

# "Executive Dashboard" Exercise

## Prerequisites:

- [Business Intelligence and Dataviz](#)
- ["Monthly Sales Reporting" Exercise](#)
- ["Chart Gallery" Exercise](#)

## Learning Objectives

- Create a tool to automate manual efforts and streamline business processes.
- Gain familiarity with CSV formatted data, and how to process it in Python.
- Explore basic data visualization capabilities in Python.
- Leverage built-in Python modules and third-party Python packages to speed development and enhance capabilities.

## Business Prompt

Assume you own and operate a successful small business, selling artisan clothing products through an online platform like Amazon, Etsy, or eBay.

The screenshot shows the Etsy website interface. At the top, there's a search bar with the text "Search for items or shops" and a "Search" button. To the right of the search bar are links for "Sell on Etsy", "Register", and "Sign in". Below these are icons for "Discover" and "Cart". A navigation bar lists various categories: "Jewelry & Accessories", "Clothing & Shoes", "Home & Living", "Wedding & Party", "Toys & Entertainment", "Art & Collectibles", "Craft Supplies", "Vintage", and "Gifts".

The main content area features a product listing for "Libra Hoodie" by the shop "ZodiaTees". The shop has a 5-star rating and 70 reviews. The product is a "Fitted Ultrasoft Woman's Hoodie. Heather Black with Metallic Gold Ink. Buy Any 2 Items Get 1 Free Tshirt". The price is "\$39.95+". There are options to "Ask a question", "Select an option" for size, and a quantity selector set to "1". A prominent "Add to cart" button is visible.

Below the product image, there's an "Overview" section with the following details:

- Handmade item
- Closure: Pullover
- Sleeve length: Long sleeve

Due to certain investment interests in your business, you are obligated to hold monthly board meetings with your investors and advisors to discuss the company's strategic direction, set priorities, and allocate resources. To aid

the board's decision-making processes, they expect you to provide them with a monthly summary report of business insights, including the aggregation of total sales and the identification of top-selling products.

Luckily, at the end of each calendar month, the online platform your company uses to sell its products makes available for download from its admin interface a CSV file representing all individual sales orders for that month.

But usually you only have a few hours from when the data becomes available to when you are required to turn it around and share the resulting insights with the board. Over the past few months, this report compilation process has been somewhat stressful and vulnerable to manual error. And the board's confidence in your operational leadership depends not only on your ability to meet sales goals, but also on your ability to provide them with accurate and timely information.

So your objective is to create a tool which automates the process of transforming monthly sales data into a report of business insights. This report should include charts and graphs to help tell a compelling story.

## Setup

### Repo Setup

Use the GitHub.com online interface to create a new remote project repository called something like "exec-dash". When prompted by the GitHub.com online interface, let's get in the habit of adding a "README.md" file and a Python-flavored ".gitignore" file (and also optionally a "LICENSE") during the repo creation process. After this process is complete, you should be able to view the repo on GitHub.com at an address like [https://github.com/YOUR\\_USERNAME/exec-dash](https://github.com/YOUR_USERNAME/exec-dash).

After creating the remote repo, use GitHub Desktop software or the command-line to download or "clone" it onto your computer. Choose a familiar download location like the Desktop.

After cloning the repo, navigate there from the command-line:

```
cd ~/Desktop/exec-dash
```

Use your text editor or the command-line to create a file in that repo called "dashboard\_generator.py", and then place the following contents inside:

```
# dashboard_generator.py

print("-----")
print("MONTH: March 2018")

print("-----")
print("CRUNCHING THE DATA...")

print("-----")
print("TOTAL MONTHLY SALES: $12,000.71")
```

```
print("-----")
print("TOP SELLING PRODUCTS:")
print("  1) Button-Down Shirt: $6,960.35")
print("  2) Super Soft Hoodie: $1,875.00")
print("  3) etc.")

print("-----")
print("VISUALIZING THE DATA...")
```

Make sure to save Python files like this whenever you're done editing them. After setting up a virtual environment, we will be ready to run this file.

Finally, download one or more of the ["monthly sales data"](#) CSV files into your exercise repository, inside a new sub-directory called "data" (e.g. "exec-dash/data/monthly-sales/sales-201803.csv").

## Environment Setup

Create and activate a new Anaconda virtual environment:

```
conda create -n dashboard-env python=3.7 # (first time only)
conda activate dashboard-env
```

From within the virtual environment, install any packages you may require:

```
pip install pandas # (only if using pandas)

pip install matplotlib # (only if using matplotlib)
pip install plotly # (only if using plotly)
pip install altair # (only if using altair)

pip install pytest # (only if writing automated tests)
```

From within the virtual environment, demonstrate your ability to run the Python script from the command-line:

```
python dashboard_generator.py
```

If you see the provided output, you're ready to move on to project development. This would be a great time to make any desired modifications to your project's "README.md" file (like adding instructions for how to setup and run the app like you've just done), and then make your first commit, with a message like "Setup the repo".

## Basic Requirements

Write a Python application which satisfies the basic requirements detailed below.

## Information Input Requirements

Your program should be able to process any one of these provided [monthly sales CSV files](#), in any order. These CSV files should reside inside a "data" sub-directory of your project repository. Instructors will use the same and/or similar files during the evaluation process. You can assume that each of these CSV files will have a name resembling "sales-YYYYMM.csv" (where "YYYY" represents the four digit year and "MM" represents the zero-padded month). And you can assume each of these CSV files will have the same header row (**date**, **product**, **unit price**, **units sold**, **sales price**).

When the user runs the program, they should be prompted to select one of these CSV files to process.

OPTION A: allow the user to pass their selection as a command-line argument or environment variable during script invocation.

OPTION B: prompt the user to input their selection.

OPTION C: use the `os` module to detect the names of all CSV files which exist in the "data" directory, then display this list to the user and prompt the user to input their selection.

OPTION D: are there any open source Python packages which provide helpful file-selection capabilities?

## Validation Requirements

If the user selects a file that doesn't exist or otherwise specifies the wrong filepath, the program should fail gracefully by: 1) displaying a friendly error message like "Hey, didn't find a file at that location", 2) avoiding runtime errors, and 3) preventing further execution.

## Calculation Requirements

The program's calculations and aggregations should be accurate and comprehensive.

To test the accuracy of your program's calculations, compare its "March 2018" results against the table of expected values below.

Product	Sum of sales price
Button-Down Shirt	\$6,960.35
Super Soft Hoodie	\$1,875.00
Khaki Pants	\$1,602.00
Vintage Logo Tee	\$941.05
Brown Boots	\$250.00
Sticker Pack	\$216.00

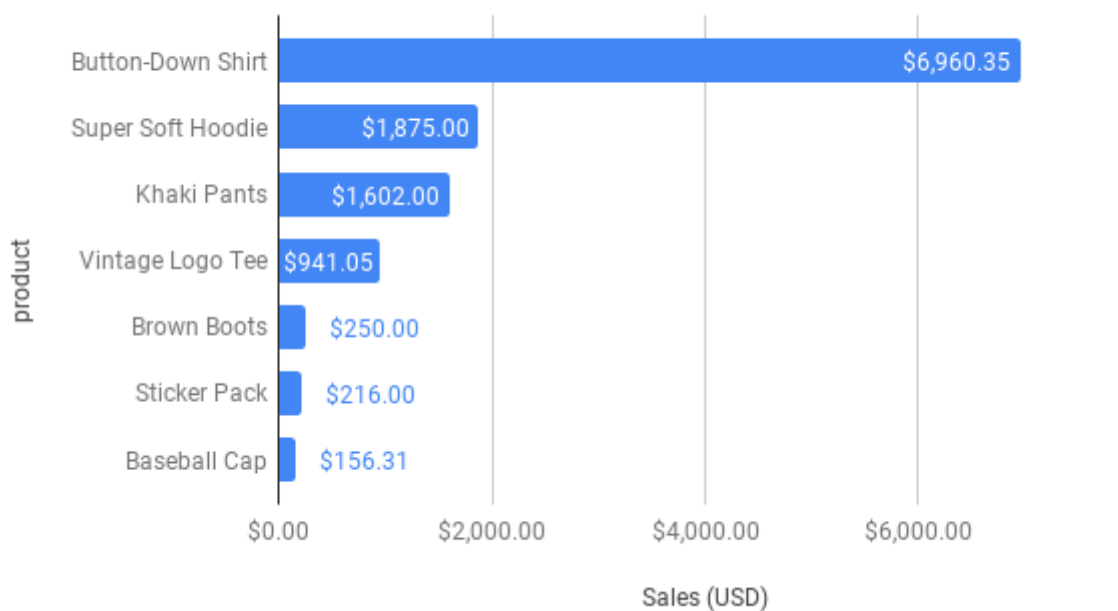
Product	Sum of sales price
Baseball Cap	\$156.31
<b>Total Monthly Sales</b>	<b>\$12,000.71</b>

## Information Output Requirements

During the course of the program's execution, it should display the following information:

1. **Total monthly sales** for the business, equivalent to the sum of total monthly sales for each product, formatted as USD with a dollar sign and two decimal places (e.g. "Total Monthly Sales: \$12,000.71").
2. A **list of the top selling products**, and total monthly sales for each, formatted as USD with a dollar sign and two decimal places (e.g. "Button-Down Shirt: \$6,960.35", "Super Soft Hoodie: \$1,875.00", etc.).
3. At least one **chart or graph** depicting this or related information to support the project objectives. Chart titles should include a human-friendly textual representation of the selected month and year (e.g. "March 2018").

### Top-selling Products (March 2018)



All displays of price-related information should be formatted as USD, with a dollar sign and two decimal places. This includes in charts and graphs.

## Further Exploration Challenges

If you're able to complete the basic project requirements with relative ease, consider addressing one or more of the ["Exec Dash" Further Exploration Challenges](#) (e.g. validating CSV files, comparing sales across months, predicting future sales).