AED	Laboratory work 1
Made by |	Marina PETICÎ
Group |	IS - 221 M
Marked by |	Corina BEȘLIU

The dataset for this laboratory work contains data on prices of homes in Boston and possible predictor variables. The tasks for the laboratory work are as follows:

crim	per capita crime rate by town
chas	Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
rm	average number of rooms per dwelling
age	proportion of owner-occupied units built prior to 1940
dis	weighted mean of distances to five Boston employment centers
tax	full-value property-tax rate per \$10,000
lstat	lower status of the population (percent)
medv	median value of owner-occupied homes in \$1000s.

TASK 1 | *Load and print the first 20 observations of the dataset. Report if you see any unusual values.*

In order to load the given dataset, the Python library for data analysis - *pandas* was used. The first 20 rows of the given dataset were shown with the help of the Python *head()* function.

Python code:

```
import pandas as pd

housingdata = pd.read_csv('HousingData.csv')

observations = housingdata.head(20)
print("\nFirst 20 rows:")
print (observations)
```

Execution:

```
First 20 rows:
       CRIM  CHAS     RM    AGE     DIS  TAX  LSTAT  MEDV
0   0.00632   0.0  6.575   65.2  4.0900  296   4.98  24.0
1   0.02731   0.0  6.421   78.9  4.9671  242   9.14  21.6
2   0.02729   0.0  7.185   61.1  4.9671  242   4.03  34.7
3   0.03237   0.0  6.998   45.8  6.0622  222   2.94  33.4
4   0.06905   0.0  7.147   54.2  6.0622  222    NaN  36.2
5   0.02985   0.0  6.430   58.7  6.0622  222   5.21  28.7
6   0.08829   NaN  6.012   66.6  5.5605  311  12.43  22.9
7   0.14455   0.0  6.172   96.1  5.9505  311  19.15  27.1
8   0.21124   0.0  5.631  100.0  6.0821  311  29.93  16.5
9   0.17004   NaN  6.004   85.9  6.5921  311  17.10  18.9
10  0.22489   0.0  6.377   94.3  6.3467  311  20.45  15.0
11  0.11747   0.0  6.009   82.9  6.2267  311  13.27  18.9
12  0.09378   0.0  5.889   39.0  5.4509  311  15.71  21.7
13  0.62976   0.0  5.949   61.8  4.7075  307   8.26  20.4
14  0.63796   NaN  6.096   84.5  4.4619  307  10.26  18.2
15  0.62739   0.0  5.834   56.5  4.4986  307   8.47  19.9
16  1.05393   0.0  5.935   29.3  4.4986  307   6.58  23.1
17  0.78420   0.0  5.990   81.7  4.2579  307  14.67  17.5
18  0.80271   0.0  5.456   36.6  3.7965  307  11.69  20.2
19  0.72580   0.0  5.727   69.5  3.7965  307  11.28  18.2
```

After the analysis of the uploaded dataset, the next observations of unusual values were made:

- the first 20 rows of columns CHAS and LSTAT contain NaN values (marked with red), thus indicating that the values in those specific places are undefined or unrepresentable, and have to be changed to reflect an accurate further analysis and prediction of the dataset. I cannot be sure the NaN values are present only in those columns, thus, when cleaning the data by dealing with any missing data, I will check the entire dataset for the presence of NaN values;
- column CHAS (the Charles River dummy variable) (marked with blue) contains only 0.0s. I can build a hypothesis around this column having an error in displaying 1.0s, thus this gives me the idea of changing the NaN values with 1.0s when I normalize the dataset, but in order to do that - I have to check the mean of the CHAS column. If the mean is 0, then my

hypothesis might be right. If the mean is higher than 0, then another approach of dataset normalization will be used (such as projecting all 1s and observing if other columns will have close-to-each-other values of data) in regard to the CHAS 1 value).

TASK 2 | *Discuss what effects you would expect to see on the med home values (medv) for each variable.*

The values of the MEDV columns are dependable variables. Following real-life logic of the variables that could affect positively or negatively the price of a house, I would expect to see a higher medv value for houses with more rooms (high RM values), low crime rats (low CRIM values), houses built recently (low AGE values), low proportions of the population that represent the lower status (low LSTAT values), short distance to Boston employment centers (low DIS values), and higher tax-rates (high TAX values).

TASK 3 | *Compile the table with summary statistics (min, max, med, etc). Add the measure of variability (var, skew) to this table. Comment on the table and report briefly if you see anything unusual in the statistics of your variables.*

In order to obtain the summary statistics, the Python *median()*, *min()*, *max()*, and *skew()* functions were used.

Python code:

```
print("\nMedian:")
print(housingdata.median())

min_data = housingdata.min()
print("\nMin:")
print(min_data)

max_data = housingdata.max()
print("\nMax:")
print(max_data)

print("\nRange:")
print(max_data - min_data)

print("\nSkew:")
print(housingdata.skew())
```

Execution:

Median values

```
Median:
CRIM         0.253715
CHAS         0.000000
RM           6.208500
AGE         76.800000
DIS          3.207450
TAX        330.000000
LSTAT       11.430000
MEDV        21.200000
dtype: float64
```

Min values

```
Min:
CRIM         0.00632
CHAS         0.00000
RM           3.56100
AGE          2.90000
DIS          1.12960
TAX        187.00000
LSTAT        1.73000
MEDV         5.00000
dtype: float64
```

Max values

```
Max:
CRIM        88.9762
CHAS         1.0000
RM           8.7800
AGE        100.0000
DIS         12.1265
TAX        711.0000
LSTAT       37.9700
MEDV        50.0000
dtype: float64
```

Range of the values

```
Range:
CRIM       88.96988
CHAS        1.00000
RM          5.21900
AGE        97.10000
DIS        10.99690
TAX       524.00000
LSTAT      36.24000
MEDV       45.00000
dtype: float64
```

Skewness of the values

```
Skew:
CRIM      5.212843
CHAS      3.382293
RM        0.403612
AGE      -0.582470
DIS       1.011781
TAX       0.669956
LSTAT     0.908892
MEDV      1.108098
dtype: float64
```

After analysing the summary statistics for each column I can state that the unusual things that I have to check are in the colums:

CRIM

The data is extremely positive skewed, with a median of 0.25 - meaning that most houses report low crime rates, and the max value of 88.97 and the values close to this one might represent outliers for this dataset (and also might be an error). To make sure this hypotesis is right, next data analysis was performed.
I took CRIM values that were higher than 30.0 from this dataset and performed and displayed them in an ascending order.

Python code

```
check_crim_data = housingdata.loc[housingdata['CRIM'] > 30.0]
print("\nCheck CRIM column for outliers:")

print(check_crim_data.sort_values(by=['CRIM']))
rows_count = check_crim_data.count()[0]

print("\nNo. of rows: " + str(rows_count))
```

Results

```
Check CRIM column for outliers:
        CRIM  CHAS     RM    AGE     DIS  TAX  LSTAT  MEDV
427  37.6619   0.0  6.202   78.7  1.8629  666  14.52  10.9
398  38.3518   0.0  5.453  100.0  1.4896  666  30.59   5.0
404  41.5292   0.0  5.531   85.4  1.6074  666  27.38   8.5
414  45.7461   0.0  4.519  100.0  1.6582  666  36.98   7.0
410  51.1358   0.0  5.757  100.0  1.4130  666  10.11  15.0
405  67.9208   0.0  5.683  100.0  1.4254  666  22.98   5.0
418  73.5341   0.0  5.957  100.0  1.8026  666  20.62   8.8
380  88.9762   0.0  6.968   91.9  1.4165  666  17.21  10.4

No. of rows: 8
```

As it can be seen, the MEDV are as well, among the lowest in regards with high crime rates, thus we can exclude the possibility of errors and define this sweness as just unusual crime rates.

CHAS

The CHAS column is as well extremely positive skewed, but since the data here alternates between 0s and 1s, we can conclude that most of the houses are not located near the Charles River.

TASK 4 | *Check the types of your data. Change the types as appropriate (if any categorical variable present change in its type to category).*

Python code

```
print("\nData Types:")
print(housingdata.dtypes)

print("\nChanged Types:")
changed_dtypes = {'TAX': float}
housingdata = housingdata.astype(changed_dtypes)
print(housingdata.dtypes)

print("\nModified dataset:")
observations = housingdata.head(30)
print(observations)
```

Results

```
Data Types:
CRIM       float64
CHAS       float64
RM         float64
AGE        float64
DIS        float64
TAX          int64
LSTAT      float64
MEDV       float64
dtype: object
```

⇒

```
Changed Types:
CRIM       float64
CHAS       float64
RM         float64
AGE        float64
DIS        float64
TAX        float64
LSTAT      float64
MEDV       float64
dtype: object
```

My first thought was to change the TAX column data type from int to float, to operate with the same datasets.

Since the CHAS variable has only 1s and 0s, a good approach is to convert those values to Boolean - thus the CHAS column for the new dataset is displaying right now 'False' for 0s and 'True' for 1s. However, since this operation means that the NaN values from the CHAS column would also be replaced with 'True' values, I think it's better to pass on this task for now, with the unchanged data, and thinking about performing the same approach once I will get rid of the NaN values.

TASK 5 | *Substitute the NaN values with appropriate measures of central tendency (mean, median, or mode – for the categorical variable – if you can't change to mode, then check what is the most frequent value of that variable (you can use value_counts) and change it to the most frequent value). You may want to do this procedure for all variables to make sure that you did not miss a variable because you were not able to see that the variable contains NaNs while inspecting the table.*

First thing to do here is to check how many columns have NaN values. This can be done with the *isnull()* (to check which variables are missing), and *sum()* (to check how many in a column) Python functions.

Python code:

```
print("\nNaN values:")
print(housingdata.isnull().sum())
```

Results:

```
NaN values:
CRIM      20
CHAS      20
RM         0
AGE       20
DIS        0
TAX        0
LSTAT     20
MEDV       0
dtype: int64
```

What can be deducted from the output is that not only the CHAS and LSTAT columns had missing values, but CRIM and AGE as well. All missing values from each column have the value of 20, however those NaN values are not intrrelated in their distribution across the dataset. After this analysis, I changed the NaN values with their median, because some columns contain extreme skeweness, and the median in this case would reflect approximation with less error, rather than if I would have used the mean.

Python code:

```
print("\nNaN values:")
print(housingdata.isnull().sum())
```

Execution:

```
Datased without NaN values:
       CRIM   CHAS     RM    AGE     DIS    TAX  LSTAT  MEDV
0   0.00632  False  6.575   65.2  4.0900  296.0   4.98  24.0
1   0.02731  False  6.421   78.9  4.9671  242.0   9.14  21.6
2   0.02729  False  7.185   61.1  4.9671  242.0   4.03  34.7
3   0.03237  False  6.998   45.8  6.0622  222.0   2.94  33.4
4   0.06905  False  7.147   54.2  6.0622  222.0  11.43  36.2
5   0.02985  False  6.430   58.7  6.0622  222.0   5.21  28.7
6   0.08829  False  6.012   66.6  5.5605  311.0  12.43  22.9
7   0.14455  False  6.172   96.1  5.9505  311.0  19.15  27.1
8   0.21124  False  5.631  100.0  6.0821  311.0  29.93  16.5
9   0.17004  False  6.004   85.9  6.5921  311.0  17.10  18.9
10  0.22489  False  6.377   94.3  6.3467  311.0  20.45  15.0
11  0.11747  False  6.009   82.9  6.2267  311.0  13.27  18.9
12  0.09378  False  5.889   39.0  5.4509  311.0  15.71  21.7
13  0.62976  False  5.949   61.8  4.7075  307.0   8.26  20.4
14  0.63796  False  6.096   84.5  4.4619  307.0  10.26  18.2
15  0.62739  False  5.834   56.5  4.4986  307.0   8.47  19.9
16  1.05393  False  5.935   29.3  4.4986  307.0   6.58  23.1
17  0.78420  False  5.990   81.7  4.2579  307.0  14.67  17.5
18  0.80271  False  5.456   36.6  3.7965  307.0  11.69  20.2
19  0.72580  False  5.727   69.5  3.7965  307.0  11.28  18.2
```

It can be observed that from the first 20 rows, the NaN values were replaced with the median of each column. To be sure tht all NaN values were replaced, I will run the same line of code as previous and check if there are any other missing values.

Python code:

```
print("\nNew check for NaN values:")
print(housingdata.isnull().sum())
```

Results:

```
New check for NaN values:
CRIM     0
CHAS     0
RM       0
AGE      0
DIS      0
TAX      0
LSTAT    0
MEDV     0
dtype: int64
```

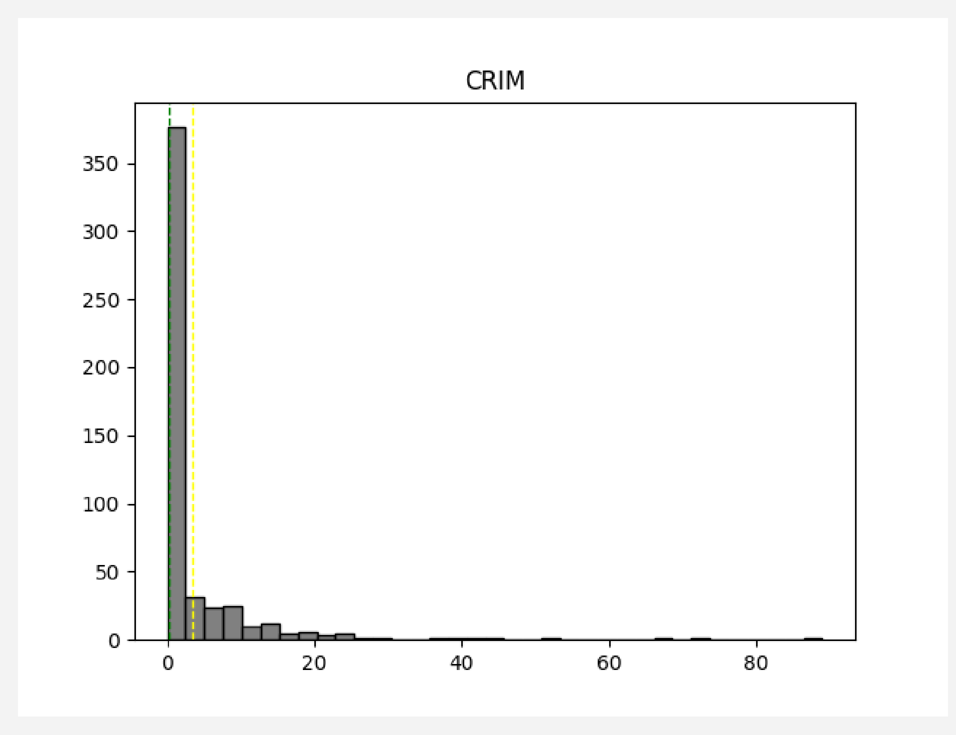
As it can be observed - all NaN values were replaced.

TASK 6 | *Produce the histograms of all variables (except Chas) and comment on their distributions (for each variable separately). Notice any outliers, or fat tails (like in the case of tax). Put this into context knowing what your variables mean. So don't just say that tax has a fat right tail, but something like "it appears that our dataset has the majority of houses with relatively low tax rates, and a set of properties that are highly taxed."*

Python code

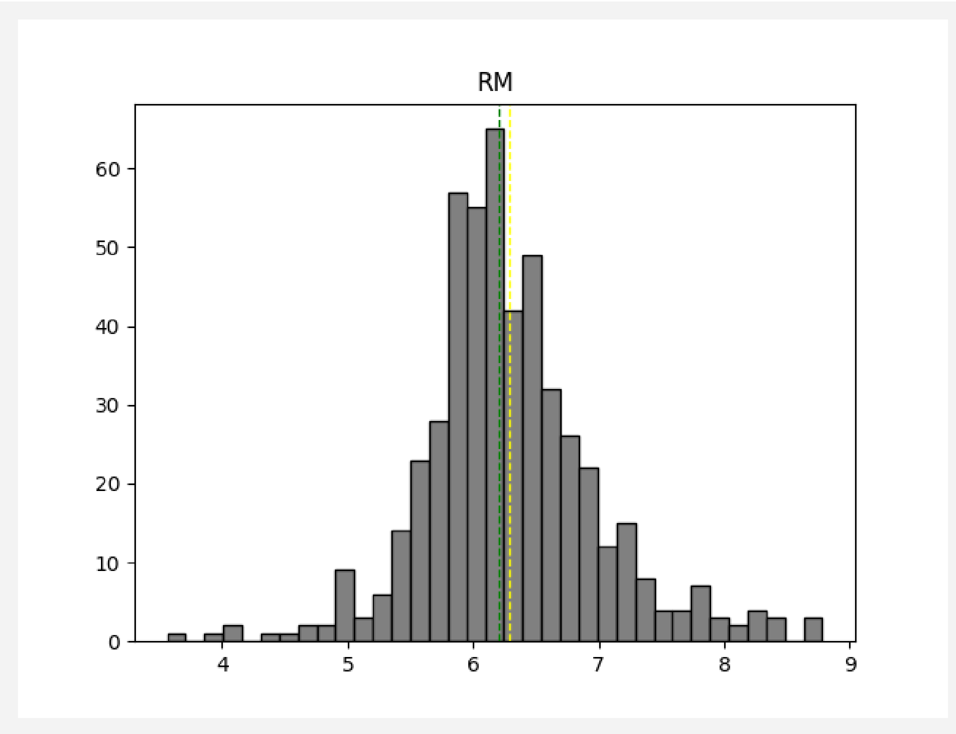
```
housingdata= housingdata.fillna(housingdata.median())
get_rows = housingdata.head(20)
print("\nFirst 20 rows:")
print (get_rows)
```

#1



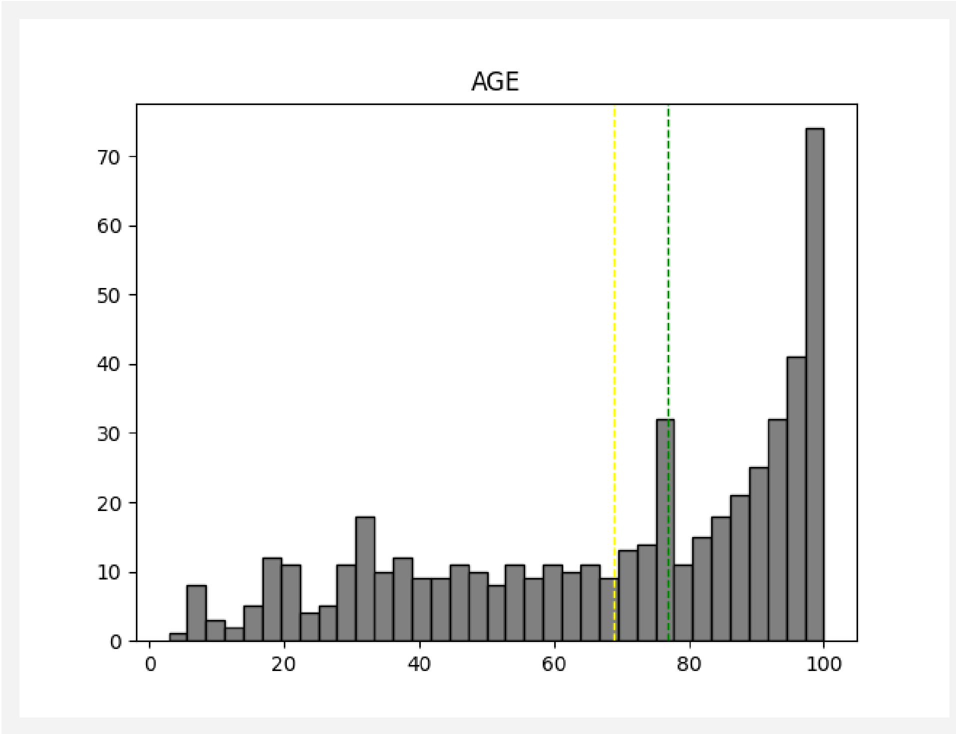
From the 1st histogram it can be deducted that the majority of Boston houses have a close-to-zero crime rates, exception being a few houses wich happens to have extremely high criminal rates. This histogram confirms the extremely positive skewness that was described in previous tasks.

#2



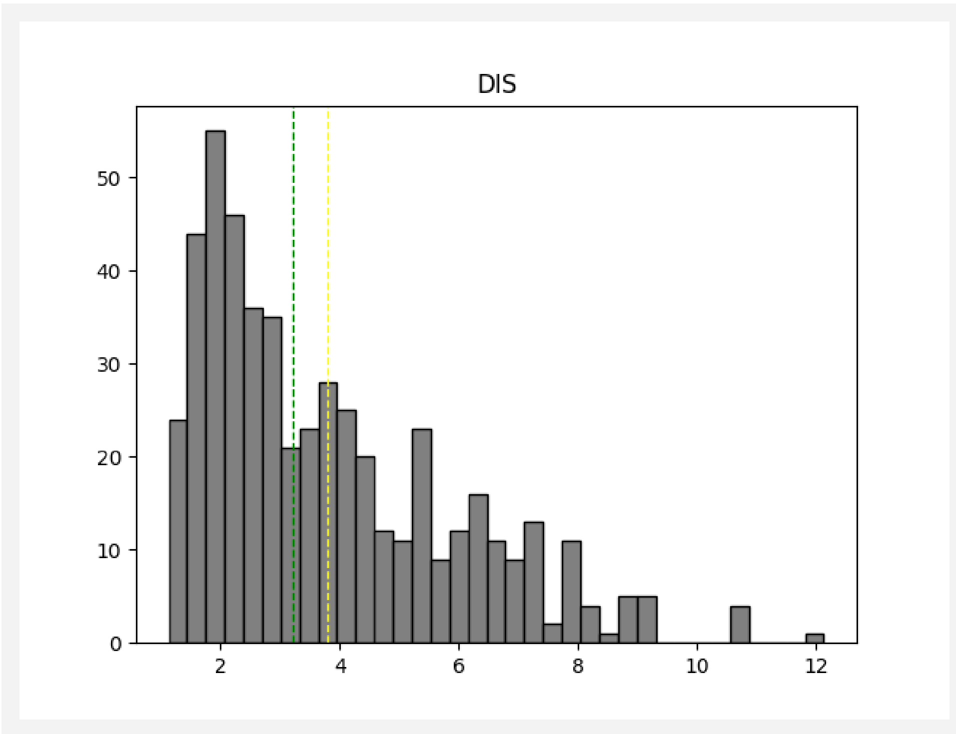
Here, we can see a normal distribution of data, with most of the houses having the average of 6 rooms, but houses with 8-9 rooms are more prevalent than those with 4-5 rooms.

#3



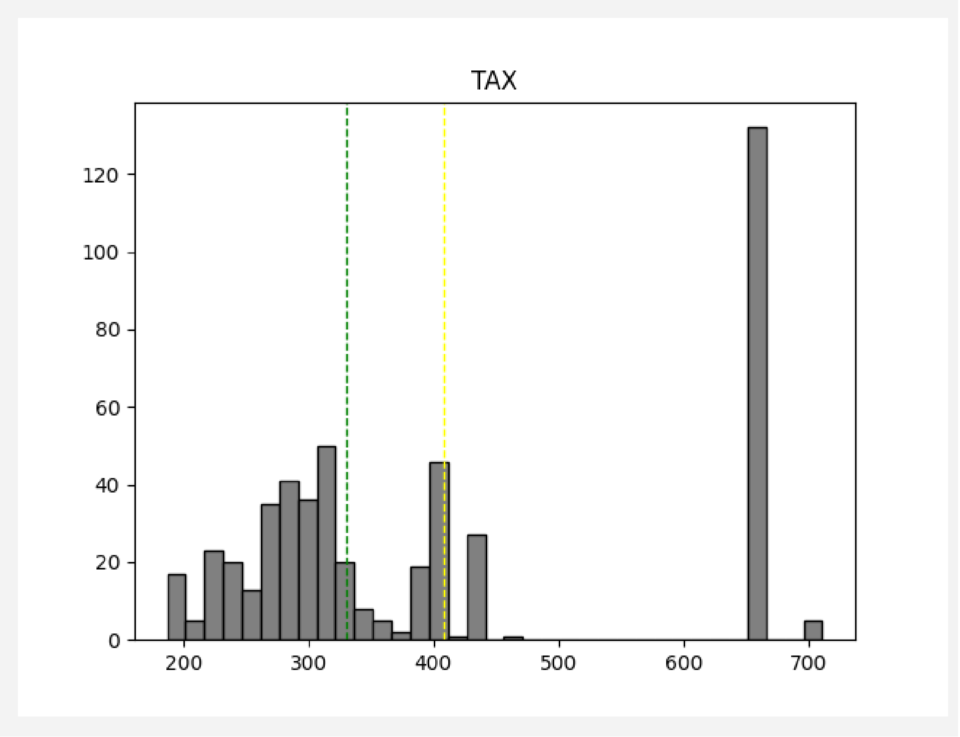
From this representation we can deduct that a significant amount of houses are old, and the remaining data is more or less normally distributed across the middle part of the graph, meaning an average antiquity.

#4



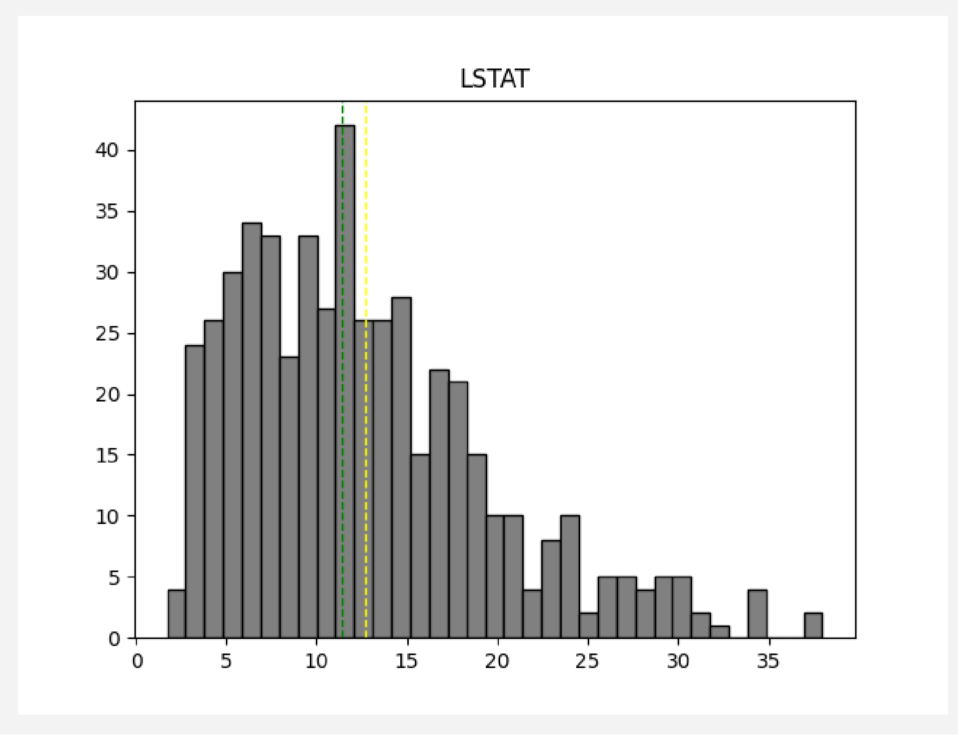
As we can see, here we have a positive skewness, and most of the houses are placed near those 5 Boston employment centers. However, there are a few houses which are placed far away.

#5



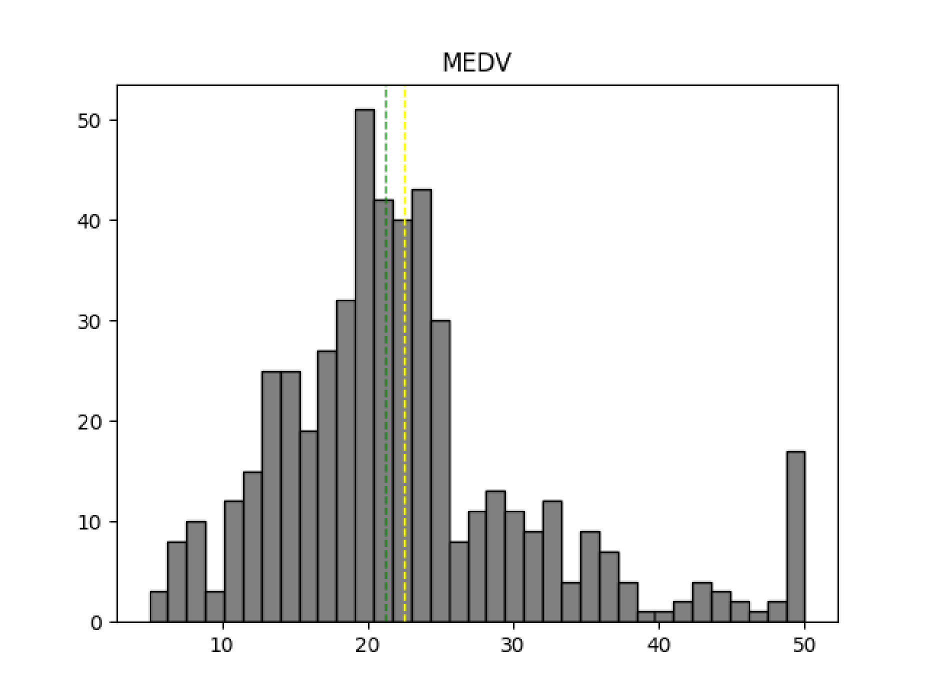
The skweness for the TAX is 0.66, meaning that the the data are moderately skewed. We can see from the values that some houses have a low tax rate, however there is a huge amount of houses with a tax rate of 666.

#6



We can see that the percentage of the lower status of the population is mostly decreased, with some exceptions that we can find on the right side of the graph, meaning a high percentage of lower status of the population.

#7



The MEDV is positively skewed, the amount of prices of those houses being lower than the middle of the graph. There are some very expensive houses and some that are cheaper and c an be found at the both tails of this graph.

TASK 7 | *Create box plots for all variables where you split by the Chas variable (make sure to adjust the number of axes). Comment shortly on each box plot separately noticing if the distributions are located higher for properties on the river versus those not on the river. What does it mean when you put it into context with what your variables mean?*
Can you make a guess if these houses are preferred by Bostoners? Are these high-end residences? How do you explain that they seem to be valued higher when it comes to price, but are in the same time on the older side when it comes to building's age?

Python code

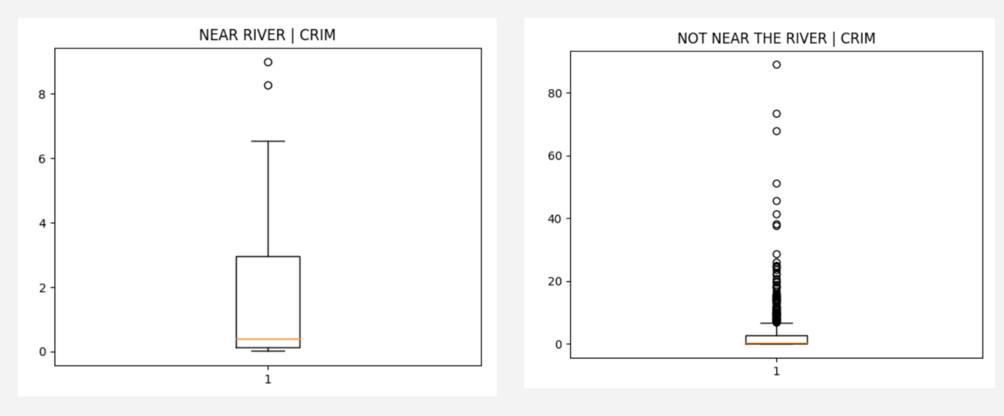
```
river_housingdata = housingdata.loc[housingdata['CHAS'] == 1]
non_river_housingdata = housingdata.loc[housingdata['CHAS'] == 0]

for x in river_housingdata.columns:
    if (x == 'CHAS'):
        continue
    plt.boxplot(housingdata[x])
    plt.title("NEAR RIVER | " + x)
    plt.show()

for x in non_river_housingdata.columns:
    if (x == 'CHAS'):
        continue
```

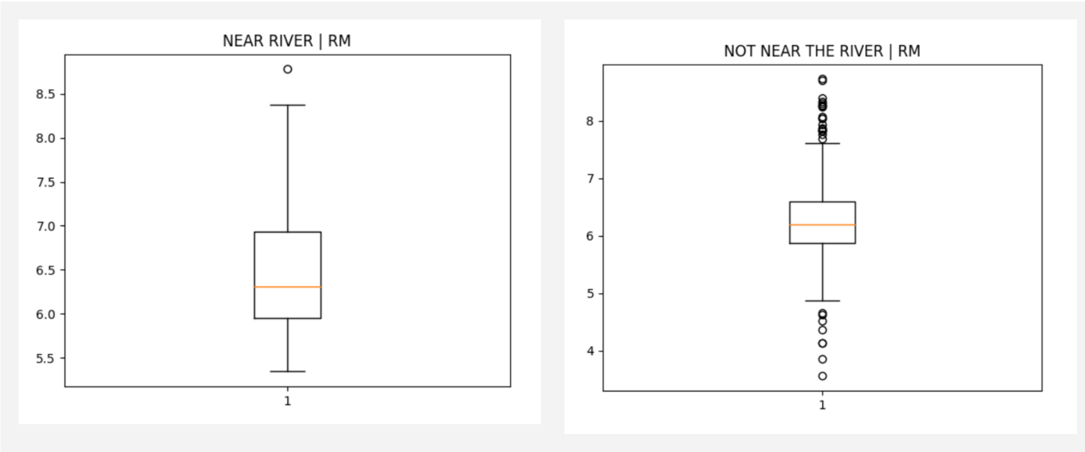
```
plt.boxplot(housingdata[x])
plt.title("NOT NEAR THE RIVER | " + x)
plt.show()
```

#1



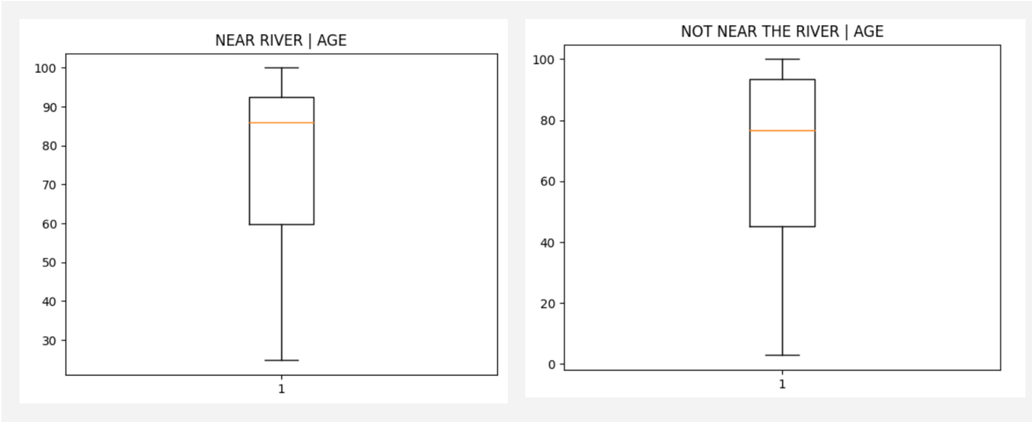
The distribution is not the same. The most crimes are not near the river.

#2



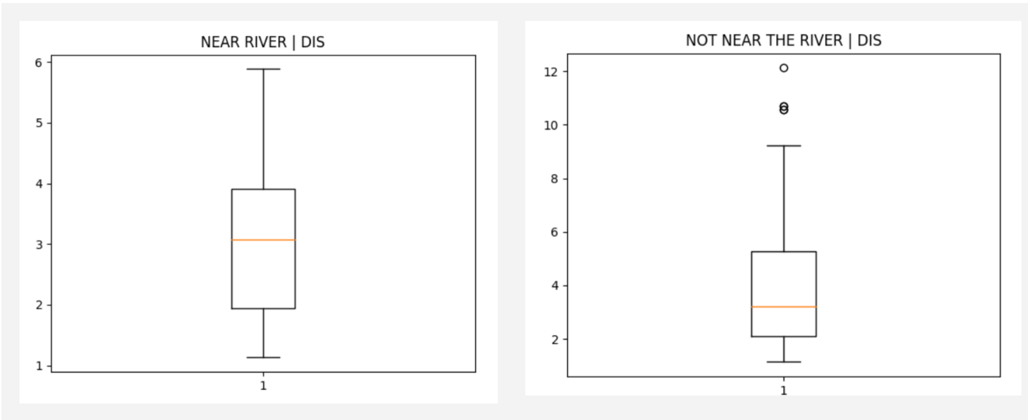
The distribution is more or less the same.

#3



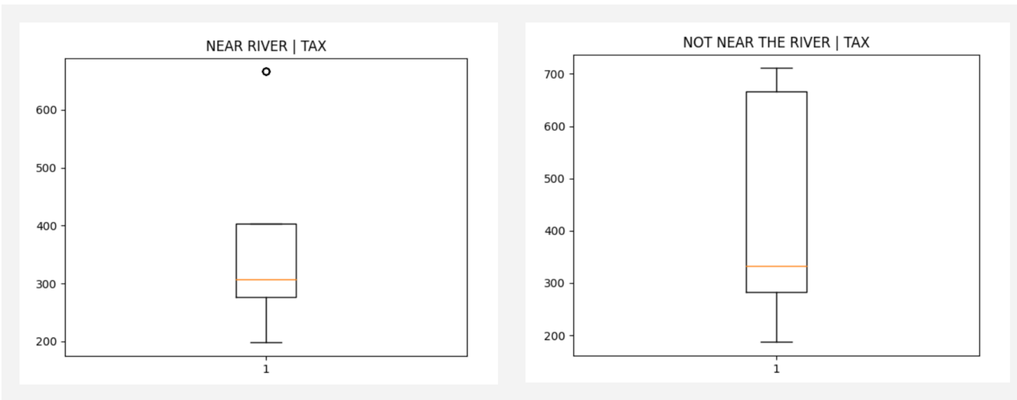
The distribution more or less the same.

#4



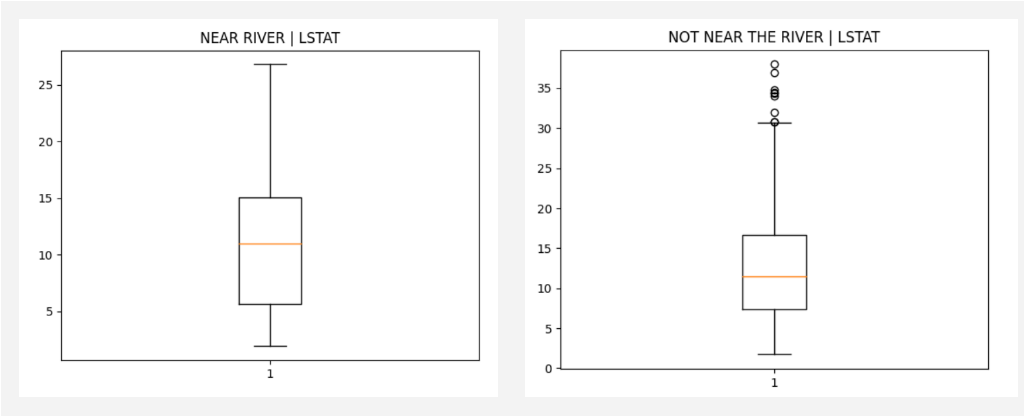
The distribution is not the same. Houses further away from the river are closer to the employment centers.

#5



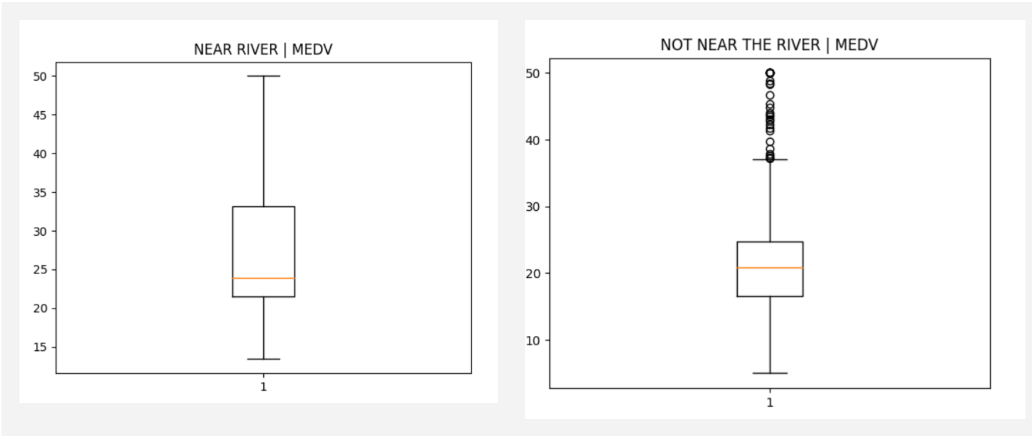
The distribution is not the same. People further away from the river pay bigger taxes that those near the river.

#6



The distribution is not the same. The percentage of the distribution of the lower status of the population seems to be higher further away from the river.

#7



The distribution is not the same. Houses near the river are pricier.

It seems like the rich Boston people have a preference for the river houses, which are pricier, but safer. Otherwise, houses further away from the river have more owners that those near the river.

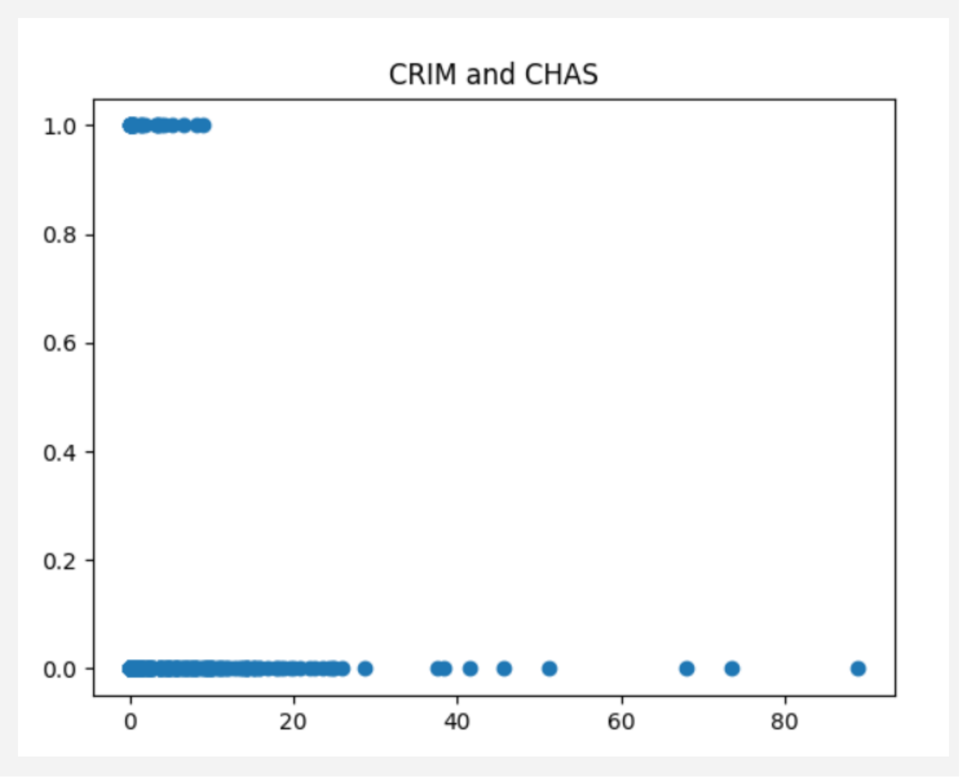
TASK 8 | *Create the scatter plots for each pair of variables. Comment on how the variables correlate with the medv variable. Do these correlations make sense? Explain why? Do they confirm what you wrote in the beginning of this work where you hypothesized how these variables will affect the houses' values? Notice any other correlations between the pairs or variables if obvious from the scatter plots.*

Python code:

```
plt.scatter(housingdata[x], housingdata[y])
plt.title("x and y")
plt.show()
```

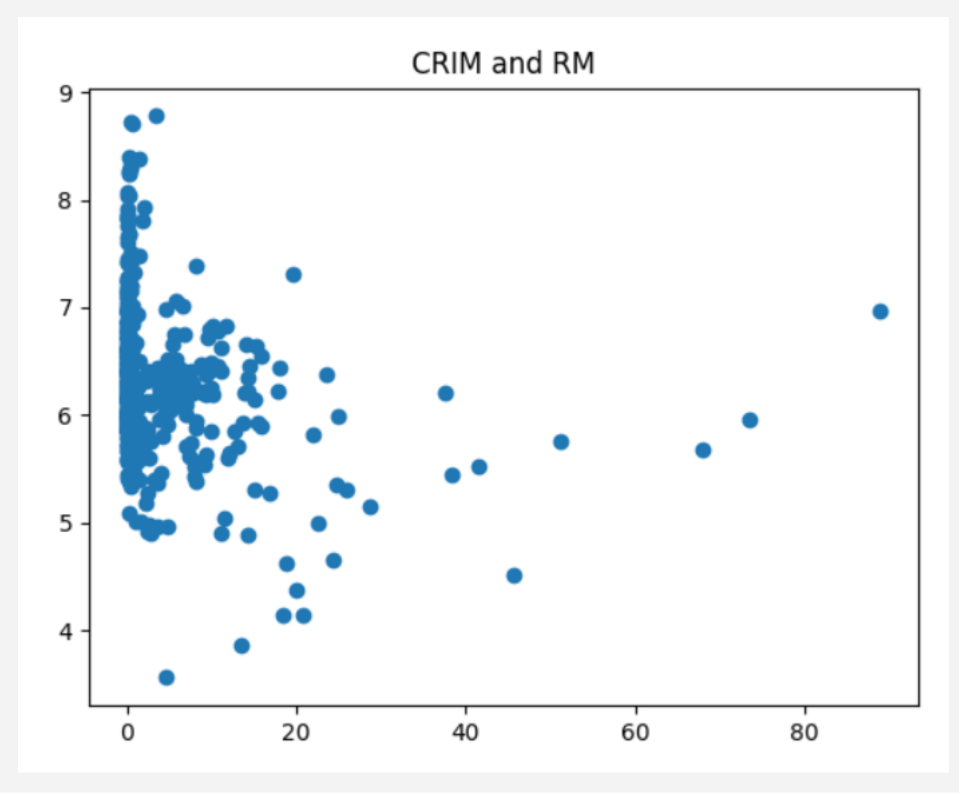
Execution:

#1



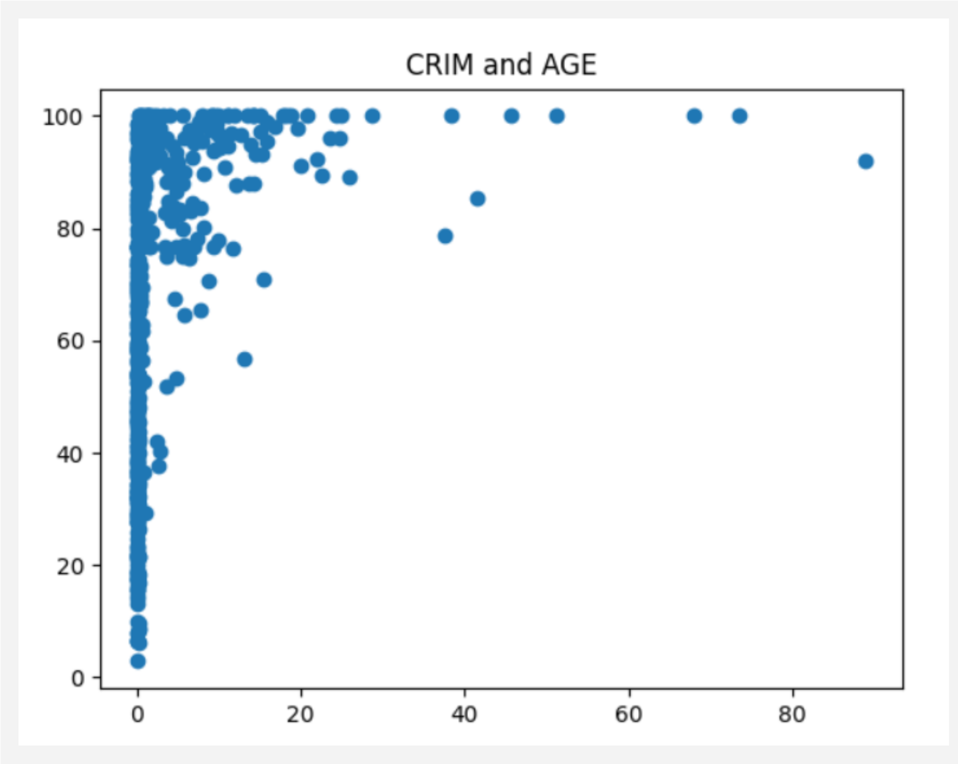
There is not correlation between the crime rate and if the house is near the river.

#2



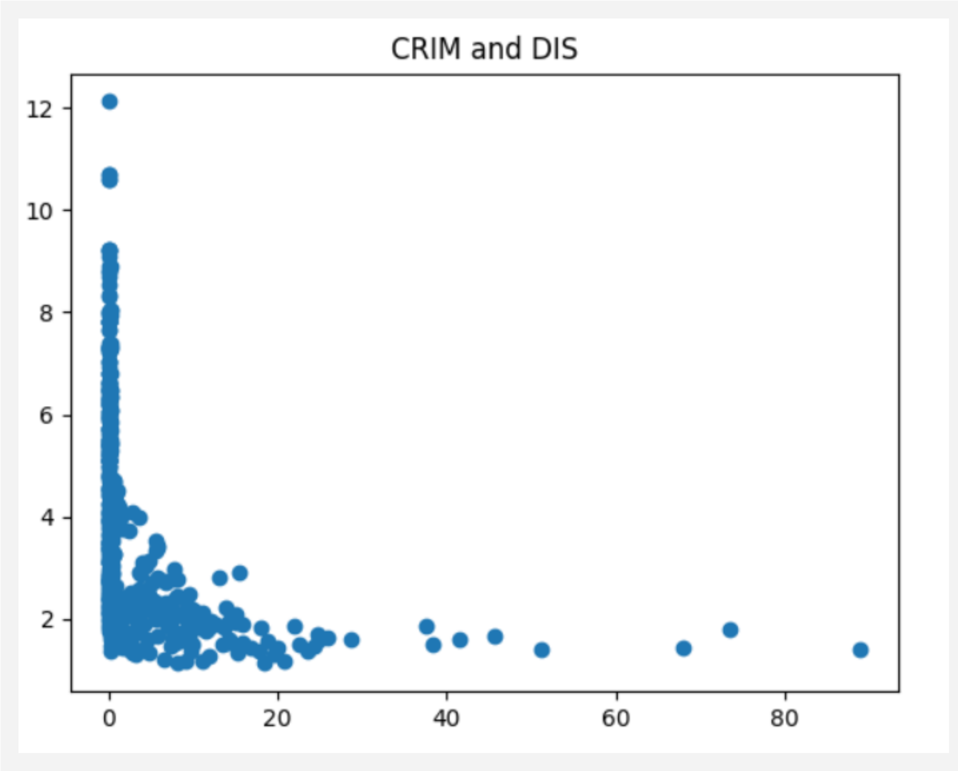
There is a correlation between the crime rate and the average no. of rooms (average no. of rooms means rich people).

#3



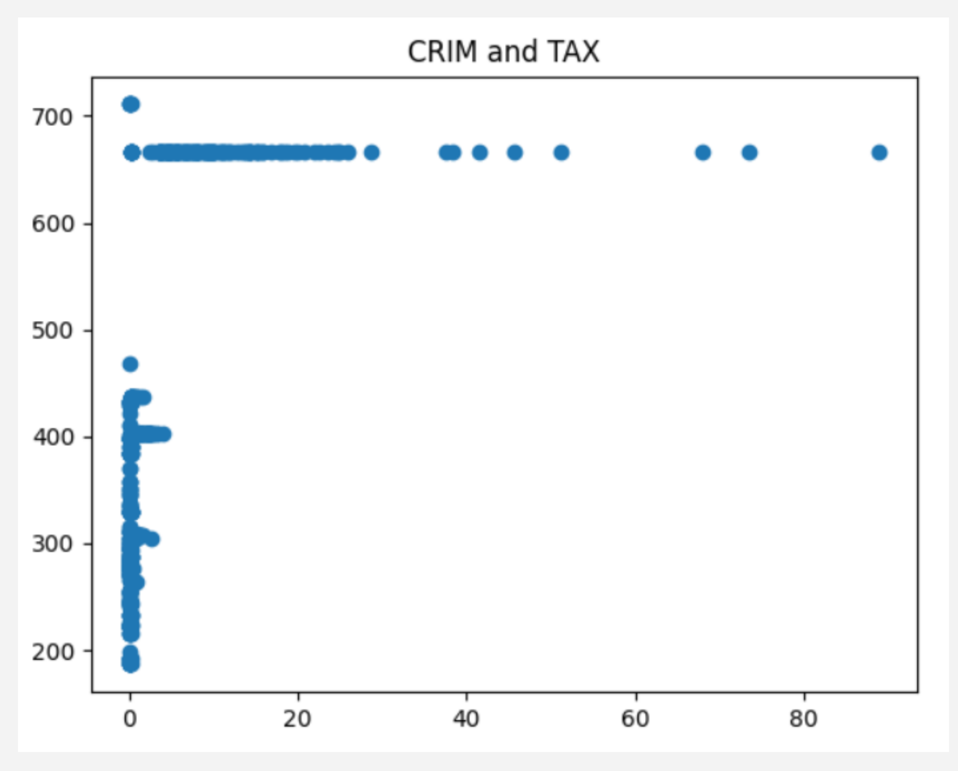
There is some correlation between the crime rate and the age of the house.

#4



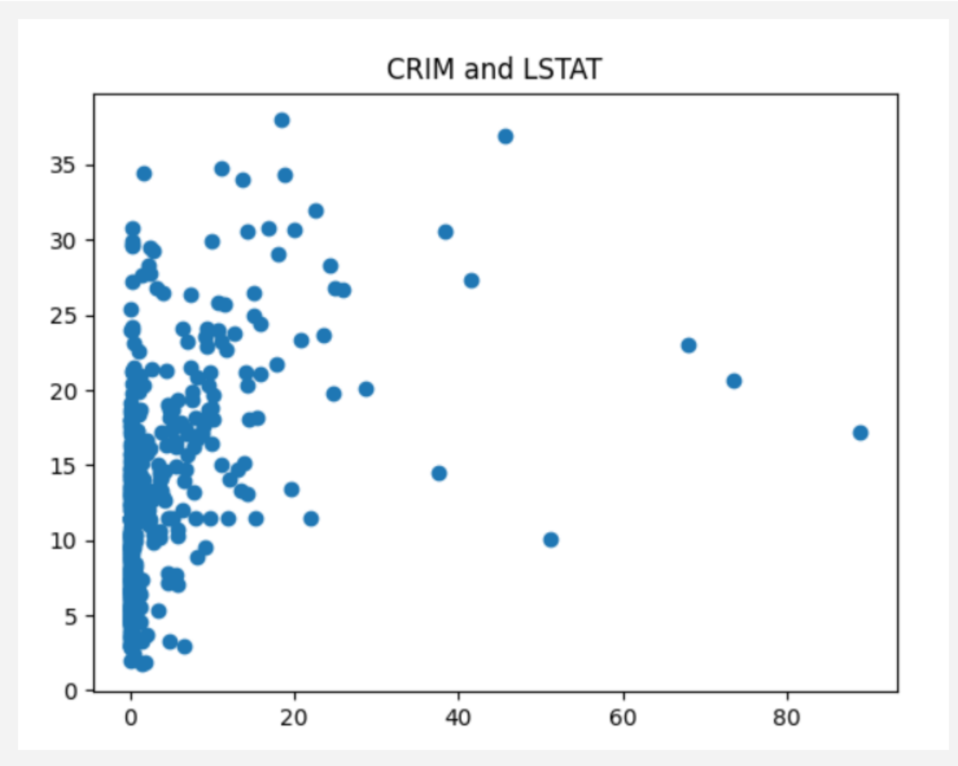
There is a some correlation between the crime rate and the distance from the house to the employment centers.

#5



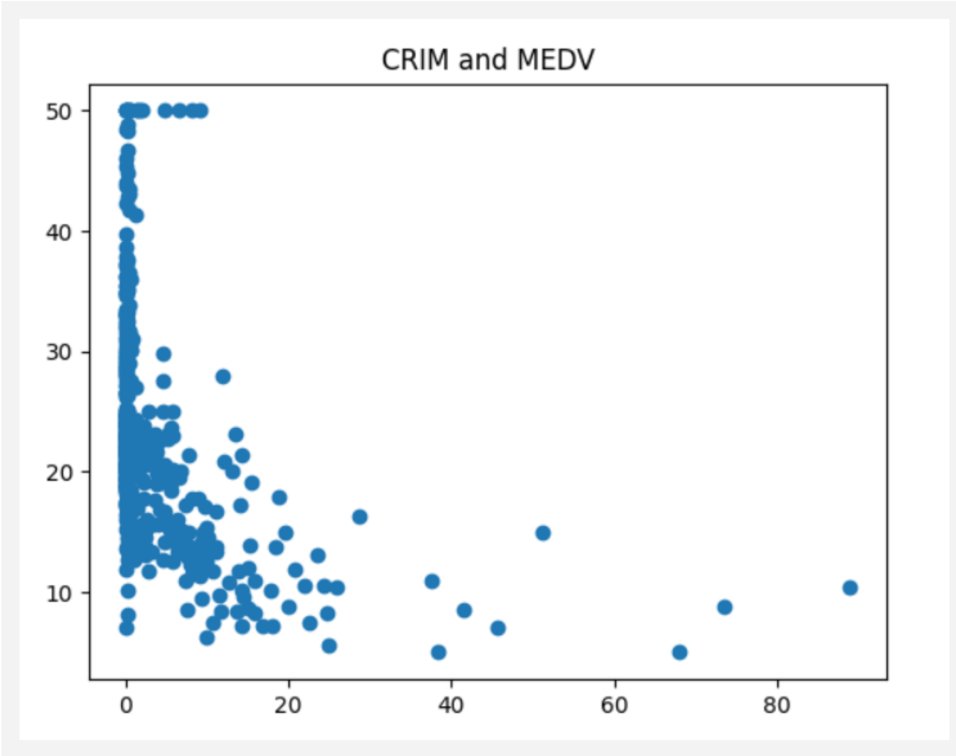
There is no correlation between the crime rate and the tax rate.

#6



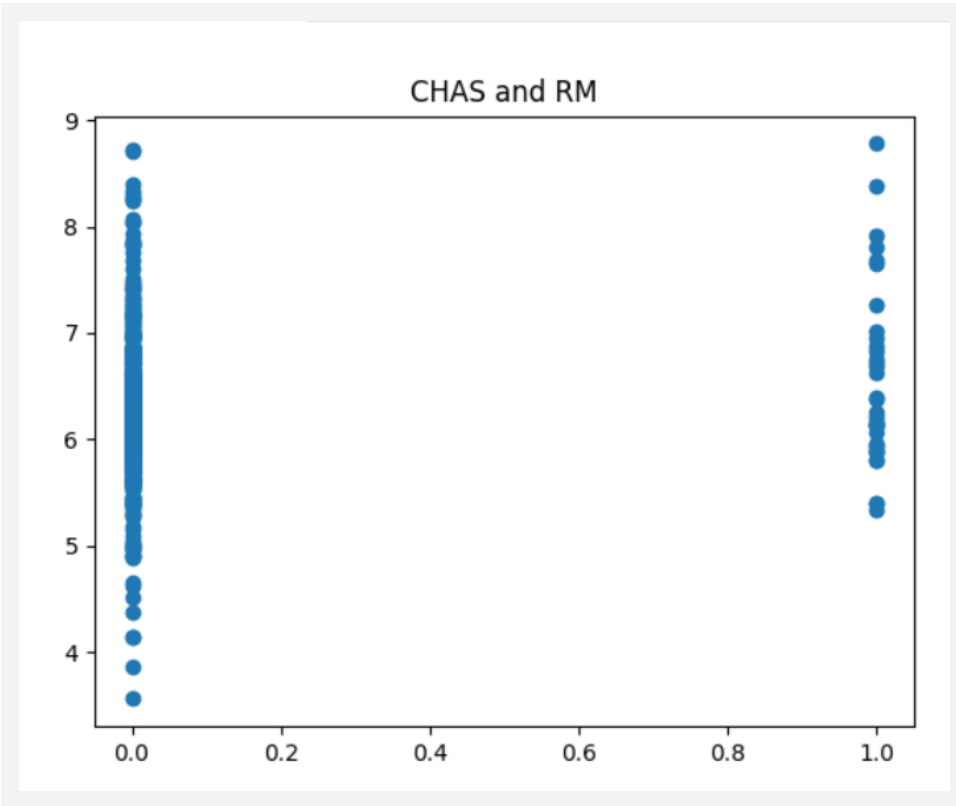
There is some corelation, but not strong, between the crime rate and the percentage of the lower status of the population.

#7



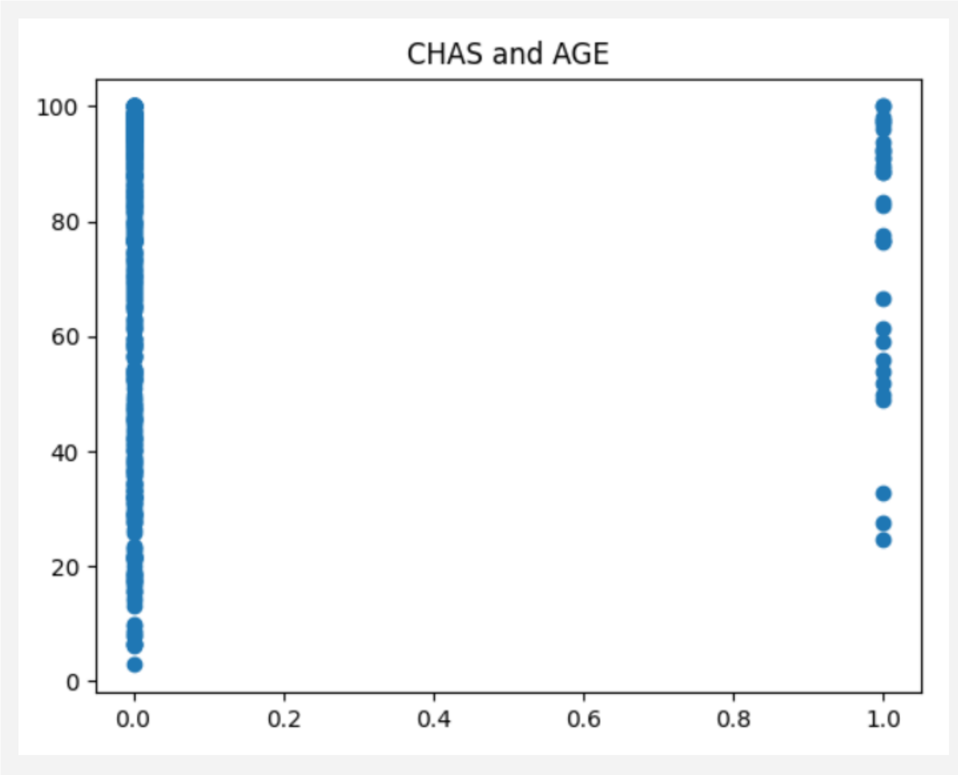
There is correlation between the crime rate and the price of the house.

#8



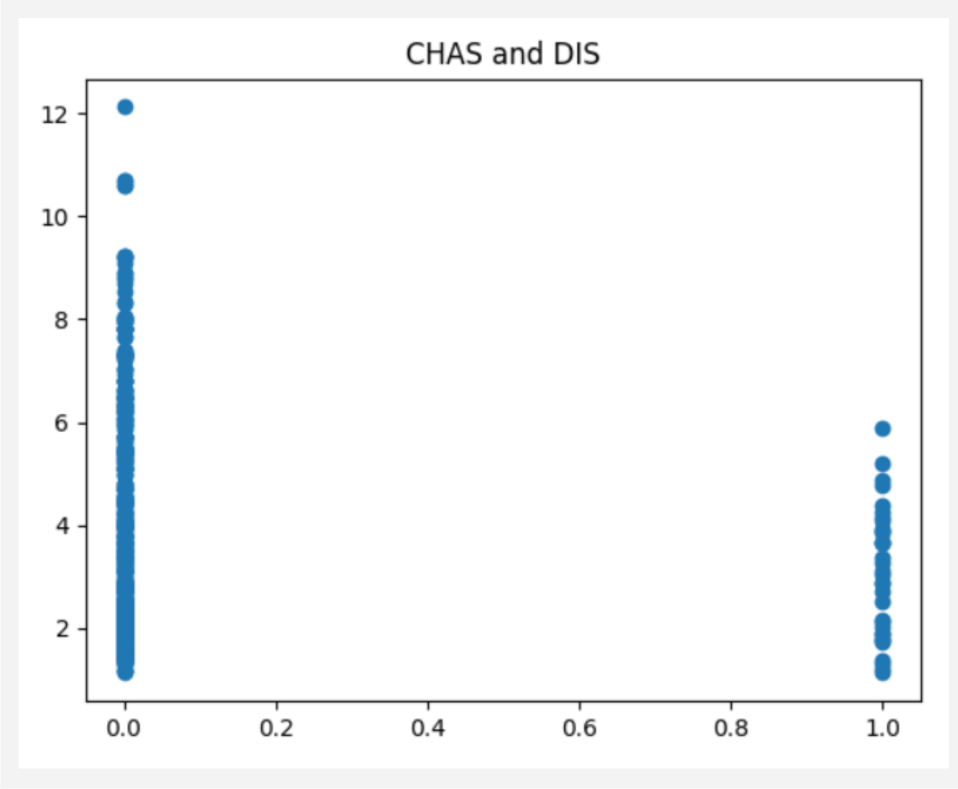
There is not correlation between the no. of rooms and if the house is near the river.

#9



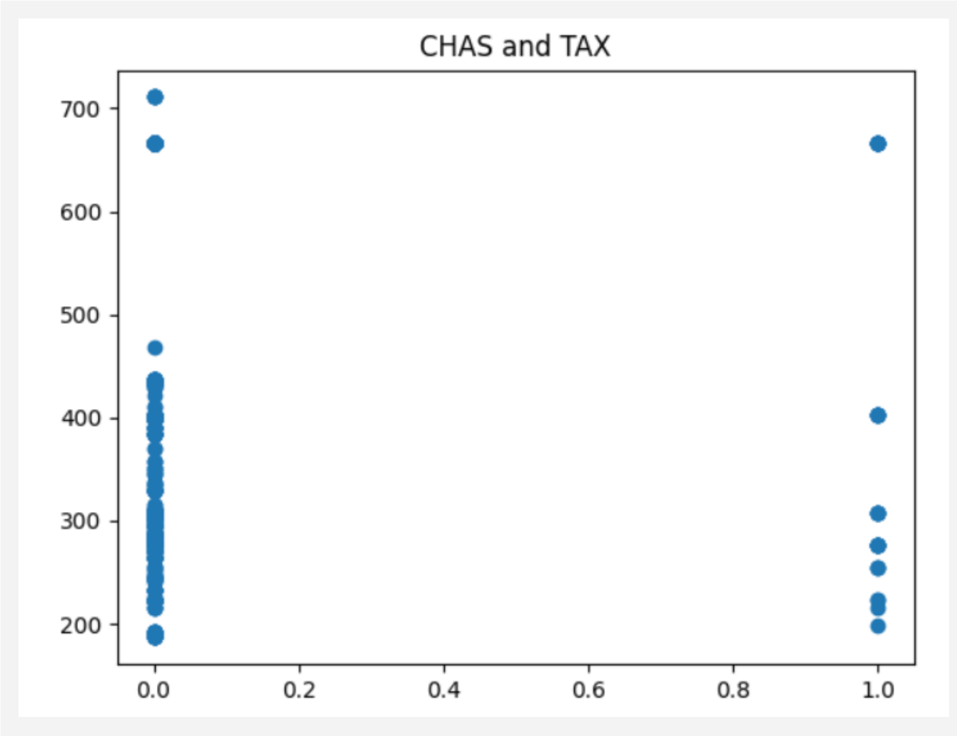
There is not correlation between the age of the house and if the house is near the river.

#10



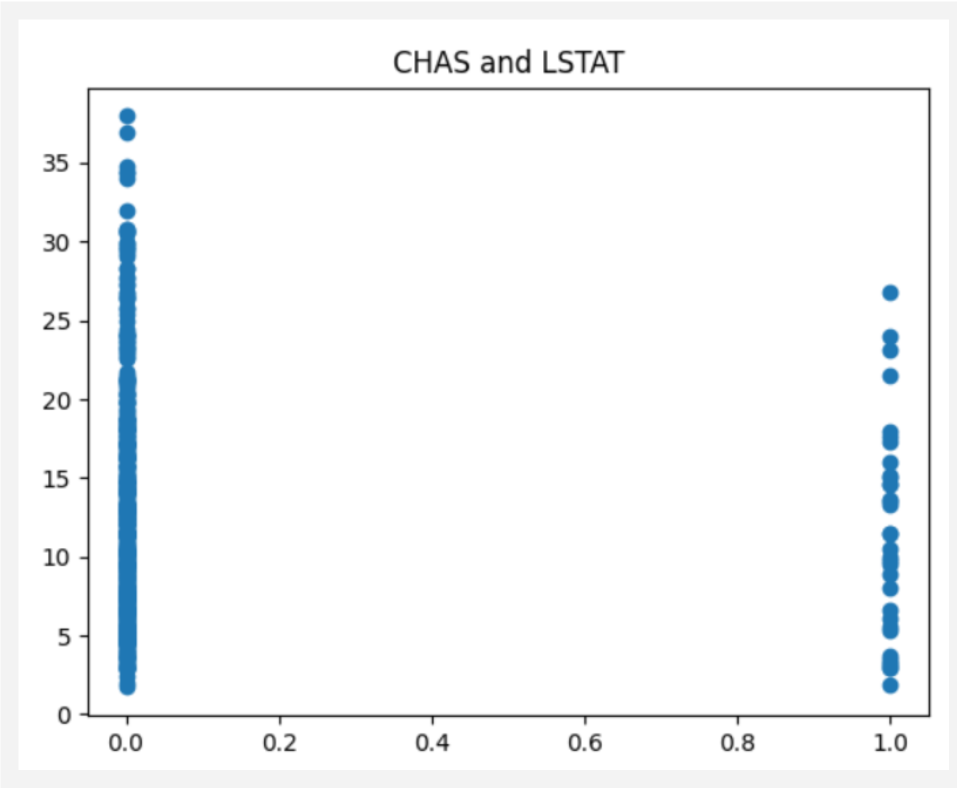
There is not correlation between the distance til work and if the house is near the river.

#11



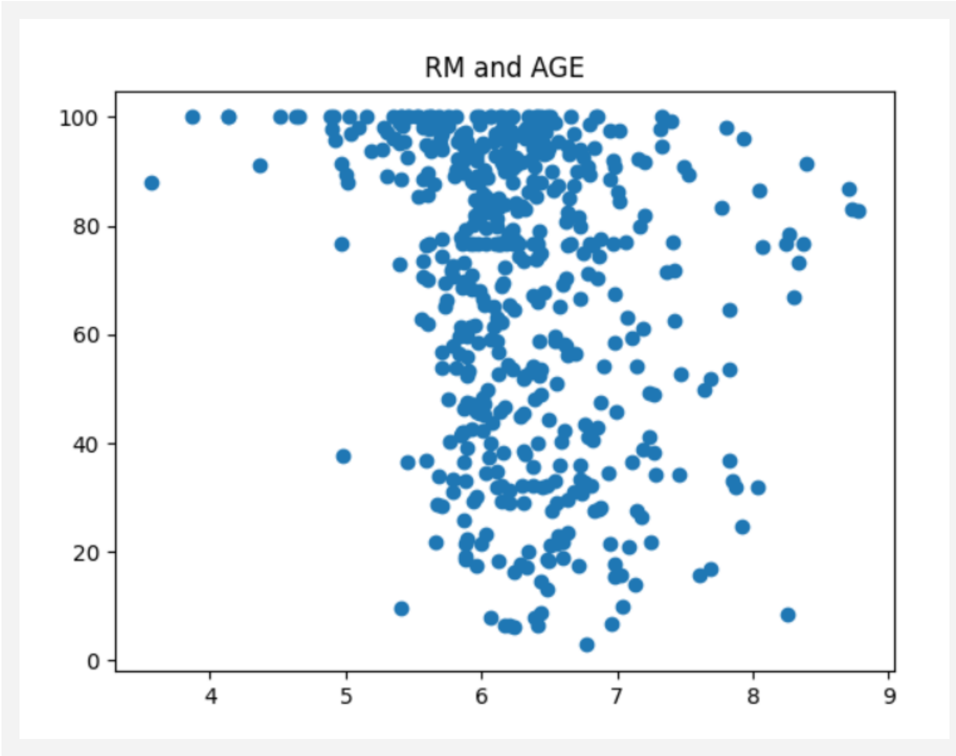
There is not correlation between the tax and if the house is near the river.

#12



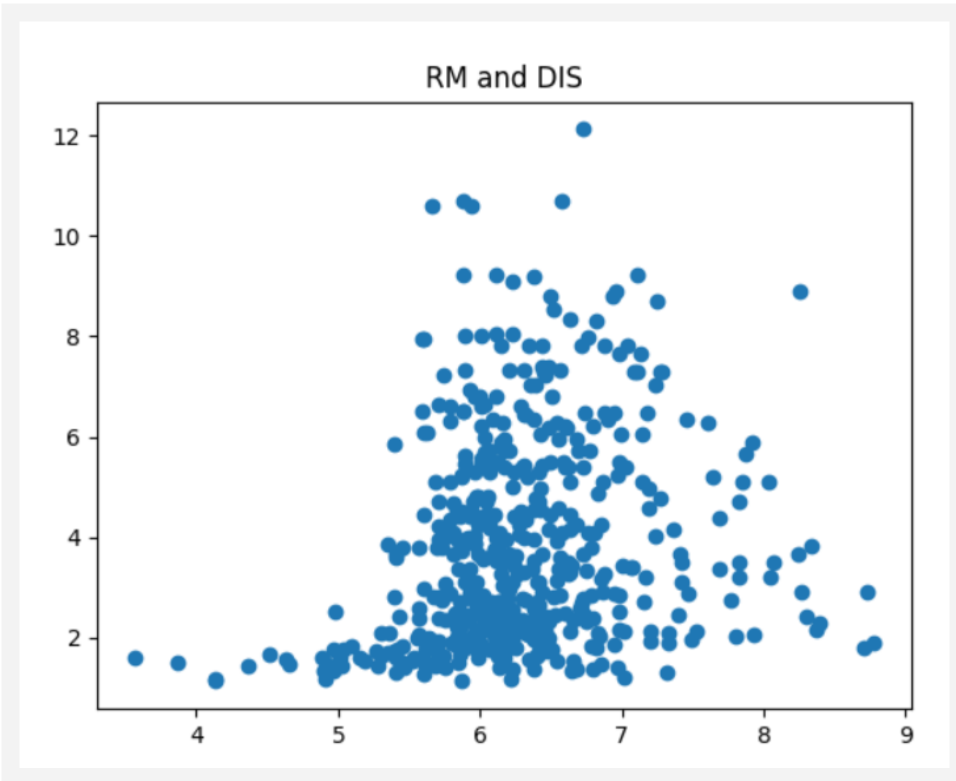
There is not correlation between the percentage of the lower status of the population and if the house is near the river.

#13



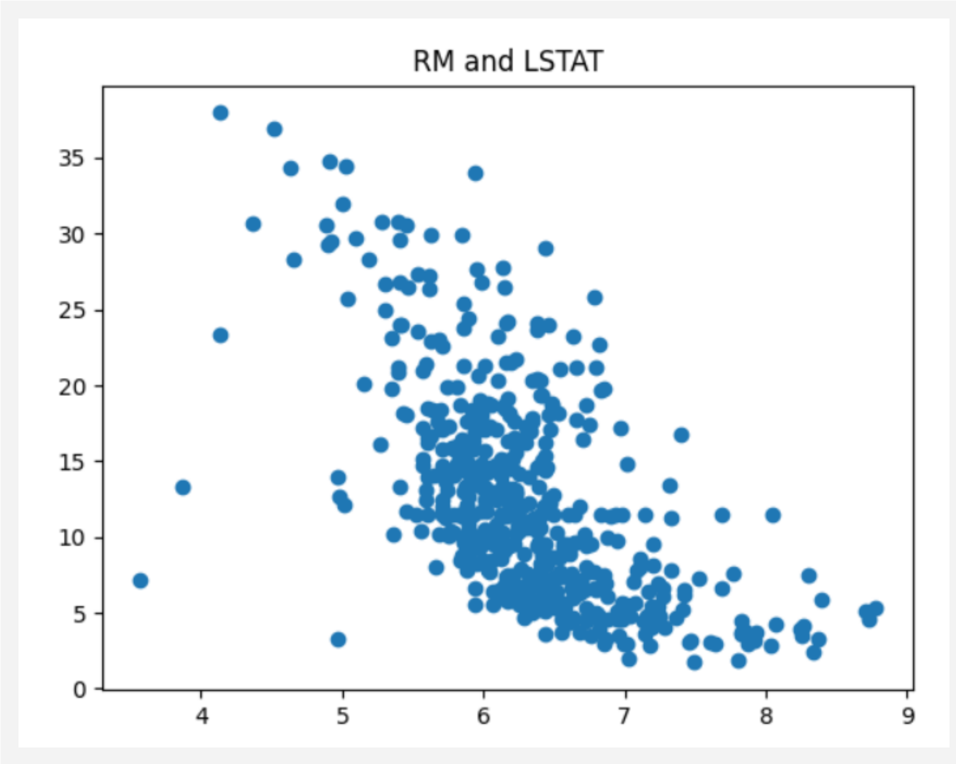
There is some correlation between the no. of rooms and the age of the house, but it's not big.

#14



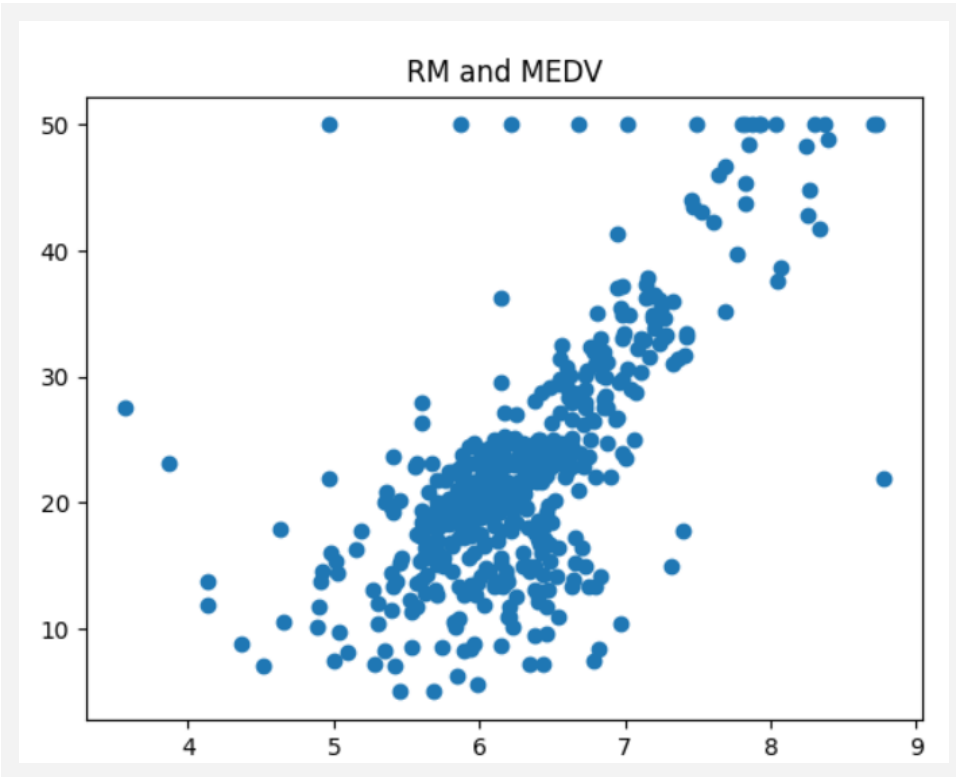
There is some correlation between the no. of rooms and the distance til work, but it's not big.

#15



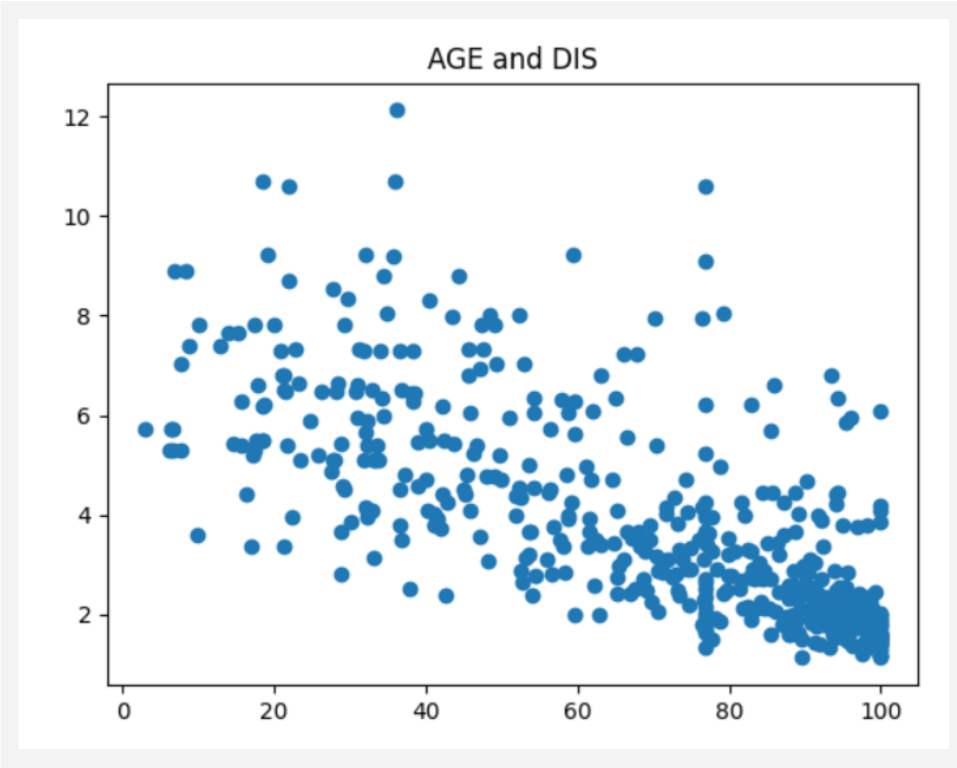
There is some correlation between the no. of rooms and the percentage of the lower status of the population.

#16



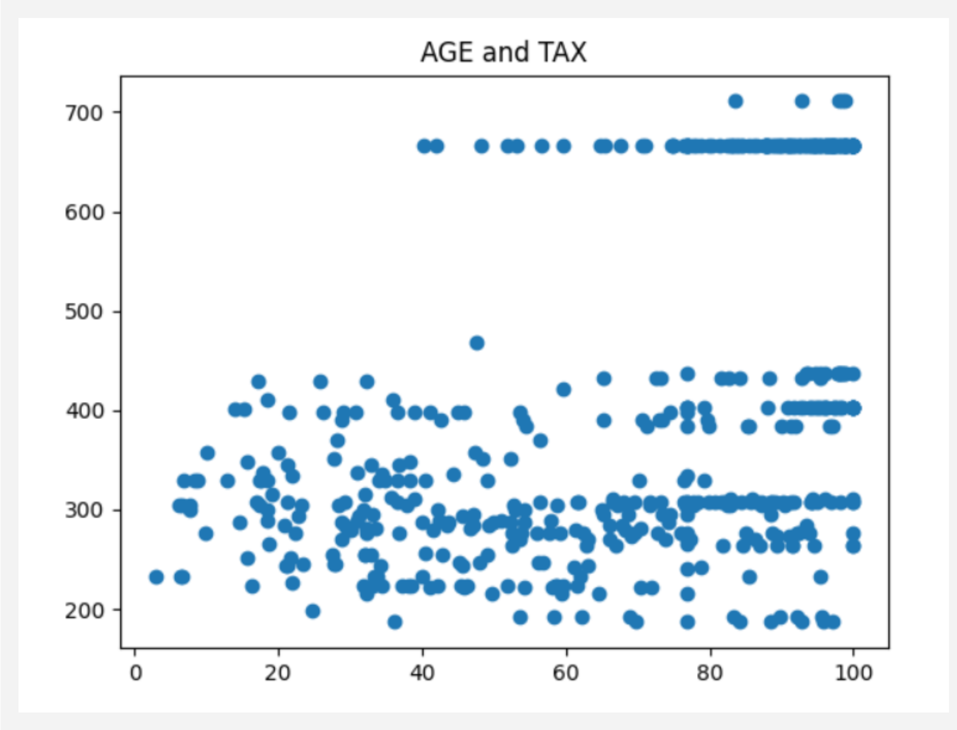
There is correlation between the no. of rooms and the price of the house.

#17



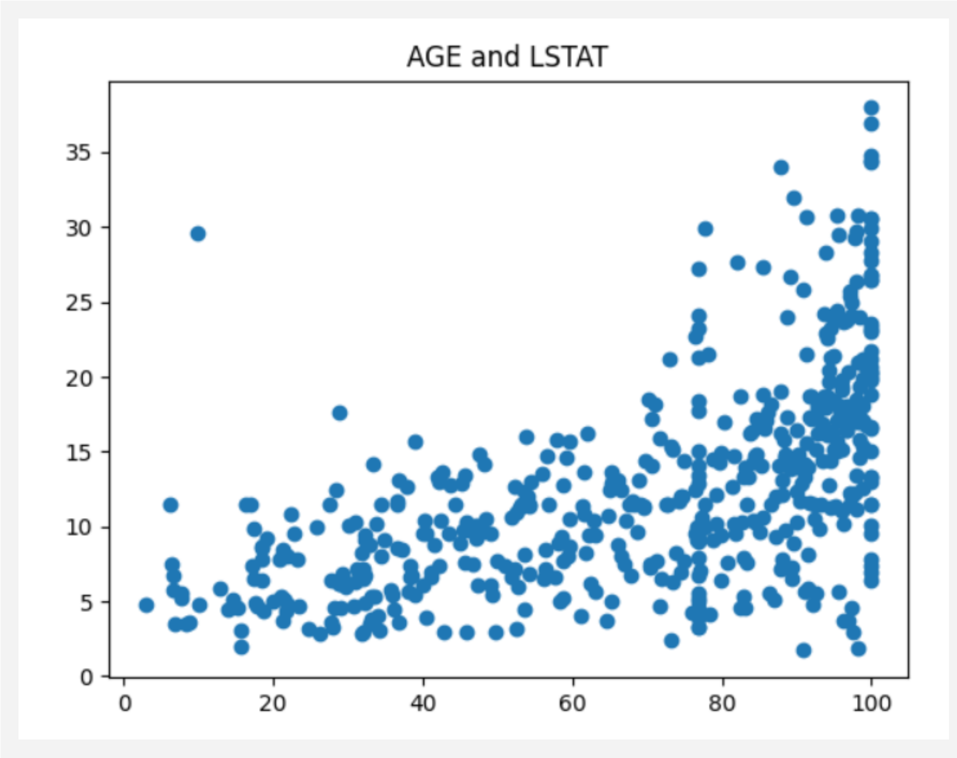
There is a little bit of correlation between the age of the house and the distance until work.

#18



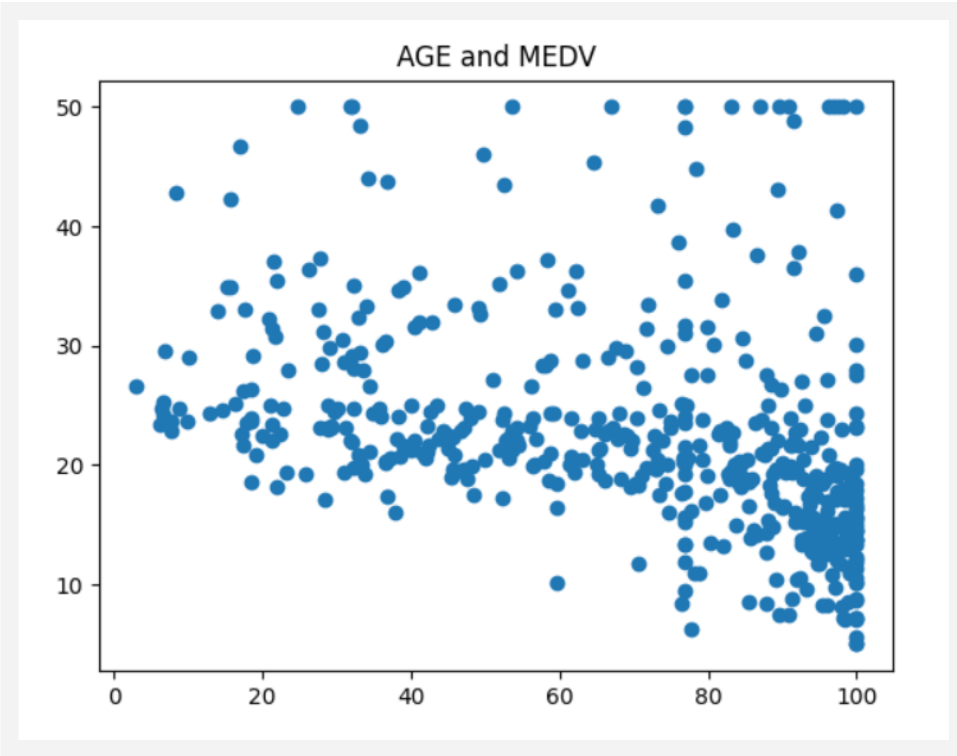
There is no correlation between the age of the house and the tax of it.

#19



There is a little bit of correlation between the age of the house and the percentage of the lower status of the population.

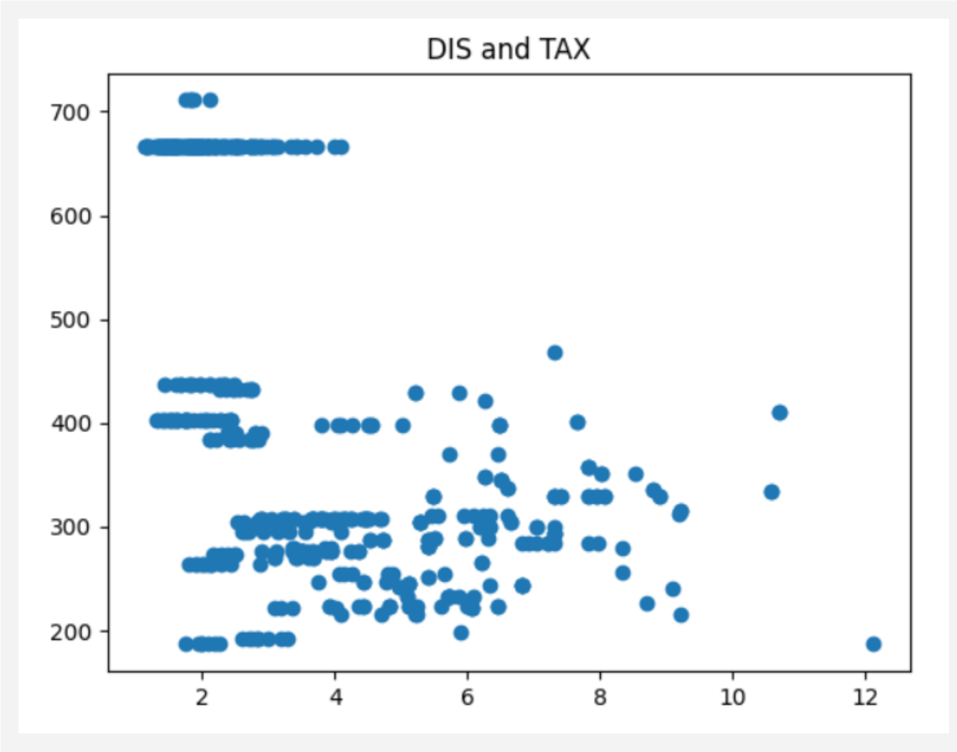
#20



There is a some correlation between the age of the house and the price of the house

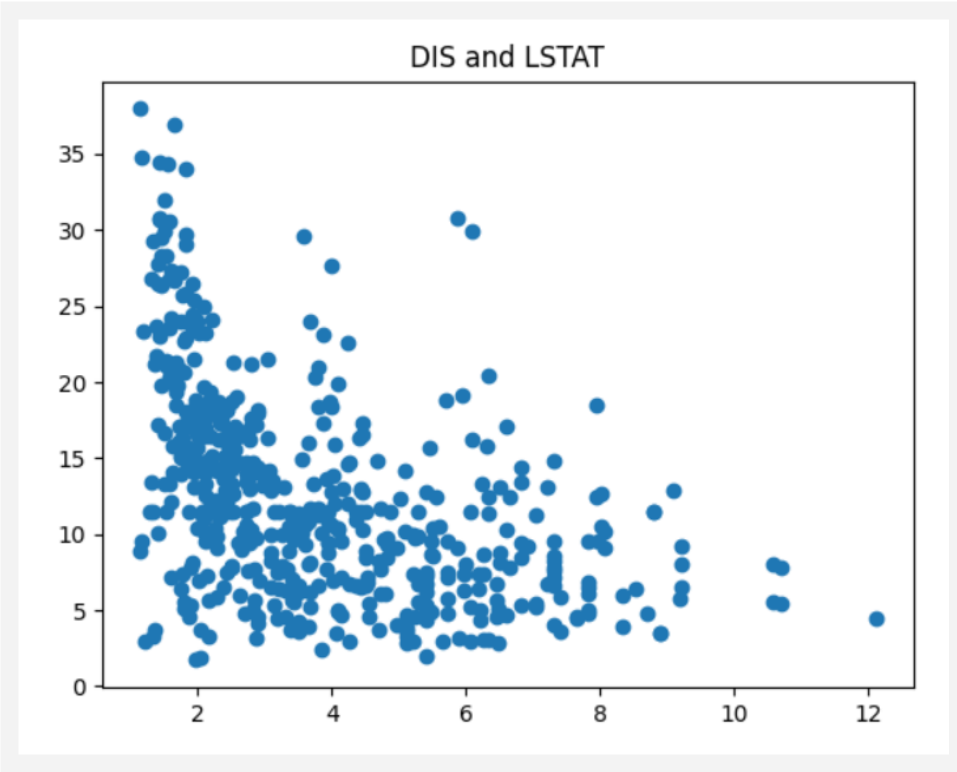
.

#21



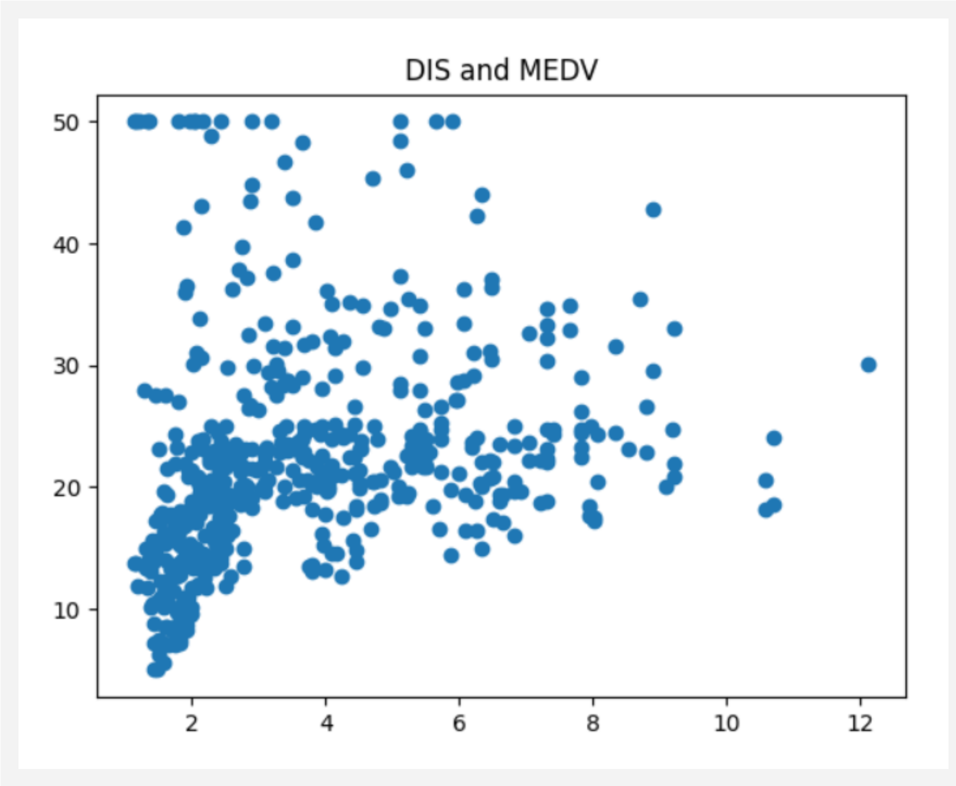
There is no correlation between the distance til work and the tax of the house.

#22



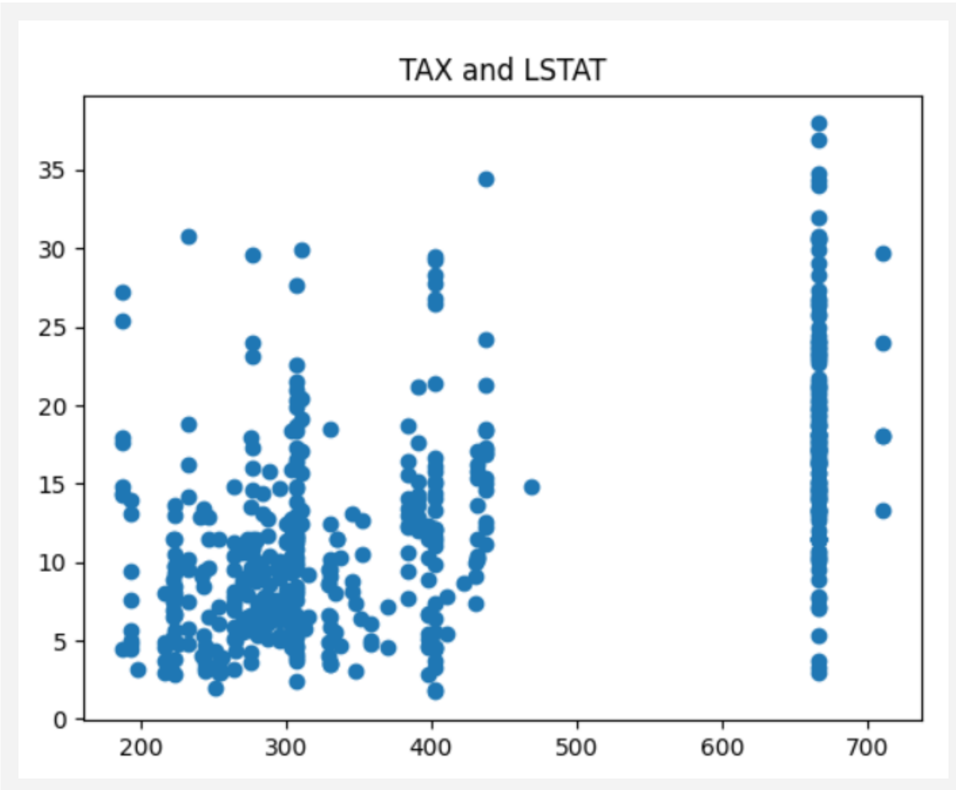
There is correlation between the distance til work and the percentage of the lower status of the population.

#23



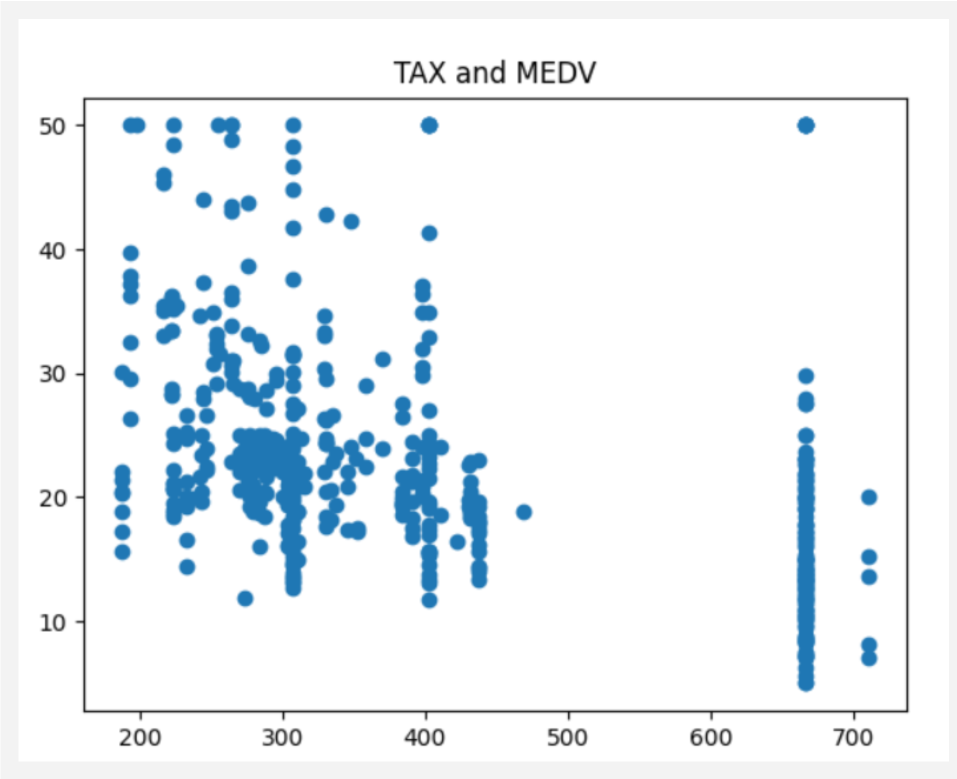
There is some correlation between the distance til work and the price of the house.

#24



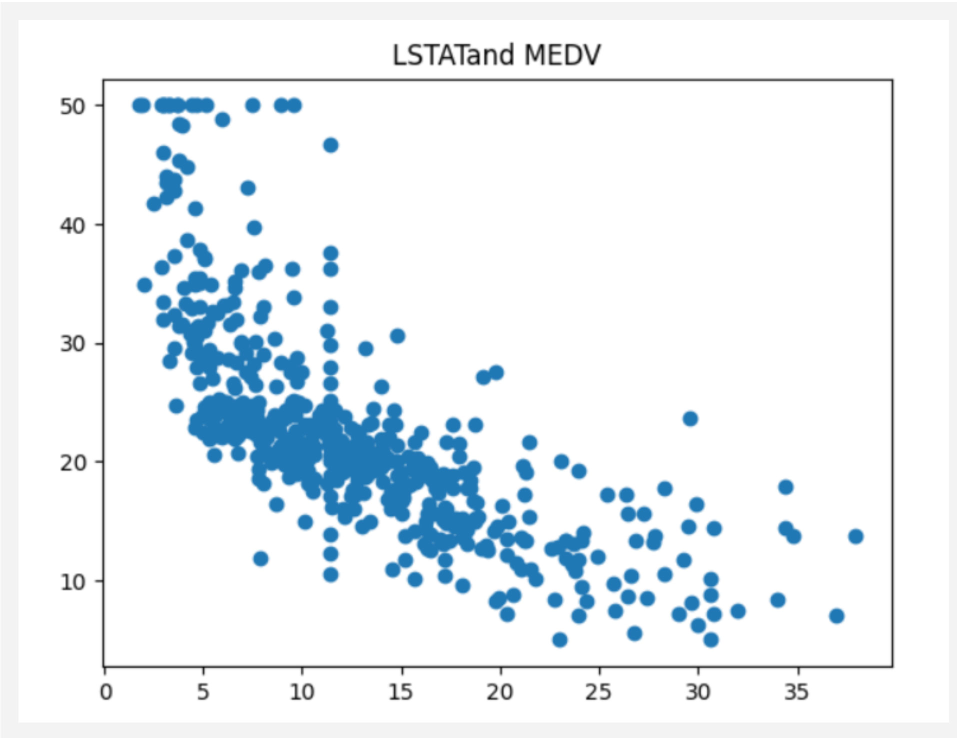
There is some correlation between the tax and the percentage of the lower status of the population.

#26



There is some correlation between the tax and the price of the house.

#27



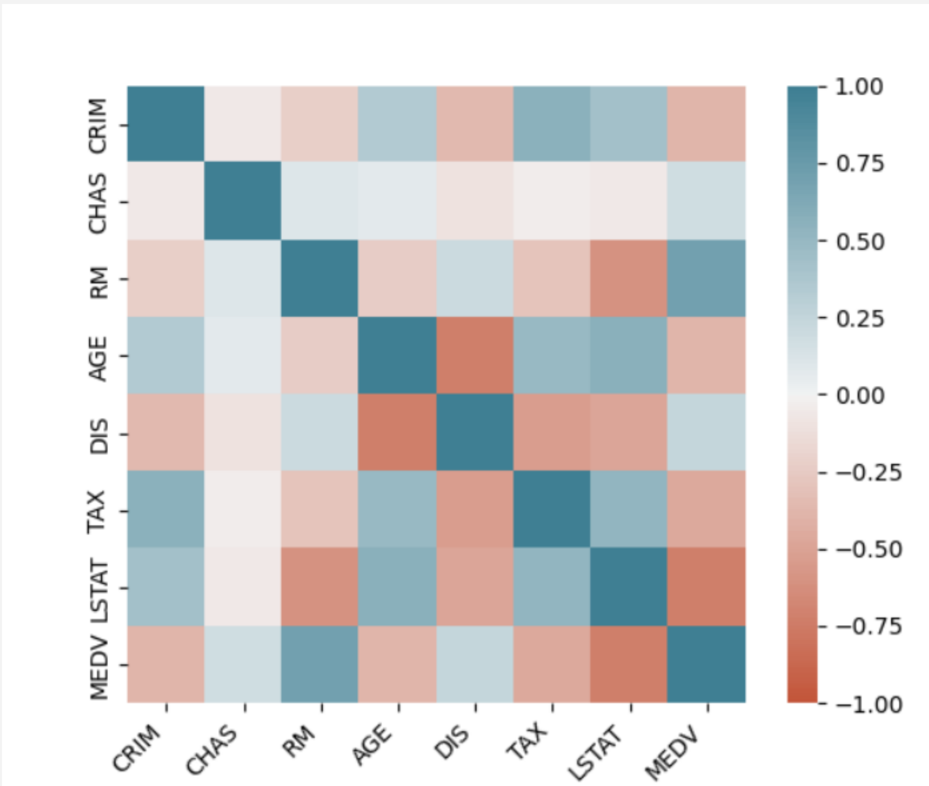
There is some correlation between the house price and the percentage of the lower status of the population.

TASK 9 | *Create the heatmap and add the correlation coefficients to it. What are the 5 strongest correlations that you see? Comment on their sign (only if not done so previously).*

Python code:

```
corr = housingdata.corr()
ax = sns.heatmap(
   corr,
   vmin=-1, vmax=1, center=0,
   cmap=sns.diverging_palette(20, 220, n=200),
   square=True
)
ax.set_xticklabels(
   ax.get_xticklabels(),
   rotation=45,
   horizontalalignment='right'
)
plt.show()
```

Execution:



As we can see from the generate heatmap, the highest correlations, marked with dark a slightly less dark shade of blue are:

1. RM and MEDV (the no. of rooms directly influences the price of the house)

2. AGE and LSTAT (the age of the house influences the percentage of the lower status of the population)
3. TAX and CRIM (the crime rate influences the tax price)
4. LSTAT and RM (the no. of rooms influences the percentage of the lower status of the population)
5. LSTAT and TAX (the tax influences the percentage of the lower status of the population)

TASK 10 | *Based on your hypotheses in (2) choose the variables that you believe could have an impact on the median value of a home. Run a regression with these variables as predictors and medv as the dependent variable. Discuss your results: interpret the coefficients, discuss if the signs of their effects are as you expected, discuss their p-values and the R-squared of the regression.*

Based on my hypothesis, and also based on the calculated correlations, the no. of rooms and the percentage of the lower status of the population have a great impact upon the predicament of the home price. Thus I used those two columns for the regression. The bigger the house - the pricier. The more adults who classify as the lower status of the population - the cheaper the house.

Python code:

```
X = housingdata[['LSTAT','RM']]
y = housingdata[['MEDV']]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,
random_state=0)
linreg = LinearRegression()
linreg = linreg.fit(X_train, y_train)
y_pred = linreg.predict(X_test)

lin_model = pd.DataFrame(y_pred, columns=['Predicted_MEDV'])
lin_model['Actual_MEDV'] = y_test.to_numpy()
print(lin_model.head(20))

print('MSE:', mean_squared_error(y_test, y_pred, squared=True))
print('RMSE:', mean_squared_error(y_test, y_pred, squared=False))
print('R2:', r2_score(y_test, y_pred))
```

Execution:

	Predicted_MEDV	Actual_MEDV
0	26.161342	22.6
1	24.108267	50.0
2	24.302166	23.0
3	12.818233	8.3
4	22.351622	21.2
5	22.784615	19.9
6	21.207158	20.6
7	22.955283	18.7
8	15.503485	16.1
9	24.386050	18.6
10	15.473489	8.8
11	18.778607	17.2
12	19.356526	14.9
13	3.559882	10.5
14	37.473698	50.0
15	31.605440	29.0
16	23.452811	23.0
17	33.337559	33.3
18	28.603666	29.4
19	22.672580	21.0

```
MSE: 35.386752358626055
RMSE: 5.948676521599242
R2: 0.5668643964814106
```

Based on this numbers, the model is capable of predicting the medv.