```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m
```

Задание

Для заданного набора данных проведите корреляционный анализ. В случае наличия пропусков в данных удалите строки или колонки, содержащие пропуски. Сделайте выводы о возможности построения моделей машинного обучения и о возможном вкладе признаков в модель. Для студентов группы ИУ5-65Б, ИУ5И-61Б - для набора данных построить "парные диаграммы".

Импорт библиотек

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

Смотрим на датасет

```
# Читаем датасет из файла
data = pd.read_csv('/content/drive/MyDrive/HousingData.csv', sep=",")
# Количество строк и колонок
print(f'В датасете {data.shape[0]} строк и {data.shape[1]} колонок.')
```

```
В датасете 506 строк и 14 колонок.
```

Что означают колонки

- CRIME - уровень преступности на душу населения в разбивке по городам
- ZN - доля жилых земель, разделенных на участки площадью более 25 000 кв.футов
- INDUS - доля акров, не связанных с розничной торговлей, в городе
- CHAS - фиктивная переменная Charles River (1, если тракт граничит с рекой; 0 в противном случае)
- NOX - концентрация оксидов азота (частей на 10 миллионов)
- RM - среднее количество комнат в доме
- AGE - доля жилых помещений, построенных владельцами до 1940 года

- AGE - доля жилых помещений, построенных владельцами до 1940 года
- DIS - Взвешенные расстояния до пяти бостонских центров занятости
- RAD - индекс доступности радиальных магистралей
- TAX - полная стоимость недвижимости-ставка налога на 10 000 долларов США
- PTRATIO - Отношение числа учащихся к числу учителей в разбивке по городам
- B - 1000(Bk - 0,63)^2, где Bk - доля чернокожих в разбивке по городам
- LSTAT - более низкий статус населения на %
- MEDV - Средняя стоимость домов, занимаемых владельцами, измеряется в 1000-ах долларов

```
# Типы колонок
print(data.dtypes)
```

```
CRIM       float64
ZN         float64
INDUS      float64
CHAS       float64
NOX        float64
RM         float64
AGE        float64
DIS        float64
RAD          int64
TAX          int64
PTRATIO    float64
B          float64
LSTAT      float64
MEDV       float64
dtype: object
```

```
# Количество пропусков
print(data.isnull().sum())
```

```
CRIM       20
ZN         20
INDUS      20
CHAS       20
NOX         0
RM          0
AGE        20
DIS         0
RAD         0
TAX         0
PTRATIO     0
B           0
LSTAT      20
MEDV        0
dtype: int64
```

```
# Количество уникальных значений
```

```
print(data.nunique())
```

```
CRIM       484
ZN          26
INDUS       76
CHAS         2
NOX         81
RM         446
AGE        348
DIS        412
RAD          9
TAX         66
PTRATIO     46
B          357
LSTAT      438
MEDV       229
dtype: int64
```

Обрабатываем пропуски

```
# Удаляем строки пропущенными значениями
data_new = data.dropna(axis=0, how='any')
print(f'Было {data.shape[0]} строк и {data.shape[1]} колонок.')
print(f'Стало {data_new.shape[0]} строк и {data_new.shape[1]} колонок.')
```

```
Было 506 строк и 14 колонок.
Стало 394 строк и 14 колонок.
```

Обрабатываем выбросы

```
# Функция для определения индексов выбросов. Выбросы - точки за пределами 3 сигм
def outliers_indices(feature):
    mid = data_new[feature].mean()
    sigma = data_new[feature].std()
    return data_new[(data_new[feature] < mid - 3 * sigma) | (data_new[feature] > mid + 3
```

```
# Определяем в каких строках есть выбросы
outliers = set()
for column in data_new.columns:
    outliers.update(outliers_indices(column))

# Удаляем строки с выбросами
data_without_outliers = data_new.drop(index=outliers)
print(f'Было {data_new.shape[0]} строк и {data_new.shape[1]} колонок.')
print(f'Стало {data_without_outliers.shape[0]} строк и {data_without_outliers.shape[1]} к

# Количество уникальных значений
```

```
print(data_without_outliers.nunique())
```

```
Было 394 строк и 14 колонок.
Стало 318 строк и 14 колонок.
CRIM       318
ZN          21
INDUS       60
CHAS         1
NOX         72
RM         289
AGE        255
DIS        269
RAD          9
TAX         52
PTRATIO     41
B          214
LSTAT      305
MEDV       184
dtype: int64
```

```
# Видим что CHAS имеет всего 1 уникальное значение, этом признак потерял информативность.
data_without_outliers = data_without_outliers.drop('CHAS', axis=1)
```

Корреляционный анализ

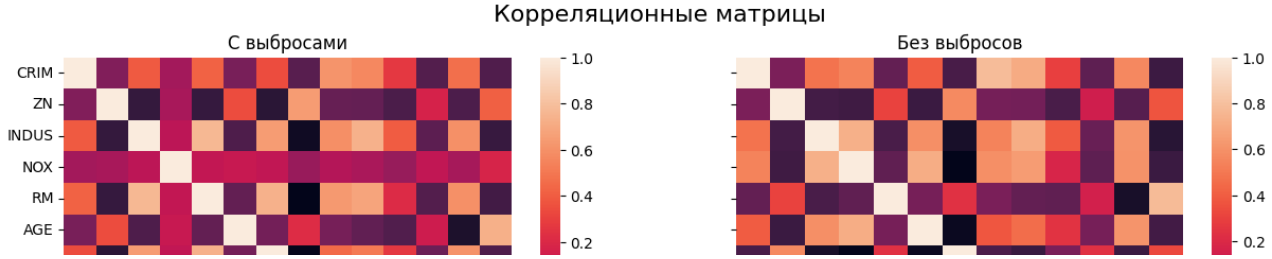
```
# Числовая корреляционная матрица
correlation_matrix = data_without_outliers.corr(method='spearman')
```

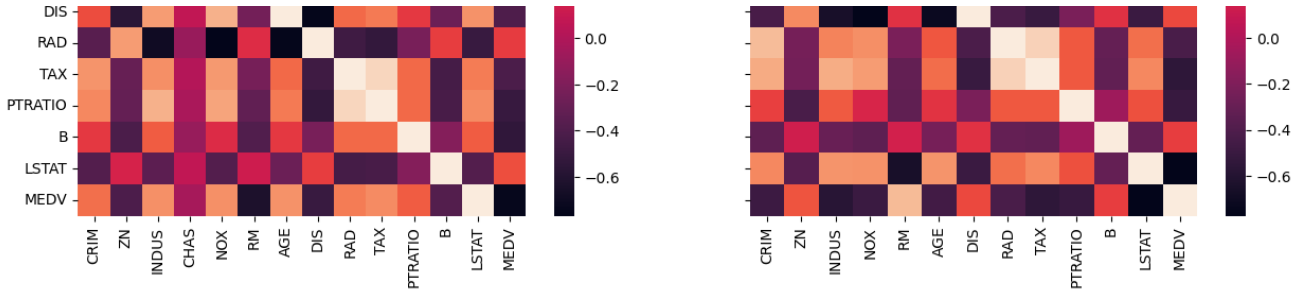
```
# Создаем фигуру с двумя подграфиками
fig, ax = plt.subplots(1, 2, sharex='col', sharey='row', figsize=(15, 5))
fig.suptitle('Корреляционные матрицы', fontsize=16) # Заголовок

# Строим первую тепловую карту на основе данных с выбросами
sns.heatmap(data_new.corr(), ax=ax[0])
ax[0].set_title('С выбросами')

# Строим вторую тепловую карту на основе данных без выбросов
sns.heatmap(data_without_outliers.corr(), ax=ax[1])
ax[1].set_title('Без выбросов')
#sns.heatmap(data_without_outliers.corr())
```

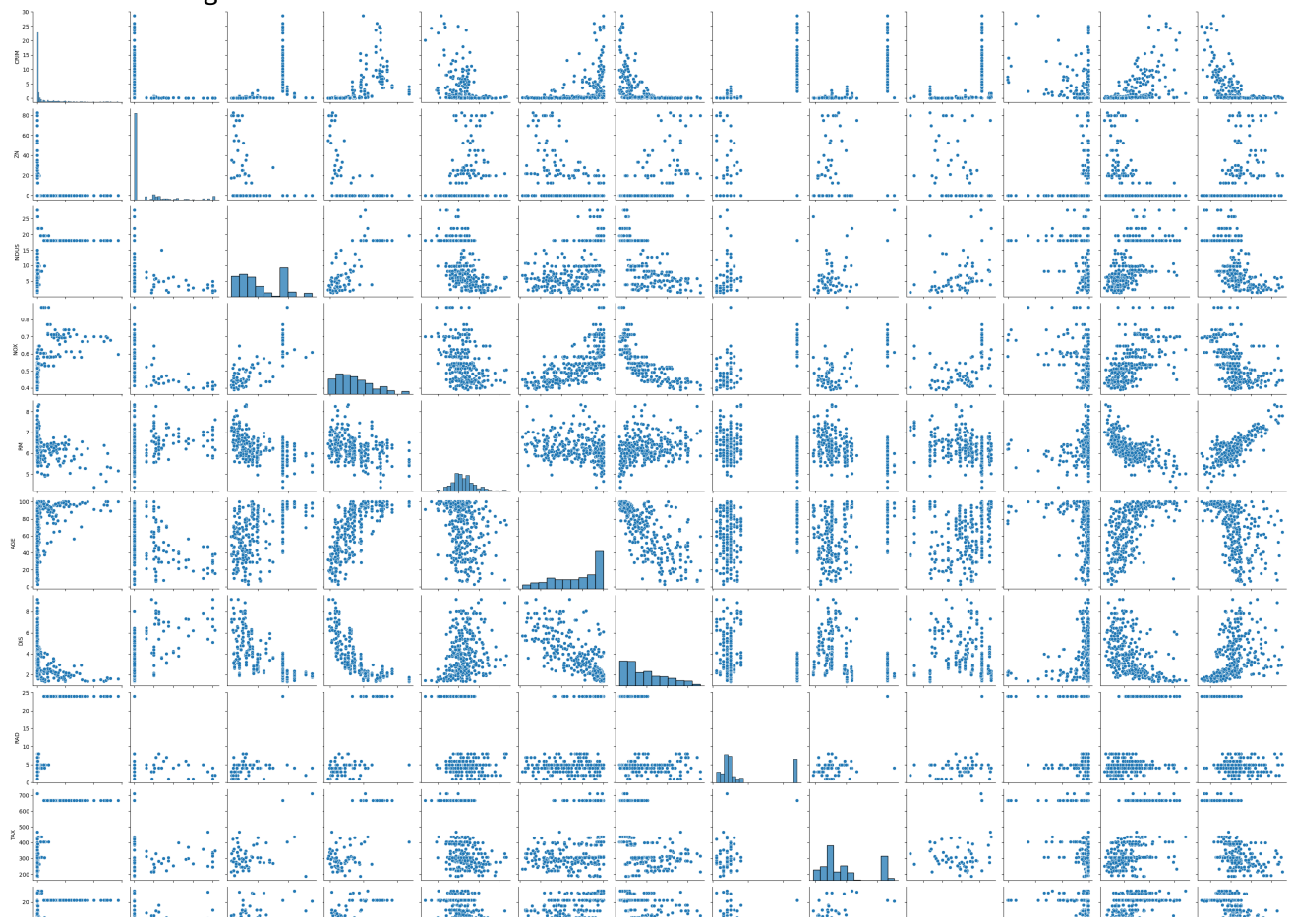
```
Text(0.5, 1.0, 'Без выбросов')
```

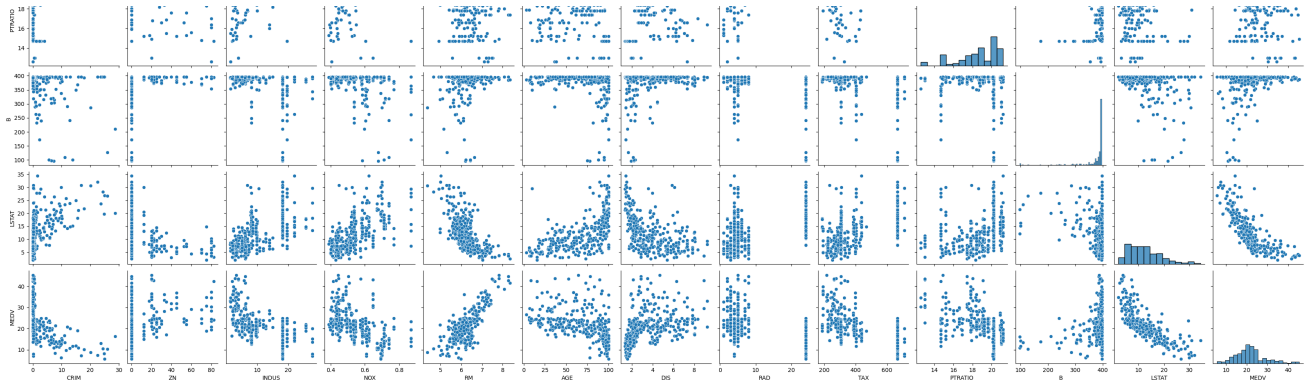





```
# Построение парных диаграмм
sns.pairplot(data_without_outliers)
```

```
<seaborn.axisgrid.PairGrid at 0x7ccd0816f690>
```




PTRATIO
B
LSTAT
MEDV
CRIM
ZN
INDUS
NOX
RM
AGE
DIS
RAD
TAX
PTRATIO
B
LSTAT
MEDV

В описании к датасету написано одно из заданий: узнать какие параметры влияют на цену жилья. Возьмем в качестве целевой переменной MEDV - медианную стоимость жилья. Выясним какие параметры влияют на неё

```
medv_correlations = correlation_matrix['MEDV'].sort_values()
print(medv_correlations)
```

```
LSTAT     -0.847752
INDUS     -0.617471
AGE       -0.593804
TAX       -0.584183
NOX       -0.578090
CRIM      -0.568515
PTRATIO   -0.564765
RAD       -0.305923
B          0.134643
ZN         0.430502
DIS        0.442623
RM         0.668183
MEDV       1.000000
Name: MEDV, dtype: float64
```

Как мы видим, целевая переменная MEDV имеет сильную обратнкую связь с переменной LSTAT, обратную связь с переменными INDUS AGE TAX NOX CRIM PTRATIO, имеет положительную связь с перменной RM.

Переменные RAD B ZN DIS влияют на целевую переменную MEDV незначительно