needs of supermarket inventory management. The incorporation of argparse ensures a refined user-tool interaction, guaranteeing consistent input. The choice of CSV for data management accentuates simplicity in data storage and accessibility. Concurrently, the date and time management features facilitate in-depth historical data analysis. Cumulatively, this design approach accentuates user ease and data precision in the realm of supermarket inventory management.

3. Installation

Before you can start using Superpy, make sure you have Python 3.7 or higher installed on your system. To install Python, you have a few options. One option is to download the latest version of Python from the Python website and run the installer. Make sure you check the option to add Python to the PATH when prompted. This will allow you to run Python from any directory in the command prompt. Another option is to install Python from the Microsoft Store. This will automatically add Python to the PATH and keeps it updated. You can search for "Python" in the Microsoft Store app and choose the version you want to install.

Additionally, you need to install the required Python packages by running the following command:

pip install prettytable openpyxl matplotlib



```
PS C:\Super.py> pip install prettytable openpyxl matplotlib
```

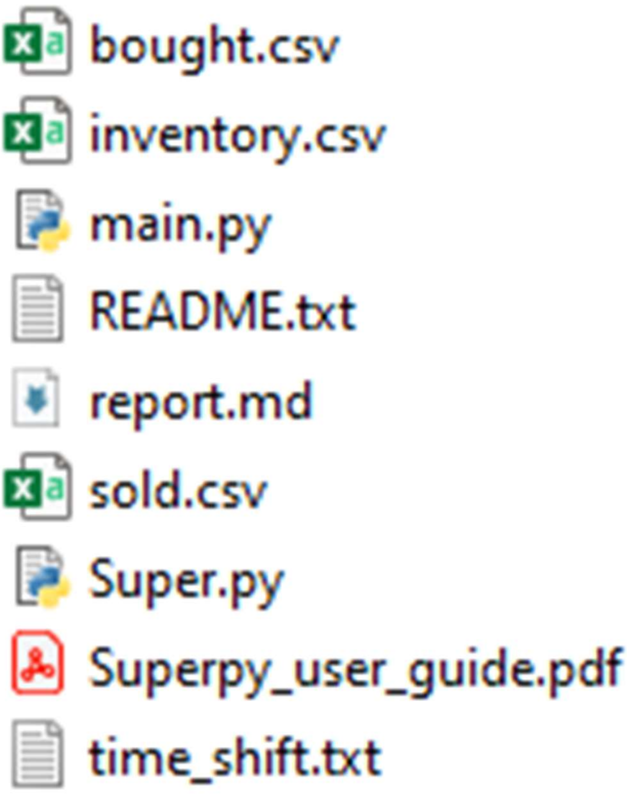
Once you have installed the necessary packages, you are ready to use Superpy.

4. File list

The basic distribution contains in the main folder:

- bought.csv
- inventory.csv
- main.py
- README.txt
- report.md
- sold.csv
- Super.py
- time_shift.txt
- Superpy_user_guide.pdf

Unless the user decides otherwise, Superpy also places files with exported data in the program's main folder during operation.



bought.csv
inventory.csv
main.py
README.txt
report.md
sold.csv
Super.py
Superpy_user_guide.pdf
time_shift.txt

5. Basic Usage

Buying Products

To register the purchase of a product, use the following command:

```
python Super.py buy <product_name> <quantity> <price> [--date <purchase_date>] --expiration_date <expiration_date>
```

<product_name>: The name of the product you are buying.

<quantity>: The quantity of the product purchased.

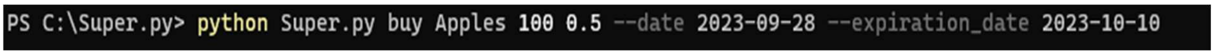
<price>: The price per unit of the product.

[--date <purchase_date>] (Optional): The purchase date in the format YYYY-MM-DD. If not provided, the current date will be used.

--expiration_date <expiration_date>: The expiration date of the product in the format YYYY-MM-DD.

Example:

```
python Super.py buy Apples 100 0.5 --date 2023-09-28 --expiration_date 2023-10-10
```



```
PS C:\Super.py> python Super.py buy Apples 100 0.5 --date 2023-09-28 --expiration_date 2023-10-10
```

Selling Products

To register the sale of a product, use the following command:

```
python Super.py sell <product_name> <quantity> <price> [--date <sale_date>]
```

<product_name>: The name of the product you are selling.

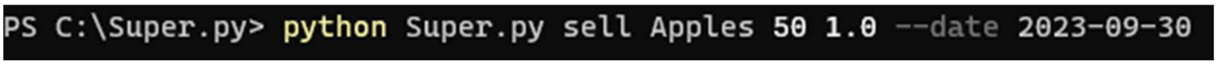
<quantity>: The quantity of the product sold.

<price>: The price per unit of the product.

[--date <sale_date>] (Optional): The sale date in the format YYYY-MM-DD. If not provided, the current date will be used.

Example:

```
python Super.py sell Apples 50 1.0 --date 2023-09-30
```



```
PS C:\Super.py> python Super.py sell Apples 50 1.0 --date 2023-09-30
```

Generating Reports

You can generate various types of reports using Superpy. Supported report types are 'bought', 'sold', and 'inventory'.

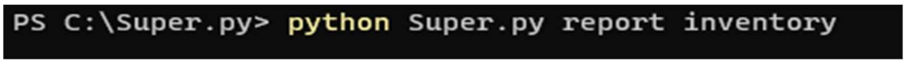
To generate a report, use the following command:

python Super.py report <report_type>

<report_type>: The type of report ('bought', 'sold', or 'inventory').

Example:

python Super.py report inventory



```
PS C:\Super.py> python Super.py report inventory
```

Exporting Data

Superpy allows you to export data to CSV or XLSX format.

To export data, use the following command:

python Super.py export <data_type> [--format <output_format>] [--output <output_file>]

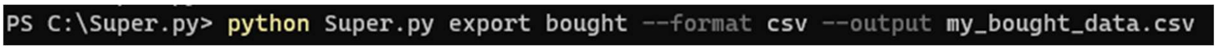
<data_type>: The type of data to export ('bought', 'sold', or 'inventory').

[--format <output_format>] (Optional): The output format ('csv' or 'xlsx'). Default is 'xlsx'.

[--output <output_file>] (Optional): The path to the output file. If not provided, a default filename will be used.

Example:

python Super.py export bought --format csv --output my_bought_data.csv



```
PS C:\Super.py> python Super.py export bought --format csv --output my_bought_data.csv
```

Advancing Time

You can change the recognized "today" date using the advance-time command.

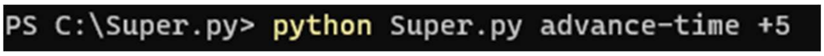
To advance time, use the following command:

python Super.py advance-time <days|now>

<days>: The number of days to advance (use '+' for forward and '-' for backward) or use 'now' to reset to the current date.

Example:

python Super.py advance-time +5



```
PS C:\Super.py> python Super.py advance-time +5
```

6. Advanced Features

Generating Revenue Reports + Line graph

You can generate revenue reports for a specific date range.

To generate a revenue report, use the following command:

python Super.py report-revenue <start_date> <end_date>

<start_date>: The start date of the revenue report in the format YYYY-MM-DD.

<end_date>: The end date of the revenue report in the format YYYY-MM-DD.

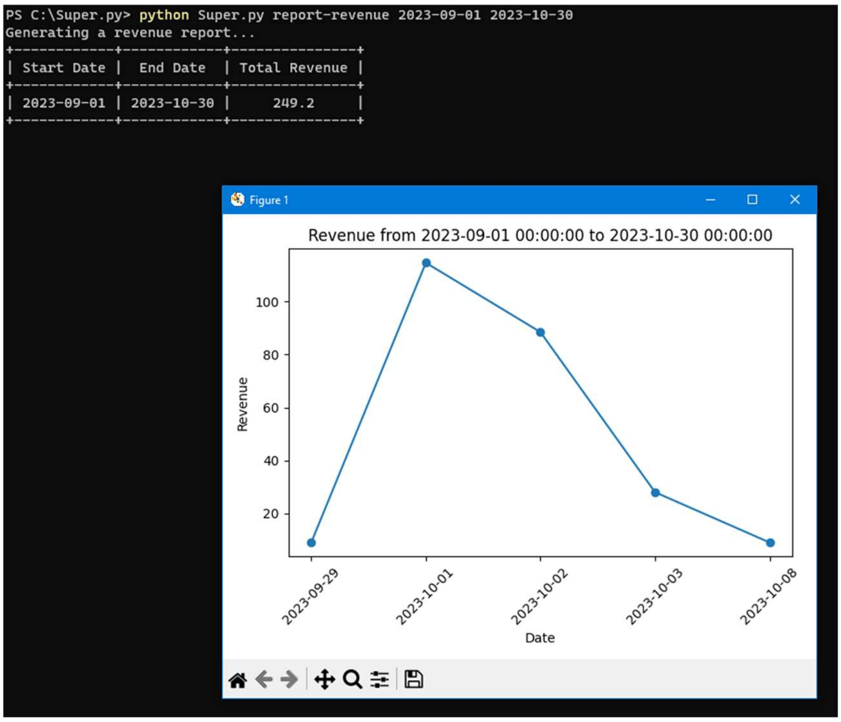
Example:

python Super.py report-revenue 2023-09-01 2023-09-30

```
PS C:\Super.py> python Super.py report-revenue 2023-09-01 2023-09-30
```

Line graph

When you generate a table report on your screen, Superpy automatically generates a bar chart.



```
PS C:\Super.py> python Super.py report-revenue 2023-09-01 2023-10-30
Generating a revenue report...
+------------+------------+---------------+
| Start Date |  End Date  | Total Revenue |
+------------+------------+---------------+
| 2023-09-01 | 2023-10-30 |     249.2     |
+------------+------------+---------------+
```

Generating Profit Reports + A bar graph

You can generate profit reports for different time periods or specific products.

To generate a profit report, use the following command:

python Super.py report-profit <period> [<value>]

<period>: The time period for the profit report ('day', 'month', 'year', 'all', or 'product').

[<value>] (Optional): The specific value for the time period (e.g., '2023-09-30' for a daily report or a product name for a product-specific report).

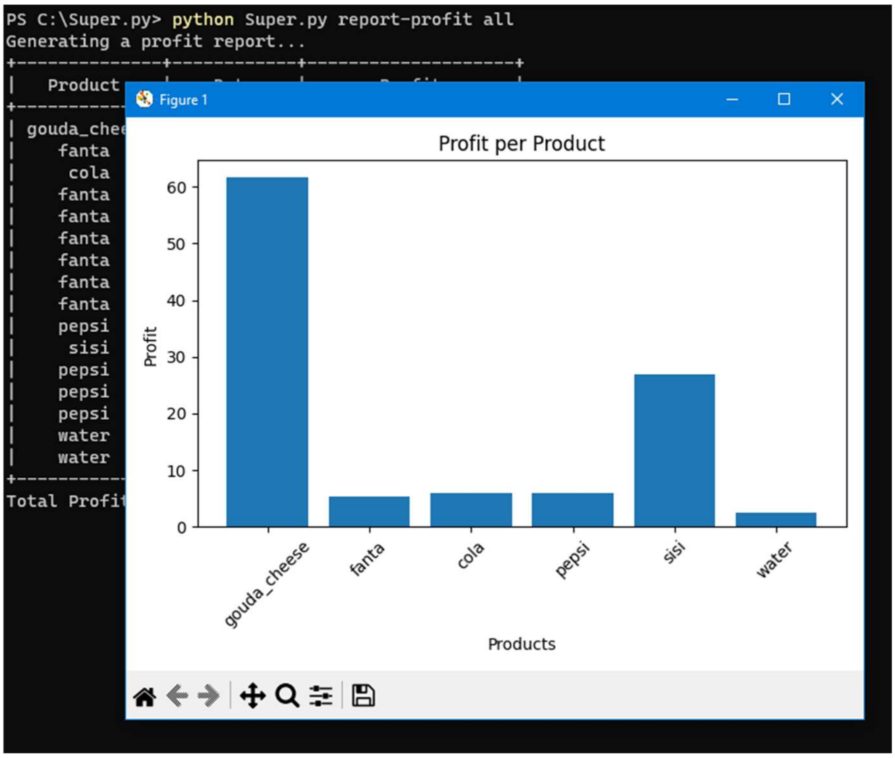
Example:

python Super.py report-profit day 2023-09-30

```
PS C:\Super.py> python Super.py report-profit day 2023-09-30
```

A bar graph

When you generate a table report on your screen, Superpy automatically generates a bar chart.



```
PS C:\Super.py> python Super.py report-profit all
Generating a profit report...
+--------------+------------+--------------------+
|   Product
+-----------
| gouda_chee
|    fanta
|     cola
|    fanta
|    fanta
|    fanta
|    fanta
|    fanta
|    fanta
|    pepsi
|     sisi
|    pepsi
|    pepsi
|    pepsi
|    water
|    water
+-----------
Total Profi
```

Exporting Revenue Data

You can export revenue data to CSV or XLSX format.

To export revenue data, use the following command:

python Super.py export-revenue <start_date> <end_date> [--format <output_format>] [--output <output_file>]

<start_date>: The start date of the revenue report in the format YYYY-MM-DD.

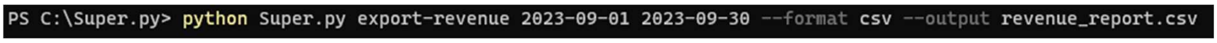
<end_date>: The end date of the revenue report in the format YYYY-MM-DD.

[--format <output_format>] (Optional): The output format ('csv' or 'xlsx'). Default is 'xlsx'.

[--output <output_file>] (Optional): The path to the output file. If not provided, a default filename will be used.

Example:

python Super.py export-revenue 2023-09-01 2023-09-30 --format csv --output revenue_report.csv



```
PS C:\Super.py> python Super.py export-revenue 2023-09-01 2023-09-30 --format csv --output revenue_report.csv
```

Exporting Profit Data

You can export profit data to CSV or XLSX format.

To export profit data, use the following command:

python Super.py export-profit <period> [<value>] [--format <output_format>] [--output <output_file>]

<period>: The time period for the profit report ('day', 'month', 'year', 'all', or 'product').

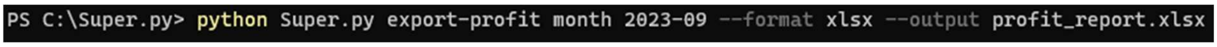
[<value>] (Optional): The specific value for the time period (e.g., '2023-09-30' for a daily report or a product name for a product-specific report).

[--format <output_format>] (Optional): The output format ('csv' or 'xlsx'). Default is 'xlsx'.

[--output <output_file>] (Optional): The path to the output file. If not provided, a default filename will be used.

Example:

python Super.py export-profit month 2023-09 --format xlsx --output profit_report.xlsx



```
PS C:\Super.py> python Super.py export-profit month 2023-09 --format xlsx --output profit_report.xlsx
```

7. Conclusion

Superpy is a powerful tool for supermarket inventory management. With its various features, you can efficiently track products, generate reports, and export data for analysis. This guide should help you get started with Superpy, but don't hesitate to explore more of its capabilities by checking out the available commands and options. For additional assistance, you can contact the author, Marcin Pietrzak, at marcinpiet@gmail.com.