МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Кафедра «Систем обработки информации и управления»

ОТЧЕТ

Лабораторная работа № 1
по курсу «Методы машинного обучения»

Тема: «Разведочный анализ данных. Исследование и визуализация данных»

ИСПОЛНИТЕЛЬ: Паршева А. М.

группа ИУ5-22М _____
подпись

"__"_____201_ г.

ПРЕПОДАВАТЕЛЬ: _____
ФИО

_____
подпись

"__"_____201_ г.

Москва - 2020

Лабораторная работа №1. Разведочный анализ данных. Исследование и визуализация данных.

Задание

1. Выбрать набор данных (датасет).
2. Создать ноутбук, который содержит следующие разделы:
    - Текстовое описание выбранного набора данных.
    - Основные характеристики датасета.
    - Визуальное исследование датасета. Необходимо использовать не менее 2 различных библиотек и не менее 5 графиков.
    - Информация о корреляции признаков.
3. Сформировать отчет и разместить его в своем репозитории на github.

Текстовое описание датасета

In [1]:

```
from sklearn.datasets import load_boston
import numpy as np
import pandas as pd
```

In [2]:

```
data = load_boston()
```

Для выполнения данной лабораторной работы использовался датасет load_boston, который включает в себя, следующие данные:

- **CRIM** уровень приступности на душу населения
- **ZN** доля жилой земли зонированая под участки более 25 000 кв.
- **INDUS** доля неторговых площадей на город
- **CHAS** фиктивная переменная (= 1, если тракт ограничивает реку; 0 в противном случае)
- **NOX** концентрация оксидов азота (частей на 10 миллионов)
- **RM** среднее количество комнат в доме
- **AGE** доля домов, построенных до 1940 года
- **DIS** взвешенные расстояния до пяти бостонских центров занятости
- **RAD** индекс доступности к радиальным магистралям
- **TAX** ставка налога на полную стоимость имущества за 10 000 долл. США
- **PTRATIO** соотношение учеников и учителей по городам
- **B** 1000 (Bk - 0,63) ^ 2, где Bk - доля чернокожих по городам
- **LSTAT** % низкого статуса среди населения
- **MEDV** средняя стоимость домов в 1000 долл. США

In [3]:

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

Основные характеристики датасета

In [4]:

```
rows = df.shape[0]
columns = df.shape[1]

"Данный датасет содержит %d строк и %d столбцов." % (rows, col
umns)
```

Out[4]:

```
'Данный датасет содержит 506 строк и 14 столбцов.'
```

In [5]:

```
df.describe()
```

Out[5]:

	CRIM	ZN	INDUS	CHAS	NOX
count	506.000000	506.000000	506.000000	506.000000	506.000000
mean	3.613524	11.363636	11.136779	0.069170	0.554695
std	8.601545	23.322453	6.860353	0.253994	0.115878
min	0.006320	0.000000	0.460000	0.000000	0.385000
25%	0.082045	0.000000	5.190000	0.000000	0.449000
50%	0.256510	0.000000	9.690000	0.000000	0.538000
75%	3.677083	12.500000	18.100000	0.000000	0.624000
max	88.976200	100.000000	27.740000	1.000000	0.871000

In [6]:

```
'Данный датасет содержит столбцы: ' + ', '.join(df.columns)
```

Out[6]:

```
'Данный датасет содержит столбцы: CRIM, ZN, INDUS,
CHAS, NOX, RM, AGE, DIS, RAD, TAX, PTRATIO, B, LST
AT, MEDV'
```

In [7]:

```
df.dtypes
```

Out[7]:

```
CRIM        float64
ZN          float64
INDUS       float64
CHAS        float64
NOX         float64
RM          float64
AGE         float64
DIS         float64
RAD         float64
TAX         float64
PTRATIO     float64
B           float64
LSTAT       float64
MEDV        float64
dtype: object
```

Визуализация данных

Для визуализации данных использовались библиотеки:

1. seaborn
2. plotly
3. matplotlib

In [ ]:

```
import seaborn as sns
import plotly as py
import plotly.express as px
import matplotlib.pyplot as plt
```

При выполнениии лабораторной работы были построены следующие виды диаграмм:

1. Парные диаграммы
2. Диаграммы рассеивания
3. Гистограммы
4. "Ящик с усами"
5. Joinplot

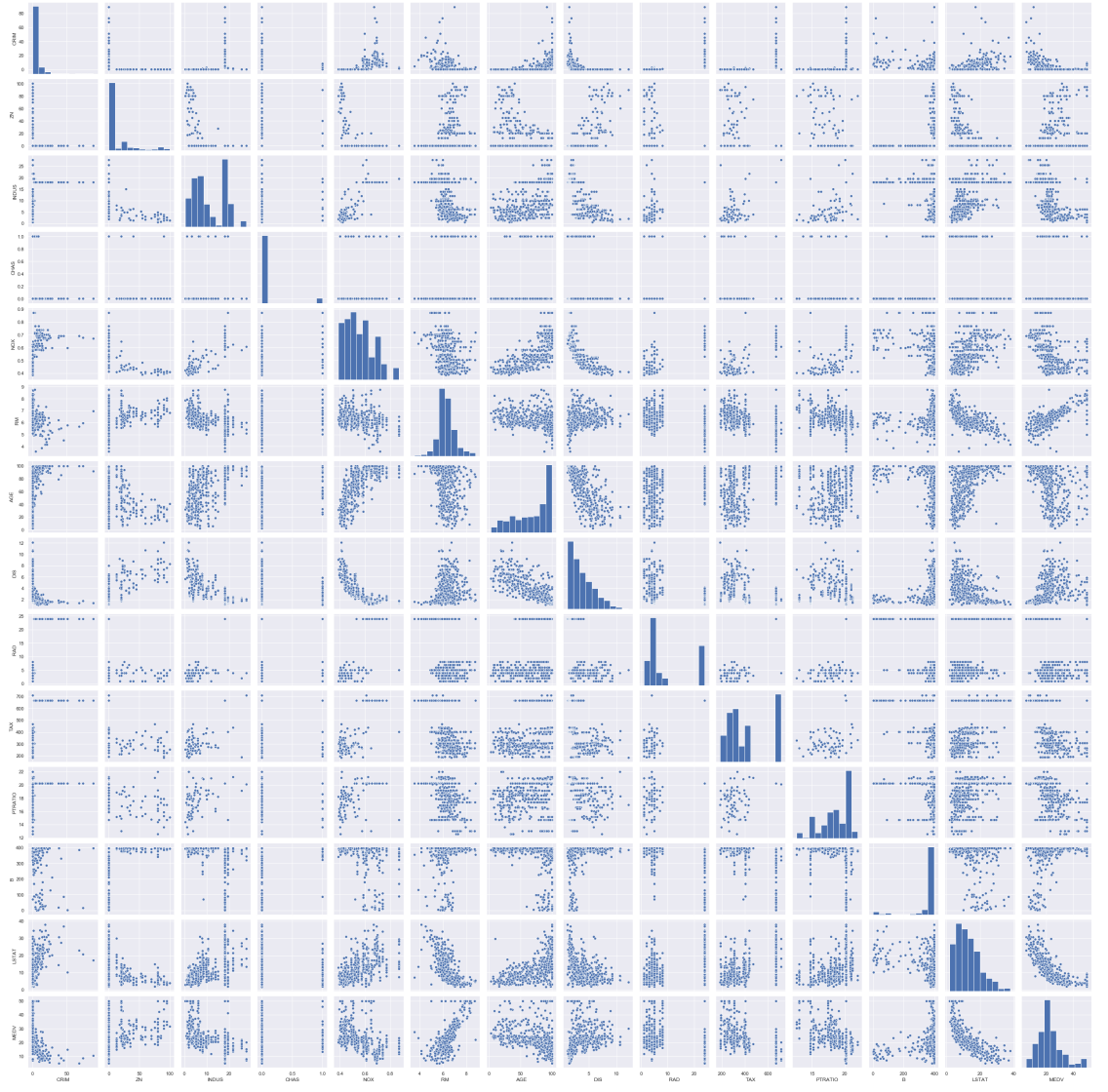
Парные диаграммы

In [39]:

```
sns.pairplot(df)
```

Out[39]:

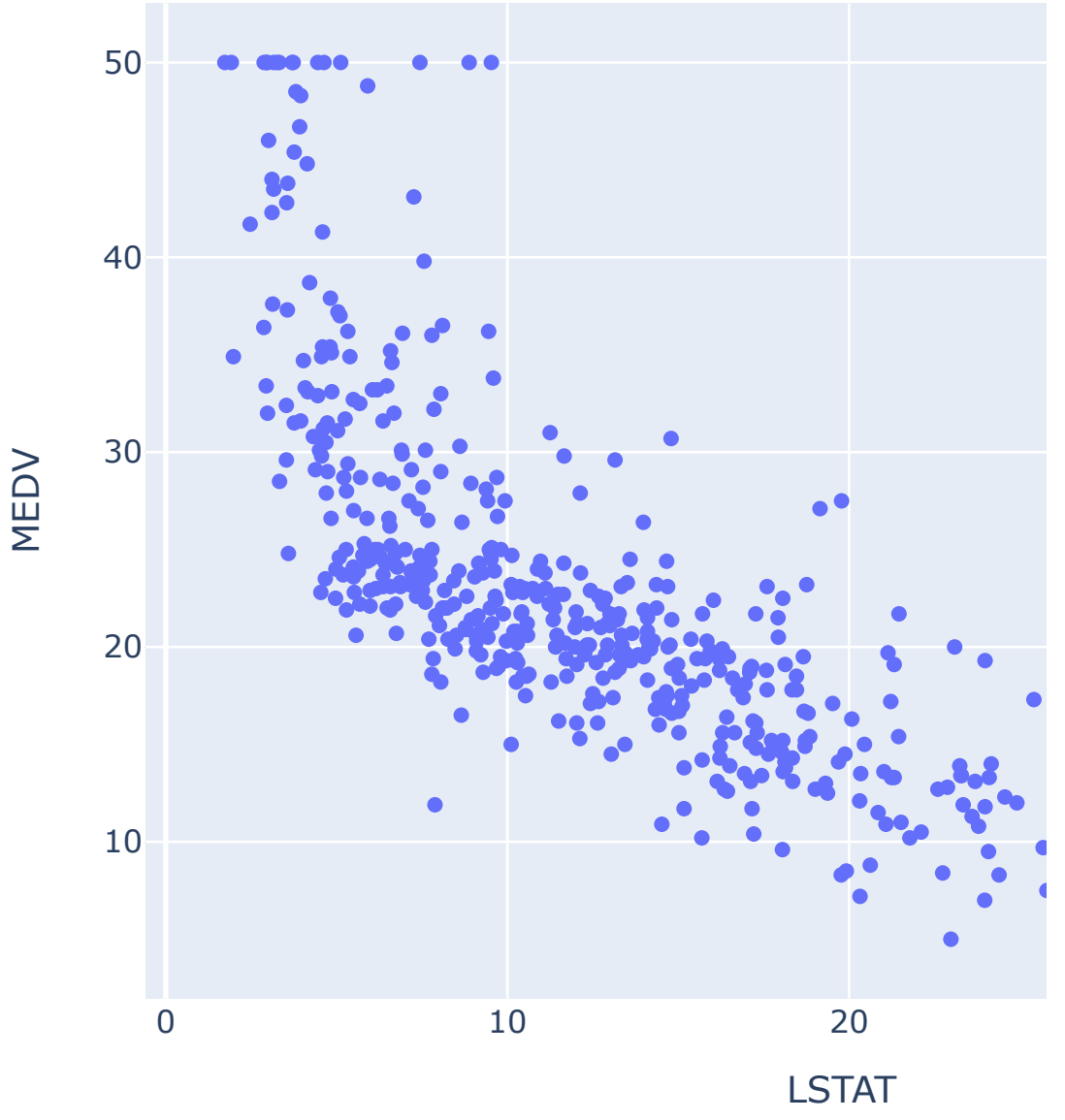
```
<seaborn.axisgrid.PairGrid at 0x132506908>
```



Диаграммы рассеивания

In [10]:

```
py.offline.init_notebook_mode(connected=True)
fig = px.scatter(df, x = 'LSTAT', y='MEDV')
fig.show()
```

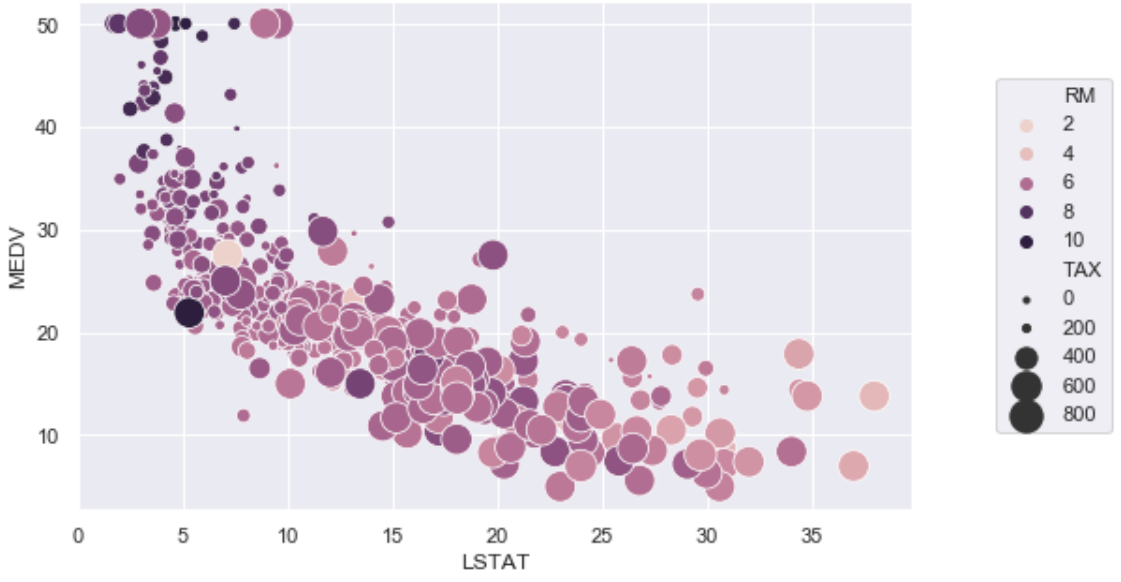


In [83]:

```
plt.figure(figsize=(8, 5))
ax = sns.scatterplot(x="LSTAT", y="MEDV",
                     hue="RM", size="TAX",
                     sizes=(10, 300),
                     data=df)
ax.legend(loc='center right', ncol=1,bbox_to_anchor=(1.25, 0.5
))
```

Out[83]:

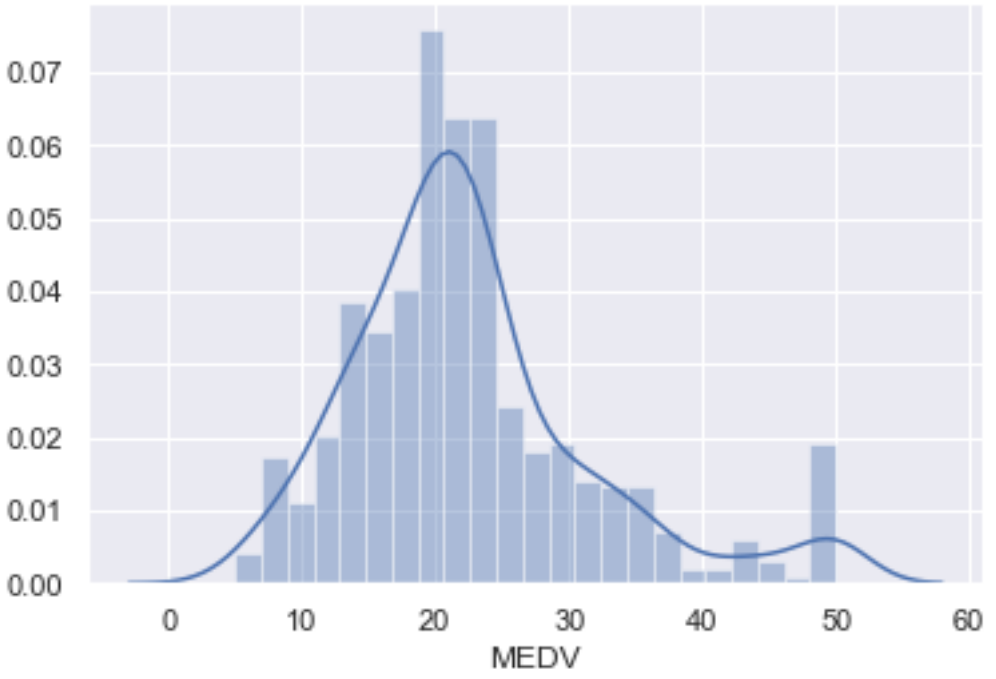
```
<matplotlib.legend.Legend at 0x13a1a5128>
```



Гистограмма

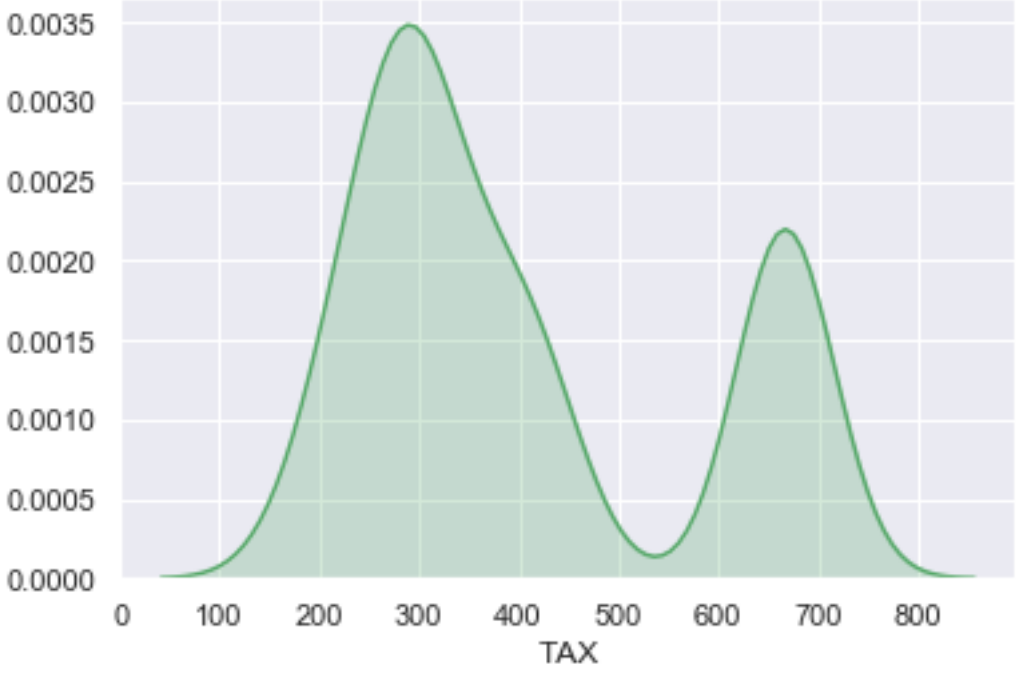
In [38]:

```
sns_plot = sns.distplot(df['MEDV'])
fig = sns_plot.get_figure()
```



In [87]:

```
sns_plot = sns.distplot(df['TAX'], hist=False, color="g", kde_
kws={"shade": True})
fig = sns_plot.get_figure()
```



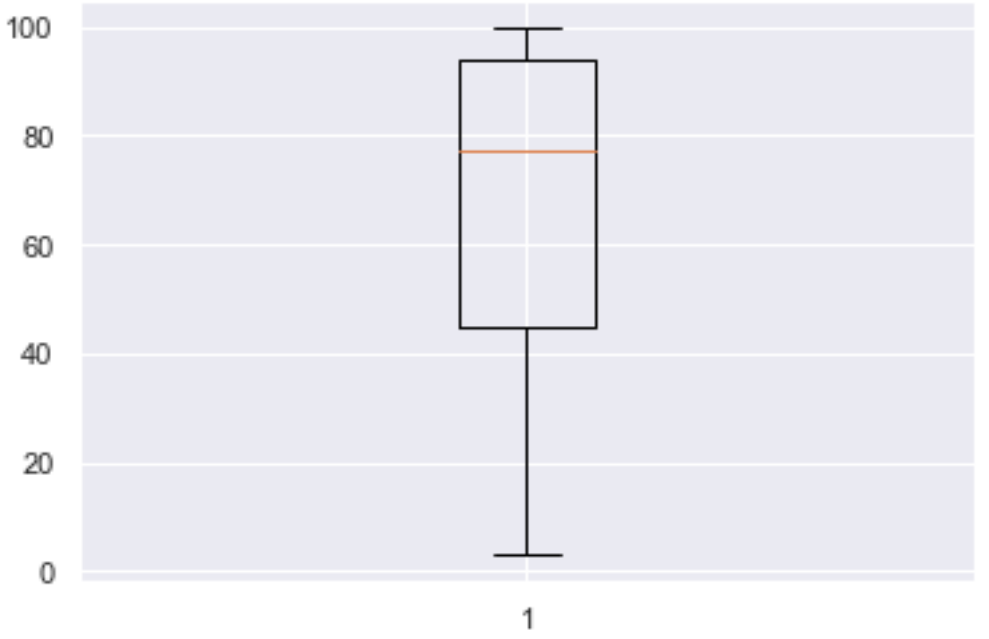
Ящик с усами

In [90]:

```
plt.boxplot(df['AGE'])
```

Out[90]:

```
{'whiskers': [<matplotlib.lines.Line2D at 0x138ef2
470>,
  <matplotlib.lines.Line2D at 0x138ef27f0>],
 'caps': [<matplotlib.lines.Line2D at 0x138ef2b70>
,
  <matplotlib.lines.Line2D at 0x138ef2ef0>],
 'boxes': [<matplotlib.lines.Line2D at 0x138ef2048
>],
 'medians': [<matplotlib.lines.Line2D at 0x138f072
b0>],
 'fliers': [<matplotlib.lines.Line2D at 0x138f0763
0>],
 'means': []}
```



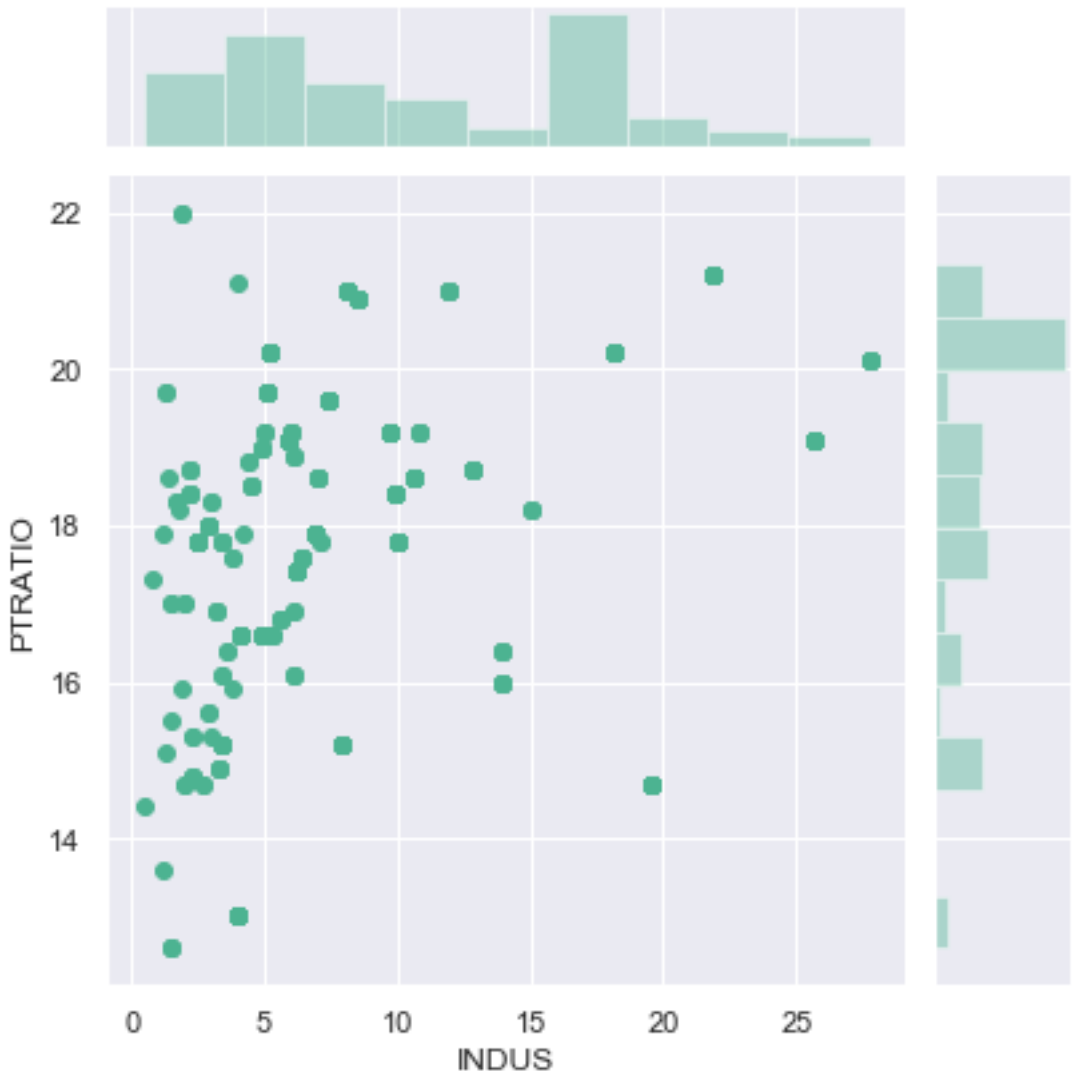
Joinplot

In [91]:

```
sns.jointplot(x='INDUS', y='PTRATIO',data = df, color="#4CB391
")
```

Out[91]:

```
<seaborn.axisgrid.JointGrid at 0x13bd83cc0>
```



Корреляционный анализ

In [92]:

```
corr_matrix_p = df.corr(method='pearson').round(2)
corr_matrix_p
```

Out[92]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RA
CRIM	1.00	-0.20	0.41	-0.06	0.42	-0.22	0.35	-0.38	0.6
ZN	-0.20	1.00	-0.53	-0.04	-0.52	0.31	-0.57	0.66	-0.3
INDUS	0.41	-0.53	1.00	0.06	0.76	-0.39	0.64	-0.71	0.6
CHAS	-0.06	-0.04	0.06	1.00	0.09	0.09	0.09	-0.10	-0.0
NOX	0.42	-0.52	0.76	0.09	1.00	-0.30	0.73	-0.77	0.6
RM	-0.22	0.31	-0.39	0.09	-0.30	1.00	-0.24	0.21	-0.2
AGE	0.35	-0.57	0.64	0.09	0.73	-0.24	1.00	-0.75	0.4
DIS	-0.38	0.66	-0.71	-0.10	-0.77	0.21	-0.75	1.00	-0.4
RAD	0.63	-0.31	0.60	-0.01	0.61	-0.21	0.46	-0.49	1.0
TAX	0.58	-0.31	0.72	-0.04	0.67	-0.29	0.51	-0.53	0.9
PTRATIO	0.29	-0.39	0.38	-0.12	0.19	-0.36	0.26	-0.23	0.4
B	-0.39	0.18	-0.36	0.05	-0.38	0.13	-0.27	0.29	-0.4
LSTAT	0.46	-0.41	0.60	-0.05	0.59	-0.61	0.60	-0.50	0.4
MEDV	-0.39	0.36	-0.48	0.18	-0.43	0.70	-0.38	0.25	-0.3

In [93]:

```
corr_matrix_k = df.corr(method='kendall').round(2)
corr_matrix_k
```

Out[93]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RA
CRIM	1.00	-0.46	0.52	0.03	0.60	-0.21	0.50	-0.54	0.5
ZN	-0.46	1.00	-0.54	-0.04	-0.51	0.28	-0.43	0.48	-0.2
INDUS	0.52	-0.54	1.00	0.08	0.61	-0.29	0.49	-0.57	0.3
CHAS	0.03	-0.04	0.08	1.00	0.06	0.05	0.06	-0.07	0.0
NOX	0.60	-0.51	0.61	0.06	1.00	-0.22	0.59	-0.68	0.4
RM	-0.21	0.28	-0.29	0.05	-0.22	1.00	-0.19	0.18	-0.0
AGE	0.50	-0.43	0.49	0.06	0.59	-0.19	1.00	-0.61	0.3
DIS	-0.54	0.48	-0.57	-0.07	-0.68	0.18	-0.61	1.00	-0.3
RAD	0.56	-0.23	0.35	0.02	0.43	-0.08	0.31	-0.36	1.0
TAX	0.54	-0.29	0.48	-0.04	0.45	-0.19	0.36	-0.38	0.5
PTRATIO	0.31	-0.36	0.34	-0.12	0.28	-0.22	0.25	-0.22	0.2
B	-0.26	0.13	-0.19	-0.03	-0.20	0.03	-0.15	0.17	-0.2
LSTAT	0.45	-0.39	0.47	-0.04	0.45	-0.47	0.49	-0.41	0.2
MEDV	-0.40	0.34	-0.42	0.12	-0.39	0.48	-0.39	0.31	-0.2

In [94]:

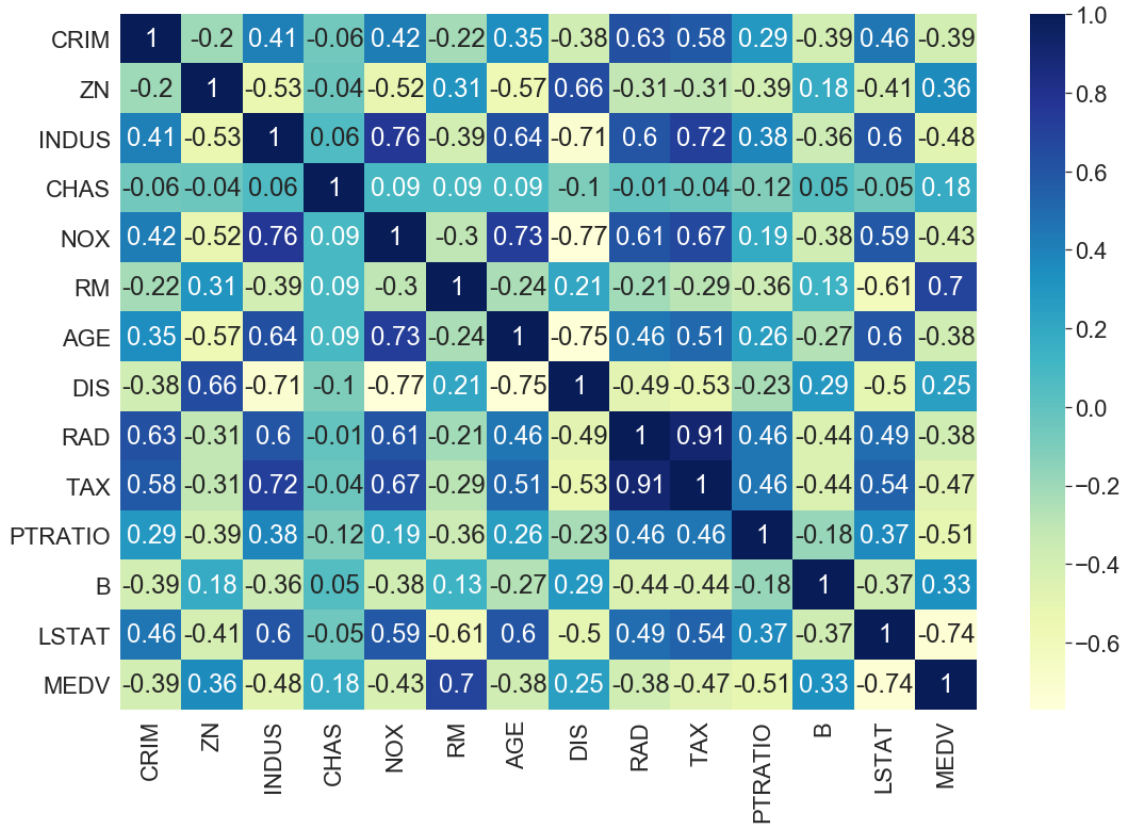
```
corr_matrix_s = df.corr(method='spearman').round(2)
corr_matrix_s
```

Out[94]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RA
CRIM	1.00	-0.57	0.74	0.04	0.82	-0.31	0.70	-0.74	0.7
ZN	-0.57	1.00	-0.64	-0.04	-0.63	0.36	-0.54	0.61	-0.2
INDUS	0.74	-0.64	1.00	0.09	0.79	-0.42	0.68	-0.76	0.4
CHAS	0.04	-0.04	0.09	1.00	0.07	0.06	0.07	-0.08	0.0
NOX	0.82	-0.63	0.79	0.07	1.00	-0.31	0.80	-0.88	0.5
RM	-0.31	0.36	-0.42	0.06	-0.31	1.00	-0.28	0.26	-0.1
AGE	0.70	-0.54	0.68	0.07	0.80	-0.28	1.00	-0.80	0.4
DIS	-0.74	0.61	-0.76	-0.08	-0.88	0.26	-0.80	1.00	-0.5
RAD	0.73	-0.28	0.46	0.02	0.59	-0.11	0.42	-0.50	1.0
TAX	0.73	-0.37	0.66	-0.04	0.65	-0.27	0.53	-0.57	0.7
PTRATIO	0.47	-0.45	0.43	-0.14	0.39	-0.31	0.36	-0.32	0.3
B	-0.36	0.16	-0.29	-0.04	-0.30	0.05	-0.23	0.25	-0.2
LSTAT	0.63	-0.49	0.64	-0.05	0.64	-0.64	0.66	-0.56	0.3
MEDV	-0.56	0.44	-0.58	0.14	-0.56	0.63	-0.55	0.45	-0.3

In [95]:

```
sns.set(font_scale=2)
plt.figure(figsize=(18, 12))
ax = sns.heatmap(corr_matrix_p, annot=True, cmap='YlGnBu')
```



In [96]:

```
sns.set(font_scale=1)
fig, ax = plt.subplots(1, 3, sharex='col', sharey='row', figsi
ze=(15,4))
sns.heatmap(corr_matrix_p, ax=ax[0])
sns.heatmap(corr_matrix_k, ax=ax[1])
sns.heatmap(corr_matrix_s, ax=ax[2])
fig.suptitle('Корреляционные матрицы, построенные различными м
етодами')
ax[0].title.set_text('Pearson')
ax[1].title.set_text('Kendall')
ax[2].title.set_text('Spearman')
```