Task-by-Task Guide

If you would like a little more support while completing this project, explore this step-by-step resource to get additional hints and resources to help you along each task.

Task 1 – Import required libraries

For this task you will gather the required libraries to include in your application to write the code. This is usually done at the top of the program.

Here, you are importing the following required libraries:

- pandas for creating the dataframe
- numpy for forming a random number from a range
- Matplotlib.pyplot for displaying graphs
- seaborn for plotting the data
- random for making a choice from a list of items

You can also use the 'as' keyword to shorten the module name for use in the program. For example:

import pandas as pd

Resources:

importing modules from a package -- Python.org

seaborn QuickStart -- seaborn.pydata.org

Task 2 – Generate random data for the social media data

Now that you have the required imports, you need to generate some random tweet data to analyze. There are many ways to generate random data in Python, but some are more convenient than others. In this case, you may use pandas date range to choose a pseudo-random date within a range, the random module's choice to create a choice from a list, and numpy's random to create a random integer.

First of all you need to define a ***list*** of categories for the social media experiment. The list may include the following, for example:
Food, Travel, 'Fashion, Fitness, Music, Culture, Family, and Health

Next, generate a Python data dictionary with fields Date, Category, and number of likes, all with random data. You will need the data to align, so the 'Date' dictionary entry should be n periods long, The

Category should be a ***list*** of random choices n entries long and the Likes category should be random integers in the range 0 to lets say 10000 also with size equal to n. For example, if n is equal to 500:
data = {'Date': pd.date_range('2021-01-01', periods=500),

. . .

… Now Use the random method called choice to gather a random category.
'Category': [random.choice(categories) for _ in range(500)]

… Then Use numpy's random randint() to form a random integer for the number of likes.
'Likes': np.random.randint(0, 10000, size=500)}

Resources:

pandas date range -- pandas.pydata.org
python random choice

numpy random integer -- numpy.org

Task 3 – Load the data into a Pandas DataFrame and Explore the data

The next step is to load the randomly generated data into the pandas dataframe and print the data.

To do so, you need to use the DataFrame method of the pandas object and pass the data to it.

Then, print the dataframe head, the dataframe information, and the dataframe description.

Finally, print the count of each 'Category'element.

See the references below for more assistance:

Resources:

creating-a-pandas-dataframe -- geeksforgeeks.org

pandas.DataFrame.value_counts. -- pandas.pydata.org

dataframe describe -- https://pandas.pydata.org

dataframe head -- pandas.pydata.org

dataframe info -- pandas.pydata.org

Task 4 – Clean the data

An important aspect of processing data is to move invalid data points so you can effectively perform statistics and visualize valid data. The pandas dataframe has built-in functionality to clean the data.

First, remove all the null data using the appropriate dataframe drop method. Next, you may want to also remove duplicate data from the dataframe. Use a dataframe method to do so.

To appropriately display the data field, convert the dataframe field to a datetime format using the pandas object (not the dataframe). Hint: you pass the dataframe's 'Date' field to the pandas conversion method.

Next, convert the dataframe 'Likes' data to an integer.

Check out the references below for more information.

Resources:

_pandas cleaning -- w3schools.com

pandas.to_datetime -- pandas.pydata.org

dataframe astype -- pandas.pydata.org

Task 5– Visualize and Analyze the data

An important aspect of data analysis is the ability to physically view it to visually observe relationships among the data using charts and graphs. The second way to analyze the data is to perform statistics on it, for example compute the average

First, visualize the data using the seaborn module in a histogram plot of the Likes. This is accomplished using the method histplot, passing in the dataframe field 'Likes' as in df['Likes'].

In order to have the histogram show up in the output, use the MathPlotLib.pyplot's show merthod.

Now, create a boxplot with the x axis as 'Category', and the y axis as 'Likes'.

Be sure to also call the pyplot's show method to see the boxplot output.

Now perform some statistics on the data. First, print out the mean of the 'Likes' catregory.

Next, use the dataframe's groupby method to print out the mean of each Category 'Likes'

Check out the references below for more information.

Resources:

seaborn boxplopt -- seaborn.pydata.org

seaborn histogram -- seaborn.pydata.org

GroupBy.mean -- pandas.pydata.org

Task 5 – Describe Conclusions

Write a conclusion about your process and any key findings.

This is your opportunity to impress your prospective employer with your critical thinking and problem-solving skills. You may want to discuss the process you followed and share your struggles and how you overcame them. What do you think sets your portfolio project apart from those of other candidates?

You may even want to offer ideas for improving the design for future business endeavors.

At this point, you can prepare the project artifacts for uploading into your portfolio. You should include:

- An image file of your Graphs and Statistics with the fields and data displayed.
- Excerpts from your code explaining the purpose of the code.
- Any improvements/changes you would make to the application.