added activities solutions and homework
ameralhomdy authored 18 hours ago

Name	Last commit	Last update
..		
Images	added activities solutions and homework	18 hours ago
PyBank/Resources	added activities solutions and homework	18 hours ago
PyRamen	added activities solutions and homework	18 hours ago
FinTech_Unit_2_Homework_Grading_Rubric -...	added activities solutions and homework	18 hours ago
FinTech_Unit_2_Homework_Grading_Rubric.pdf	added activities solutions and homework	18 hours ago
README.md	added activities solutions and homework	18 hours ago

README.md

Unit 2 | Homework Assignment: Automate Your Day Job with Python

Background

You've made it! It's time to put away the Excel sheet and join the big leagues. Welcome to the world of programming with Python. In this homework assignment, you'll be using the concepts you've learned to complete **two** Python activities, PyBank and PyRamen. Both activities present a real-world situation in which your newfound Python skills will come in handy. These activities are far from easy, though, so expect some hard work ahead!

Before You Begin

1. Create a new GitHub repo called `python-homework`. Then, clone it to your computer.
2. In your local git repository, create a directory for both of the Python activities. Use folder names that correspond to the activities: **PyBank** and **PyRamen**.
3. In each folder you just created, add a new file called `main.ipynb`. Remember that to create this file you will need to use JupyterLab to correctly generate the .ipynb file format. This will be the main notebook to run for each analysis.
4. Push the above changes to GitHub.

PyBank

In this activity, you are tasked with creating a Python script for analyzing the financial records of your company. You will be provided with a financial dataset in this file: budget_data.csv. This dataset is composed of two columns, Date and Profit/Losses. (Thankfully, your company has rather lax standards for accounting, so the records are simple.)

Your task is to create a Python script that analyzes the records to calculate each of the following:

- The total number of months included in the dataset.
- The net total amount of Profit/Losses over the entire period.
- The average of the changes in Profit/Losses over the entire period.
- The greatest increase in profits (date and amount) over the entire period.
- The greatest decrease in losses (date and amount) over the entire period.

Your resulting analysis should look similar to the following:

```
Financial Analysis
----------------------------
Total Months: 86
Total: $38382578
Average  Change: $-2315.12
Greatest Increase in Profits: Feb-2012 ($1926159)
Greatest Decrease in Profits: Sep-2013 ($-2196167)
```

Your final script should print the analysis to the terminal and export a text file with the results.

PyRamen (Optional)

Background

Welcome to Ichiban Ramen!

Opening a ramen shop has always been your dream, and now it's finally been realized—you're closing out on your second year of sales! Like last year, you need to analyze your business's financial performance by cross-referencing your sales data with your internal menu data to figure out revenues and costs for the year.

This year, you also want to analyze how well your business did on a per-product basis (as you have several choices of ramen) in order to better understand which products are doing well, which are doing poorly, and, ultimately, which products may need to be removed or changed.

You tried doing this type of per-product analysis last year in Excel, but you were not able to keep your reports up-to-date with your current sales data. Therefore, you need to innovate. With more customers and more data to process, you'll need a tool that will allow you to automate your calculations in a manner that scales with your business.

Enter Python! Python provides a wide range of capabilities for handling data, harnessing the power of low-level Python data structures and high-level development libraries, all the while supporting the automation and scalability needs for a growing enterprise.

In this homework assignment, you will need to:

1. Read the Data
2. Manipulate the Data

Instructions

Read the Data

Complete the following:

- Read in `menu_data.csv` and set its contents to a separate list object. (This way, you can cross-reference your menu data with your sales data as you read in your sales data in the coming steps.)
    - Initialize an empty `menu` list object to hold the contents of `menu_data.csv`.
    - Use a `with` statement and open the `menu_data.csv` by using its file path.
    - Use the `reader` function from the `csv` library to begin reading `menu_data.csv`.
    - Use the `next` function to skip the header (first row of the CSV).
    - Loop over the rest of the rows and append every row to the `menu` list object (the outcome will be a list of lists).
- Set up the same process to read in `sales_data.csv`. However, instead append every row of the sales data to a new `sales` list object.

Manipulate the Data

Complete the following:

- Initialize an empty `report` dictionary to hold the future aggregated per-product results. The `report` dictionary will eventually contain the following metrics:
    - `01-count` : the total quantity for each ramen type
    - `02-revenue` : the total revenue for each ramen type
    - `03-cogs` : the total cost of goods sold for each ramen type
    - `04-profit` : the total profit for each ramen type
- Then, loop through every row in the `sales` list object.
    - For each row of the `sales` data, set the following columns of the sales data to their own variables:
        - Quantity
        - Menu_Item
    - Perform a quick check if the `sales_item` is already included in the `report` . If not, initialize the key-value pairs for the particular `sales_item` in the report. Then, set the `sales_item` as a new key to the `report` dictionary and the values as a nested dictionary containing the following:

```
{
"01-count": 0,
"02-revenue": 0,
"03-cogs": 0,
"04-profit": 0,
}
```

- Create a nested loop by looping through every record in `menu` .
    - For each row of the `menu` data, set the following columns of the menu data to their own variables:
        - Item
        - Price
        - Cost
    - If the `sales_item` in sales is equal to the `item` in `menu` , capture the `quantity` from the sales data and the `price` and `cost` from the menu data to calculate the `profit` for each item.
        - Cumulatively add the values to the corresponding metrics in the report like so:

```
report[sales_item]["01-count"] += quantity
report[sales_item]["02-revenue"] += price * quantity
report[sales_item]["03-cogs"] += cost * quantity
report[sales_item]["04-profit"] += profit * quantity
```

    - Else print the message "{sales_item} does not equal {item}! NO MATCH!".
- Write out the contents of the `report` dictionary to a text file. The report should output each ramen type as the keys and `01-count` , `02-revenue` , `03-cogs` , and `04-profit` metrics as the values for every ramen type as shown:

```
spicy miso ramen {'01-count': 9238, '02-revenue': 110856.0, '03-cogs': 46190.0, '04-profit': 64666.0}
tori paitan ramen {'01-count': 9156, '02-revenue': 119028.0, '03-cogs': 54936.0, '04-profit': 64092.0}
truffle butter ramen {'01-count': 8982, '02-revenue': 125748.0, '03-cogs': 62874.0, '04-profit': 62874.0}
tonkotsu ramen {'01-count': 9288, '02-revenue': 120744.0, '03-cogs': 55728.0, '04-profit': 65016.0}
vegetarian spicy miso {'01-count': 9216, '02-revenue': 110592.0, '03-cogs': 46080.0, '04-profit': 64512.0}
shio ramen {'01-count': 9180, '02-revenue': 100980.0, '03-cogs': 45900.0, '04-profit': 55080.0}
miso crab ramen {'01-count': 8890, '02-revenue': 106680.0, '03-cogs': 53340.0, '04-profit': 53340.0}
nagomi shoyu {'01-count': 9132, '02-revenue': 100452.0, '03-cogs': 45660.0, '04-profit': 54792.0}
soft-shell miso crab ramen {'01-count': 9130, '02-revenue': 127820.0, '03-cogs': 63910.0, '04-profit': 63910.
burnt garlic tonkotsu ramen {'01-count': 9070, '02-revenue': 126980.0, '03-cogs': 54420.0, '04-profit': 72560
vegetarian curry + king trumpet mushroom ramen {'01-count': 8824, '02-revenue': 114712.0, '03-cogs': 61768.0,
```

Resources

- Stack Overflow: A wealth of community-driven questions and answers, particularly effective for IT solution seekers.
- Python Basics: Contains example materials and exercises for the Python 3 programming language.
- Python Documentation: Official Python documentation

Hints and Considerations

- Consider what we've learned so far. To date, we've learned how to import modules like `csv` ; to read and write files in various formats; to store contents in variables, lists, and dictionaries; to iterate through basic data structures; and to debug along the way. Using what we've learned, try to break down you tasks into discrete mini-objectives. This will be a *much* better course of action than attempting to Google search for a miracle.
- As you will discover, for some of these activities, the datasets are quite large. This was done purposefully, as it showcases one of the limits of Excel-based analysis. While our first instinct as data analysts is often to head straight to Excel, creating scripts in Python can provide us with more robust options for handling "big data."
- Your scripts should work for each dataset provided. Run your script for each dataset separately to make sure that the code works for different data.
- Feel encouraged to work in groups, but don't shortchange yourself by copying someone else's work. Dig your heels in, burn the night oil, and learn this while you can! These are skills that will pay dividends in your future career.
- **Start early**, and reach out for help often! Challenge yourself to identify *specific* questions for your instructors and TAs. Don't resign yourself to simply saying, "I'm totally lost." Come prepared to show your effort and thought patterns, we'll be happy to help along the way.
- Always commit your work (and do it often!) and back it up with GitHub pushes. You don't want to lose hours of your work because you didn't push it to GitHub every half hour or so.

Submission

- Upload homework files to your GitHub repo.
- Submit the link to your GitHub repo on Bootcamp Spot.