<script src="https://gist.github.com/Shanils123/a09f0c4f241abb37a375a6c1b39c548d.js"></script>

|  |
| --- |
| #!/usr/bin/env python |
|  | # coding: utf-8 |
|  |  |
|  | # # Introduction |
|  | # |
|  | # In this project, you will act as a data visualization developer at Yahoo Finance! You will be helping the "Netflix Stock Profile" team visualize the Netflix stock data. In finance, a \_stock profile\_ is a series of studies, visualizations, and analyses that dive into different aspects a publicly traded company's data. |
|  | # |
|  | # For the purposes of the project, you will only visualize data for the year of 2017. Specifically, you will be in charge of creating the following visualizations: |
|  | # + The distribution of the stock prices for the past year |
|  | # + Netflix's earnings and revenue in the last four quarters |
|  | # + The actual vs. estimated earnings per share for the four quarters in 2017 |
|  | # + A comparison of the Netflix Stock price vs the Dow Jones Industrial Average price in 2017 |
|  | # |
|  | # Note: We are using the Dow Jones Industrial Average to compare the Netflix stock to the larter stock market. Learn more about why the Dow Jones Industrial Average is a general reflection of the larger stock market [here](https://www.investopedia.com/terms/d/djia.asp). |
|  | # |
|  | # During this project, you will analyze, prepare, and plot data. Your visualizations will help the financial analysts asses the risk of the Netflix stock. |
|  | # |
|  | # After you complete your visualizations, you'll be creating a presentation to share the images with the rest of the Netflix Stock Profile team. Your slides should include: |
|  | # |
|  | # - A title slide |
|  | # - A list of your visualizations and your role in their creation for the "Stock Profile" team |
|  | # - A visualization of the distribution of the stock prices for Netflix in 2017 |
|  | # - A visualization and a summary of Netflix stock and revenue for the past four quarters and a summary |
|  | # - A visualization and a brief summary of their earned versus actual earnings per share |
|  | # - A visualization of Netflix stock against the Dow Jones stock (to get a sense of the market) in 2017 |
|  | # |
|  | # Financial Data Source: [Yahoo Finance](https://finance.yahoo.com/quote/DATA/) |
|  | # |
|  |  |
|  | # ## Step 1 |
|  | # |
|  | # Let's get our notebook ready for visualizing! Import the modules that you'll be using in this project: |
|  | # - `from matplotlib import pyplot as plt` |
|  | # - `import pandas as pd` |
|  | # - `import seaborn as sns` |
|  |  |
|  | # In[77]: |
|  |  |
|  |  |
|  | from matplotlib import pyplot as plt |
|  | import pandas as pd |
|  | import seaborn as sns |
|  |  |
|  |  |
|  | # ## Step 2 |
|  |  |
|  | # Let's load the datasets and inspect them. |
|  |  |
|  | # Load \*\*NFLX.csv\*\* into a DataFrame called `netflix\_stocks`. Then, quickly inspect the DataFrame using `print()`. |
|  | # |
|  | # Hint: Use the `pd.read\_csv()`function). |
|  | # |
|  | # Note: In the Yahoo Data, `Adj Close` represents the adjusted close price adjusted for both dividends and splits. This means this is the true closing stock price for a given business day. |
|  |  |
|  | # In[78]: |
|  |  |
|  |  |
|  | netflix\_stocks = pd.read\_csv("NFLX.csv") |
|  | print= (netflix\_stocks) |
|  |  |
|  |  |
|  | # Load \*\*DJI.csv\*\* into a DataFrame called `dowjones\_stocks`. Then, quickly inspect the DataFrame using `print()`. |
|  | # |
|  | # Note: You can learn more about why the Dow Jones Industrial Average is a industry reflection of the larger stock market [here](https://www.investopedia.com/terms/d/djia.asp). |
|  | # |
|  |  |
|  | # In[79]: |
|  |  |
|  |  |
|  | dowjones\_stocks = pd.read\_csv("DJI.csv") |
|  | print= (dowjones\_stocks) |
|  |  |
|  |  |
|  | # Load \*\*NFLX\_daily\_by\_quarter.csv\*\* into a DataFrame called `netflix\_stocks\_quarterly`. Then, quickly inspect the DataFrame using `print()`. |
|  | # |
|  |  |
|  | # In[80]: |
|  |  |
|  |  |
|  | netflix\_stocks\_quarterly = pd.read\_csv('NFLX\_daily\_by\_quarter.csv') |
|  | print= (netflix\_stocks\_quarterly) |
|  |  |
|  |  |
|  | # ## Step 3 |
|  |  |
|  | # Let's learn more about our data. The datasets are large and it may be easier to view the entire dataset locally on your computer. Open the CSV files directly from the folder you downloaded for this project. |
|  | # - `NFLX` is the stock ticker symbol for Netflix and `^DJI` is the stock ticker symbol for the Dow Jones industrial Average, which is why the CSV files are named accordingly |
|  | # - In the Yahoo Data, `Adj Close` is documented as adjusted close price adjusted for both dividends and splits. |
|  | # - You can learn more about why the Dow Jones Industrial Average is a industry reflection of the larger stock market [here](https://www.investopedia.com/terms/d/djia.asp). |
|  | # |
|  | # Answer the following questions by inspecting the data in the \*\*NFLX.csv\*\*,\*\*DJI.csv\*\*, and \*\*NFLX\_daily\_by\_quarter.csv\*\* in your computer. |
|  |  |
|  | # What year is represented in the data? Look out for the latest and earliest date. |
|  |  |
|  | # In[81]: |
|  |  |
|  |  |
|  | #The year represented in that data is 2017. |
|  |  |
|  |  |
|  | # + Is the data represented by days, weeks, or months? |
|  | # + In which ways are the files different? |
|  | # + What's different about the columns for `netflix\_stocks` versus `netflix\_stocks\_quarterly`? |
|  |  |
|  | # In[82]: |
|  |  |
|  |  |
|  | ##The data for NFLX.csv and DJI.csv is represented by months while NFLX\_daily\_by\_quarter.csv is represented by days |
|  | ##Theres are quarter column added at the end of NFLX\_daily\_by\_quarter.csv |
|  |  |
|  |  |
|  | # ## Step 4 |
|  | # |
|  | # Great! Now that we have spent sometime looking at the data, let's look at the column names of the DataFrame `netflix\_stocks` using `.head()`. |
|  |  |
|  | # In[83]: |
|  |  |
|  |  |
|  | netflix\_stocks = pd.read\_csv("NFLX.csv") |
|  | print= (netflix\_stocks.head()) |
|  |  |
|  |  |
|  | # What do you notice? The first two column names are one word each, and the only one that is not is `Adj Close`! |
|  | # |
|  | # The term `Adj Close` is a confusing term if you don't read the Yahoo Documentation. In Yahoo, `Adj Close` is documented as adjusted close price adjusted for both dividends and splits. |
|  | # |
|  | # This means this is the column with the true closing price, so these data are very important. |
|  | # |
|  | # Use Pandas to change the name of of the column to `Adj Close` to `Price` so that it is easier to work with the data. Remember to use `inplace=True`. |
|  | # |
|  | # Do this for the Dow Jones and Netflix Quarterly pandas dataframes as well. |
|  | # Hint: Use [`.rename()`](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.rename.html)). |
|  | # |
|  |  |
|  | # In[84]: |
|  |  |
|  |  |
|  | netflix\_stocks.rename(columns={"Adj Close": "Price"}, inplace=True ) |
|  | netflix\_stocks\_quarterly.rename(columns={"Adj Close": "Price"}, inplace=True) |
|  | dowjones\_stocks.rename(columns={"Adj Close": "Price"}, inplace=True) |
|  |  |
|  |  |
|  | # Run `netflix\_stocks.head()` again to check your column name has changed. |
|  |  |
|  | # In[85]: |
|  |  |
|  |  |
|  | netflix\_stocks.head() |
|  |  |
|  |  |
|  | # Call `.head()` on the DataFrame `dowjones\_stocks` and `netflix\_stocks\_quarterly`. |
|  |  |
|  | # In[86]: |
|  |  |
|  |  |
|  | dowjones\_stocks.head() |
|  | netflix\_stocks\_quarterly.head() |
|  |  |
|  |  |
|  | # ## Step 5 |
|  | # |
|  | # In this step, we will be visualizing the Netflix quarterly data! |
|  | # |
|  | # We want to get an understanding of the distribution of the Netflix quarterly stock prices for 2017. Specifically, we want to see in which quarter stock prices flucutated the most. We can accomplish this using a violin plot with four violins, one for each business quarter! |
|  | # |
|  | # |
|  | # 1. Start by creating a variable `ax` and setting it equal to `sns.violinplot()`. This will instantiate a figure and give us access to the axes through the variable name `ax`. |
|  | # 2. Use `sns.violinplot()` and pass in the following arguments: |
|  | # + The `Quarter` column as the `x` values |
|  | # + The `Price` column as your `y` values |
|  | # + The `netflix\_stocks\_quarterly` dataframe as your `data` |
|  | # 3. Improve the readability of the chart by adding a title of the plot. Add `"Distribution of 2017 Netflix Stock Prices by Quarter"` by using `ax.set\_title()` |
|  | # 4. Change your `ylabel` to "Closing Stock Price" |
|  | # 5. Change your `xlabel` to "Business Quarters in 2017" |
|  | # 6. Be sure to show your plot! |
|  | # |
|  |  |
|  | # In[87]: |
|  |  |
|  |  |
|  | ax =plt.subplot() |
|  | sns.violinplot(data= netflix\_stocks\_quarterly, x="Quarter", y="Price") |
|  | ax.set\_title("Distribution of 2017 Netflix Stock Prices by Quarter") |
|  | plt.ylabel("Closing Stock Price") |
|  | plt.xlabel("Business Quarters in 2017") |
|  | plt.savefig('Netflixviolin.png') |
|  | plt.show() |
|  |  |
|  |  |
|  | # ## Graph Literacy |
|  | # - What are your first impressions looking at the visualized data? |
|  | # |
|  | # - In what range(s) did most of the prices fall throughout the year? |
|  | # |
|  | # - What were the highest and lowest prices? |
|  |  |
|  | # |
|  |  |
|  | # |
|  |  |
|  | # ## Step 6 |
|  | # |
|  | # Next, we will chart the performance of the earnings per share (EPS) by graphing the estimate Yahoo projected for the Quarter compared to the actual earnings for that quarters. We will accomplish this using a scatter chart. |
|  | # |
|  | # 1. Plot the actual EPS by using `x\_positions` and `earnings\_actual` with the `plt.scatter()` function. Assign `red` as the color. |
|  | # 2. Plot the actual EPS by using `x\_positions` and `earnings\_estimate` with the `plt.scatter()` function. Assign `blue` as the color |
|  | # |
|  | # 3. Often, estimates and actual EPS are the same. To account for this, be sure to set your transparency `alpha=0.5` to allow for visibility pf overlapping datapoint. |
|  | # 4. Add a legend by using `plt.legend()` and passing in a list with two strings `["Actual", "Estimate"]` |
|  | # |
|  | # 5. Change the `x\_ticks` label to reflect each quarter by using `plt.xticks(x\_positions, chart\_labels)` |
|  | # 6. Assing "`"Earnings Per Share in Cents"` as the title of your plot. |
|  | # |
|  |  |
|  | # In[88]: |
|  |  |
|  |  |
|  | ax =plt.subplot() |
|  | x\_positions = [1, 2, 3, 4] |
|  | chart\_labels = ["1Q2017","2Q2017","3Q2017","4Q2017"] |
|  | earnings\_actual =[.4, .15,.29,.41] |
|  | earnings\_estimate = [.37,.15,.32,.41 ] |
|  | plt.scatter(x\_positions, earnings\_actual, color = 'red') |
|  | plt.scatter(x\_positions, earnings\_estimate, color = 'blue', alpha=0.5) |
|  | plt.legend(["Actual", "Estimate"]) |
|  | plt.xticks(x\_positions, chart\_labels) |
|  | ax.set\_title("Earnings Per Share in Cents") |
|  | plt.savefig('EPS.png') |
|  | plt.show() |
|  |  |
|  |  |
|  | # ## Graph Literacy |
|  | # |
|  | # + What do the purple dots tell us about the actual and estimate earnings per share in this graph? Hint: In color theory red and blue mix to make purple. |
|  | # |
|  |  |
|  | # |
|  |  |
|  | # |
|  |  |
|  | # ## Step 7 |
|  |  |
|  | # Next, we will visualize the earnings and revenue reported by Netflix by mapping two bars side-by-side. We have visualized a similar chart in the second Matplotlib lesson [Exercise 4](https://www.codecademy.com/courses/learn-matplotlib/lessons/matplotlib-ii/exercises/side-by-side-bars). |
|  | # |
|  | # As you may recall, plotting side-by-side bars in Matplotlib requires computing the width of each bar before hand. We have pasted the starter code for that exercise below. |
|  | # |
|  | # 1. Fill in the `n`, `t`, `d`, `w` values for the revenue bars |
|  | # 2. Plot the revenue bars by calling `plt.bar()` with the newly computed `x\_values` and the `revenue\_by\_quarter` data |
|  | # 3. Fill in the `n`, `t`, `d`, `w` values for the earnings bars |
|  | # 4. Plot the revenue bars by calling `plt.bar()` with the newly computed `x\_values` and the `earnings\_by\_quarter` data |
|  | # 5. Create a legend for your bar chart with the `labels` provided |
|  | # 6. Add a descriptive title for your chart with `plt.title()` |
|  | # 7. Add labels to each quarter by assigning the position of the ticks through the code provided. Hint: `plt.xticks(middle\_x, quarter\_labels)` |
|  | # 8. Be sure to show your plot! |
|  | # |
|  |  |
|  | # In[89]: |
|  |  |
|  |  |
|  | # The metrics below are in billions of dollars |
|  | revenue\_by\_quarter = [2.79, 2.98,3.29,3.7] |
|  | earnings\_by\_quarter = [.0656,.12959,.18552,.29012] |
|  | quarter\_labels = ["2Q2017","3Q2017","4Q2017", "1Q2018"] |
|  | ax = plt.subplot() |
|  |  |
|  | # Revenue |
|  | n = 1 # This is our first dataset (out of 2) |
|  | t = 2 # Number of dataset |
|  | d = 4 # Number of sets of bars |
|  | w = .8 # Width of each bar |
|  | bars1\_x = [t\*element + w\*n for element |
|  | in range(d)] |
|  | plt.bar(bars1\_x, revenue\_by\_quarter) |
|  |  |
|  |  |
|  | # Earnings |
|  | n = 2 # This is our second dataset (out of 2) |
|  | t = 2 # Number of dataset |
|  | d = 4 # Number of sets of bars |
|  | w = .8 # Width of each bar |
|  | bars2\_x = [t\*element + w\*n for element |
|  | in range(d)] |
|  | plt.bar(bars2\_x, earnings\_by\_quarter) |
|  |  |
|  |  |
|  | middle\_x = [ (a + b) / 2.0 for a, b in zip(bars1\_x, bars2\_x)] |
|  | plt.xticks(middle\_x, quarter\_labels) |
|  | plt.legend(['Revenue', 'Earnings']) |
|  | plt.title('Revenue and Earnings in 2017') |
|  | plt.savefig('Revenue\_n\_Earnings2017.png') |
|  | plt.show() |
|  |  |
|  |  |
|  | # ## Graph Literacy |
|  | # What are your first impressions looking at the visualized data? |
|  | # |
|  | # - Does Revenue follow a trend? |
|  | # - Do Earnings follow a trend? |
|  | # - Roughly, what percentage of the revenue constitutes earnings? |
|  |  |
|  | # In[ ]: |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | # ## Step 8 |
|  | # |
|  | # In this last step, we will compare Netflix stock to the Dow Jones Industrial Average in 2017. We will accomplish this by plotting two line charts side by side in one figure. |
|  | # |
|  | # Since `Price` which is the most relevant data is in the Y axis, let's map our subplots to align vertically side by side. |
|  | # - We have set up the code for you on line 1 in the cell below. Complete the figure by passing the following arguments to `plt.subplots()` for the first plot, and tweaking the third argument for the second plot |
|  | # - `1`-- the number of rows for the subplots |
|  | # - `2` -- the number of columns for the subplots |
|  | # - `1` -- the subplot you are modifying |
|  | # |
|  | # - Chart the Netflix Stock Prices in the left-hand subplot. Using your data frame, access the `Date` and `Price` charts as the x and y axes respectively. Hint: (`netflix\_stocks['Date'], netflix\_stocks['Price']`) |
|  | # - Assign "Netflix" as a title to this subplot. Hint: `ax1.set\_title()` |
|  | # - For each subplot, `set\_xlabel` to `"Date"` and `set\_ylabel` to `"Stock Price"` |
|  | # - Chart the Dow Jones Stock Prices in the left-hand subplot. Using your data frame, access the `Date` and `Price` charts as the x and y axes respectively. Hint: (`dowjones\_stocks['Date'], dowjones\_stocks['Price']`) |
|  | # - Assign "Dow Jones" as a title to this subplot. Hint: `plt.set\_title()` |
|  | # - There is some crowding in the Y axis labels, add some space by calling `plt.subplots\_adjust(wspace=.5)` |
|  | # - Be sure to `.show()` your plots. |
|  | # |
|  |  |
|  | # In[90]: |
|  |  |
|  |  |
|  | #creating Dates for the labels on X lines and ticks |
|  | Date = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'] |
|  | xline = range(len(Date)) |
|  |  |
|  | # Left plot Netflix |
|  | plt.figure(figsize=(14,10)) |
|  | ax1 = plt.subplot(1, 2, 1) |
|  | ax1.plot(netflix\_stocks['Date'], netflix\_stocks['Price']) |
|  | ax1.set\_title("Netflix") |
|  | ax1.set\_xlabel("Date") |
|  | ax1.set\_ylabel("Stock Price") |
|  | ax1.set\_xticks(xline) |
|  | ax1.set\_xticklabels(Date) |
|  | plt.savefig('Netflix\_stocks2017') |
|  | plt.show() |
|  |  |
|  |  |
|  |  |
|  | # Right plot Dow Jones |
|  | plt.figure(figsize=(14,10)) |
|  | ax2 = plt.subplot(1, 2, 2) |
|  | ax2.plot(dowjones\_stocks['Date'], dowjones\_stocks['Price']) |
|  | ax2.set\_title("Dow Jones") |
|  | ax2.set\_xlabel("Date") |
|  | ax2.set\_ylabel("Stock Price") |
|  | ax2.set\_xticks(xline) |
|  | ax2.set\_xticklabels(Date) |
|  | plt.savefig('Dow\_Jones2017') |
|  | plt.show() |
|  |  |
|  |  |
|  | # - How did Netflix perform relative to Dow Jones Industrial Average in 2017? |
|  | # - Which was more volatile? |
|  | # - How do the prices of the stocks compare? |
|  |  |
|  | # |
|  |  |
|  | # # Step 9 |
|  | # |
|  | # It's time to make your presentation! Save each of your visualizations as a png file with `plt.savefig("filename.png")`. |
|  | # |
|  | # As you prepare your slides, think about the answers to the graph literacy questions. Embed your observations in the narrative of your slideshow! |
|  | # |
|  | # Remember that your slideshow must include: |
|  | # - A title slide |
|  | # - A list of your visualizations and your role in their creation for the "Stock Profile" team |
|  | # - A visualization of the distribution of the stock prices for Netflix in 2017 |
|  | # - A visualization and a summary of Netflix stock and revenue for the past four quarters and a summary |
|  | # - A visualization and a brief summary of their earned versus actual earnings per share |
|  | # - A visualization of Netflix stock against the Dow Jones stock (to get a sense of the market) in 2017 |
|  | # |