Python Proficency for Statistics

Feb 21, 2024

Zander Bonnet

In [1]:
```
import numpy as np
import pandas as pd
import random
from scipy import stats
```

Task 1

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".

To generate the dataset I 1) Assigned sequental numbers for participant id

2) Randomly assigned each individual to one of the three groups

3) Assigned a starting blood pressure to each of the people between 100 and 140

4) Assigned a value between -20 and 20 to be added to the persons existing blood presure to simulate change

In [2]:
```
##set our basic parameters
num_participants = 100
groups = ["jogging", "weightlifting","yoga"]
```

In [3]:
```
##Generate the values for the dataset
participant_id = list(range(1,num_participants + 1))
participant_group = random.choices(groups,k = num_participants)
pre_bp = [random.randint(100,140) for x in range(num_participants)]
post_bp = [x + random.randint(-20,20) for x in pre_bp]
```

In [4]:
```
##Generates and saves the Dataset
df = pd.DataFrame({'Participant ID': participant_id, 'Exercise Group': participa
                   'Pre-exercise systolic blood pressure': pre_bp,
                   'Post-exercise systolic blood pressure': post_bp})
df.to_csv('exercise_data.csv')
df
```

Out[4]:

	Participant ID	Exercise Group	Pre-exercise systolic blood pressure	Post-exercise systolic blood pressure
0	1	jogging	136	149

	Participant ID	Exercise Group	Pre-exercise systolic blood pressure	Post-exercise systolic blood pressure
1	2	jogging	135	154
2	3	yoga	121	103
3	4	weightlifting	105	113
4	5	jogging	137	121
...	...	...	...	...
95	96	yoga	127	110
96	97	yoga	106	102
97	98	yoga	132	126
98	99	weightlifting	134	152
99	100	yoga	116	108

100 rows × 4 columns

Task 2

Highest Pre-Exercise Blood Pressure by Group

1. 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.

To accomplish this is found the index of the max value for each group and then located them in the dataset

In [5]:
```
exer_data = pd.read_csv('exercise_data.csv', index_col= 0)
#Finds the index of the max of each group and converts it to a list
group_max = list(exer_data.groupby('Exercise Group')['Pre-exercise systolic bloc
#uses the index list to find the hoghest starting bp
exer_data.iloc[group_max]
```

Out[5]:

	Participant ID	Exercise Group	Pre-exercise systolic blood pressure	Post-exercise systolic blood pressure
4	5	jogging	137	121
73	74	weightlifting	139	157
80	81	yoga	138	147

Task 3

Extract the 5 Participants with Highest Blood Pressure

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

I sorted the list by the pre exercise blood pressure and used the head function to get the first 5 entries

In [6]:
```
#sorts the dataframe by the pre bp and then gets the top 5 values
top_5 = exer_data.sort_values(by = 'Pre-exercise systolic blood pressure', ascen
top_5
```

Out[6]:

	Participant ID	Exercise Group	Pre-exercise systolic blood pressure	Post-exercise systolic blood pressure
73	74	weightlifting	139	157
91	92	weightlifting	138	144
80	81	yoga	138	147
8	9	weightlifting	137	157
30	31	weightlifting	137	145

Task 4

Monthly Blood Pressure Changes

1. 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.

To accomplish this I calculated the average weekly change in blood pressure for each participant and then used that to calculate the estimated monthly change. I then grouped them into exercise groups and calculated the estimated average monthly change.

In [7]:
```
#Gets total bp change over 6 weeks then adds the values to the data frame
change_over_6weeks = (exer_data['Post-exercise systolic blood pressure']
                      - exer_data['Pre-exercise systolic blood pressure'])
exer_data.loc[:, 'Change over 6 weeks'] = change_over_6weeks

#Gets weekly average bp change then adds the values to the data frame
average_weekly_change = exer_data['Change over 6 weeks']/6
exer_data.loc[:, 'Average weelky Change'] = average_weekly_change

#Estimates the monthly change based on the average weekly change and rounds it t
#then adds the values to the data frame
estimated_monthly_change = round(exer_data['Average weelky Change']*4)
exer_data.loc[:, 'Estimated monthly change'] = estimated_monthly_change

#Groups them by exercise group and finds the mean of the monthly change
exer_data.groupby('Exercise Group')['Estimated monthly change'].mean()
```

Out[7]:
```
Exercise Group
jogging          1.800000
weightlifting    3.064516
yoga            -3.617647
Name: Estimated monthly change, dtype: float64
```

Task 5

Compare Pre- and Post-Exercise Blood Pressure

1. 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.

I had added the change over 6 weeks in task 4, so i just sorted the dataset again and displayed the top five entries.

In [8]:
```
#sorts by pre bp and gets first 5
sort_by_pre = exer_data.sort_values(by = 'Pre-exercise systolic blood pressure',
sort_by_pre.loc[:,'Participant ID': 'Change over 6 weeks']
```

Out[8]:

	Participant ID	Exercise Group	Pre-exercise systolic blood pressure	Post-exercise systolic blood pressure	Change over 6 weeks
73	74	weightlifting	139	157	18
91	92	weightlifting	138	144	6
80	81	yoga	138	147	9
8	9	weightlifting	137	157	20
30	31	weightlifting	137	145	8

Task 6

Total Blood Pressure Reduction for Each Exercise Group

1. 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.

To calculate the measures of central tendancy for the totla blood pressure reduction for each group I split the groups up so I could run them through a loop. In the loop I calculate the mean, standard deviation and mode for each group. I then create a new data frame to display the data

In [9]:
```
#creates empty lists
means = []
stdv = []
mode = []
#seperates the groups
groups = list(exer_data.groupby('Exercise Group')['Change over 6 weeks'])
for x in range(len(groups)):
    temp = groups[x][1]
    means.append(np.average(temp))
    stdv.append(np.std(temp))
    mode.append(stats.mode(temp))
#gets the names of the groups
gnames = list(exer_data['Exercise Group'].unique())
ctdf = pd.DataFrame({'Exercise Group': gnames, 'means':means, 'modes':mode, 'Sta
ctdf
```

Out[9]:

	Exercise Group	means	modes	Standard Deviations
0	jogging	2.742857	([11], [3])	11.364966
1	yoga	4.548387	([8], [4])	11.691846
2	weightlifting	-5.352941	([-6], [4])	11.416918