МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Кафедра «Систем обработки информации и управления»

ОТЧЕТ

Рубежный контроль №1
по курсу «Методы машинного обучения»

ИСПОЛНИТЕЛЬ: _Бабин В.Е._____
ФИО
группа ИУ5-22М _____
подпись
"__"_____2020 г.

ПРЕПОДАВАТЕЛЬ: _____
ФИО
_____
подпись
"__"_____2020 г.

Москва - 2020

РК №1

ИУ5-22М Бабин В.Е.

Номер варианта: 1

Номер задачи: 1

Номер набора данных, указанного в задаче: 1

CRIM per capita crime rate by town

ZN proportion of residential land zoned for lots over 25,000 sq.ft.

INDUS proportion of non-retail business acres per town

CHAS Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)

NOX nitric oxides concentration (parts per 10 million)

RM average number of rooms per dwelling

AGE proportion of owner-occupied units built prior to 1940

DIS weighted distances to five Boston employment centres

RAD index of accessibility to radial highways

TAX full-value property-tax rate per $10,000

PTRATIO pupil-teacher ratio by town

B 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town

LSTAT % lower status of the population

MEDV Median value of owner-occupied homes in $1000’s

In [2]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_boston
```

In [22]:
```
# Boston house-prices dataset
data = load_boston()

dataset = pd.DataFrame(data.data, columns=data.feature_names)
dataset['MEDV'] = data.target
dataset
```

Out[22]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90
...	...	...	...	...	...	...	...	...	...	...	...	...
501	0.06263	0.0	11.93	0.0	0.573	6.593	69.1	2.4786	1.0	273.0	21.0	391.99
502	0.04527	0.0	11.93	0.0	0.573	6.120	76.7	2.2875	1.0	273.0	21.0	396.90
503	0.06076	0.0	11.93	0.0	0.573	6.976	91.0	2.1675	1.0	273.0	21.0	396.90
504	0.10959	0.0	11.93	0.0	0.573	6.794	89.3	2.3889	1.0	273.0	21.0	393.45
505	0.04741	0.0	11.93	0.0	0.573	6.030	80.8	2.5050	1.0	273.0	21.0	396.90

506 rows × 14 columns

In [23]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 506 entries, 0 to 505
Data columns (total 14 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   CRIM     506 non-null    float64
 1   ZN       506 non-null    float64
 2   INDUS    506 non-null    float64
 3   CHAS     506 non-null    float64
 4   NOX      506 non-null    float64
 5   RM       506 non-null    float64
 6   AGE      506 non-null    float64
 7   DIS      506 non-null    float64
 8   RAD      506 non-null    float64
 9   TAX      506 non-null    float64
 10  PTRATIO  506 non-null    float64
 11  B        506 non-null    float64
 12  LSTAT    506 non-null    float64
 13  MEDV     506 non-null    float64
dtypes: float64(14)
memory usage: 55.5 KB
```

In [24]:
```
# Проверим наличие пустых значений
for col in dataset.columns:
    temp_null_count = dataset[dataset[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))
```

```
CRIM - 0
ZN - 0
INDUS - 0
CHAS - 0
NOX - 0
RM - 0
AGE - 0
DIS - 0
RAD - 0
TAX - 0
PTRATIO - 0
B - 0
LSTAT - 0
MEDV - 0
```

Пустые значения в датасете отсутствуют. Для устранения пустых значений можно было бы: 1) удалить строки и столбцы с пустыми значениями; 2) проставить нулевые значения; 3) Сделать "внедрение значений" - импьютацию

In [25]:
```
# Основные статистические характеристки набора данных
dataset.describe()
```

Out[25]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	50
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284634	68.574901	
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617	28.148861	
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.900000	
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500	45.025000	
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500	77.500000	
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.623500	94.075000	
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000000	1

In [26]:
```
# Корреляционный анализ
dataset.corr()
```

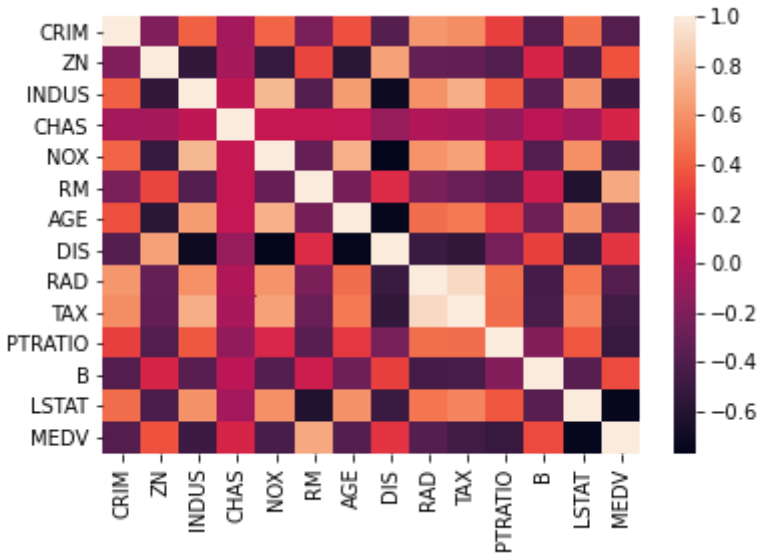
Out[26]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS
CRIM	1.000000	-0.200469	0.406583	-0.055892	0.420972	-0.219247	0.352734	-0.379670
ZN	-0.200469	1.000000	-0.533828	-0.042697	-0.516604	0.311991	-0.569537	0.664408
INDUS	0.406583	-0.533828	1.000000	0.062938	0.763651	-0.391676	0.644779	-0.708027
CHAS	-0.055892	-0.042697	0.062938	1.000000	0.091203	0.091251	0.086518	-0.099176
NOX	0.420972	-0.516604	0.763651	0.091203	1.000000	-0.302188	0.731470	-0.769230
RM	-0.219247	0.311991	-0.391676	0.091251	-0.302188	1.000000	-0.240265	0.205246
AGE	0.352734	-0.569537	0.644779	0.086518	0.731470	-0.240265	1.000000	-0.747881
DIS	-0.379670	0.664408	-0.708027	-0.099176	-0.769230	0.205246	-0.747881	1.000000
RAD	0.625505	-0.311948	0.595129	-0.007368	0.611441	-0.209847	0.456022	-0.494588
TAX	0.582764	-0.314563	0.720760	-0.035587	0.668023	-0.292048	0.506456	-0.534432
PTRATIO	0.289946	-0.391679	0.383248	-0.121515	0.188933	-0.355501	0.261515	-0.232471
B	-0.385064	0.175520	-0.356977	0.048788	-0.380051	0.128069	-0.273534	0.291512
LSTAT	0.455621	-0.412995	0.603800	-0.053929	0.590879	-0.613808	0.602339	-0.496996
MEDV	-0.388305	0.360445	-0.483725	0.175260	-0.427321	0.695360	-0.376955	0.249929

In [27]:
```
# Тепловая диаграмма корреляции
sns.heatmap(dataset.corr())
```

Out[27]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2d9492d0080>
```

Out[27]:

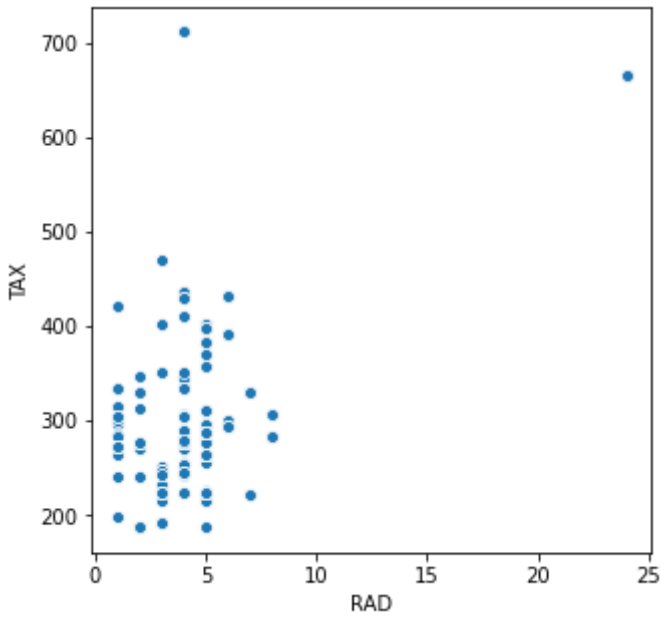


С помощью полученной диаграммы корреляции и таблицы с данными о корреляции, можно заключить, что наиболее коррелирующими из всего датасета являются показатели RAD (index of accessibility to radial highways) и TAX (full-value property-tax rate per $10,000)

In [9]:
```
# Диаграмма рассеивания для этих показателей
fig, ax = plt.subplots(figsize=(5,5))
sns.scatterplot(ax=ax, x='RAD', y='TAX', data=dataset)
```

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x2d94924eef0>

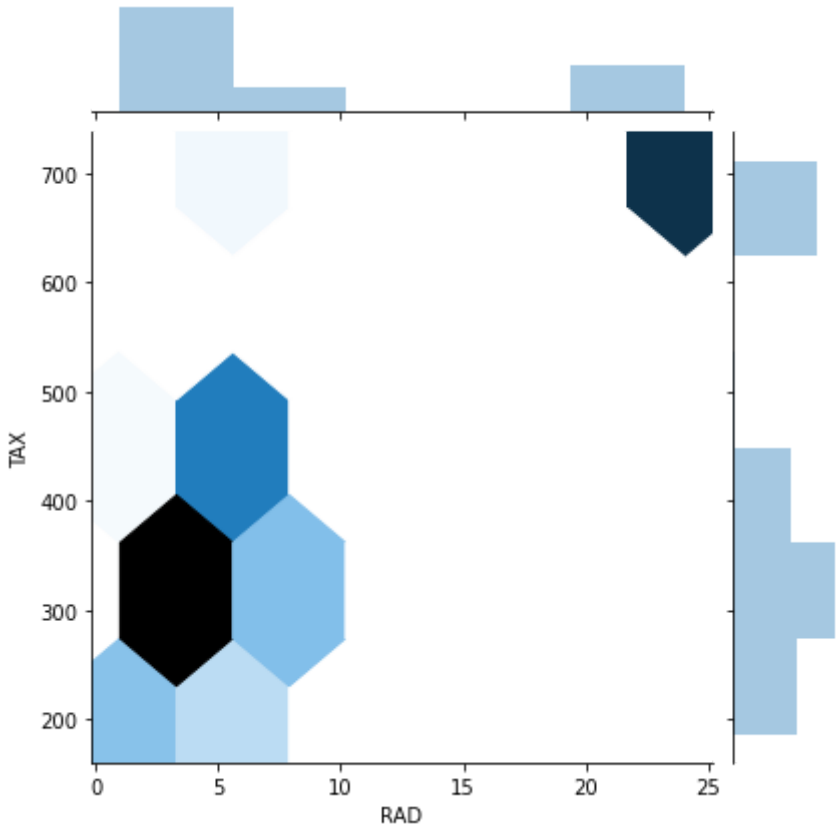
Out[9]:



In [23]:
```
sns.jointplot(x='RAD', y='TAX', data=dataset, kind="hex")
```

Out[23]: <seaborn.axisgrid.JointGrid at 0x1b46b20efd0>

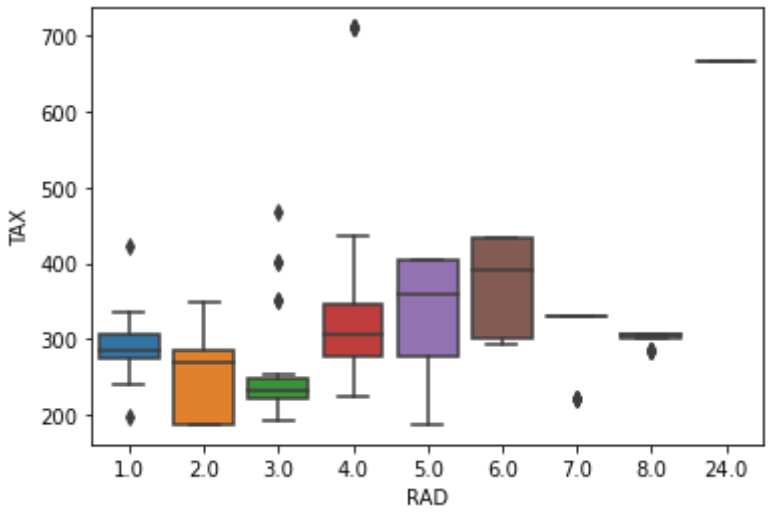
Out[23]:



In [8]:
```
sns.boxplot(x=dataset["RAD"], y=dataset["TAX"])
```

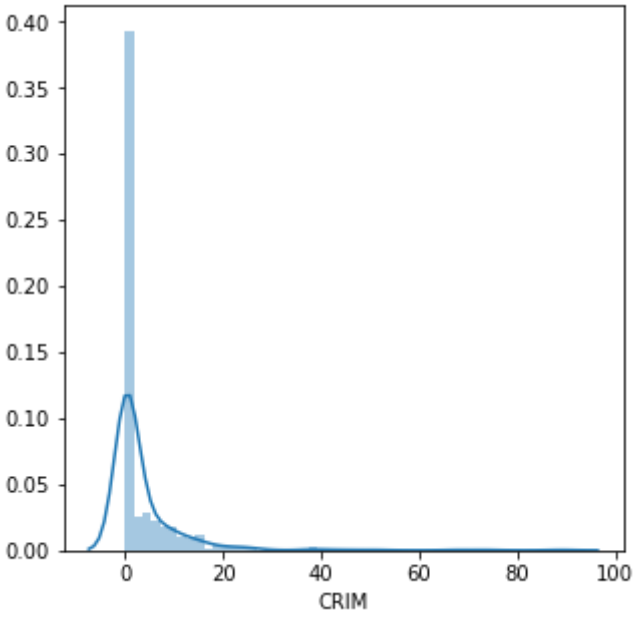
Out[8]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2d94913ec18>
```

Out[8]:

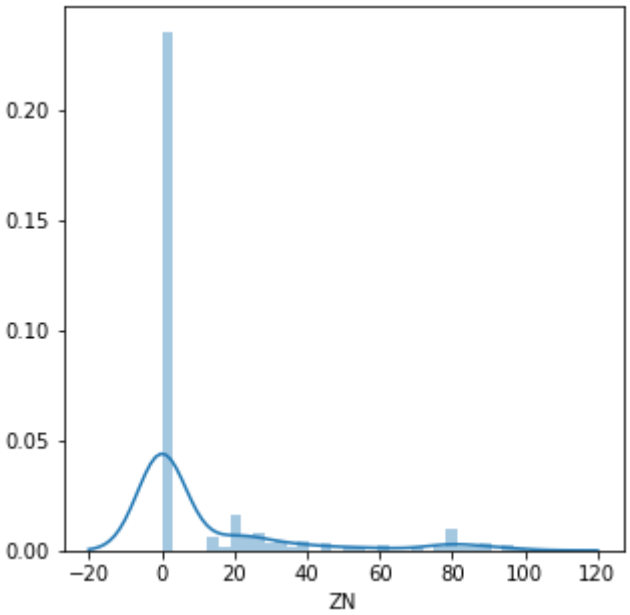


In [29]:
```
# Рассмотрим распределение показателей
for column in dataset.columns:
    fig, ax = plt.subplots(figsize=(5,5))
    sns.distplot(dataset[column])
```

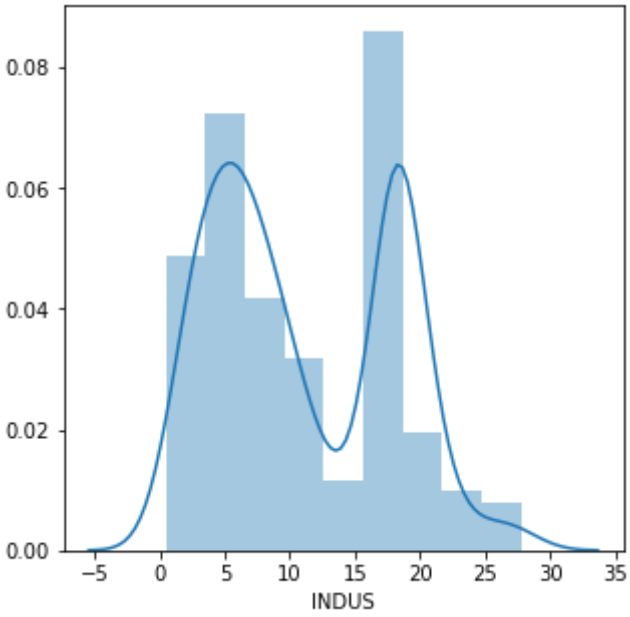
Out[29]:


0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
20
40
60
80
100
CRIM

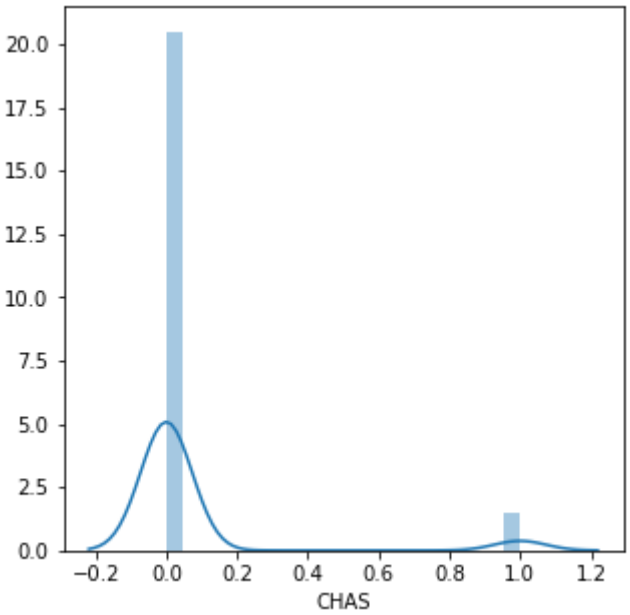
Out[29]:


0.20
0.15
0.10
0.05
0.00
−20
0
20
40
60
80
100
120
ZN

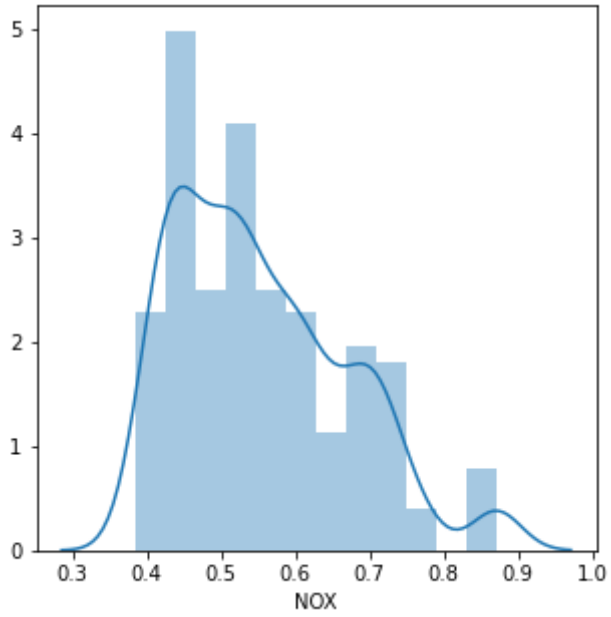
Out[29]:


0.08
0.06
0.04
0.02
0.00
−5
0
5
10
15
20
25
30
35
INDUS

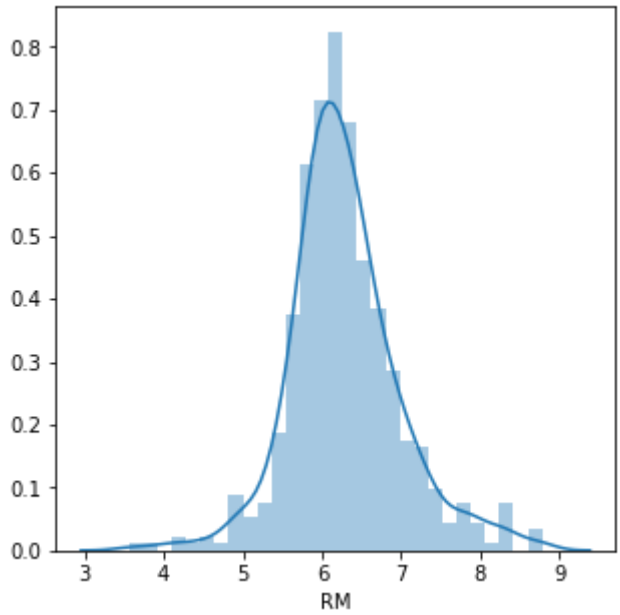
Out[29]:


20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
CHAS

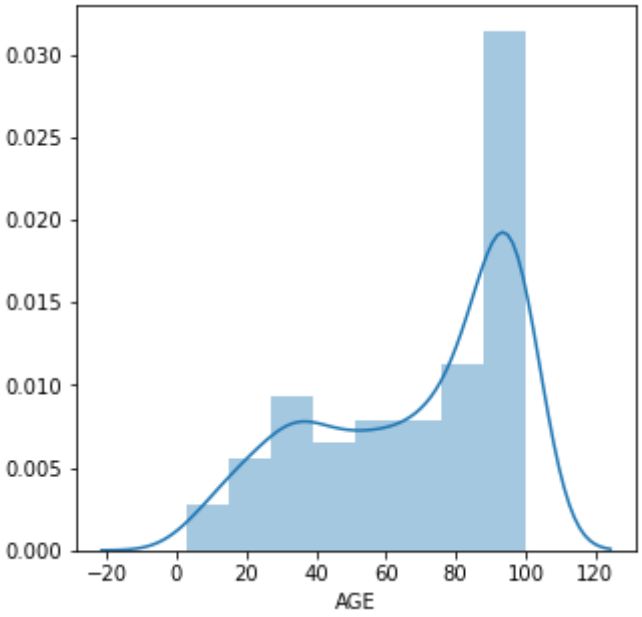
Out[29]:


5
4
3
2
1
0
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
NOX

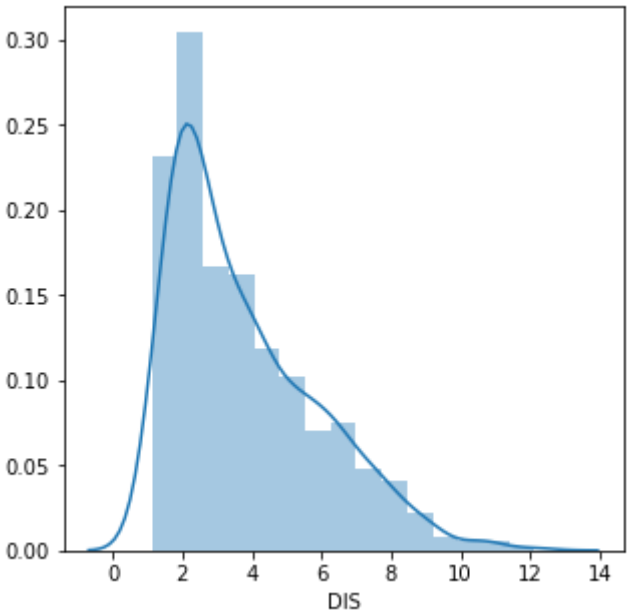
Out[29]:


0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
3
4
5
6
7
8
9
RM

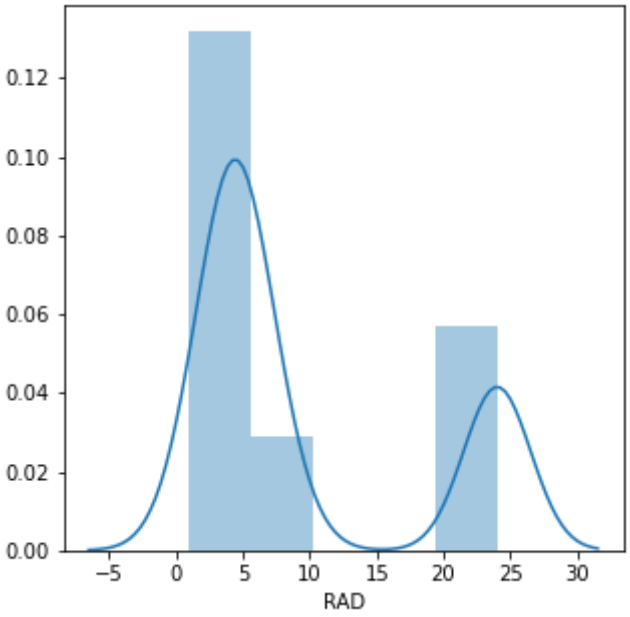
Out[29]:


0.030
0.025
0.020
0.015
0.010
0.005
0.000
−20
0
20
40
60
80
100
120
AGE

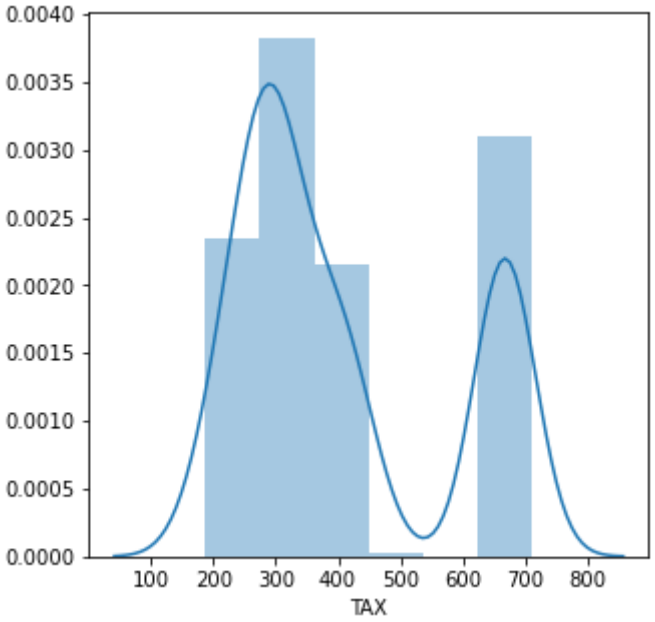
Out[29]:


0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
2
4
6
8
10
12
14
DIS

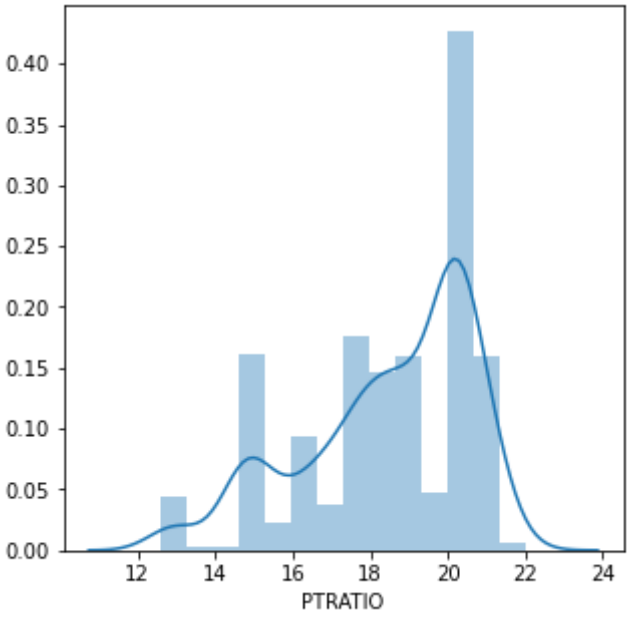
Out[29]:


0.12
0.10
0.08
0.06
0.04
0.02
0.00
−5
0
5
10
15
20
25
30
RAD

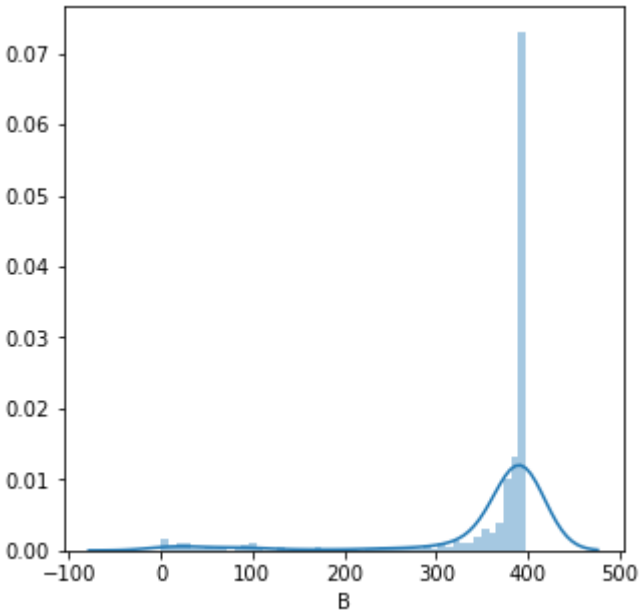
Out[29]:


0.0040
0.0035
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
100
200
300
400
500
600
700
800
TAX

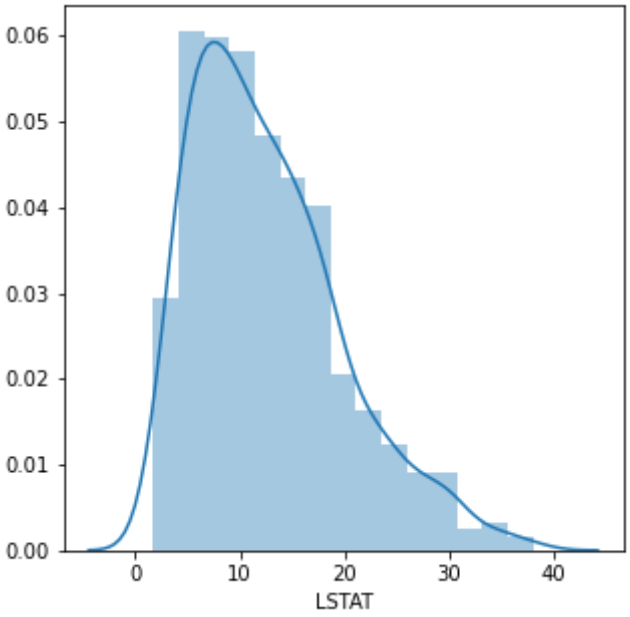
Out[29]:


0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
12
14
16
18
20
22
24
PTRATIO

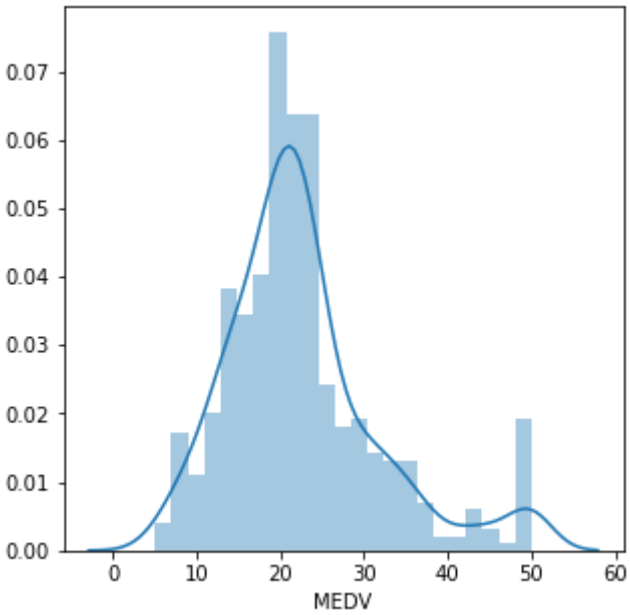
Out[29]:


0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
−100
0
100
200
300
400
500
B

Out[29]:



Out[29]:

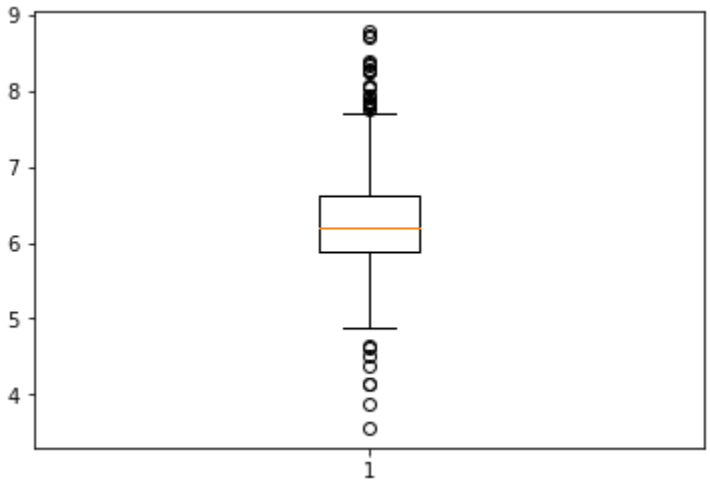


Видно, что показатель RM (average number of rooms per dwelling) является наиболее близким к нормальному распределению

In [27]:
```
plt.boxplot(dataset['RM'])
```

Out[27]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1b46ce29d30>,
  <matplotlib.lines.Line2D at 0x1b46ce360f0>],
 'caps': [<matplotlib.lines.Line2D at 0x1b46ce36470>,
  <matplotlib.lines.Line2D at 0x1b46ce367f0>],
 'boxes': [<matplotlib.lines.Line2D at 0x1b46ce29a20>],
 'medians': [<matplotlib.lines.Line2D at 0x1b46ce36b70>],
 'fliers': [<matplotlib.lines.Line2D at 0x1b46ce36ef0>],
 'means': []}
```

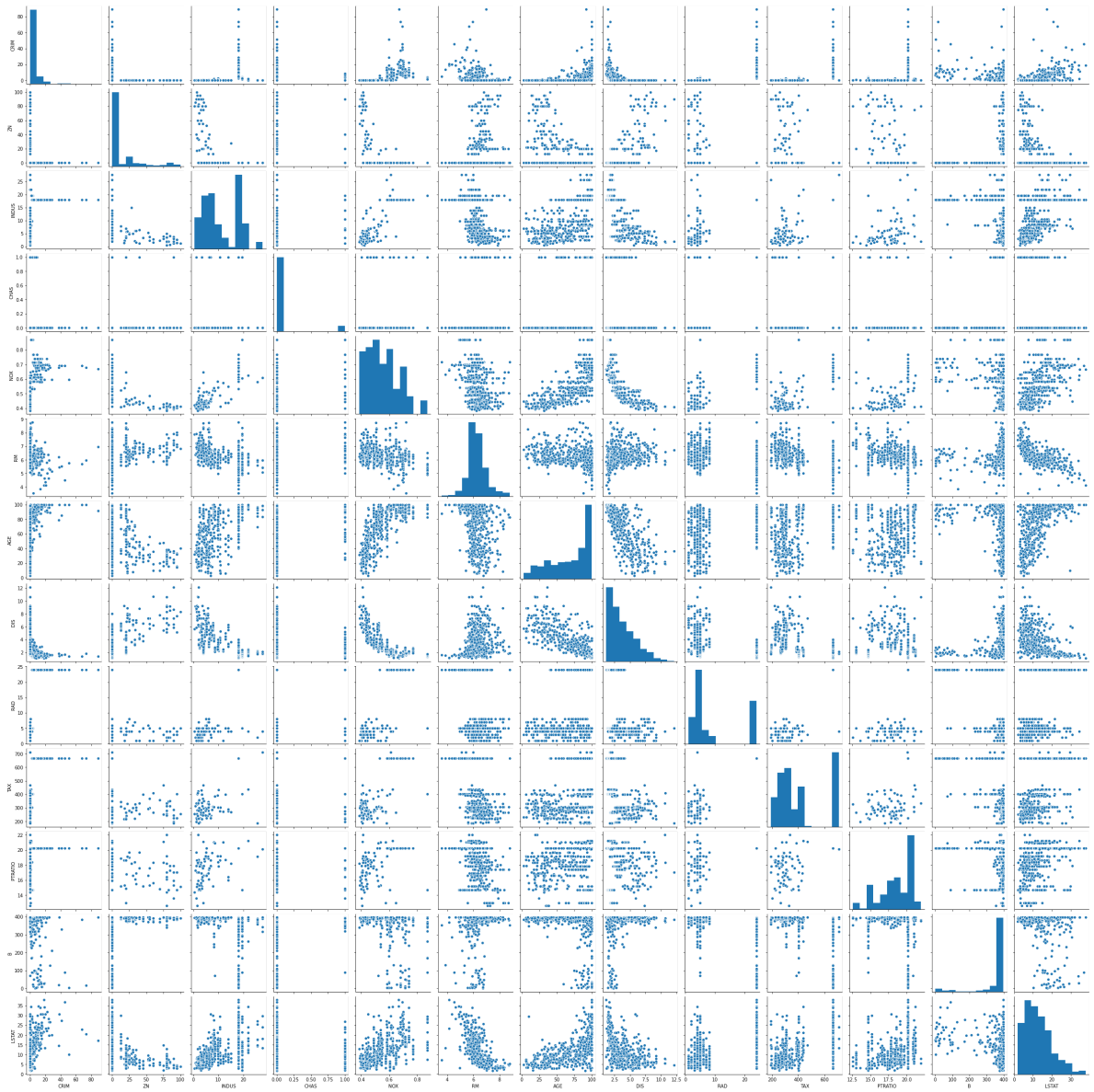
Out[27]:



In [14]:
```
# Парные диаграммы
sns.pairplot(dataset)
```

Out[14]:
```
<seaborn.axisgrid.PairGrid at 0x1b461fb7e80>
```

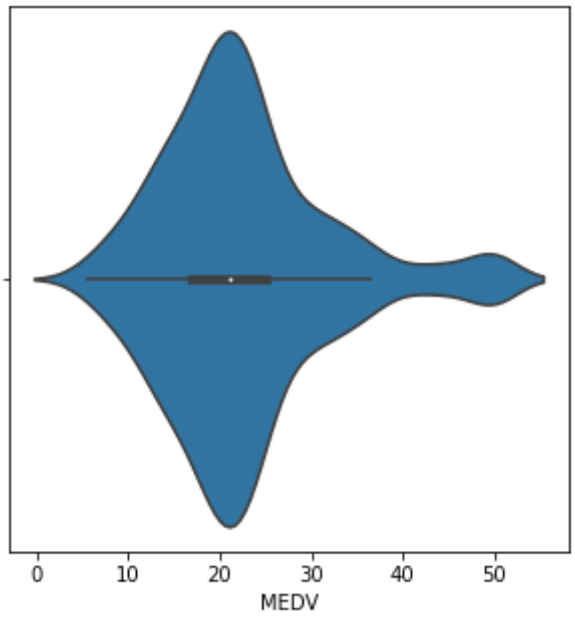
Out[14]:



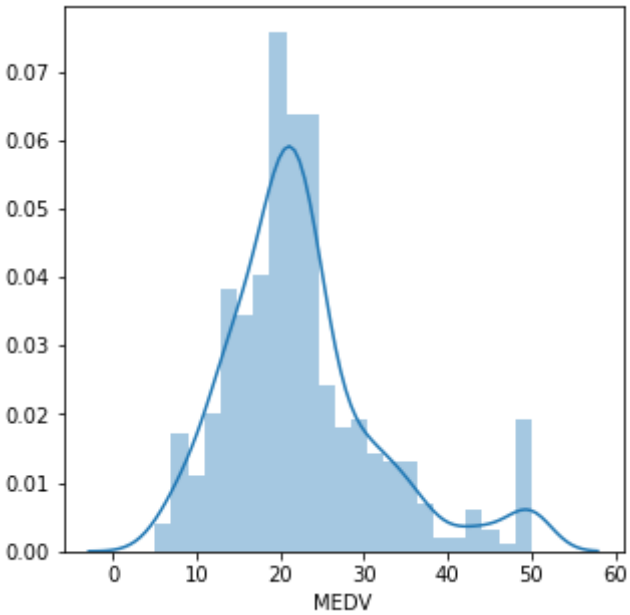
In [39]:
```
# Violin plot и гистограмма (доп задание для ИУ5-22М) для целевого призна
ка MEDV (средняя стоимость домов в 1000 долл. США)
fig, ax = plt.subplots(figsize=(5,5))
sns.violinplot(x=dataset['MEDV'])

fig, ax = plt.subplots(figsize=(5,5))
sns.distplot(dataset['MEDV'])
```

Out[39]: <matplotlib.axes._subplots.AxesSubplot at 0x2d9495e0cf8>

Out[39]:



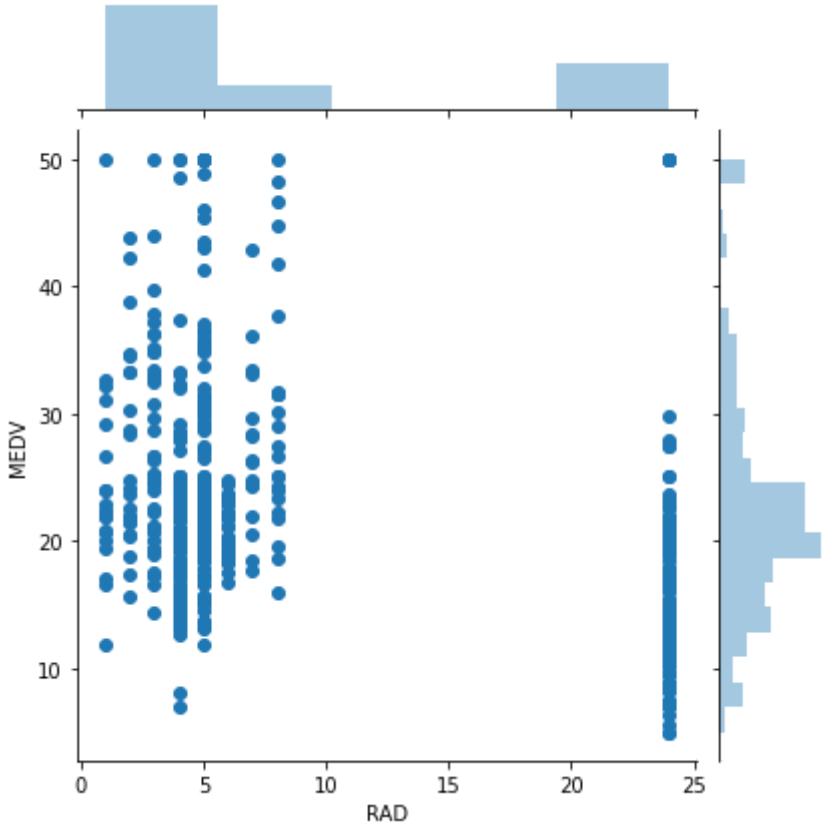
Out[39]:



In [41]:
```
# зависимость индекса доступности к радиальным магистралям от средней стоимости домов в 1000 долл. США
sns.jointplot(x='RAD', y='MEDV', data=dataset)
```

Out[41]: <seaborn.axisgrid.JointGrid at 0x2d9499ab390>

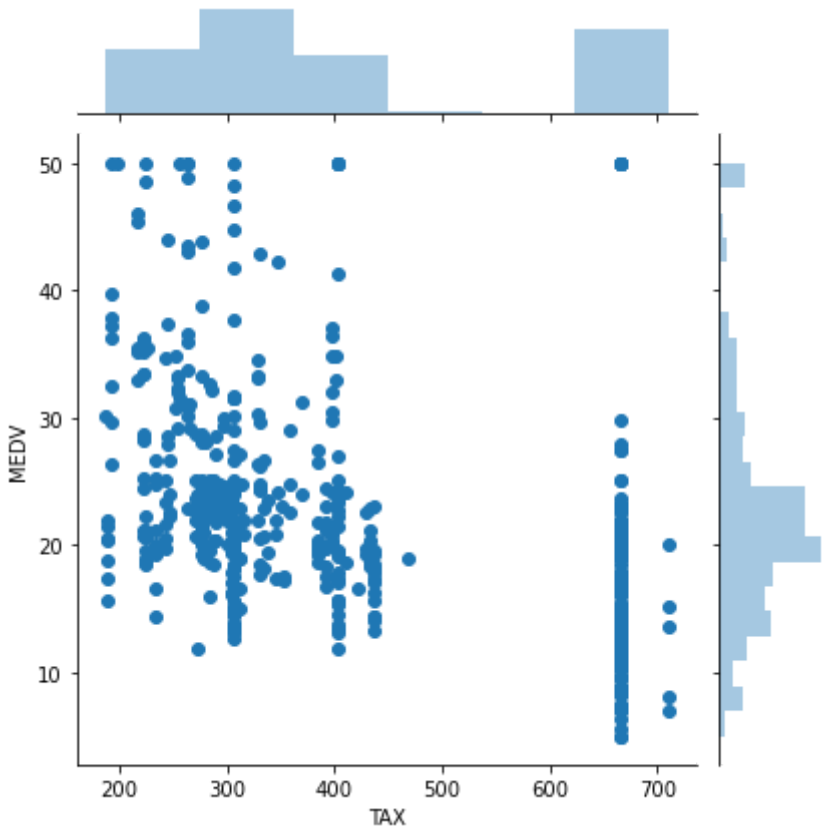
Out[41]:



In [42]:
```
# зависимость ставки налога на полную стоимость имущества за 10 000 долл. США от средней стоимости домов в 1000 долл. США
sns.jointplot(x='TAX', y='MEDV', data=dataset)
```

Out[42]: <seaborn.axisgrid.JointGrid at 0x2d94ab796d8>

Out[42]:



В ходе выполнения РК1 был проведен разведочный анализ данных о стоимости домов в районе Бостона. Были построены графики: violin plot (по краям отображаются распределения плотности) - средняя стоимость домов в районе Бостона = 20000$, гистограммы распределения (позволяет оценить плотность вероятности распределения данных) - показатель ср кол-ва комнат в доме близок к нормальному распределению, диаграмма рассеивания (позволяет построить распределение двух колонок данных и визуально обнаружить наличие зависимости), диаграмма "ящик с усами" (показывает одномерное распределение вероятности) - ср кол-во комнат в доме близко к норм распред, тепловая диаграмма корреляции (демонтрирует зависимости между признаками набора данных) - наиболее зависимыми оказались "индекс доступности к радиальным магистралям" и "ставка налога на полную стоимость имущества за 10 000 долл. США."