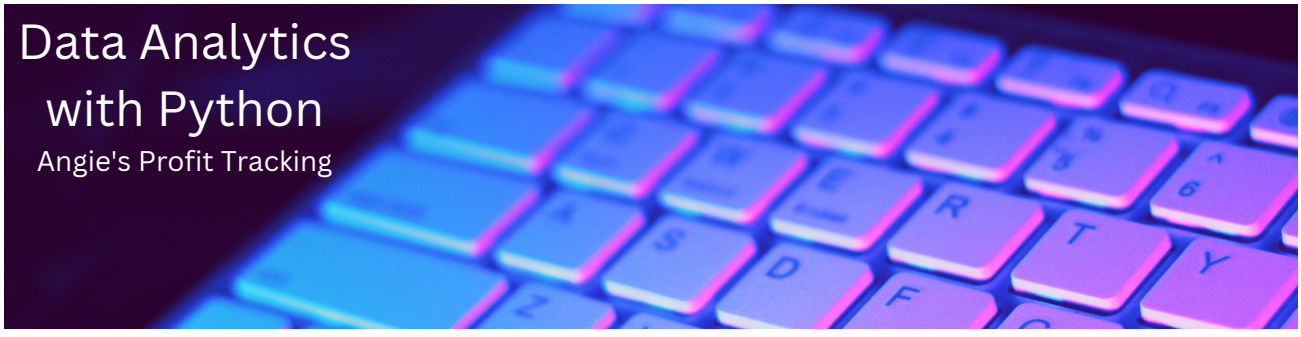

Python Works like a Powerful Calculator. We use it to Clean/ and Grab pretty much any Information from Large Data Sets. So if you have an Excel Workbook with Multiple Sheets each containing thousands of rows and columns. We can use Python to help us grab and only calculate the things we need.

Here I used Python script to analyze the financial records of a company. I was given a set of financial data called budget_data.csv.

My task was 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 changes in "Profit/Losses" over the entire period, and then the average of those changes
- The greatest increase in profits (date and amount) over the entire period
- The greatest decrease in profits (date and amount) over the entire period

The first Step in the Process is to import your CSV.. Then you define your variables. Since I'm only looking at the Profit fluctuations through out the 86 months of data. I only needed to grab the Months and Profits.

```
12  import csv
13
14  # Define PyBank's variables
15  months = []
16  profit_loss_changes = []
17
18  count_months = 0
19  net_profit_loss = 0
20  previous_month_profit_loss = 0
21  current_month_profit_loss = 0
22  profit_loss_change = 0
23
24
25  # Change directory to the directory of current python script
26  #os.chdir(r'C:\Users\Angiescomputer\Python\Resources')
27
28  # Path to collect data from the Resources folder
29  budget_data_csv_path = os.path.join("budget_data.csv")
```

By Using Python I was able to grab only the information I needed to Answers the questions above. By Submitting Functions for Python to perform.

Example.

```
# Read through each row of data after the header
for row in csv_reader:

    # Count of months
    count_months += 1

    # Net total amount of "Profit/Losses" over the entire period
    current_month_profit_loss = int(row[1])
    net_profit_loss += current_month_profit_loss

    if (count_months == 1):
        # Make the value of previous month to be equal to current month
        previous_month_profit_loss = current_month_profit_loss
        continue

    else:

        # Compute change in profit loss
        profit_loss_change = current_month_profit_loss - previous_month_profit_loss

        # Append each month to the months[]
        months.append(row[0])
```

The Function below is how I would ask for:

- The greatest increase in profits (date and amount) over the entire period
- The greatest decrease in profits (date and amount) over the entire period

After inputting this functions, python will fetch your inquiries.

Results :

The code below : Is telling Python how to display your output. Pretty much how you want the answer to print.

```
    # Locate the index value of highest and lowest changes in "Profit/Losses" over the entire period
    highest_month_index = profit_loss_changes.index(highest_change)
    lowest_month_index = profit_loss_changes.index(lowest_change)

    # Assign best and worst month
    best_month = months[highest_month_index]
    worst_month = months[lowest_month_index]

# -->>  Print the analysis to the terminal
print("Financial Analysis")
print("----------------------------")
print(f"Total Months:  {count_months}")
print(f"Total:  ${net_profit_loss}")
print(f"Average Change:  ${average_profit_loss}")
print(f"Greatest Increase in Profits:  {best_month} (${highest_change})")
print(f"Greatest Decrease in Losses:  {worst_month} (${lowest_change})")


# -->>  Export a text file with the results
budget_file = os.path.join("Output", "budget_data.txt")
with open(budget_file, "w") as outfile:
```

Summary: Below are the results to my Python Inquiry. Out of the 86 months that this Data Covered these where the results.

Total Months: 86

Total Net Gross: $383,825.78

Average Change: $-2315.12

Greatest Increase in Profits: Feb-2012 ($19,261.59)

Greatest Decrease in Profits: Sep-2013 ($-21,961.67)

This is just one Example, you can use this method to answer anything as long as you have the right dataset.