**A00573055 Rodrigo Martinez Vallejo** 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)

Haz doble clic (o pulsa Intro) para editar

## Mean, median, and standard deviation

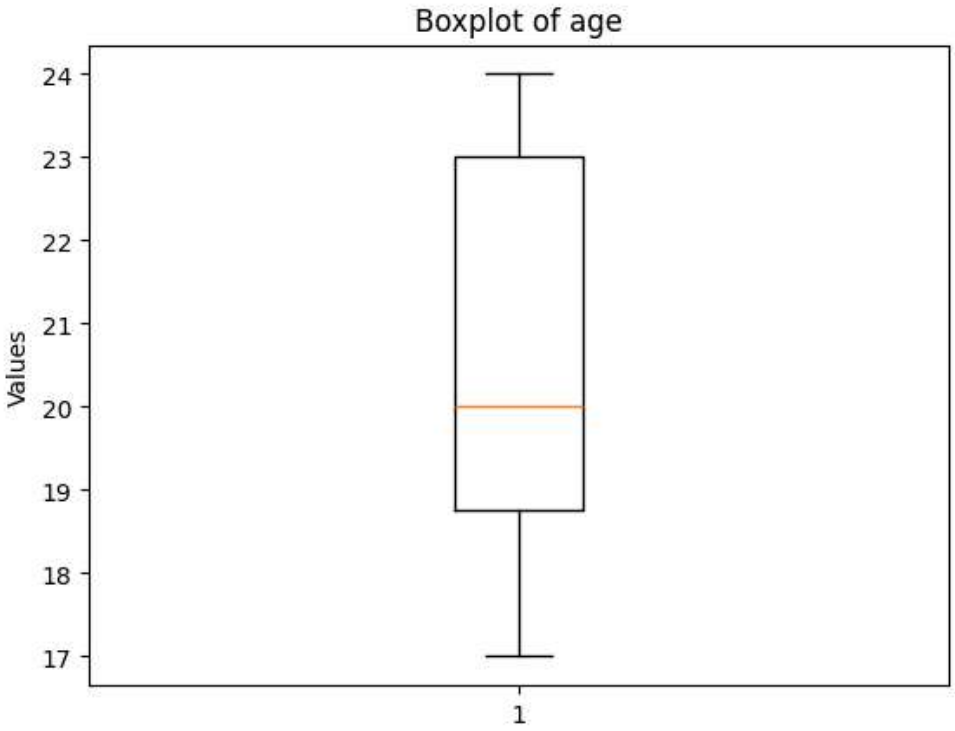
The first thing that I notice from this data is for example how the results given from Mean and Median are almost exactly the same which suggests a very symetrical distribution of the data.

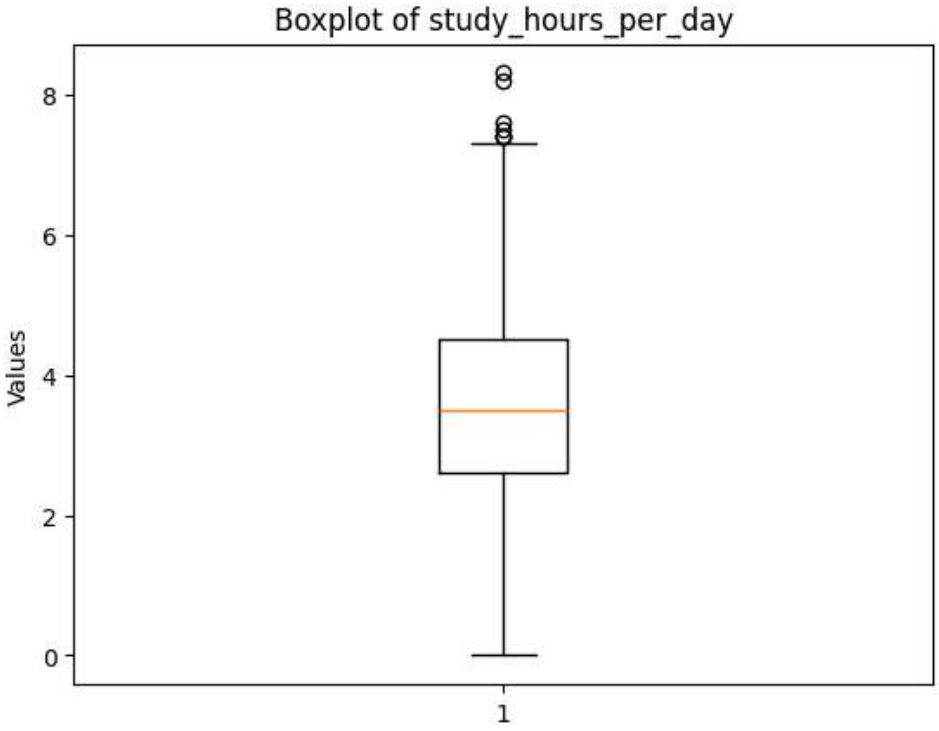
```
print("-->Mean")
print(numeric_data.mean())
print("-->Median")
print(numeric_data.median())
print("-->Standard deviation")
print(numeric_data.std())
```

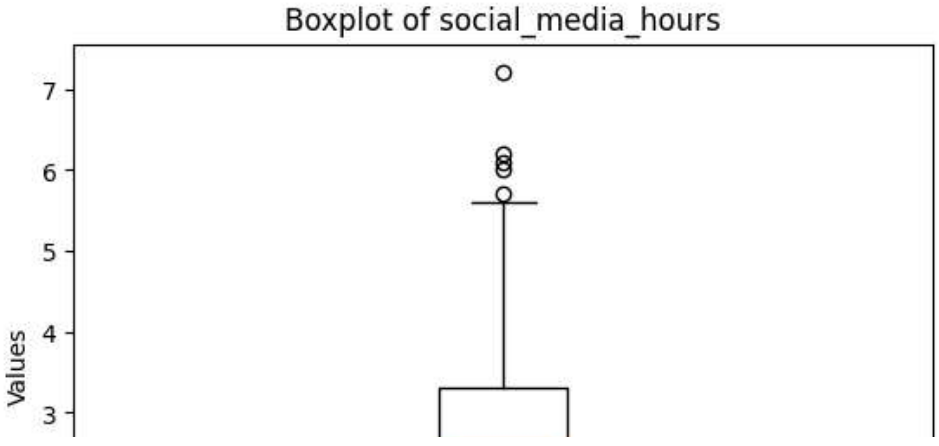
```
-->Mean
age                     20.4980
study_hours_per_day      3.5501
social_media_hours       2.5055
netflix_hours            1.8197
attendance_percentage   84.1317
sleep_hours              6.4701
exercise_frequency       3.0420
mental_health_rating     5.4380
exam_score              69.6015
dtype: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

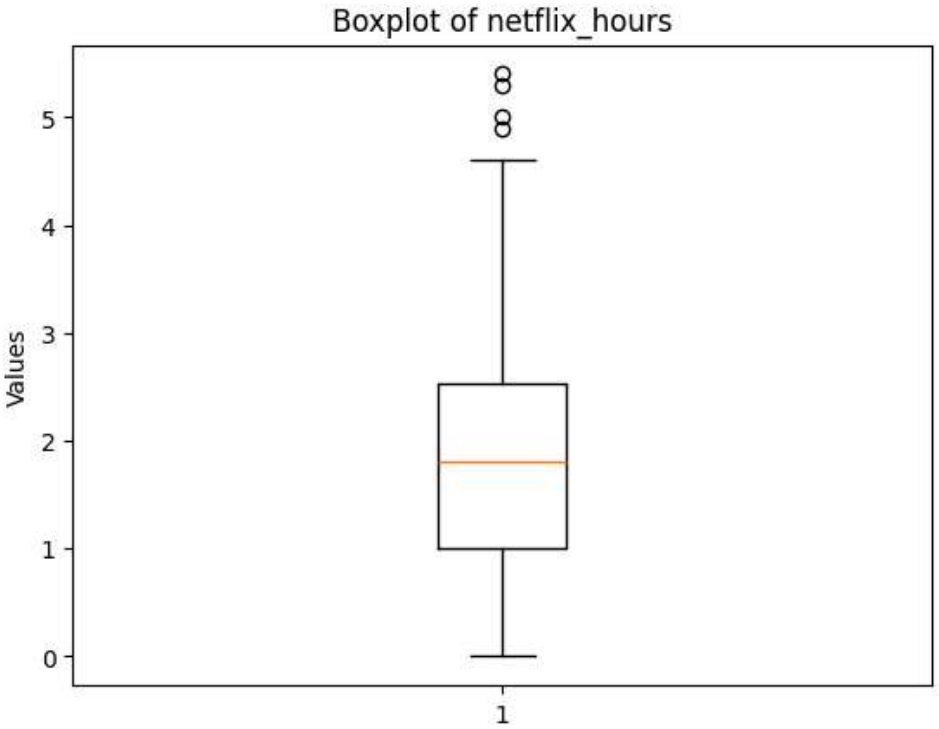
```
col_names = ['age', 'study_hours_per_day', 'social_media_hours', 'netflix_hours',
       'attendance_percentage', 'sleep_hours', 'exercise_frequency',
       'mental_health_rating', 'exam_score']

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

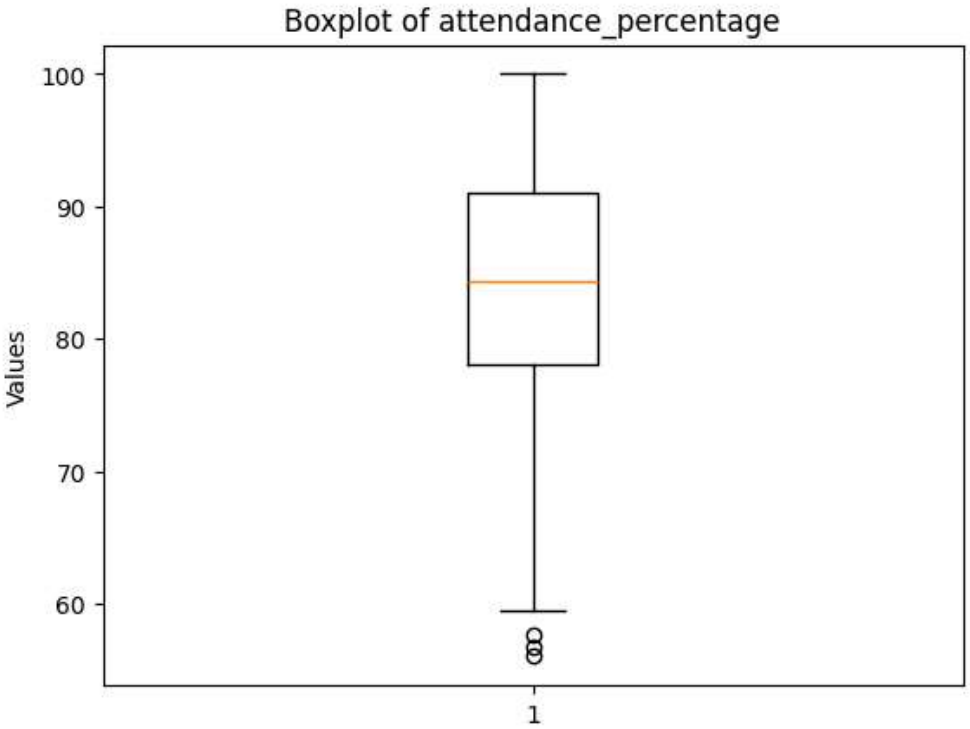

Boxplot of age
Values
24
23
22
21
20
19
18
17
1

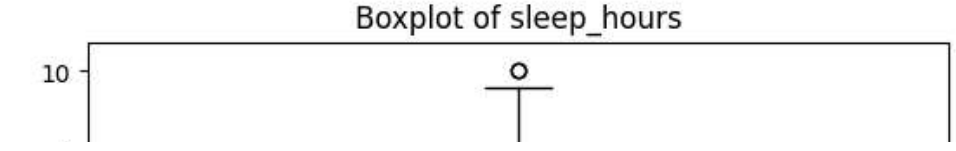

Boxplot of study_hours_per_day
Values
8
6
4
2
0
1

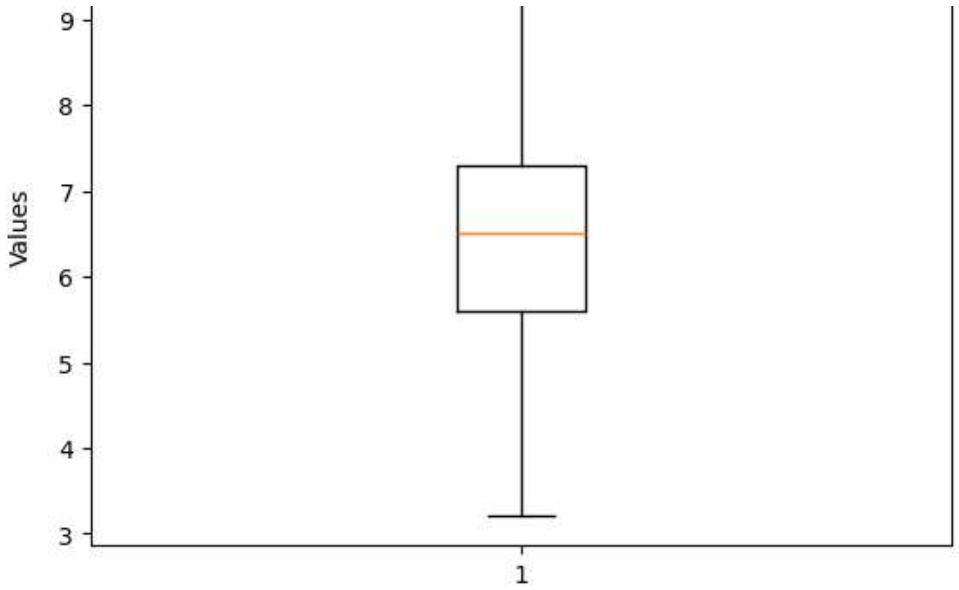

Boxplot of social_media_hours
Values
7
6
5
4
3

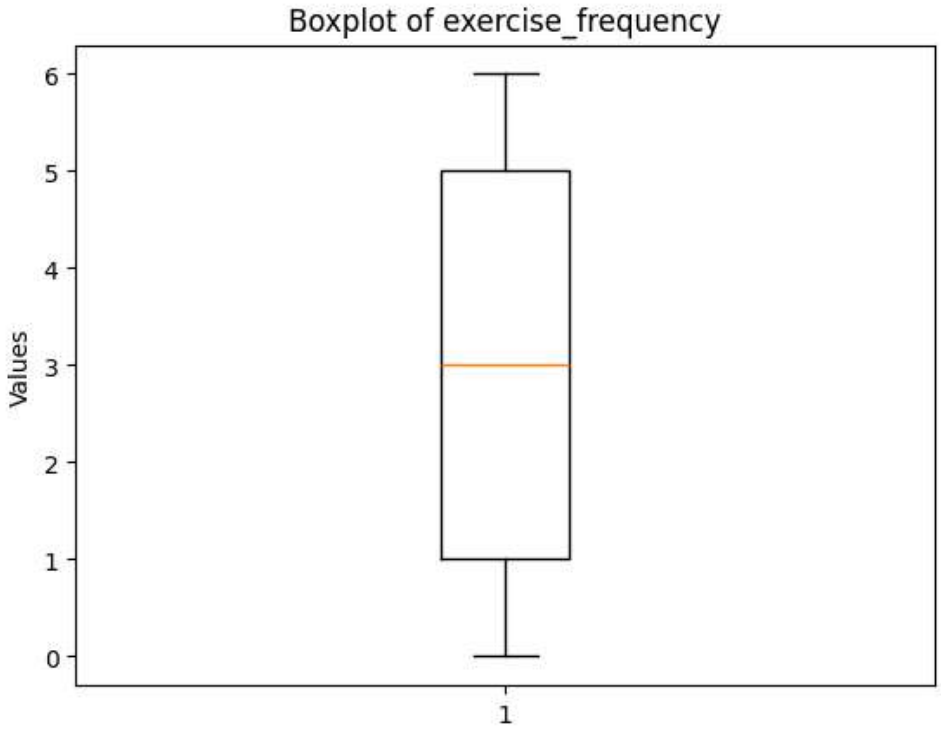

2
1
0
1

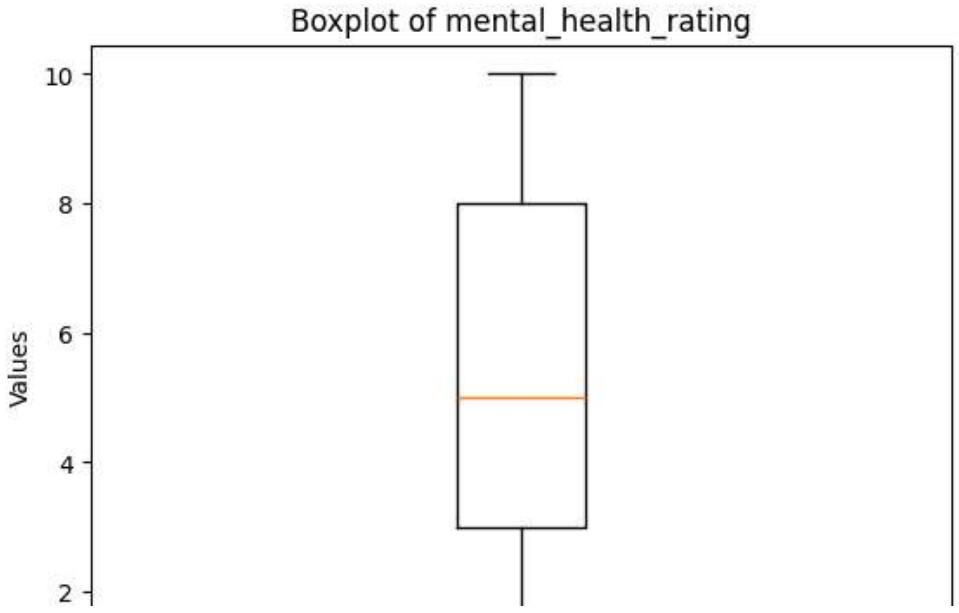

Boxplot of netflix_hours
Values
5
4
3
2
1
0
1

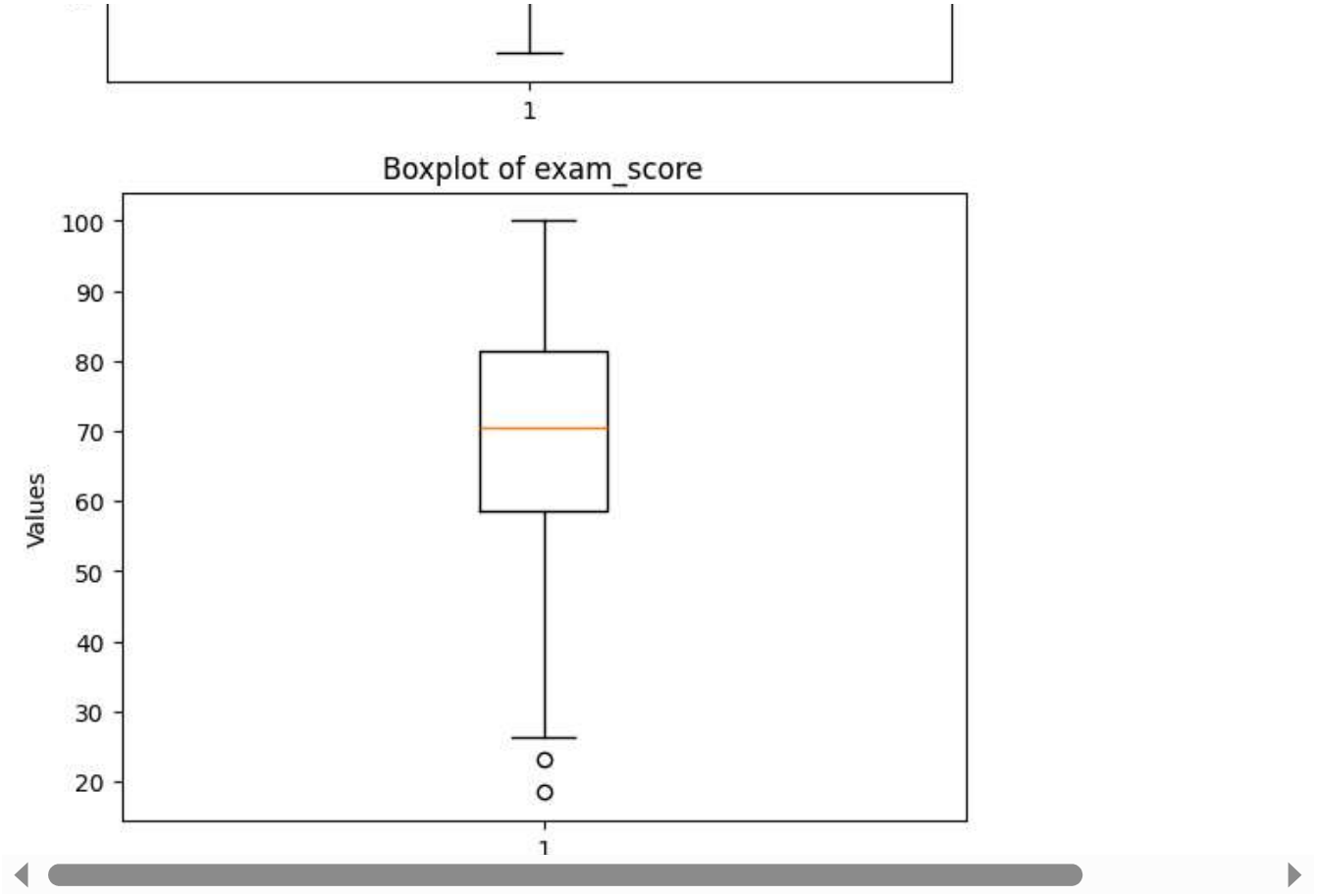

Boxplot of attendance_percentage
Values
100
90
80
70
60
1


Boxplot of sleep_hours
10


Values
9
8
7
6
5
4
3
1

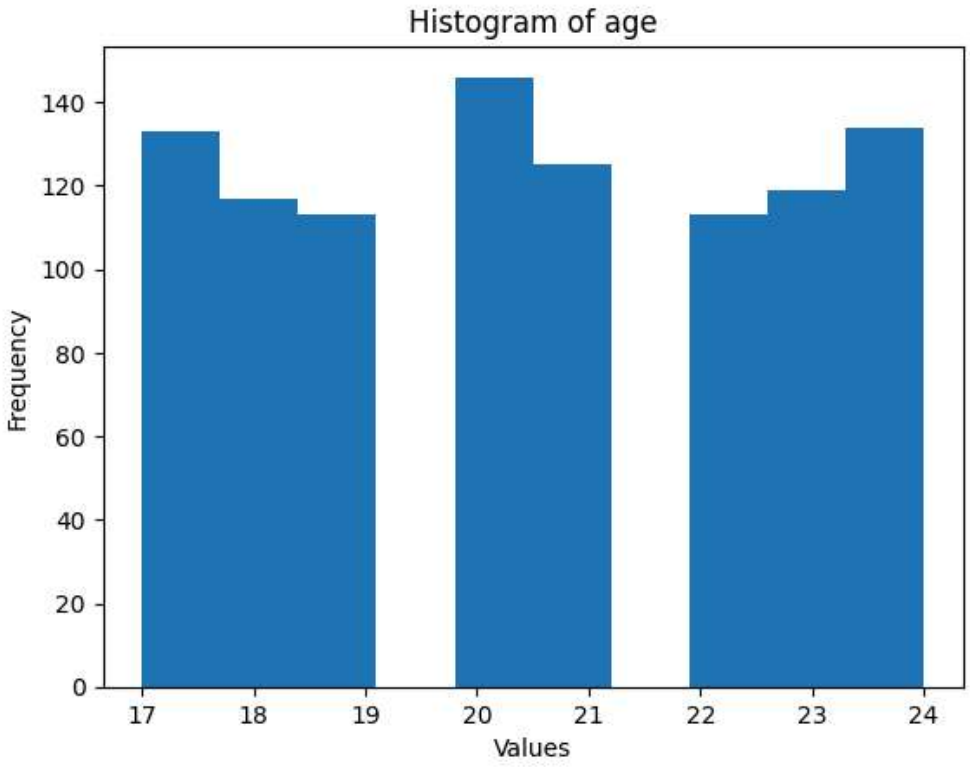

Boxplot of exercise_frequency
Values
6
5
4
3
2
1
0
1

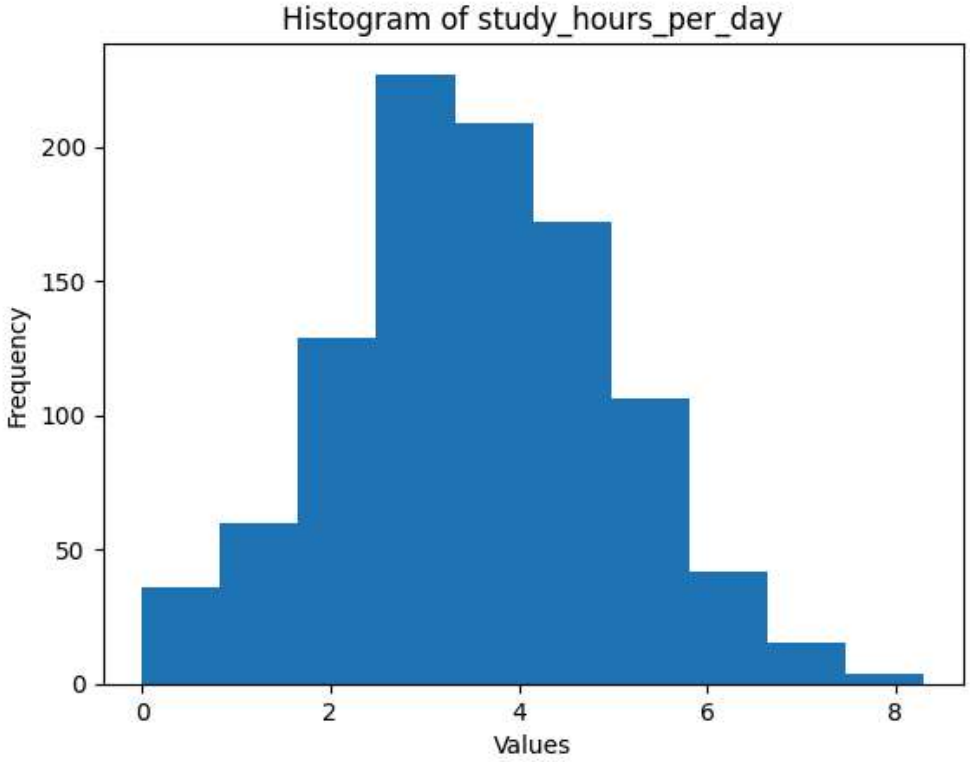

Boxplot of mental_health_rating
Values
10
8
6
4
2

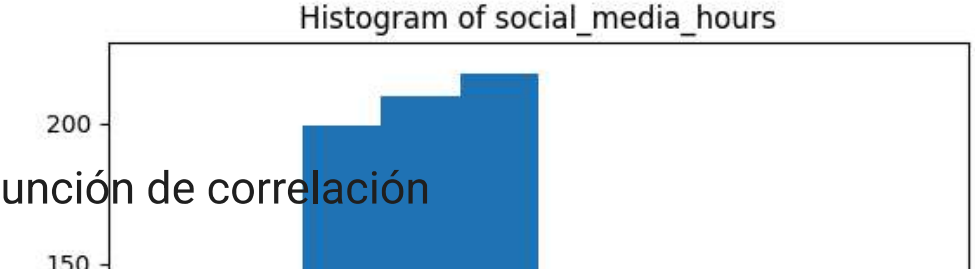

1
Boxplot of exam_score
Values
100
90
80
70
60
50
40
30
20
1

## .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()
```




## Función de correlación

```
num_data_corr = numeric_data.corr()
print(num_data_corr)
```

```
                              age  study_hours_per_day  social_media_hours  \
age                      1.000000             0.003971           -0.009151
study_hours_per_day      0.003971             1.000000            0.020282
social_media_hours      -0.009151             0.020282            1.000000
netflix_hours           -0.001174            -0.031158            0.011477
attendance_percentage   -0.026055             0.026264            0.040479
sleep_hours              0.037482            -0.027757            0.018236
exercise_frequency      -0.003836            -0.028701           -0.037319
mental_health_rating    -0.045101            -0.003768            0.001496
exam_score              -0.008907             0.825419           -0.166733

                        netflix_hours  attendance_percentage  sleep_hours  \
age                         -0.001174              -0.026055     0.037482
study_hours_per_day         -0.031158               0.026264    -0.027757
social_media_hours           0.011477               0.040479     0.018236
netflix_hours                1.000000              -0.002092    -0.000935
attendance_percentage       -0.002092               1.000000     0.013756
sleep_hours                 -0.000935               0.013756     1.000000
exercise_frequency          -0.006448              -0.007857     0.019769
mental_health_rating         0.008034              -0.018745    -0.006508
exam_score                  -0.171779               0.089836     0.121683

                        exercise_frequency  mental_health_rating  exam_score
age                              -0.003836             -0.045101   -0.008907
```