



**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»**

**Отчет по лабораторной работе №1
по дисциплине «Методы машинного обучения»
по теме «Создание “истории о данных”»**

**Выполнил:
студент группы № ИУ5-21М
Торжков М.С.
подпись, дата**

**Проверила:
Балашов А.М.
подпись, дата**

2023 г.

Задание.

Создать "историю о данных" в виде юпитер-ноутбука, с учетом следующих требований:

1. История должна содержать не менее 5 шагов (где 5 - рекомендуемое количество шагов). Каждый шаг содержит график и его текстовую интерпретацию.
2. На каждом шаге наряду с удачным итоговым графиком рекомендуется в юпитер-ноутбуке оставлять результаты предварительных "неудачных" графиков.
3. Не рекомендуется повторять виды графиков, желательно создать 5 графиков различных видов.
4. Выбор графиков должен быть обоснован использованием методологии data-to-viz. Рекомендуется учитывать типичные ошибки построения выбранного вида графика по методологии data-to-viz. Если методология Вами отвергается, то просьба обосновать Ваше решение по выбору графика.
5. История должна содержать итоговые выводы. В реальных "историях о данных" именно эти выводы представляют собой основную ценность для предприятия.

```
from datetime import datetime
import pandas as pd
import seaborn as sns

# Enable inline plots
%matplotlib inline

# Set plot style
sns.set(style="ticks")

# Set plots formats to save high resolution PNG
from IPython.display import set_matplotlib_formats
set_matplotlib_formats("retina")

pd.set_option("display.width", 70)

data = pd.read_csv("/content/StudentsPerformance.csv")
```

data.dtypes

```
gender                object
race/ethnicity         object
parental level of education  object
lunch                 object
test preparation course  object
math score            int64
reading score         int64
writing score         int64
dtype: object
```

data.head()

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	78
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93

data.shape

(1000, 8)

data.describe()

	math score	reading score	writing score
count	1000.00000	1000.00000	1000.00000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000



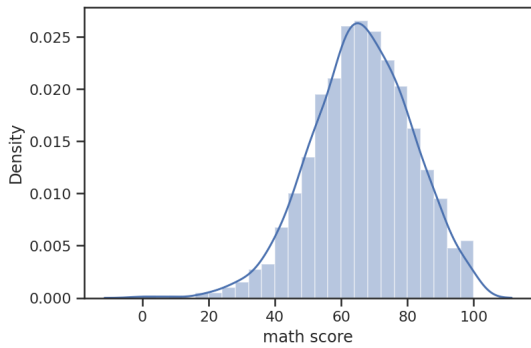
```
from sklearn.preprocessing import LabelEncoder
```

Визуальное исследование датасета

Оценим распределение целевого признака — оценки по математике:

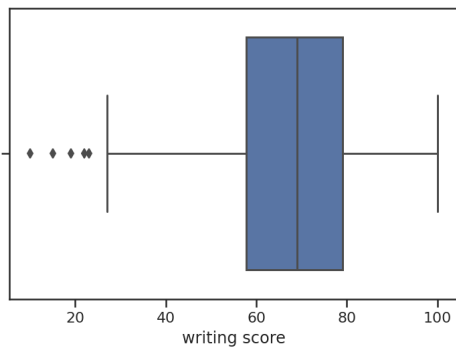
```
sns.distplot(data["math score"]);
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning:
warnings.warn(msg, FutureWarning)
```



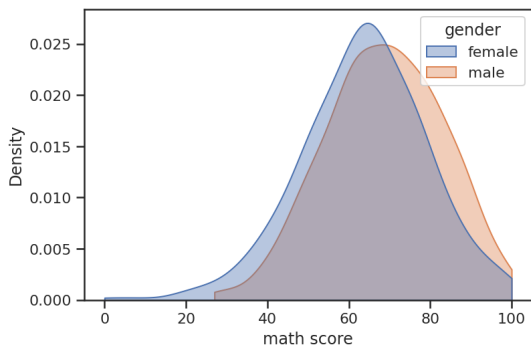
```
sns.boxplot(x=data['writing score'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f624cfc4070>
```



```
sns.kdeplot(data=data, x="math score", hue="gender", cut=0, fill=True, common_norm=False, alpha=0.4)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f624d1bbb80>
```



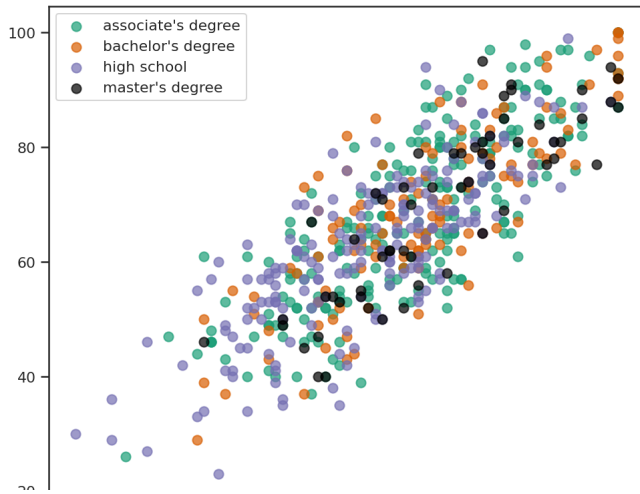
```
import numpy as np
import matplotlib.pyplot as plt
FLIPPER_LENGTH = data["reading score"].values
BILL_LENGTH = data["math score"].values

SPECIES = data["parental level of education"].values
SPECIES_ = np.unique(SPECIES)

COLORS = ["#1B9E77", "#D95F02", "#7570B3", "#000000"]

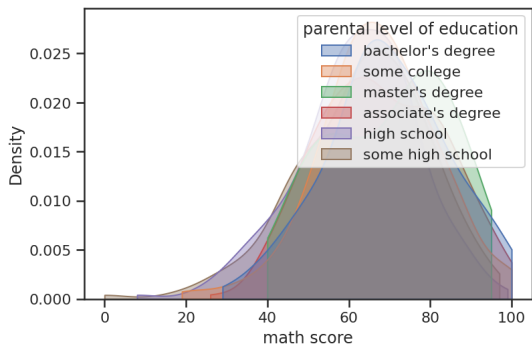
fig, ax = plt.subplots(figsize=(8,8))
for species, color in zip(SPECIES_, COLORS):
    idxs = np.where(SPECIES == species)
    # No legend will be generated if we don't pass label=species
    ax.scatter(
        FLIPPER_LENGTH[idxs], BILL_LENGTH[idxs], label=species,
        s=50, color=color, alpha=0.7
    )

ax.legend();
```



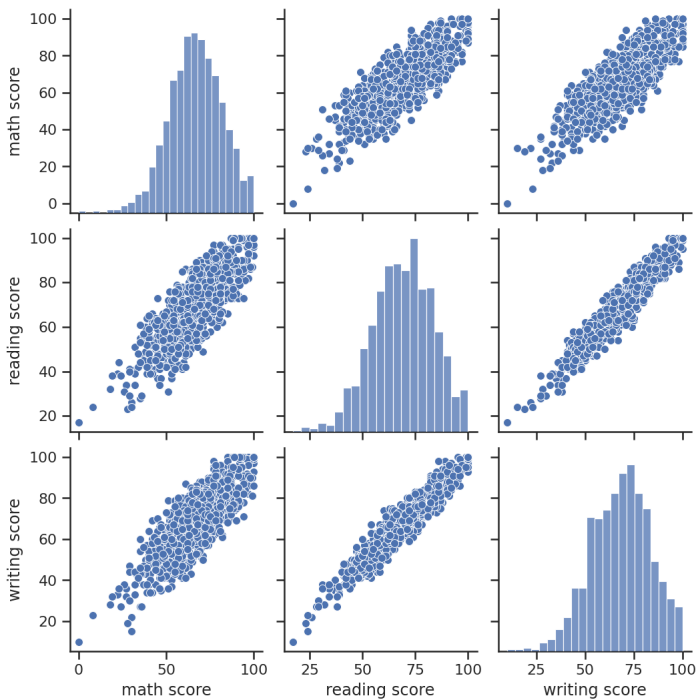
```
sns.kdeplot(data=data, x="math score", hue="parental level of education", cut=0, fill=True, common_norm=False, alpha=0.4)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f624d631880>
```



```
sns.pairplot(data)
```

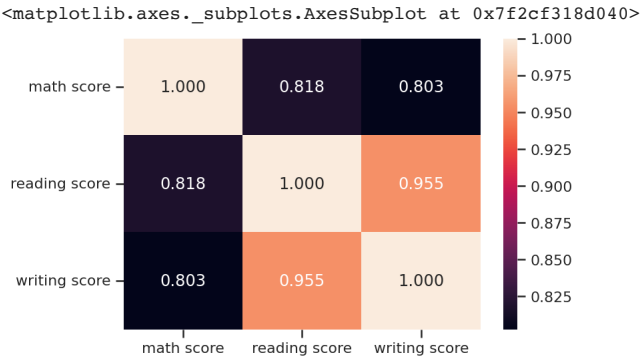
```
<seaborn.axisgrid.PairGrid at 0x7f624d112a00>
```



▼ Корреляции признаков

```
corr_matrix = data.corr()
```

```
sns.heatmap(data.corr(), annot=True, fmt='.3f')
```



[Вспомогательный Colab](#) [Открыть в новом окне](#)

✓ 4 сек. выполнено в 00:27

