

▼ Белоусов Евгений Александрович

ИУ5-23м

РК1

```
1 import sklearn
2 import numpy as np
3 import pandas as pd
4 import seaborn as sns
5 import matplotlib.pyplot as plt
6 %matplotlib inline
```

```
1 !unzip /content/drive/MyDrive/Colab_data/MMO/archive.zip
```

📁 Archive: /content/drive/MyDrive/Colab_data/MMO/archive.zip
inflating: autoru_total.csv

+ Код

+ Текст

```
1 data = pd.read_csv('/content/autoru_total.csv', sep=',')
```

```
1 data.head()
```

	Model	Year	Mileage	V_engine	EngineType	HorsePower	Tax	State	
0	Subaru Forester IV	2013	83800.0	2.5	Бензин	171.0	8379.0	Не требует ремонта	влад
1	Opel Zafira B Рестайлинг	2014	97265.0	1.8	Бензин	140.0	4900.0	Не требует ремонта	влад
2	Kia Rio IV	2017	48000.0	1.6	Бензин	123.0	3075.0	Не требует ремонта	влад
3	Skoda Octavia II	2014	88000.0	1.4	Бензин	102.0	4440.0	Не требует ремонта	влад

▼ Задача №4

Для набора данных проведите кодирование одного (произвольного) категориального признака с использованием метода "label encoding".

```
1 from sklearn.preprocessing import LabelEncoder
```

```
1 data['Selector'].unique()

array(['вариатор', 'механическая', 'автоматическая', 'роботизированная'],
      dtype=object)
```

```
1 le = LabelEncoder()
2 cat_le = le.fit_transform(data['Selector'])
3 np.unique(cat_le)

array([0, 1, 2, 3])
```

```
1 data['Selector_le'] = cat_le
2 data.head()
```

	Model	Year	Mileage	V_engine	EngineType	HorsePower	Tax	State
0	Subaru Forester IV	2013	83800.0	2.5	Бензин	171.0	8379.0	Не требует ремонта
1	Opel Zafira B Рестайлинг	2014	97265.0	1.8	Бензин	140.0	4900.0	Не требует ремонта
2	Kia Rio IV	2017	48000.0	1.6	Бензин	123.0	3075.0	Не требует ремонта
3	Skoda Octavia II	2014	80000.0	1.4	Бензин	102.0	4440.0	Не требует ремонта

```
1 le.inverse_transform([0, 1, 2, 3])

array(['автоматическая', 'вариатор', 'механическая', 'роботизированная'],
      dtype=object)
```

▼ Задача 24

Для набора данных для одного (произвольного) числового признака проведите обнаружение и удаление выбросов на основе 5% и 95% квантилей.

```
1 data.shape
```

```
(36906, 18)
```

```
1 lower_boundary = data['Price'].quantile(0.05)
```

```
2 upper_boundary = data['Price'].quantile(0.95)
```

```
3
```

```
4 outliers_temp = np.where(data['Price'] > upper_boundary, True,
```

```
5                             np.where(data['Price'] < lower_boundary, True, False))
```

```
6 data_trimmed = data.loc[~(outliers_temp), ]
```

```
7 data_trimmed.shape
```

```
(33226, 18)
```

▼ Дополнительное требование по группам

Для студентов групп ИУ5-23М, ИУ5И-23М - для произвольной колонки данных построить график "Ящик с усами (boxplot)".

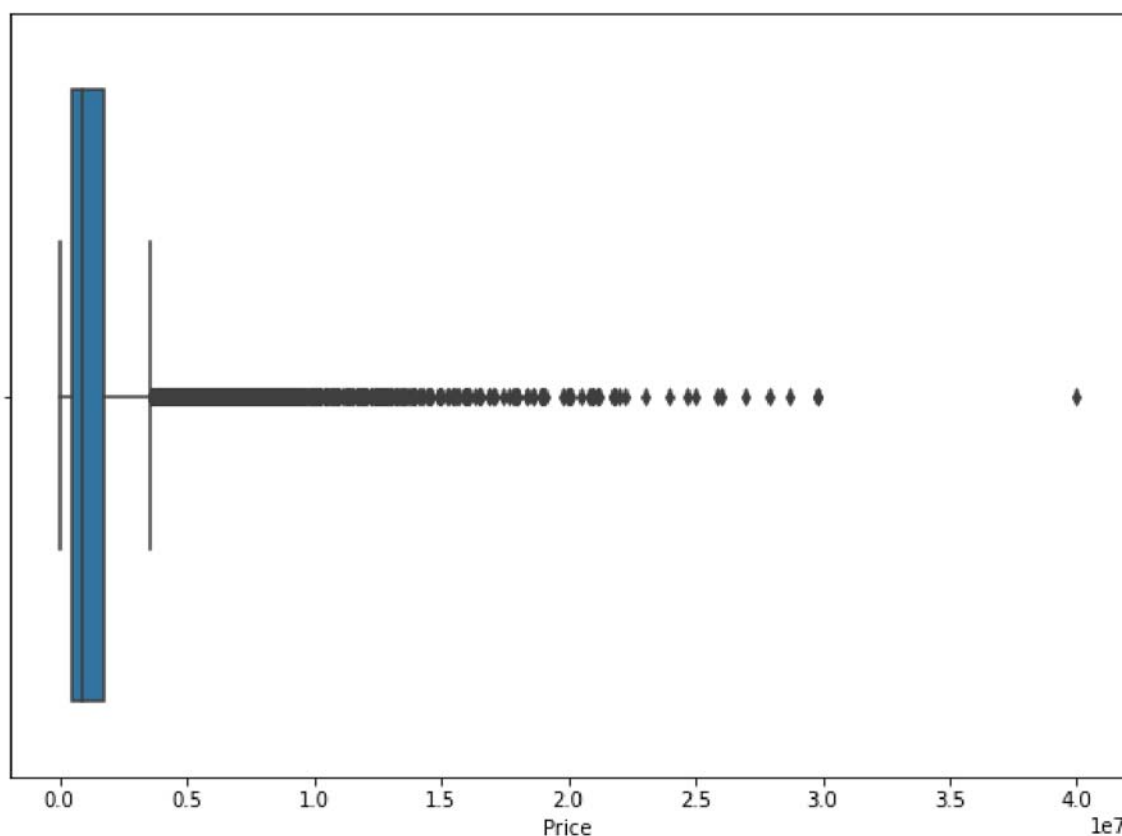
```
1 fig, ax = plt.subplots(figsize=(10,7))
```

```
2 sns.boxplot(x=data['Price'])
```

```
3 fig.suptitle('Ящик с усами для Price')
```

```
4 plt.show()
```

Ящик с усами для Price



```
1 fig, ax = plt.subplots(figsize=(10,7))
2 sns.boxplot(x=data_trimmed['Price'])
3 fig.suptitle('Ящик с усами после удаления выбросов для Price')
4 plt.show()
```

Ящик с усами после удаления выбросов для Price

