

## Background

programming with Python. this assignment, use the concepts learned to complete **two** Python challenges, PyBank and PyPoll. Both tasks present a real-world situation.

## Before You Begin

Before starting the assignment, be sure to complete the following steps:

- Create a new repository for this project called `python-challenge`.
- Clone the new repository to your computer.
- Inside your local Git repository, create a folder for each Python assignment and name them `PyBank` and `PyPoll`.
- In each folder that you just created, add the following content:
  - A new file called `main.py`. This will be the main script to run for each analysis.
  - A `Resources` folder that contains the CSV files you used. Make sure that your script has the correct path to the CSV file.
  - An `analysis` folder that contains your text file that has the results from your analysis.

## Files

Download the following files to help you get started:

[Module 3 Challenge files](#)[Links to an external site.](#)

## PyBank Instructions

In this Challenge, you are tasked with creating a Python script to analyze the financial records of your company. You will be given a financial dataset called `budget_data.csv`. The dataset is composed of two columns: "Date" and "Profit/Losses".

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

- 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

Your analysis should align with the following results:

```
Financial Analysis
```

```
-----
```

```
Total Months: 86
```

```
Total: $22564198
```

```
Average Change: $-8311.11
```

```
Greatest Increase in Profits: Aug-16 ($1862002)
```

```
Greatest Decrease in Profits: Feb-14 ($-1825558)
```

In addition, your final script should both print the analysis to the terminal and export a text file with the results.

## PyPoll Instructions

In this Challenge, you are tasked with helping a small, rural town modernize its vote-counting process.

You will be given a set of poll data called `election_data.csv`. The dataset is composed of three columns: "Voter ID", "County", and "Candidate". Your task is to create a Python script that analyzes the votes and calculates each of the following values:

- The total number of votes cast
- A complete list of candidates who received votes
- The percentage of votes each candidate won
- The total number of votes each candidate won
- The winner of the election based on popular vote

Your analysis should align with the following results:

Election Results

-----

Total Votes: 369711

-----

Charles Casper Stockham: 23.049% (85213)

Diana DeGette: 73.812% (272892)

Raymon Anthony Doane: 3.139% (11606)

-----

Winner: Diana DeGette

-----

In addition, your final script should both print the analysis to the terminal and export a text file with the results.

## Hints and Considerations

- Consider what you've learned so far. You've learned how to import modules like `csv`. You've learned how to read and write files in various formats. You've learned how to store content in variables, lists, and dictionaries. You've learned how to iterate through basic data structures. And you've learned how to debug along the way. Using all that you've learned, try to break down your tasks into discrete mini-objectives.
- The datasets for these Challenges are quite large. This was done purposefully to showcase one of the limits of Excel-based analysis. As data analysts, our first instinct is often to go straight to Excel, but creating scripts in Python can provide us with more powerful options for handling big data.
- Write one script for each of the provided datasets. Run each script separately to make sure that the code works for its respective dataset.

## Requirements

### Correctly Reads in the CSV

- Reads in the CSVs for both PyBank and PyPoll using Python
- Successfully stores the header row

### Results Printed out to correctly to terminal

- Results correctly display for PyBank:
  - Total Months

- Total
- Average Change
- Greatest Increase
- Greatest Decrease
- Results correctly display for PyPoll:
  - Total Votes
  - Each candidate's total votes and percent of votes
  - Winner

### **Code Runs Error Free**

- Error Free
- Producing consistent results

### **Exports results to text file**

- The text file contains for PyBank:
  - Total Months
  - Total
  - Average Change
  - Greatest Increase
  - Greatest Decrease
- The text file contains for Pypoll:
  - Total Votes
  - Each candidate's total votes and percent of votes
  - Winner

### **Code is cleaned and commented**

- Has additional tests and debugging removed (
- Commented