Assignment 1 - Descriptive statistics

INSTRUCTIONAL DETAILS

There are several ways to compute descriptive statistics in Python. One way is to use the statistics module in the Python Standard Library. This module provides functions for calculating mathematical statistics of numeric (Real-valued) data.

Note - the following examples will require modification in order to be used w/ the below mentioned data set. Specifically, you'll need to import that wine dataset as a data frame and read that frame in order to power your calculations.

For example, to compute the mean of a dataset, you can use the mean() function:

```
import statistics

data = [1, 2, 2, 3, 3, 3, 4, 4, 4, 4]
mean = statistics.mean(data)
print(mean)  # Output: 3.0
```

To compute the median of a dataset, you can use the median() function:

```
import statistics

data = [1, 2, 2, 3, 3, 3, 4, 4, 4, 4]
median = statistics.median(data)
print(median)  # Output: 3.0
```

To compute the mode of a dataset, you can use the mode() function:

```
import statistics

data = [1, 2, 2, 3, 3, 3, 4, 4, 4, 4]
mode = statistics.mode(data)
print(mode)  # Output: 4
```

You can also use the describe() function from the statistics module to compute a variety of summary statistics for a dataset, including the mean, median, mode, standard deviation, and several percentiles:

```
import statistics

data = [1, 2, 2, 3, 3, 3, 4, 4, 4, 4]
summary = statistics.describe(data)
print(summary)

DescribeResult(mean=3.0, median=3.0, mode=4, variance=2.5, stdev=1.5811388300841898, minmax=(1, 4),
sum=30)
```

Another option is to use the numpy module, which provides a wide range of functions for working with numerical data in Python. For example, to compute the mean of a dataset using numpy, you can use the mean() function:

```
import numpy as np

data = [1, 2, 2, 3, 3, 3, 4, 4, 4, 4]
mean = np.mean(data)
print(mean)  # Output: 3.0
```

To compute the median of a dataset using numpy, you can use the median() function:

```
import numpy as np

data = [1, 2, 2, 3, 3, 3, 4, 4, 4, 4]
median = np.median(data)
print(median)  # Output: 3.0
```

You can also use the std() function from numpy to compute the standard deviation of a dataset:

```
import numpy as np

data = [1, 2, 2, 3, 3, 3, 4, 4, 4, 4]
std = np.std(data)
print(std)  # Output: 1.5811388300841898
```

Finally, you can use the scipy module, which provides more advanced statistical functions, including functions for calculating percentiles, skewness, and kurtosis.

BACKGROUND

Descriptive statistics are a set of tools used to describe and summarize data. They are important to businesses because they allow them to understand the characteristics of their data and make informed decisions based on that understanding.

For example, a business might use descriptive statistics to understand the distribution of customer satisfaction scores. They might use measures such as the mean, median, and mode to understand the central tendency of the data, and measures such as the standard deviation to understand the spread of the data. This can help the business understand how satisfied their customers are overall and identify any areas that may need improvement.

Descriptive statistics can also be used to visualize data and make it easier to understand. For example, a business might use a histogram to visualize the distribution of customer satisfaction scores, or a scatterplot to understand the relationship between two variables such as customer satisfaction and repeat business.

Overall, descriptive statistics are an important tool for businesses because they help them understand the characteristics of their data and make informed decisions based on that understanding. This can ultimately lead to improved performance and increased profitability.

RESEARCH QUESTION

The dataset describes the wine cellar of a prestigious wine collector in the greater Boston metropolitan area. The collector would like your analysis of their collection (the data) through the lens of descriptive statistics.

Data analysis has become an increasingly important tool in the wine industry in recent years. Here are a few ways in which the wine industry has used data analysis:

Quality control: Winemakers can use data analysis to track the quality of their wines over time and identify any issues that may arise during the production process. This can help them improve the quality of their wines and ensure consistency from batch to batch.

Marketing and sales: Wineries can use data analysis to better understand their target market and identify trends in consumer behavior. This can help them optimize their marketing and sales strategies and target their efforts more effectively.

Supply chain management: Data analysis can help wineries optimize their supply chain by identifying bottlenecks and inefficiencies, and by forecasting demand more accurately.

Vineyard management: Wineries can use data analysis to optimize their vineyard management practices by analyzing factors such as soil quality, weather conditions, and pest and disease control.

Wine pairing: Data analysis can be used to identify patterns and trends in the flavors and characteristics of different wines, which can help restaurants and other wine sellers make more informed recommendations to their customers.

REQUIREMENTS FOR SUBMISSION

Write-up

Please create and submit a document w/ your thoughts, analysis, and any necessary visualizations.

Syntax

Please save your syntax as a text document (or as a Google Colab file) and submit.

FORMATTING

Here are some basic formatting guidelines that you can follow when submitting a Wentworth graduate school assignment through Google Docs:

Use a standard font: Use a standard font such as Times New Roman or Arial, and make sure the font size is easy to read (e.g. 12 point).

Use double spacing: Use double spacing between lines to make the document easier to read.

Use appropriate margins: Use appropriate margins (e.g. 1 inch) to give the document a professional appearance.

Use headings and subheadings: Use headings and subheadings to organize the content of the document and make it easier to read.

Use bulleted or numbered lists: Use bulleted or numbered lists to highlight important points or to organize information in a logical way.

Use proper citation style: Use the appropriate citation style (e.g. APA, MLA, Chicago) for the assignment and make sure to properly cite any sources you use.

Use page numbers: Use page numbers to help the reader navigate the document and to make it easier to reference specific sections.

Here are some basic formatting guidelines that you can follow when creating data visualizations for a Wentworth graduate school assignment:

Use appropriate chart types: Choose the chart type that is most appropriate for the data you are visualizing. For example, use a bar chart to compare categories, a line chart to show trends over time, and a pie chart to show proportions.

Use clear labels: Use clear and concise labels on the axes and legend of the chart to help the reader understand the data.

Use appropriate scales: Use appropriate scales on the axes of the chart to make it easy to read and compare the data.

Use appropriate colors: Use colors appropriately to help the reader understand the data. For example, use different colors to distinguish between different categories or use a sequential color scheme to show a progression from low to high values.

Use appropriate titles: Use a clear and concise title to describe the main message of the chart.

Use appropriate font sizes: Use a font size that is easy to read and makes the chart easy to interpret.

Use appropriate chart size: Use a chart size that is appropriate for the data you are visualizing and the size of the document.

Here are some syntax formatting guidelines for Python:

Use four spaces for indentation: Use four spaces to indent the code blocks within a function, loop, or control statement. Do not use tabs for indentation.

Use blank lines to separate code blocks: Use blank lines to separate code blocks within a function, loop, or control statement. This helps to make the code more readable and easier to understand.

Use descriptive variable names: Use descriptive variable names to make the code more readable and easier to understand. Avoid using short or cryptic names.

Use comments to document the code: Use comments to document the code and explain what the code is doing. Start comments with a "#" symbol and use them to describe the purpose of the code and any important details.

Use docstrings to document functions: Use docstrings to document functions and provide a brief description of what the function does and how it should be used. Docstrings should be placed immediately after the function definition and should be enclosed in triple quotes.

Use inline comments sparingly: Use inline comments sparingly, as they can make the code more cluttered and difficult to read. Only use inline comments to explain particularly complex or confusing sections of code.

Follow PEP 8 style guidelines: Follow the PEP 8 style guidelines, which provide recommendations on how to format and style Python code. Adhering to these guidelines can help to make your code more readable and easier to understand.

DATASET DETAILS (all data sets can be found here or here)
Wine.csv

DATASET FIELDS

unique_id
class
alcohol_percentage
malic_acid
ash
alcalinity
magnesium
phenols
flavanoids
nonflavanoids
proanthocyanins
color
hue
price_usd