HOMEWORK

Question1

Suppose you have a set of data $x_1 , x_2 , ..., x_n$, show that the the sum of deviations(not squared) is zero.

1. suppose I have a set of data $x_1 , x_2 , ..., x_n$ and I will show that the the sum of deviations(not squared) is zero.
2. $\overline{x} = \frac{x_1 + x_2 + ... + x_n}{n}$
3. SUM = $\sum_{i=1}^{n}(x_i - \overline{x})$ = $\sum_{i=1}^{n}(x_i - \frac{x_1 + x_2 + ... + x_n}{n})$ = $\sum_{i=1}^{n}x_i - n\frac{x_1 + x_2 + ... + x_n}{n}$ = $\sum_{i=1}^nx_i -(x_1 + x_2 + ... + x_n)$ = 0
4. Therefore, the sum of deviations is zero.

Question2

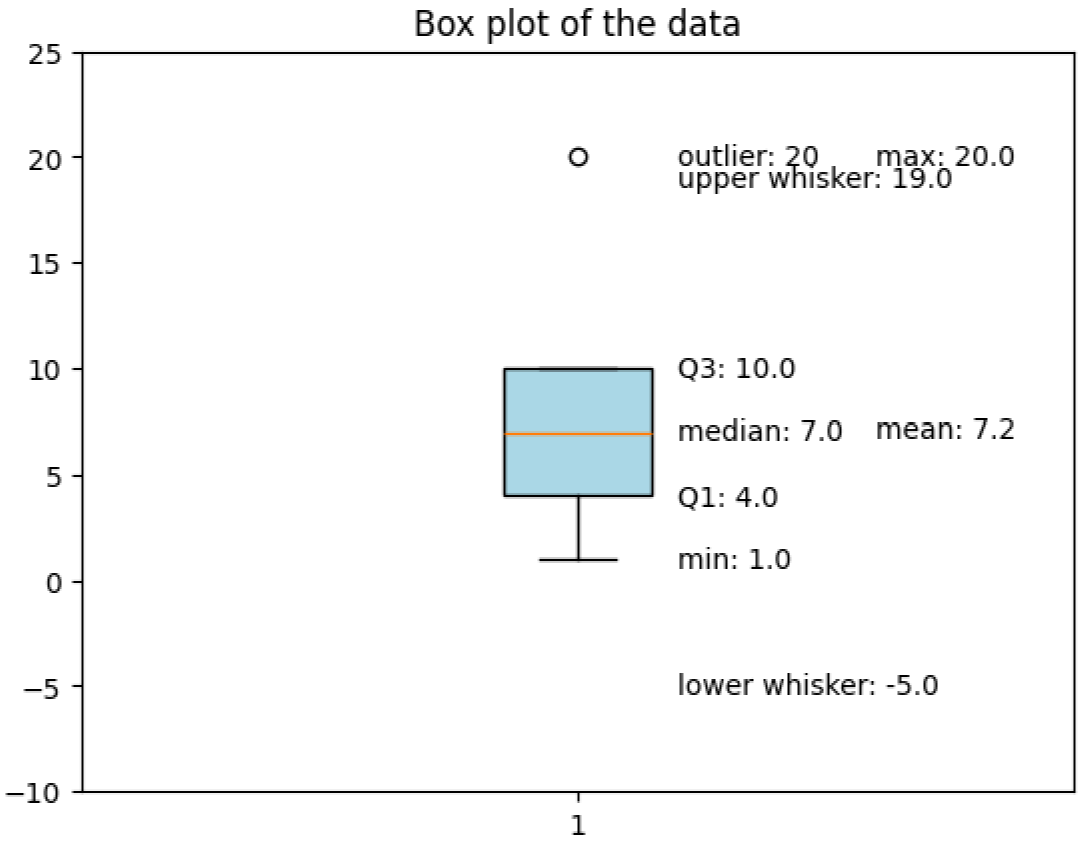
For the following set of data
2, 3, 7, 7, 10, 9, 7, 10, 6, 10, 3, 10, 20, 3, 10, 8, 5, 1, 5
provide:
a) Tukey’s 5-number summaries
b) Interquartile range
c) A box plot (mark the important values on the plot.)

In [1]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data_original = np.array([2, 3, 7, 7, 10, 9, 7, 10, 6, 10, 3, 10, 20, 3, 10, 8
data_sorted = np.sort(data_original)
print("sorted data:", data_sorted)

n = len(data_sorted)
print("number of data points:", n)

minimum = data_sorted[0]
maximum = data_sorted[-1]
mean = np.mean(data_sorted)
print("mean:", mean)
print("minimum:", minimum)
print("maximum:", maximum)
```

```
if n % 2 == 0:
    median = (data_sorted[n//2 - 1] + data_sorted[n//2]) / 2
else:
    median = data_sorted[n//2]
print("median:", median)

if n % 2 == 0:
    Q1 = np.median(data_sorted[:n//2])
    Q3 = np.median(data_sorted[n//2:])
else:
    Q1 = np.median(data_sorted[:(n + 1)//2])
    Q3 = np.median(data_sorted[(n +1)//2+1:])
print("Q1:", Q1)
print("Q3:", Q3)

IQR = Q3 - Q1
print("IQR:", IQR)

lower_whisker = Q1 - 1.5 * IQR
upper_whisker = Q3 + 1.5 * IQR
print("lower whisker:", lower_whisker)
print("upper whisker:", upper_whisker)

print("boxplot of the data:")
fig, ax = plt.subplots()
box = ax.boxplot(data_sorted, patch_artist=True)  # patch_artist=True 便于给箱体

for patch in box['boxes']:
    patch.set_facecolor('lightblue')

ax.text(1.1, median, f"median: {median:.1f}", va="center")
ax.text(1.3, mean, f"mean: {mean:.1f}", va="center")
ax.text(1.1, minimum, f"min: {minimum:.1f}", va="center")
ax.text(1.3, maximum, f"max: {maximum:.1f}", va="center")
ax.text(1.1, Q1, f"Q1: {Q1:.1f}", va="center")
ax.text(1.1, Q3, f"Q3: {Q3:.1f}", va="center")
ax.text(1.1, lower_whisker, f"lower whisker: {lower_whisker:.1f}", va="center"
ax.text(1.1, upper_whisker, f"upper whisker: {upper_whisker:.1f}", va="center"

outliers = [f"outlier: {x}" for x in data_sorted if x < lower_whisker or x > u
for i, outlier in enumerate(outliers):
    ax.text(1.1, data_sorted[np.where(data_sorted == float(outlier.split(": ")

ax.set_ylim(-10, 25)
plt.title("Box plot of the data")
plt.show()
```

```
sorted data: [ 1  2  3  3  3  5  5  6  7  7  7  8  9 10 10 10 10 10 20]
number of data points: 19
mean: 7.157894736842105
minimum: 1
maximum: 20
median: 7
Q1: 4.0
Q3: 10.0
IQR: 6.0
lower whisker: -5.0
upper whisker: 19.0
boxplot of the data:
```



Question 3

Python problem: Looking at the dataset Housing.csv. The variables are

- medv: median home price in different neighborhoods
- crim: per capita crime rate
- rm: average number of rooms per dwelling
- zn: proportion of large lots (zoned for > 25, 000 feet)
- river: whether a home is near a river (0: No, 1: yes)
- ptratio: pupil-teacher ratio by town

Questions:

a) Show first 8 rows of the dataset.

b) Given the median and mean of medv.

c) Calculate 1st and 3rd quantile of crim.

d) Plot histogram of crime.

e) On the same plot, draw histograms of medv near a river and not near a river, respectively.

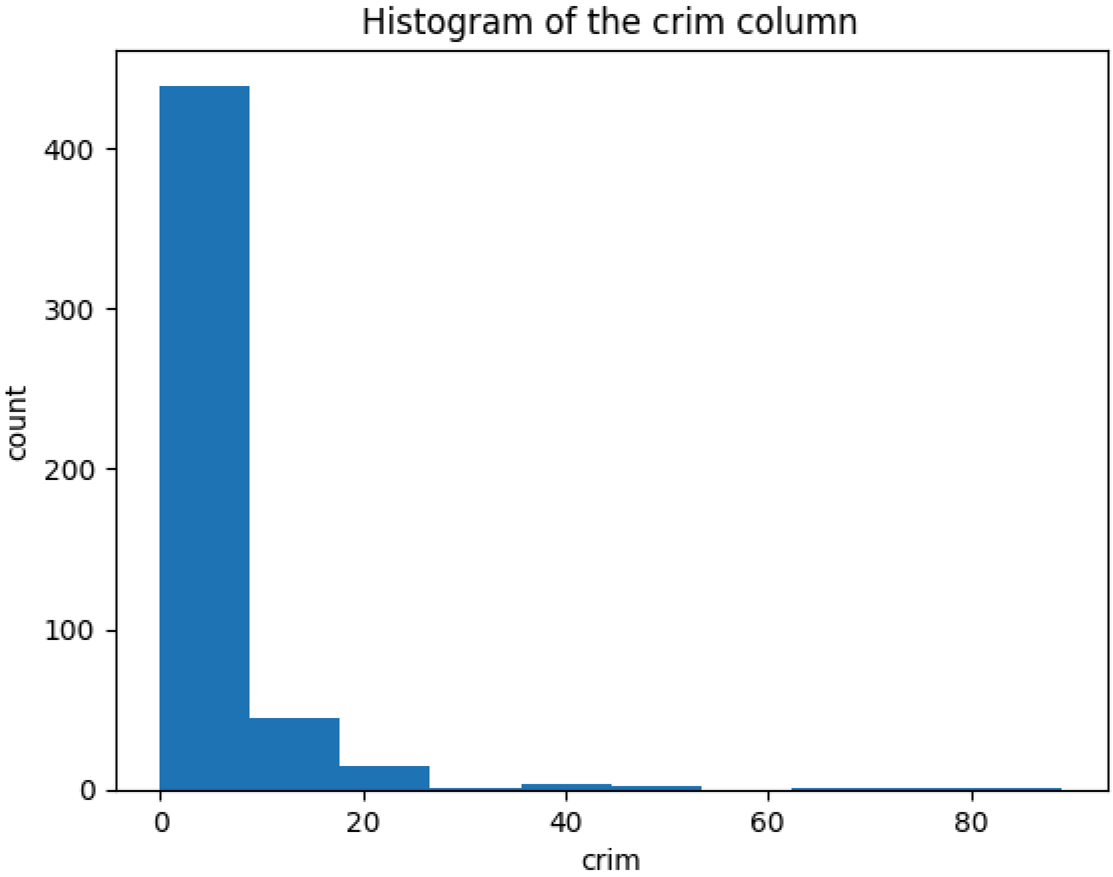
f) Provide correlation matrix and the corresponding heatmap for the dataset.

g) Fit a density curve using 10 bins for medv.

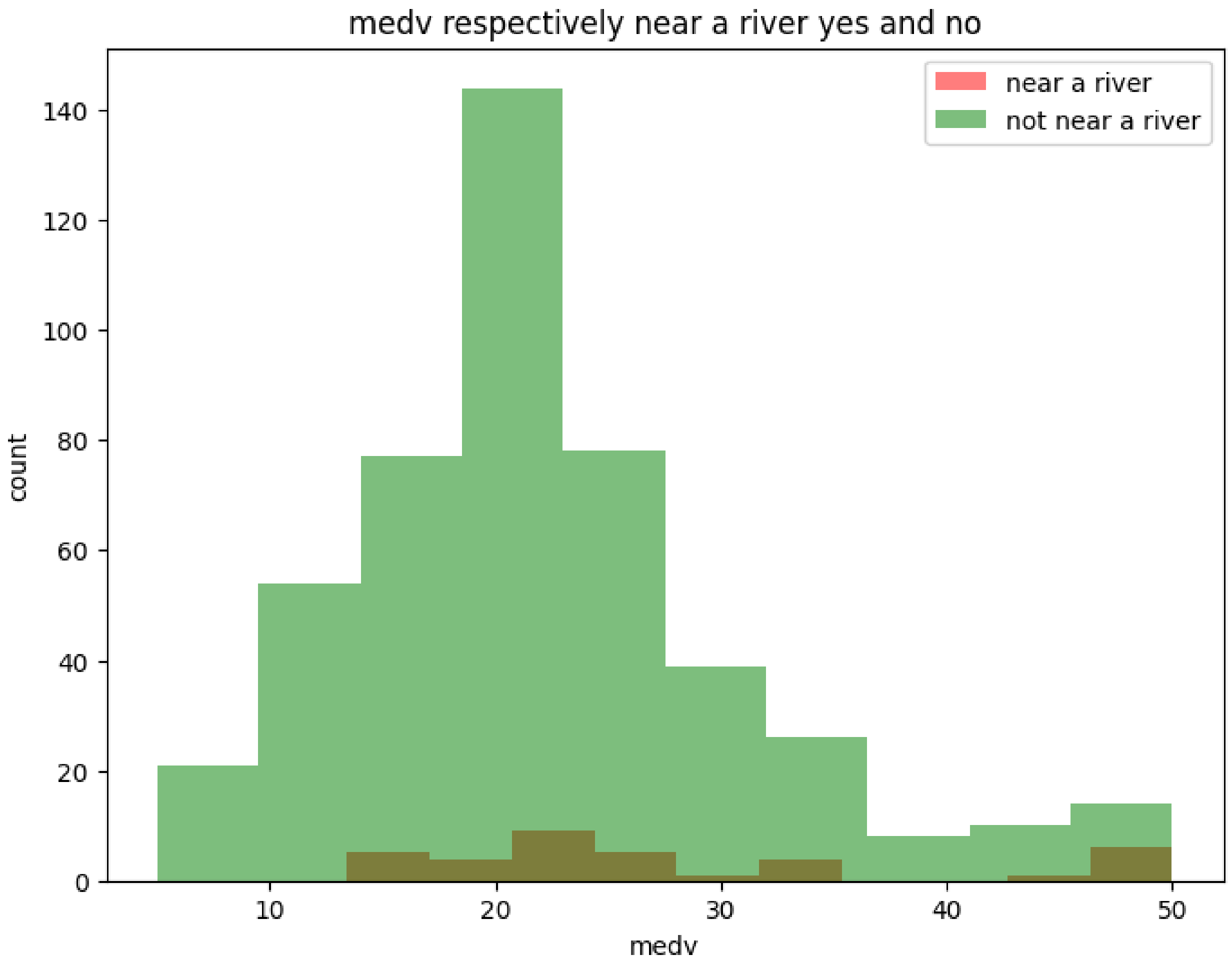
In [ ]:

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

# Read the data from the csv file
data_Housing = pd.read_csv('Housing.csv')
try:
    data_Housing = pd.read_csv('Housing.csv')
    print("data_Housing loaded successfully.")
except FileNotFoundError:
    print("Error: 'Housing.csv' file not found.")

# show the shape of the data
print("shape of the data:", data_Housing.shape)

# show the first 8 rows of the data
print("first 8 rows of the data:\n", data_Housing.head(8))

# show the nedian and mean of the 'medv' column
medv_median = data_Housing['medv'].median()
medv_mean = data_Housing['medv'].mean()
print(f"median of the 'medv' column: {medv_median:.2f}")
print(f"mean of the 'medv' column: {medv_mean:.2f}")

# show the 1sd, 3sd quantiles of the 'crim' column
data_crim = data_Housing['crim']
data_crim_sorted = np.sort(data_crim)
n = len(data_crim_sorted)
if n % 2 == 0:
    Q1_crim = np.median(data_crim_sorted[:n//2])
    Q3_crim = np.median(data_crim_sorted[n//2:])
else:
    Q1_crim = np.median(data_crim_sorted[:n//2])
    Q3_crim = np.median(data_crim_sorted[(n//2)+1:])
print(f"Q1 of the 'crim' column: {Q1_crim:.2f}")
print(f"Q3 of the 'crim' column: {Q3_crim:.2f}")

# show the histogram of the 'crim' column
print("\n the histogram of the 'crim' column:")
plt.hist(data_crim, bins=10)
plt.xlabel('crim')
plt.ylabel('count')
plt.title('Histogram of the crim column')
plt.show()

#on the same plot, draw histograms of medv near a river and not near a river,
print("\n the histogram of the medv column, near a river and not near a river,
plt.figure(figsize=(8, 6))
river_yes = data_Housing[data_Housing['river'] == 1]['medv']
river_no = data_Housing[data_Housing['river'] == 0]['medv']
plt.hist(river_yes, bins=10, alpha=0.5, label='near a river', color='red')
plt.hist(river_no, bins=10, alpha=0.5, label='not near a river', color='green'
plt.xlabel('medv')
plt.ylabel('count')
plt.title('medv respectively near a river yes and no')
plt.legend()
plt.show()

#Provide correlation matrix and the corresponding heatmap for the dataset
corr_matrix = data_Housing.corr()
print("\n Correlation matrix:\n", corr_matrix)
plt.figure(figsize=(8, 6))
seaborn.heatmap(corr_matrix, annot=True)
plt.title('Heatmap of the correlation matrix')
plt.show()

#the density plot of the 'medv' column(10bins)
print("\n the density plot of the 'medv' column:")
plt.figure(figsize=(8, 6))
sns.histplot(data_Housing['medv'], bins=10, kde=True, color='red')
plt.xlabel('medv')
plt.ylabel('density')
plt.title('Density plot of the medv column')
plt.show()
```

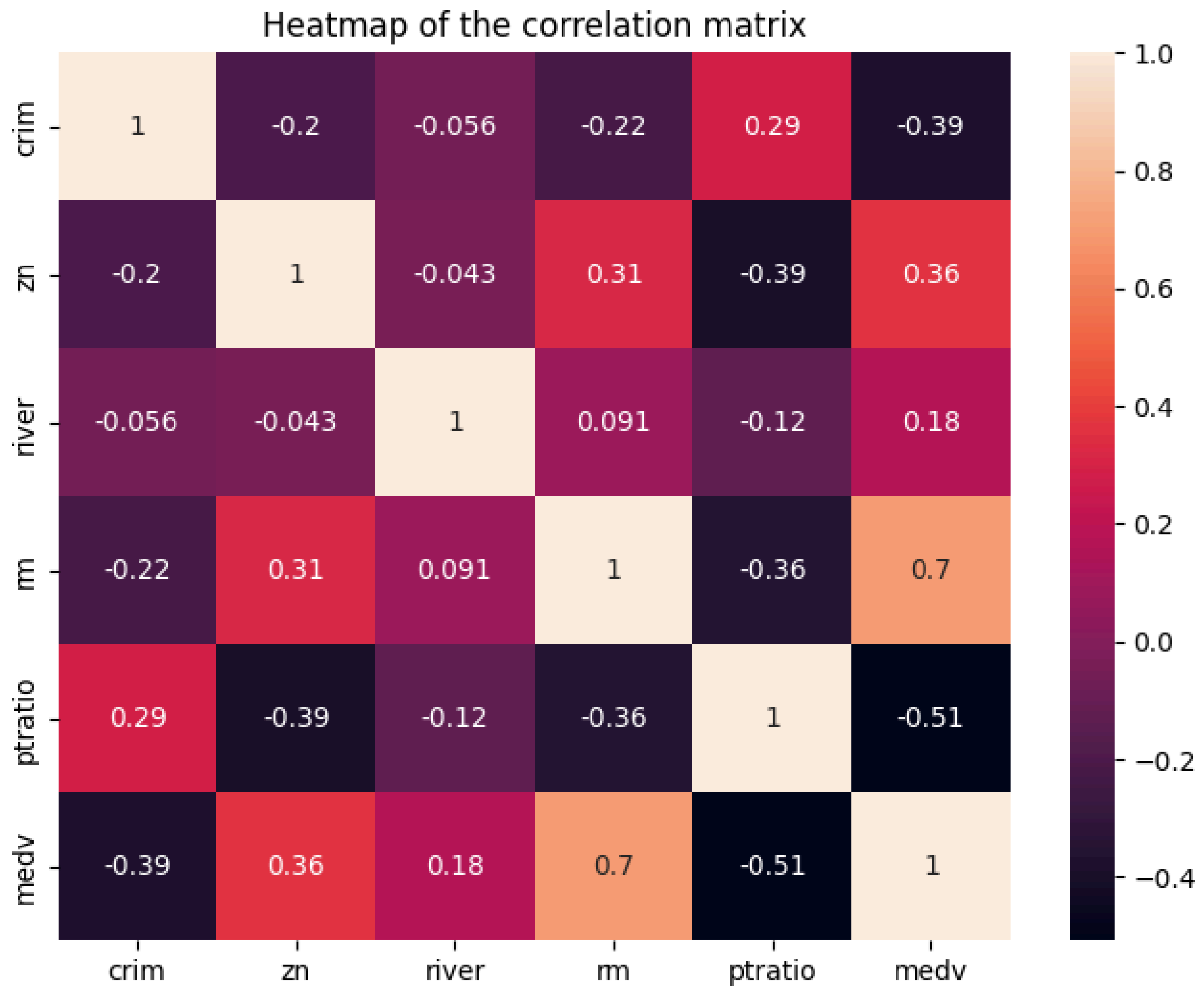
```
data_Housing loaded successfully.
shape of the data: (506, 6)
first 8 rows of the data:
       crim    zn  river     rm  ptratio  medv
0  0.00632  18.0      0  6.575     15.3  24.0
1  0.02731   0.0      0  6.421     17.8  21.6
2  0.02729   0.0      0  7.185     17.8  34.7
3  0.03237   0.0      0  6.998     18.7  33.4
4  0.06905   0.0      0  7.147     18.7  36.2
5  0.02985   0.0      0  6.430     18.7  28.7
6  0.08829  12.5      0  6.012     15.2  22.9
7  0.14455  12.5      0  6.172     15.2  27.1
median of the 'medv' column: 21.20
mean of the 'medv' column: 22.53
Q1 of the 'crim' column: 0.08
Q3 of the 'crim' column: 3.68

 the histogram of the 'crim' column:
```



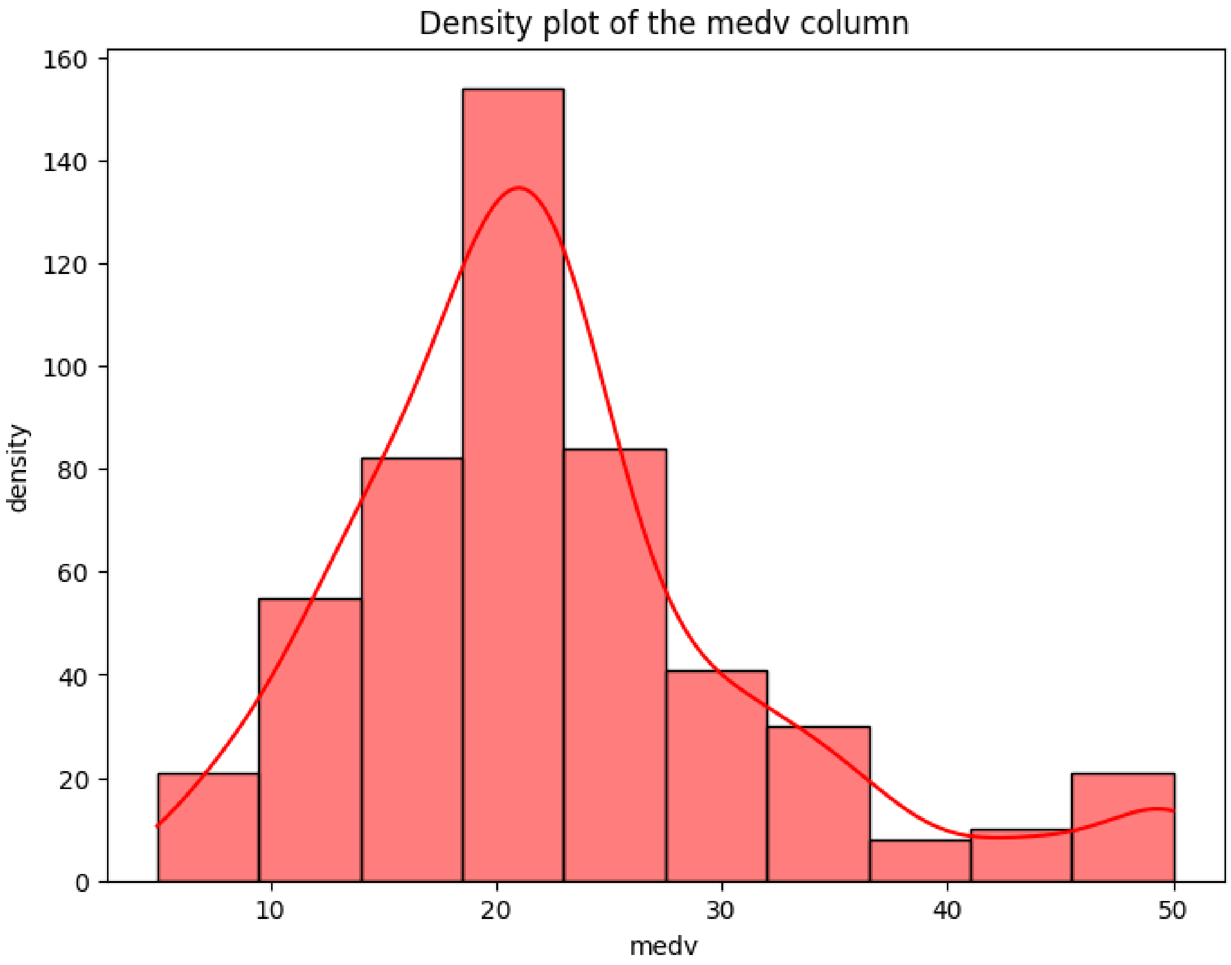
```
 the histogram of the medv column, near a river and not near a river, respectiv
ely:
```



```
 Correlation matrix:
              crim        zn     river        rm   ptratio      medv
crim      1.000000 -0.200469 -0.055892 -0.219247  0.289946 -0.388305
zn       -0.200469  1.000000 -0.042697  0.311991 -0.391679  0.360445
river    -0.055892 -0.042697  1.000000  0.091251 -0.121515  0.175260
rm       -0.219247  0.311991  0.091251  1.000000 -0.355501  0.695360
ptratio   0.289946 -0.391679 -0.121515 -0.355501  1.000000 -0.507787
medv     -0.388305  0.360445  0.175260  0.695360 -0.507787  1.000000
```



```
the density plot of the 'medv' column:
```


Density plot of the medv column
density
0
20
40
60
80
100
120
140
160
medv
10
20
30
40
50