"Chart Gallery" Exercise

> Prerequisites:
>
> - Business Intelligence and Dataviz

Learning Objectives

- Learn how to create data visualizations in Python.
- Practice an active learning process to perform research and investigate capabilities of third-party Python packages.

Setup

Repo Setup

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

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/chart-gallery
```

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

```
# three_charts.py

#
# CHART 1 (PIE)
#

pie_data = [
    {"company": "Company X", "market_share": 0.55},
    {"company": "Company Y", "market_share": 0.30},
    {"company": "Company Z", "market_share": 0.15}
]
```

```
print("----------------")
print("GENERATING PIE CHART...")
print(pie_data) # TODO: create a pie chart based on the pie_data

#
# CHART 2 (LINE)
#

line_data = [
    {"date": "2019-01-01", "stock_price_usd": 100.00},
    {"date": "2019-01-02", "stock_price_usd": 101.01},
    {"date": "2019-01-03", "stock_price_usd": 120.20},
    {"date": "2019-01-04", "stock_price_usd": 107.07},
    {"date": "2019-01-05", "stock_price_usd": 142.42},
    {"date": "2019-01-06", "stock_price_usd": 135.35},
    {"date": "2019-01-07", "stock_price_usd": 160.60},
    {"date": "2019-01-08", "stock_price_usd": 162.62},
]

print("----------------")
print("GENERATING LINE GRAPH...")
print(line_data) # TODO: create a line graph based on the line_data

#
# CHART 3 (HORIZONTAL BAR)
#

bar_data = [
    {"genre": "Thriller", "viewers": 123456},
    {"genre": "Mystery", "viewers": 234567},
    {"genre": "Sci-Fi", "viewers": 987654},
    {"genre": "Fantasy", "viewers": 876543},
    {"genre": "Documentary", "viewers": 283105},
    {"genre": "Action", "viewers": 544099},
    {"genre": "Romantic Comedy", "viewers": 121212}
]

print("----------------")
print("GENERATING BAR CHART...")
print(bar_data) # TODO: create a horizontal bar chart based on the bar_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.

Environment Setup

Create and activate a new Anaconda virtual environment:

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

From within the virtual environment, install any packages you might need (see "Instructions" below):

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

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

```
python three_charts.py
```

If you see the provided data structures, 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

You'll need to write Python code to convert the provided data structures into charts.

The instructions in the "Research" and "Investigation" sections below can help you get started on your own. Otherwise, you might consult the notes for one of the following Python charting packages to help you get started:

- The `matplotlib` Package
- The `plotly` Package (recommended)
- The `altair` Package

Generally the approach you should try to follow is something like:

1. Identify the problem or goal (e.g. "how to make charts in Python?").
2. Identify potential solutions (e.g. various charting packages).
3. Evaluate potential solutions to determine popularity and usability.
4. Choose a package to further investigate, and complete each of the following steps in order before moving on to the next:
    1. See if you can make one of the example charts provided by that package's documentation (without changing any or much of the example code).
    2. See if you can make the specific kind of chart you're interested in making (e.g. bar chart), after finding a corresponding example provided by that package's documentation (without changing any or much of the example code).
    3. See if you can chart your desired dataset instead of the dataset provided by the chart example.

4. For chart configuration and styling efforts, search the package documentation for information about the kind of change you'd like to make (e.g. "how to format x axis labels as USD?" or "how to display a chart title?"). Then implement the suggested changes one step at a time, using scientific method to make a small change to the working chart code and seeing how the change affects the chart display. Repeat until the chart looks the way you'd like!

Research Phase

Search the Internet to identify a handful of potential third-party Python packages which can produce data visualizations. Consult their official documentation, GitHub source code repositories, and a variety of credible third-party sources to familiarize yourself with the relative popularity and usability of each package.

As you find helpful online resources and documentation, assemble a list of URLs for future reference and attribution purposes.

Investigation Phase

After identifying a few potential dataviz packages, use your judgment about their relative popularity and usability to choose one to investigate further.

From within your project's virtual environment, use Pip to install your selected package according to its installation instructions.

Either in the "three_charts.py" file or in a separate scratch-work file called something like "investigate.py", see if you can write Python code to implement some of the simpler examples from your chosen package's documentation. Commit your code incrementally as you make progress and reach certain milestones.

If at any time you think you'd like to investigate a different package, be open to doing so, perhaps in a separate scratch-work file. You might have to try a few different options before you find a package that works best for you. Take your time and enjoy the investigation process.

Development Phase

Once you're comfortable in your ability to make an example chart using your chosen package, adapt the contents of the "three_charts.py" script to process the provided data structures into three respective charts (i.e. pie, line, horizontal bar).

Focus on one chart at a time. Commit your code incrementally as you make progress and reach certain milestones. Make at least one commit per chart.

Prefer to generate a draft version of each chart before moving on to customizing the intricate details of each (e.g. chart formatting and style).