

The "Monthly Sales Reporting" Exercise

Prerequisites:

- [Datastores, and Processing CSV Data](#)
- The [csv](#) and [itertools](#) Modules, or [The pandas Package](#)

Learning Objectives

- Create a Python application to automate a business process.
- Practice using Python to process CSV files.
- Practice researching and leveraging the capabilities provided by Python modules and third-party packages.

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". Further right are icons for "Discover" and "Cart". Below the search bar is a horizontal navigation menu with categories: "Jewelry & Accessories", "Clothing & Shoes", "Home & Living", "Wedding & Party", "Toys & Entertainment", "Art & Collectibles", "Craft Supplies", "Vintage", and "Gifts". The main content area features a shop profile for "ZodiaTees" with a 5-star rating and 70 reviews. Below the shop name is a "Favorite shop" button. To the right of the shop name are four small thumbnail images of the hoodie and a box indicating "139 items". The main product image shows a woman wearing a dark grey hoodie with a gold Libra zodiac symbol. To the right of the image is the product title: "Libra Hoodie. Fitted Ultrasoft Woman's Hoodie. Heather Black with Metallic Gold Ink. Buy Any 2 Items Get 1 Free Tshirt". Below the title is the price "\$39.95+" and an "Ask a question" button. There are dropdown menus for "Size" (with the text "Select an option") and "Quantity" (set to 1). Below these is a black "Add to cart" button. At the bottom, there's an "Overview" section with bullet points: "Handmade item", "Closure: Pullover", and "Sleeve length: Long sleeve".

At the end of each calendar month, the online platform makes available for download from its admin interface a CSV file representing all individual sales orders for that month. See the ["monthly sales data"](#) examples (row per day per product sold that day).

To aid your ability to make data-driven decisions, you decide to create a Python program which will automate the process of gleaning business insights from the monthly sales data.

Setup

Repo Setup

Use the GitHub.com online interface to create a new remote project repository called something like "monthly-sales". 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/monthly-sales.

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/monthly-sales
```

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

```
# reporter.py

def to_usd(my_price):
    """
    Converts a numeric value to usd-formatted string, for printing and
    display purposes.
    Source: https://github.com/prof-rossetti/intro-to-
python/blob/master/notes/python/datatypes/numbers.md#formatting-as-currency
    Param: my_price (int or float) like 4000.444444
    Example: to_usd(4000.444444)
    Returns: $4,000.44
    """
    return f"${my_price:,.2f}" #> $12,000.71

print("GENERATING SALES REPORT FOR MONTH OF OCTOBER 2013...")
```

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. "monthly-sales/data/monthly-sales/sales-201803.csv"). 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).

Environment Setup

Create and activate a new Anaconda virtual environment:

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

From within the virtual environment, install any packages you think you'll need:

```
pip install pandas # (only if you're using pandas to parse the CSV files)
```

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

```
python reporter.py
```

If you see the "GENERATING SALES REPORT FOR MONTH OF OCTOBER 2013..." message, 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".

Instructions

Adapt the contents of the "reporter.py" script to process the CSV file to display a human-readable representation of the given month and year and the total sales for that month.

Example output:

```
SALES REPORT (MARCH 2018)
```

```
TOTAL SALES: $12,000.71
```

HINT: use either [the csv module](#) or [the pandas package](#) to read the CSV file contents into memory for further processing

Further Exploration Challenges

Adapt your script to also identify the top-selling products for that month and the total sales for each. Example output:

SALES REPORT (MARCH 2018)

TOTAL SALES: \$12,000.71

TOP SELLING PRODUCTS:

1. Button-Down Shirt: \$6,960.35
2. Super Soft Hoodie: \$1,875.00
3. etc.

HINT: use the `groupby()` method provided by either the [the itertools module](#) or the [the pandas package](#) to group the rows by product for further processing

Solutions

There are many many ways to solve the challenges. Here are some example solutions via two different approaches (i.e. the `csv` module vs the `pandas` package):

- [Solution](#)
- [Further Exploration Solution](#)
- [Solution \(using Pandas\)](#)
- [Further Exploration Solution \(using Pandas\)](#)