

## Assessment Description

A researcher is conducting a study on the effects of different exercise regimens on blood pressure. The study involves 100 participants who are randomly assigned to one of three exercise groups: jogging, weightlifting, or yoga. Each participant's blood pressure is measured before and after the 6-week exercise program.

The researcher has collected the data and stored it in a CSV file. The file contains the following columns:

Participant ID (numeric) Exercise group (text: "jogging," "weightlifting," or "yoga") Pre-exercise systolic blood pressure (numeric)  
Post-exercise blood pressure (numeric) The researcher wants to analyze the data using Python and NumPy. Complete the following tasks as part of the initial statistical analysis of the scenario above.

## Generate Synthetic Dataset on Exercise and Blood Pressure

1. Create a Python script that generates a synthetic dataset matching the description of your study. The dataset should be saved as a CSV file named "exercise\_data.csv"

```
In [ ]: import random
import pandas as pd
import numpy as np
```

```
In [ ]: number_of_participants = 100

np.random.seed(0) # For reproducibility
participant_ids = np.arange(1, number_of_participants + 1)
exercise_groups = np.random.choice(['jogging', 'weightlifting', 'yoga'], number_of_participants)
pre_exercise_bp = np.random.normal(120, 15, number_of_participants) # Assume normal distribution around 120 mmHg
post_exercise_bp = pre_exercise_bp - np.random.normal(5, 10, number_of_participants) # Decrease with some variability

data = {
    'Participant ID': participant_ids,
    'Exercise group': exercise_groups,
    'Pre-exercise BP': pre_exercise_bp,
    'Post-exercise systolic BP': post_exercise_bp
}
df = pd.DataFrame(data)
```

```
csv_file_path = 'exercise_data.csv'  
df.to_csv(csv_file_path, index=False)
```

Explanation: The script creates a synthetic dataset by generating random values for participant IDs, exercise groups, and pre/post-exercise blood pressure. It uses NumPy's random functions for generating these values, ensuring a level of randomness and variability similar to real-world data.

## Highest Pre-Exercise Blood Pressure by Group

2. Write a Python script to read the "exercise\_data.csv" file and print the participant with the highest pre-exercise systolic blood pressure in each exercise group.

```
In [ ]: file = 'exercise_data.csv'  
read_df = pd.read_csv(file)  
  
max_pre_bp_jogging = df[df['Exercise group'] == 'jogging']['Pre-exercise systolic BP'].idxmax()  
max_pre_bp_weightlifting = df[df['Exercise group'] == 'weightlifting']['Pre-exercise systolic BP'].idxmax()  
max_pre_bp_yoga = df[df['Exercise group'] == 'yoga']['Pre-exercise systolic BP'].idxmax()  
  
print("Participant ID with highest pre-exercise systolic BP in each group:")  
print("Jogging: ", read_df.loc[max_pre_bp_jogging, 'Participant ID'])  
print("Weightlifting: ", read_df.loc[max_pre_bp_weightlifting, 'Participant ID'])  
print("Yoga: ", read_df.loc[max_pre_bp_yoga, 'Participant ID'])
```

Participant ID with highest pre-exercise systolic BP in each group:

Jogging: 39

Weightlifting: 94

Yoga: 82

Explanation: The script reads the dataset and identifies the participant with the highest pre-exercise blood pressure in each exercise group. It uses pandas to filter and sort the data.

## Extract the 5 Participants with Highest Blood Pressure

3. Write a Python function that sorts the list based on blood pressure and displays the full record of the top 5.

```
In [ ]: highest_bp = read_df.sort_values(by=['Pre-exercise systolic BP'], ascending=True)
highest_bp.head(5)
```

```
Out[ ]:
```

	Participant ID	Exercise group	Pre-exercise systolic BP	Post-exercise systolic BP
<b>44</b>	45	jogging	83.343260	69.896999
<b>7</b>	8	yoga	86.633946	82.599707
<b>19</b>	20	weightlifting	87.694881	75.635597
<b>61</b>	62	yoga	87.849668	84.086577
<b>75</b>	76	jogging	88.126196	74.743088

Explanation: The script sorts the data based on pre-exercise blood pressure and displays the records of the top 5 participants. This demonstrates data sorting and extraction capabilities in pandas.

## Monthly Blood Pressure Changes

4. Write a Python script that assumes that blood pressure measurements were taken monthly. Compute and print the average change in blood pressure for each exercise group. Note: This is hypothetical as the original study is for 6 weeks only.

```
In [ ]: read_df['BP Change'] = read_df['Post-exercise systolic BP'] - read_df['Pre-exercise systolic BP']
average_change_bp = read_df.groupby('Exercise group')['BP Change'].mean()

print("Average change in bp for each exercise group: ")
print(average_change_bp)
```

```
Average change in bp for each exercise group:
Exercise group
jogging          -5.837068
weightlifting    -4.503938
yoga             -4.293088
Name: BP Change, dtype: float64
```

Explanation: The script computes the average change in blood pressure for each exercise group, assuming monthly measurements. It involves calculating the difference between pre- and post-exercise blood pressure and then finding the average of these

differences.

## Compare Pre- and Post-Exercise Blood Pressure

5. Search for the 5 participants from the pre-exercise (Topic 4) and find their post-exercise blood pressure. Produce a table that compares their pre- and post-exercise pressure and displays the difference.

```
In [ ]: top_5_pre_bp = df.nlargest(5, 'Pre-exercise systolic BP')

comparison_table = top_5_pre_bp[['Participant ID', 'Pre-exercise systolic BP', 'Post-exercise systolic BP']]
comparison_table['BP Difference'] = comparison_table['Pre-exercise systolic BP'] - comparison_table['Post-exercise systolic BP']

print("Comparison of Pre- and Post-Exercise Systolic Blood Pressure: ")
print(comparison_table)
```

Comparison of Pre- and Post-Exercise Systolic Blood Pressure:

	Participant ID	Pre-exercise systolic BP	Post-exercise systolic BP \
93	94	152.953365	136.789620
33	34	148.626932	134.332937
38	39	146.341605	146.390568
81	82	144.889150	147.170429
42	43	144.071641	140.119173

	BP Difference
93	16.163745
33	14.293994
38	-0.048962
81	-2.281280
42	3.952468

Explanation: The script compares pre- and post-exercise blood pressure for the top 5 participants and displays the differences. This task involves data selection, comparison, and computation of differences.

## Total Blood Pressure Reduction for Each Exercise Group

6. Write a Python script to read the "exercise\_data.csv" file and compute the measures of central tendency for each exercise group: mean, mode, standard deviation.

```
In [ ]: exercise_groups = df['Exercise group'].unique()

for group in exercise_groups:
    group_data = df[df['Exercise group'] == group]
    print(f"{group} group:\n")
    for column in ['Pre-exercise systolic BP', 'Post-exercise systolic BP']:
        mean = group_data[column].mean()
        mode = group_data[column].mode().iloc[0]
        std_dev = group_data[column].std()
        print(f"{column}:\nMean = {mean},\nMode = {mode},\nStandard Deviation = {std_dev}\n")
    print()
```

jogging group:

Pre-exercise systolic BP:

Mean = 117.08430665682569,

Mode = 83.34325974232178,

Standard Deviation = 14.845603589239683

Post-exercise systolic BP:

Mean = 111.2472387215338,

Mode = 69.89699944029005,

Standard Deviation = 18.908900518538886

weightlifting group:

Pre-exercise systolic BP:

Mean = 120.7447816702838,

Mode = 87.69488120463342,

Standard Deviation = 15.356891288871793

Post-exercise systolic BP:

Mean = 116.24084344034989,

Mode = 75.6355973146187,

Standard Deviation = 22.003051008590212

yoga group:

Pre-exercise systolic BP:

Mean = 120.0799992795688,

Mode = 86.63394596681371,

Standard Deviation = 15.316872208225165

Post-exercise systolic BP:

Mean = 115.78691144643656,

Mode = 82.59970724122813,

Standard Deviation = 17.185228833241585

Explanation: The script calculates the mean, mode, and standard deviation for pre- and post-exercise blood pressure in each exercise group. It demonstrates the use of NumPy for statistical calculations.

