A00573055 Rodrigo Martinez Vallejo

- Elemento de lista
- Elemento de lista

Link del repositorio: https://github.com/a00573055/Mastering-Analytics-

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Data Preparation

```
data = pd.read_csv('data.csv')
```

```
numeric_data = data.select_dtypes(include='number')
```

Data Description

In this section I review the type of data each columns holds, as long as what it represents and the upper and lower limits of those columns.

```
print(data.dtypes)
```

```
student_id                        object
age                                int64
gender                            object
study_hours_per_day              float64
social_media_hours               float64
netflix_hours                    float64
part_time_job                     object
attendance_percentage            float64
sleep_hours                      float64
diet_quality                      object
exercise_frequency                 int64
parental_education_level          object
internet_quality                  object
mental_health_rating               int64
extracurricular_participation     object
exam_score                       float64
dtype: object
```

In total there are 15 variables and a total number of 1000 rows

- **student_id** *object* it is the unique identifier of each student
- **age** *int64* Age of student. Goes from 17 to 24
- **gender** *object* Male/Female/Other.
- **study_hours_per_day** *float64* Avg. daily study time. Goes from 0 to 8.3
- **social_media_hours** *float* Daily social media time. Goes from 0 to 7.2
- **netflix_hours** *float64* Avg. daily Netflix/binging time. Goes from 0 to 5.4
- **part_time-job** Yes/No.
- **attendance_percentage** *flaot64* Class attendace (0-100%).
- **sleep_hours** *float64* Avg. daily sleep
- **diet_quality** *object* Poor/Fair/Good. Goes form 3.2 to 10
- **exercise_frequency** *int64* Times per week. Goes from 0 to 7.
- **parental_education_level** *object* HighSchool/Bachellor/Other
- **internet_quality** *object* Good/Average/Other
- **mental_health_rating** *int64* Scale of 1 to 10
- **extracurricular_participation** *object* Yes/No
- **exam_scores** *float64* Final exam score (0-100)

Double-click (or enter) to edit

Mean, median, and standard deviation

The first thing that I notice from this data is for example how the results given from Mean and Median are almost exactly the same which suggests a very symetrical distribution of the data.

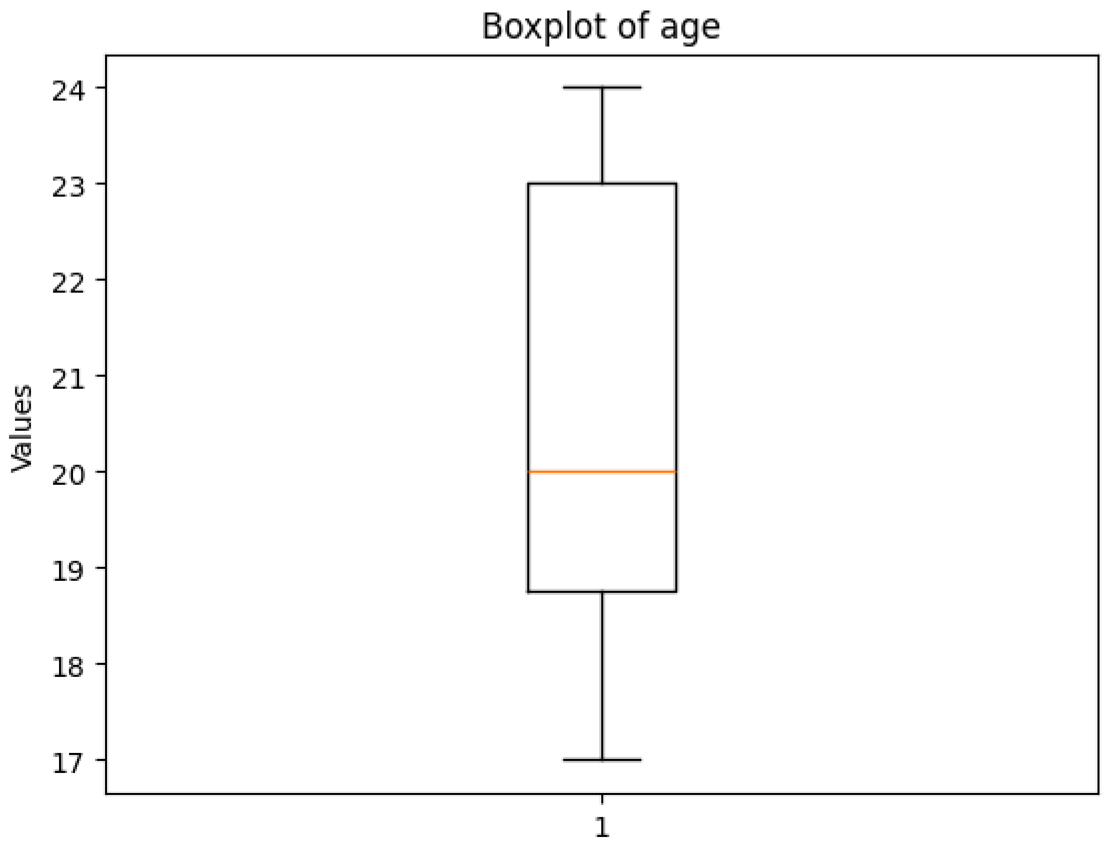
```
print("-->Mean")
print(numeric_data.mean())
print("-->Median")
print(numeric_data.median())
print("-->Standard deviation")
print(numeric_data.std())
```

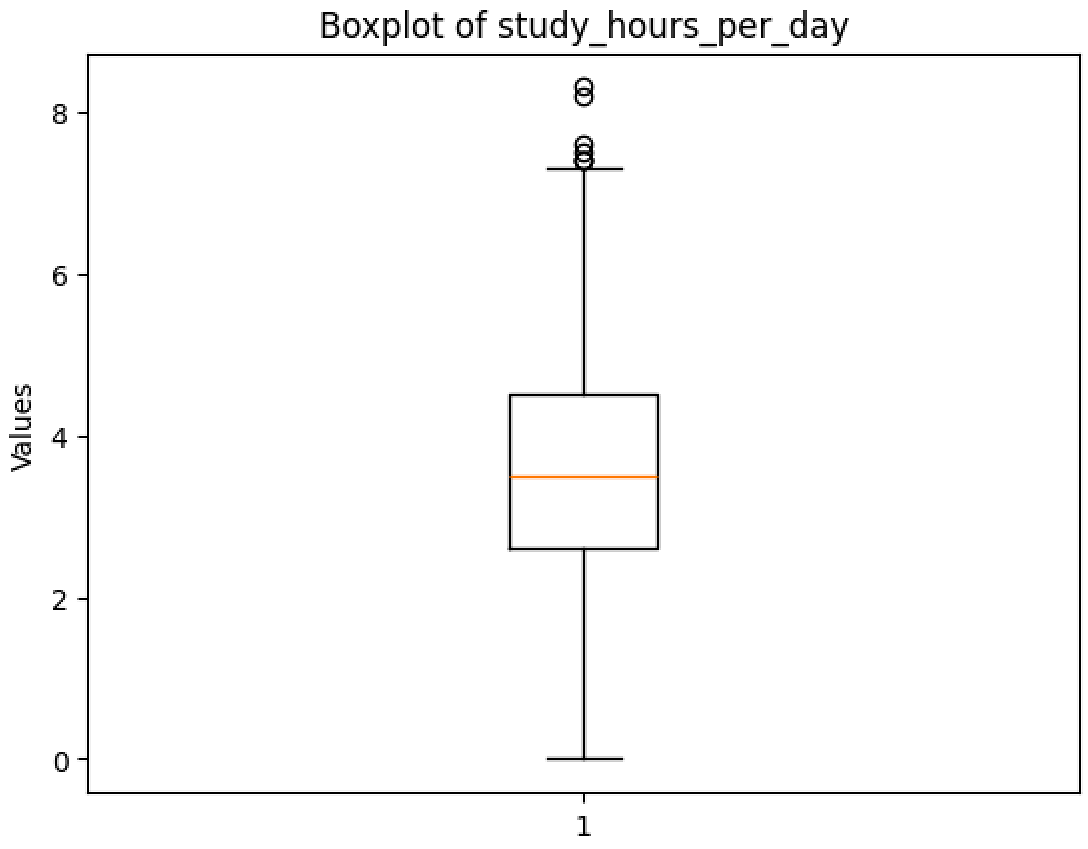
```
-->Mean
age                      20.4980
study_hours_per_day       3.5501
social_media_hours        2.5055
netflix_hours             1.8197
attendance_percentage    84.1317
sleep_hours               6.4701
exercise_frequency        3.0420
mental_health_rating      5.4380
exam_score               69.6015
dtype: float64
```

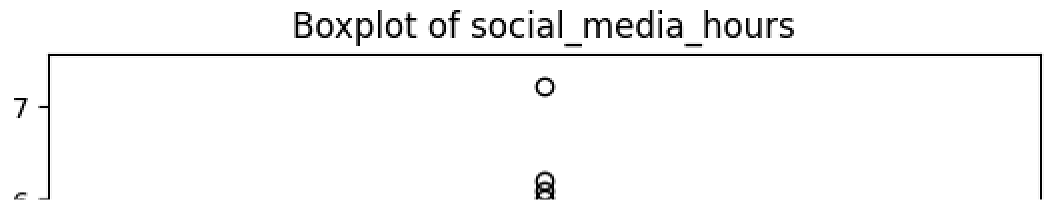
```
-->Median
age                     20.0
study_hours_per_day      3.5
social_media_hours       2.5
netflix_hours            1.8
attendance_percentage   84.4
sleep_hours              6.5
exercise_frequency       3.0
mental_health_rating     5.0
exam_score              70.5
dtype: float64
-->Standard deviation
age                      2.308100
study_hours_per_day      1.468890
social_media_hours       1.172422
netflix_hours            1.075118
attendance_percentage    9.399246
sleep_hours              1.226377
exercise_frequency       2.025423
mental_health_rating     2.847501
exam_score              16.888564
dtype: float64
```

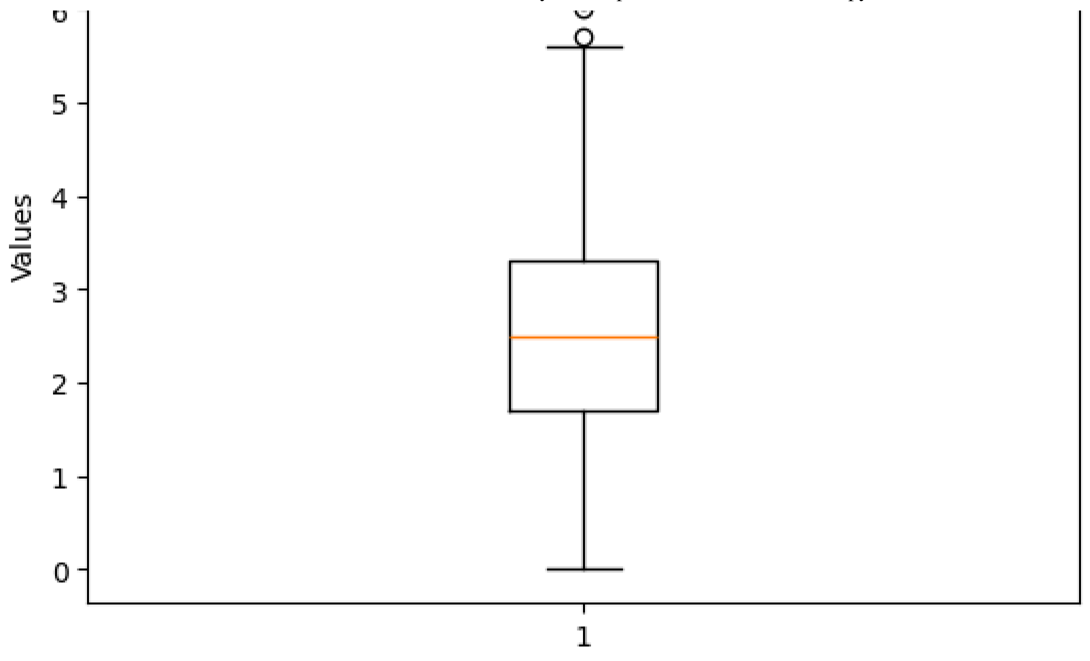
Box Diagram

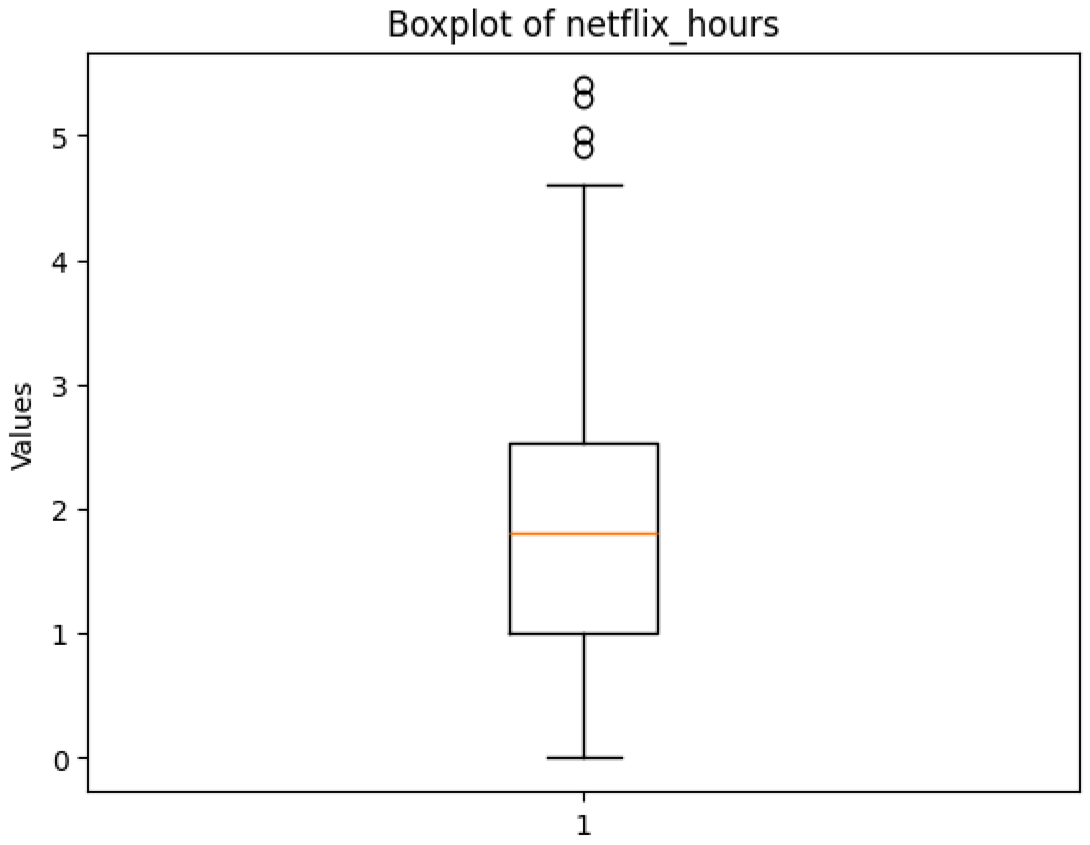
```
col_names = ['age', 'study_hours_per_day', 'social_media_hours', 'netflix_hours',
       'attendance_percentage', 'sleep_hours', 'exercise_frequency',
       'mental_health_rating', 'exam_score']

for name in col_names:
  plt.boxplot(numeric_data[name])
  plt.title(f'Boxplot of {name}')
  plt.ylabel('Values')
  plt.show()
```

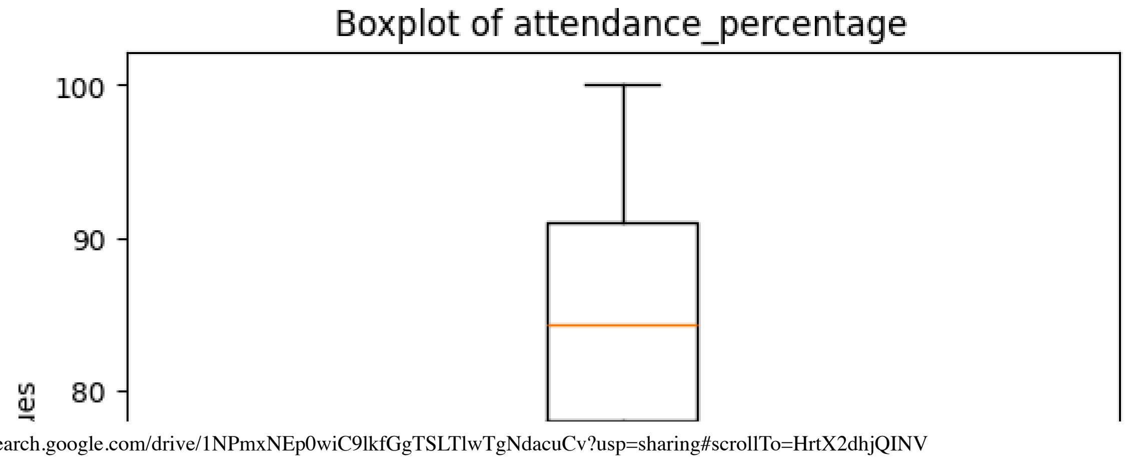

Boxplot of age
Values
24
23
22
21
20
19
18
17
1

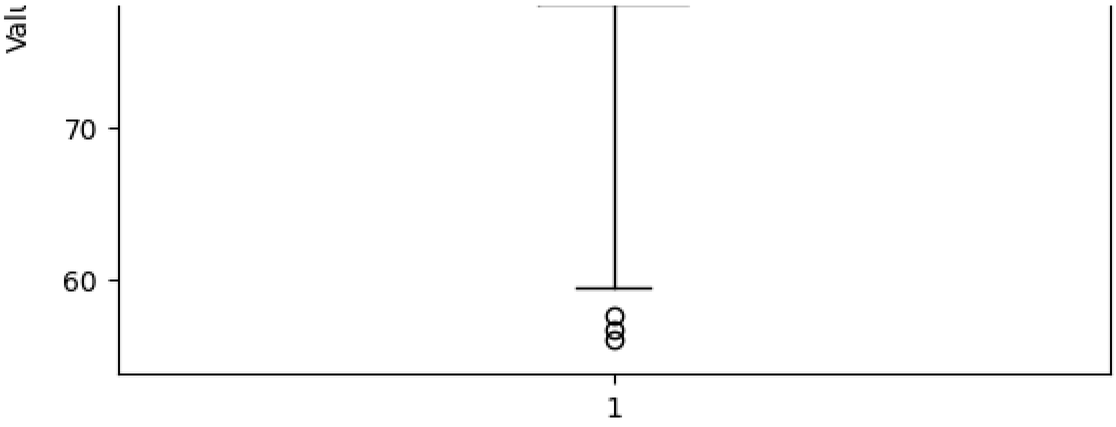

Boxplot of study_hours_per_day
Values
8
6
4
2
0
1

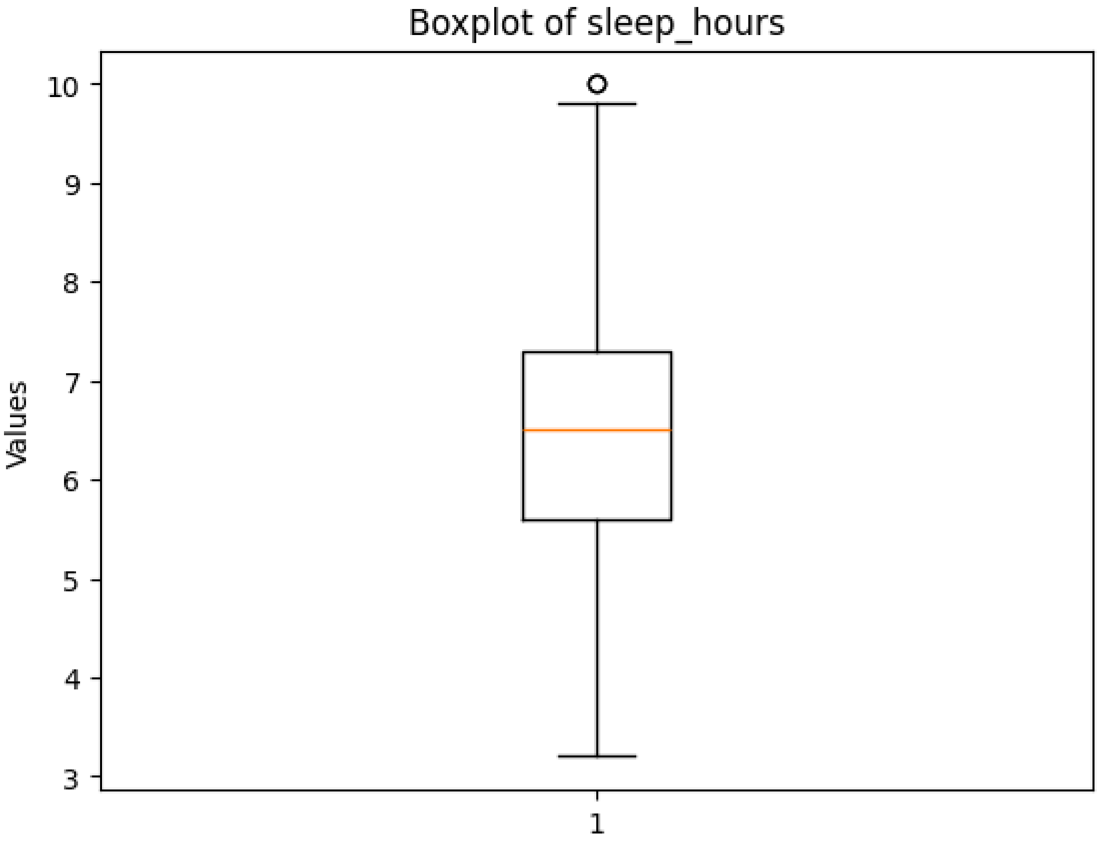

Boxplot of social_media_hours
7

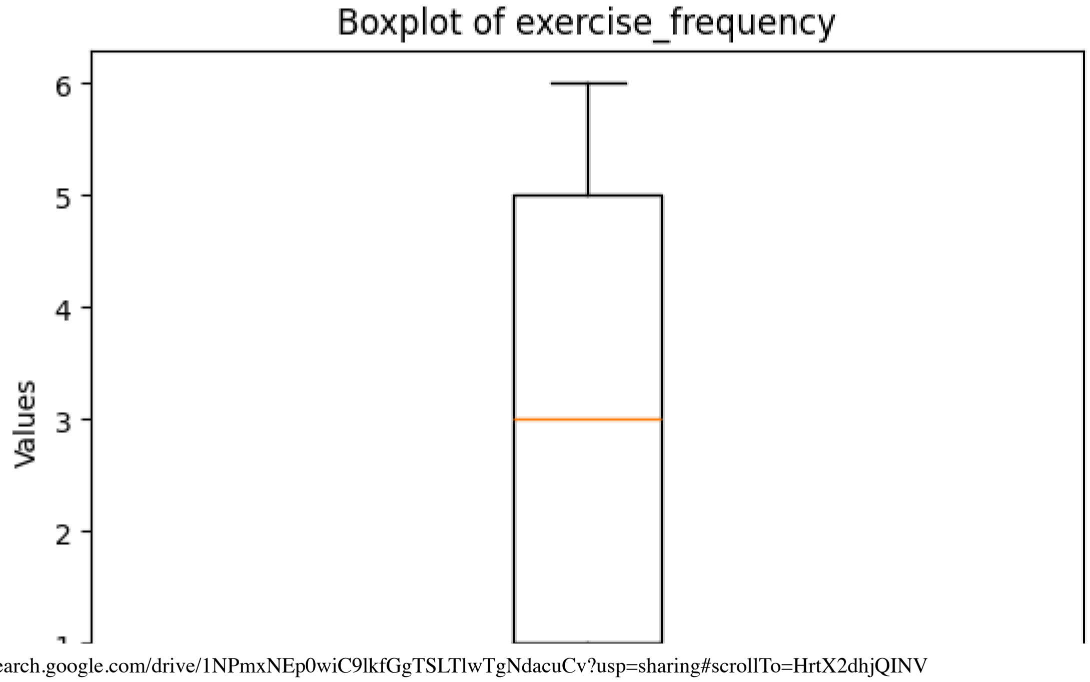

Values
0
1
2
3
4
5
6
1

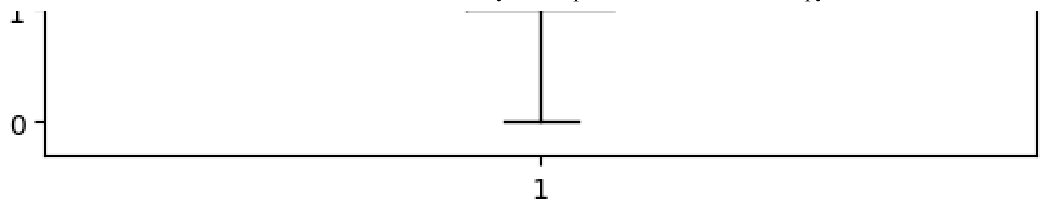

Boxplot of netflix_hours
Values
0
1
2
3
4
5
1

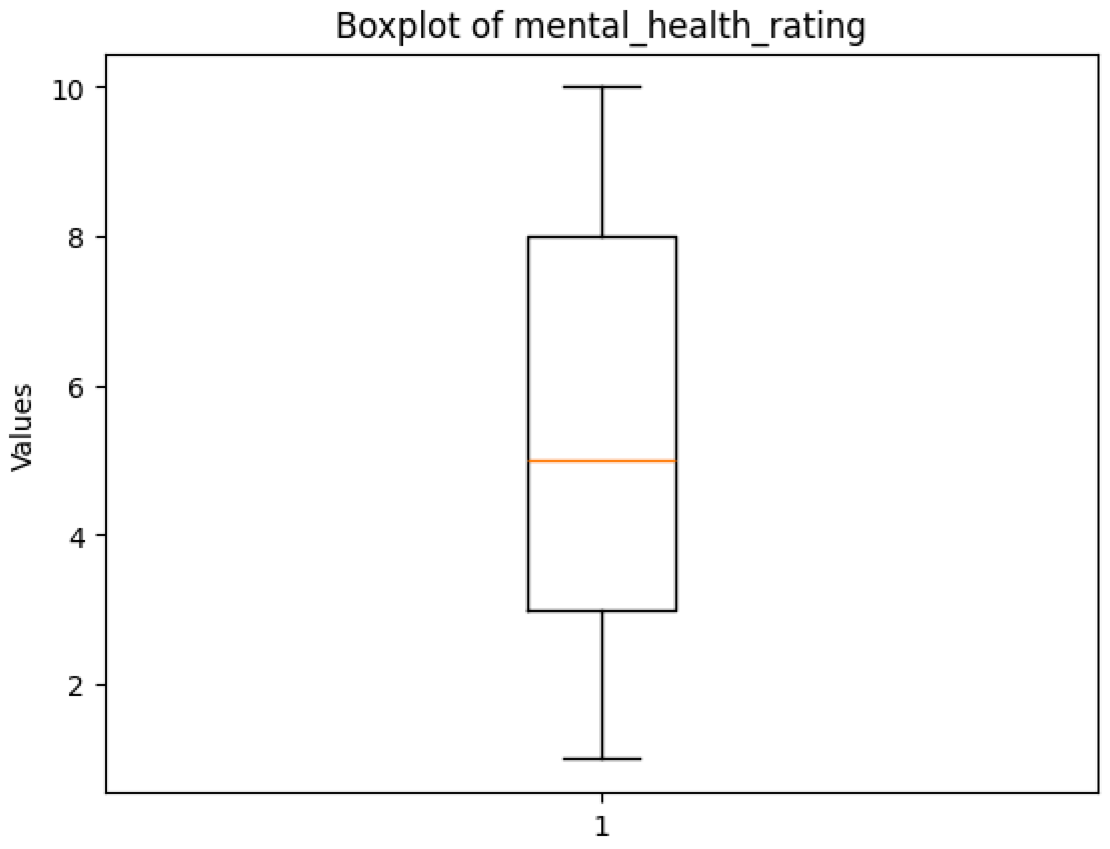

Boxplot of attendance_percentage
100
90
80

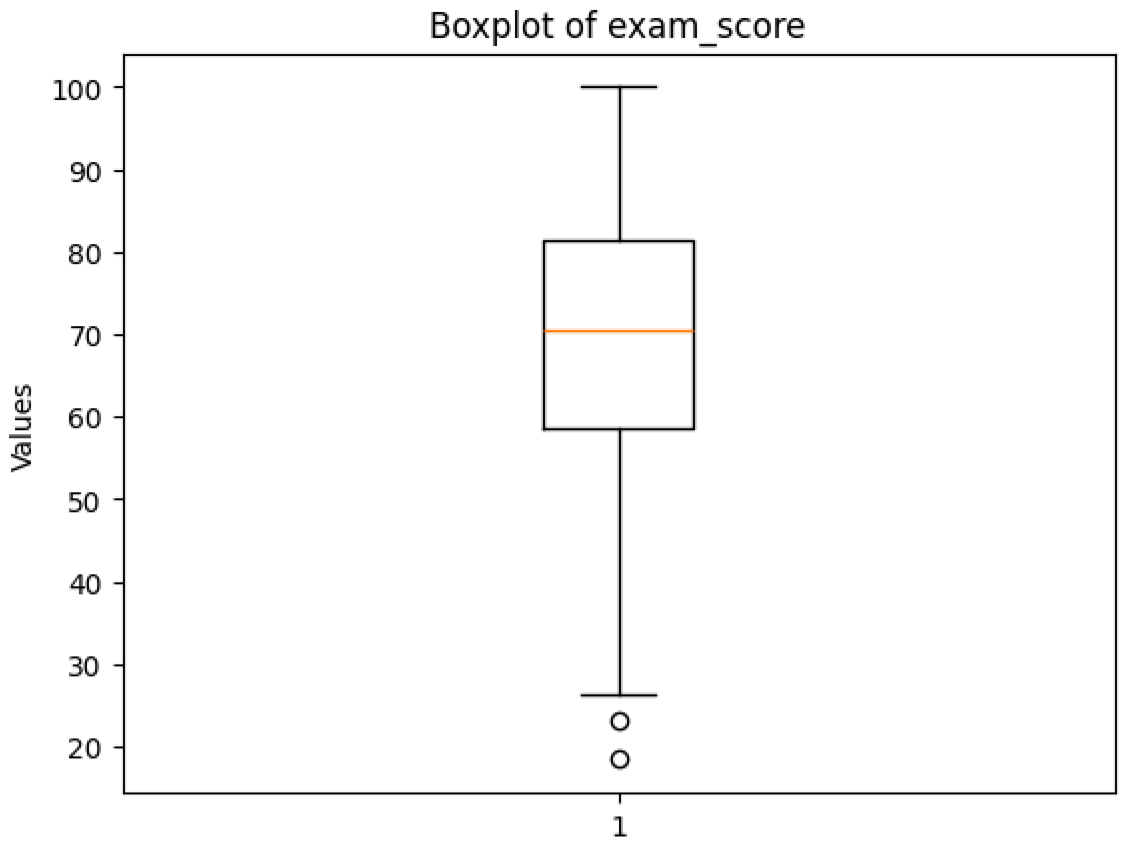

Valu
70
60
1


Boxplot of sleep_hours
Values
10
9
8
7
6
5
4
3
1


Boxplot of exercise_frequency
Values
6
5
4
3
2

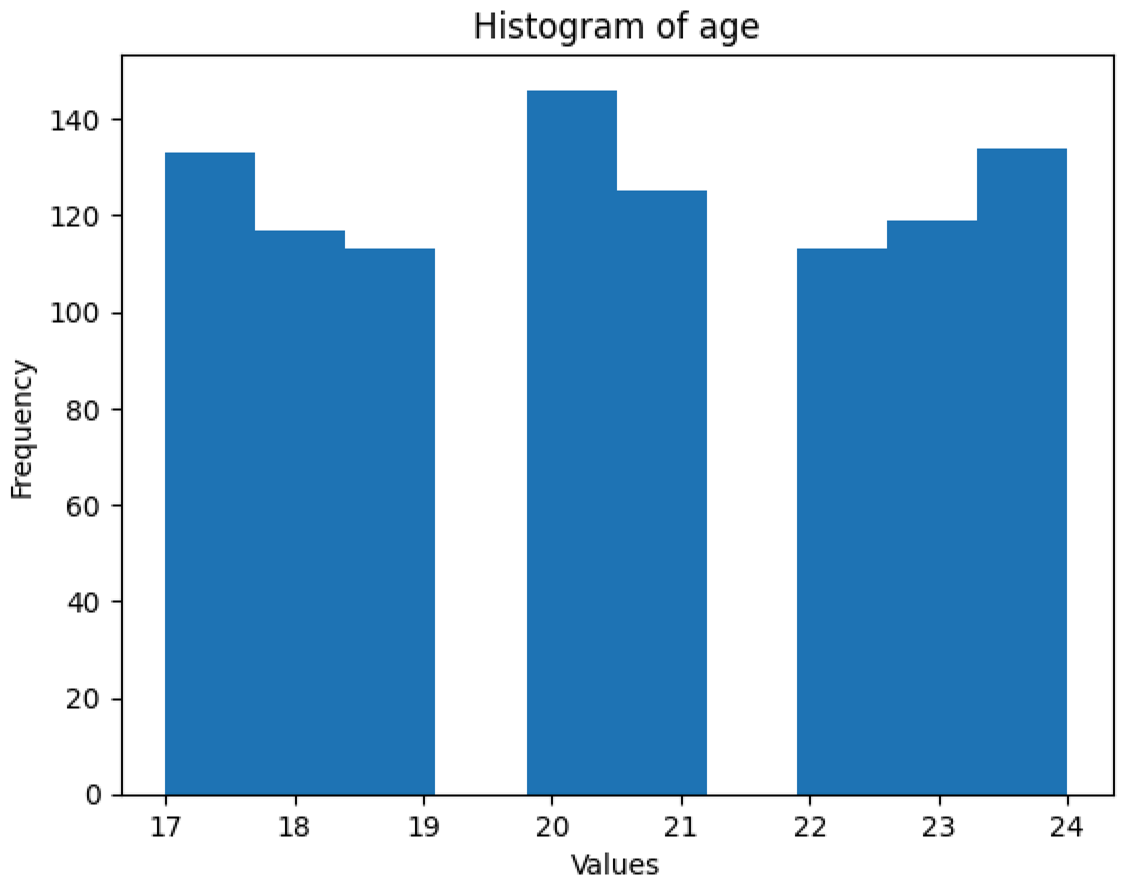

1
0
1

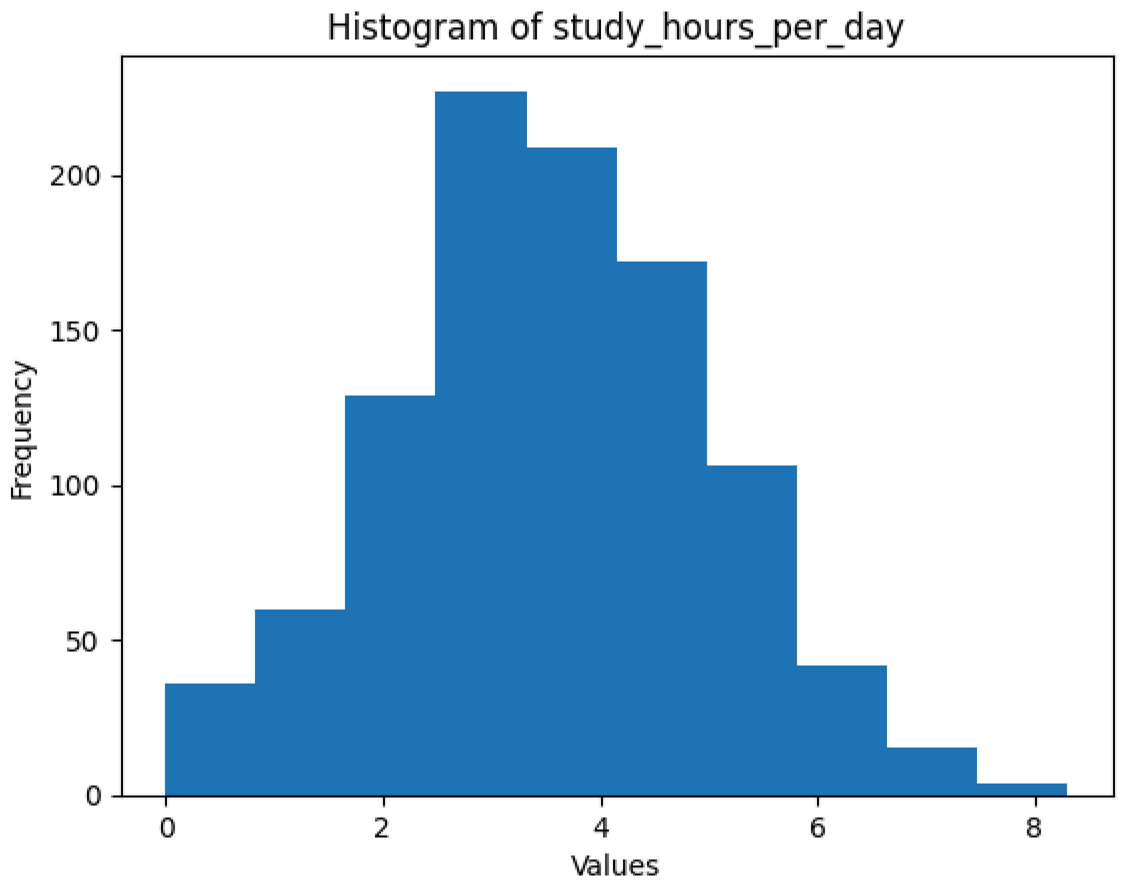

Boxplot of mental_health_rating
Values
10
8
6
4
2
1


Boxplot of exam_score
Values
100
90
80
70
60
50
40
30
20
1

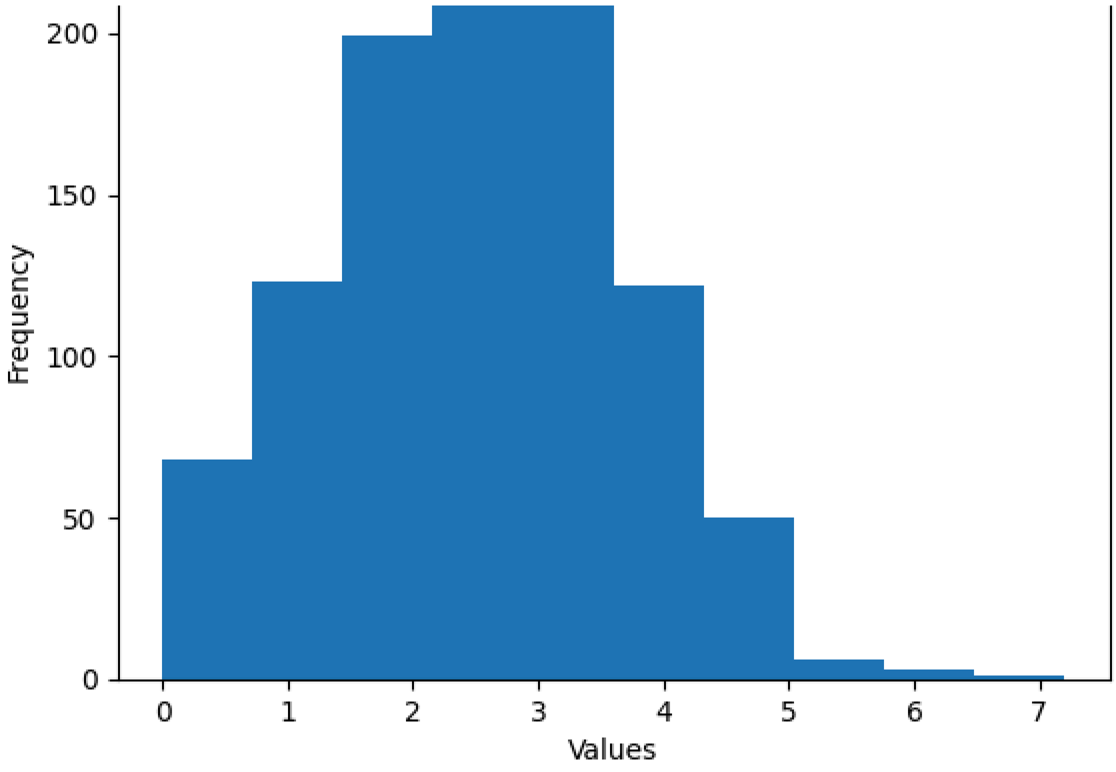
.hits() function

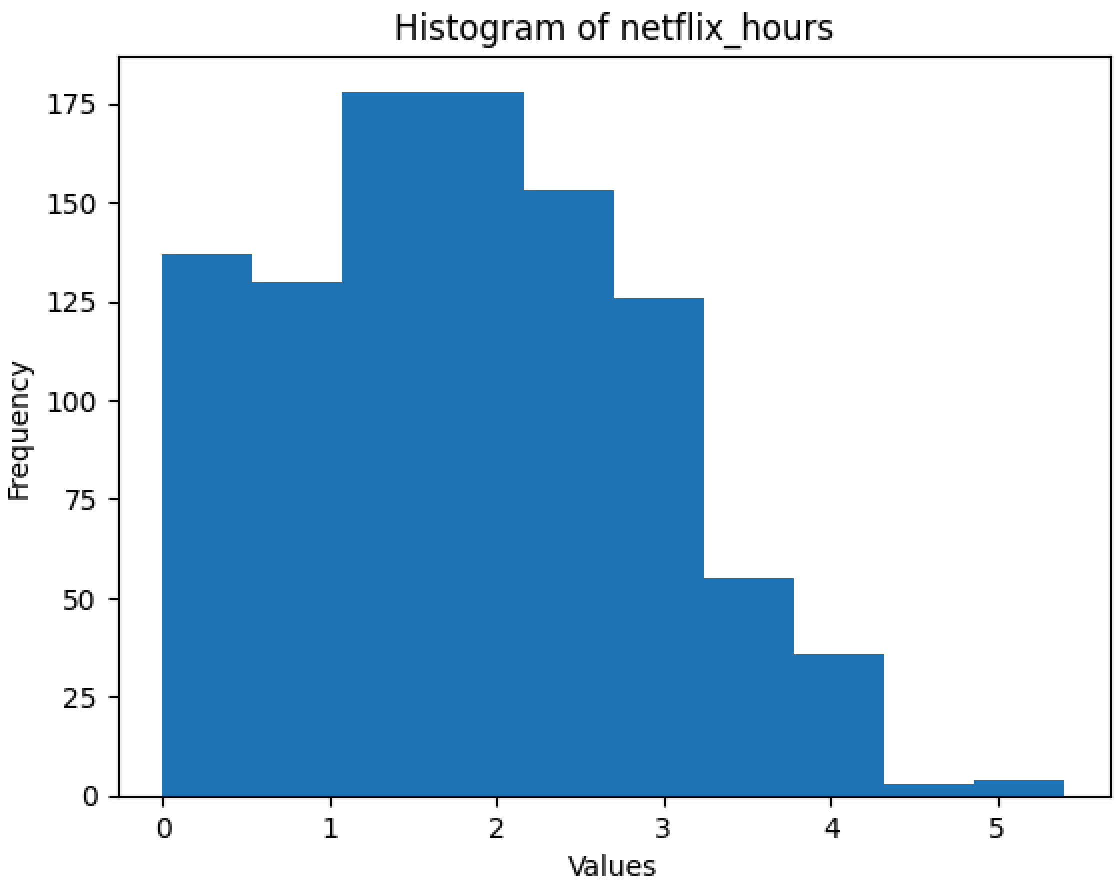
```
for col in col_names:
  plt.hist(numeric_data[col])
  plt.title(f'Histogram of {col}')
  plt.xlabel('Values')
  plt.ylabel('Frequency')
  plt.show()
```

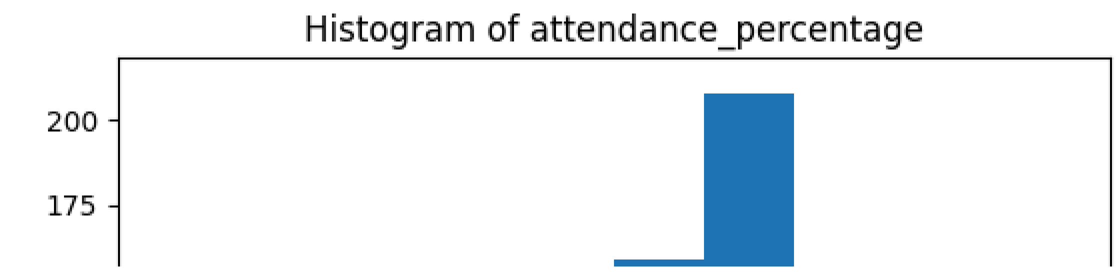

Histogram of age
Frequency
Values
0
20
40
60
80
100
120
140
17
18
19
20
21
22
23
24


Histogram of study_hours_per_day
Frequency
Values
0
50
100
150
200
0
2
4
6
8

Histogram of social_media_hours


Frequency
Values
0
50
100
150
200
0
1
2
3
4
5
6
7


Histogram of netflix_hours
Frequency
Values
0
25
50
75
100
125
150
175
0
1
2
3
4
5


Histogram of attendance_percentage
175
200