Московский государственный технический университет
им. Н.Э. Баумана
Факультет «Информатика и системы управления»
Кафедра «Системы обработки информации и управления»

Лабораторная работа № 1

По курсу «методы машинного обучения в АСОИУ»

«Создание «истории о данных» (Data Storytelling)»

Выполнил:

студент ИУ5-23М
Семенов И.А.

Проверил:

Гапанюк Ю.Е.

Подпись:

29.02.2024

Москва, 2024

Описание задания

- Выбрать набор данных (датасет). Вы можете найти список свободно распространяемых датасетов здесь.
- Для лабораторных работ не рекомендуется выбирать датасеты очень большого размера.
- Создать "историю о данных" в виде юпитер-ноутбука, с учетом следующих требований: история должна содержать не менее 5 шагов (где 5 - рекомендуемое количество шагов). Каждый шаг содержит график и его текстовую интерпретацию.
- На каждом шаге наряду с удачным итоговым графиком рекомендуется в юпитер-ноутбуке оставлять результаты предварительных "неудачных" графиков.
- Не рекомендуется повторять виды графиков, желательно создать 5 графиков различных видов.
- 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.
- История должна содержать итоговые выводы. В реальных "историях о данных" именно эти выводы представляют собой основную ценность для предприятия.
- Сформировать отчет и разместить его в своем репозитории на github.

Текст программы и экранные формы с примерами выполнения программы

```
[40] import numpy as np
     import pandas as pd
     import seaborn as sns
     import matplotlib.pyplot as plt
     %matplotlib inline
     sns.set(style="ticks")
```

```
[41] data = pd.read_csv('apple_quality.csv', encoding='windows-1251')
```

Основные характеристики датасета

```
# Первые 5 строк датасета
data.head()
```

	A_id	Size	Weight	Sweetness	Crunchiness	Juiciness	Ripeness	Acidity	Quality
0	0.0	-3.970049	-2.512336	5.346330	-1.012009	1.844900	0.329840	-0.491590483	good
1	1.0	-1.195217	-2.839257	3.664059	1.588232	0.853286	0.867530	-0.722809367	good
2	2.0	-0.292024	-1.351282	-1.738429	-0.342616	2.838636	-0.038033	2.621636473	bad
3	3.0	-0.657196	-2.271627	1.324874	-0.097875	3.637970	-3.413761	0.790723217	good
4	4.0	1.364217	-1.296612	-0.384658	-0.553006	3.030874	-1.303849	0.501984036	good

Next steps: View recommended plots

```
[43] # Размер датасета - 150 строк, 6 колонок
     data.shape
```

```
(4001, 9)
```

```
# Список колонок с типами данных
data.dtypes
```

```
A_id           float64
Size           float64
Weight         float64
Sweetness      float64
Crunchiness    float64
Juiciness      float64
Ripeness       float64
Acidity         object
Quality         object
dtype: object
```

```
# Проверим наличие пустых значений
# Цикл по колонкам датасета
for col in data.columns:
    # Количество пустых значений - все значения заполнены
    temp_null_count = data[data[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))
```

```
A_id - 1
Size - 1
Weight - 1
Sweetness - 1
Crunchiness - 1
Juiciness - 1
Ripeness - 1
Acidity - 0
Quality - 1
```

```
data.drop( 'A_id', axis = 1, inplace=True)
data.dropna(inplace=True)
```

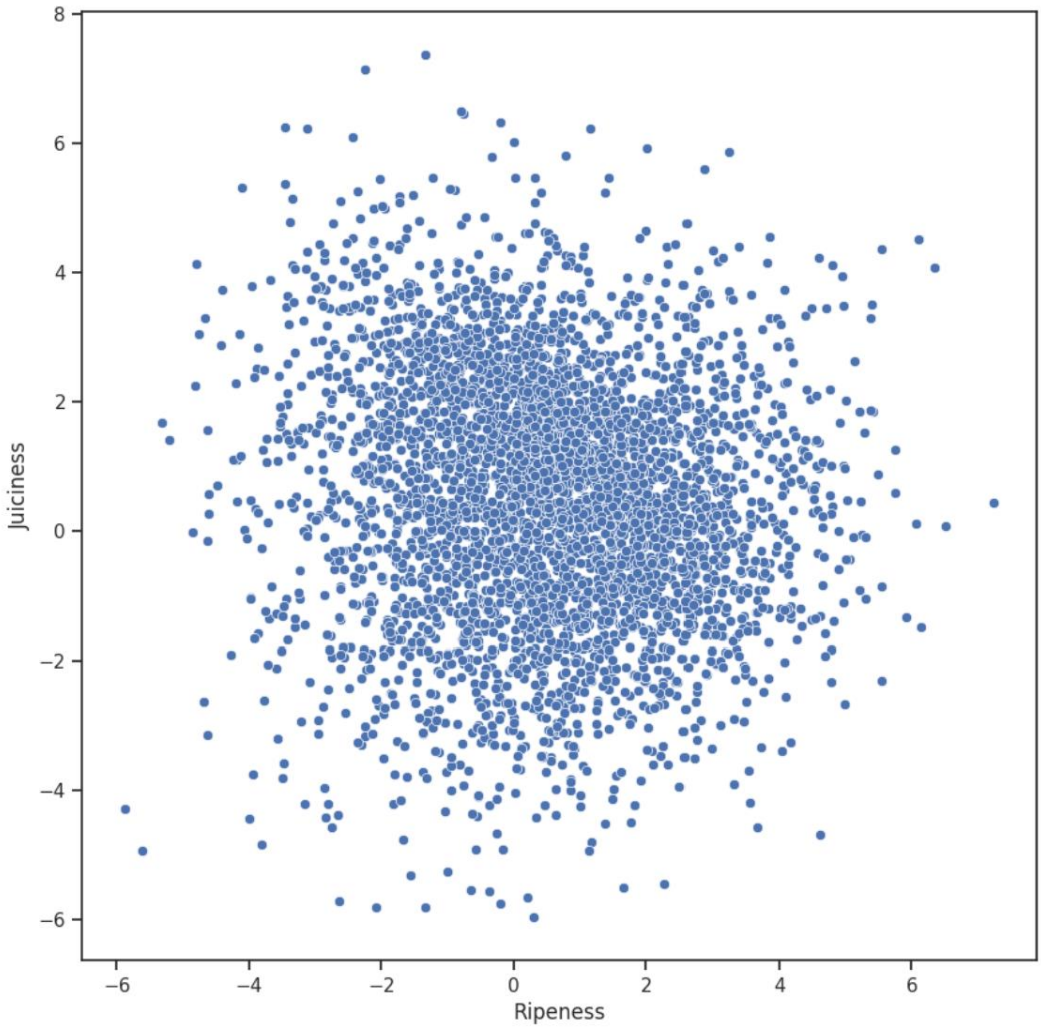
[47]
```
# Основные статистические характеристки набора данных
data.describe()
```

	Size	Weight	Sweetness	Crunchiness	Juiciness	Ripeness
count	4000.000000	4000.000000	4000.000000	4000.000000	4000.000000	4000.000000
mean	-0.503015	-0.989547	-0.470479	0.985478	0.512118	0.498277
std	1.928059	1.602507	1.943441	1.402757	1.930286	1.874427
min	-7.151703	-7.149848	-6.894485	-6.055058	-5.961897	-5.864599
25%	-1.816765	-2.011770	-1.738425	0.062764	-0.801286	-0.771677
50%	-0.513703	-0.984736	-0.504758	0.998249	0.534219	0.503445
75%	0.805526	0.030976	0.801922	1.894234	1.835976	1.766212
max	6.406367	5.790714	6.374916	7.619852	7.364403	7.237837

[48]
```
#  'Acidity' to float64
data['Acidity'] = pd.to_numeric(data['Acidity'], errors='coerce')

# Quality to 1 0
data['Quality'].value_counts()
data['Quality'] = data['Quality'].replace({'good': 1, 'bad': 0}).astype(int)
```

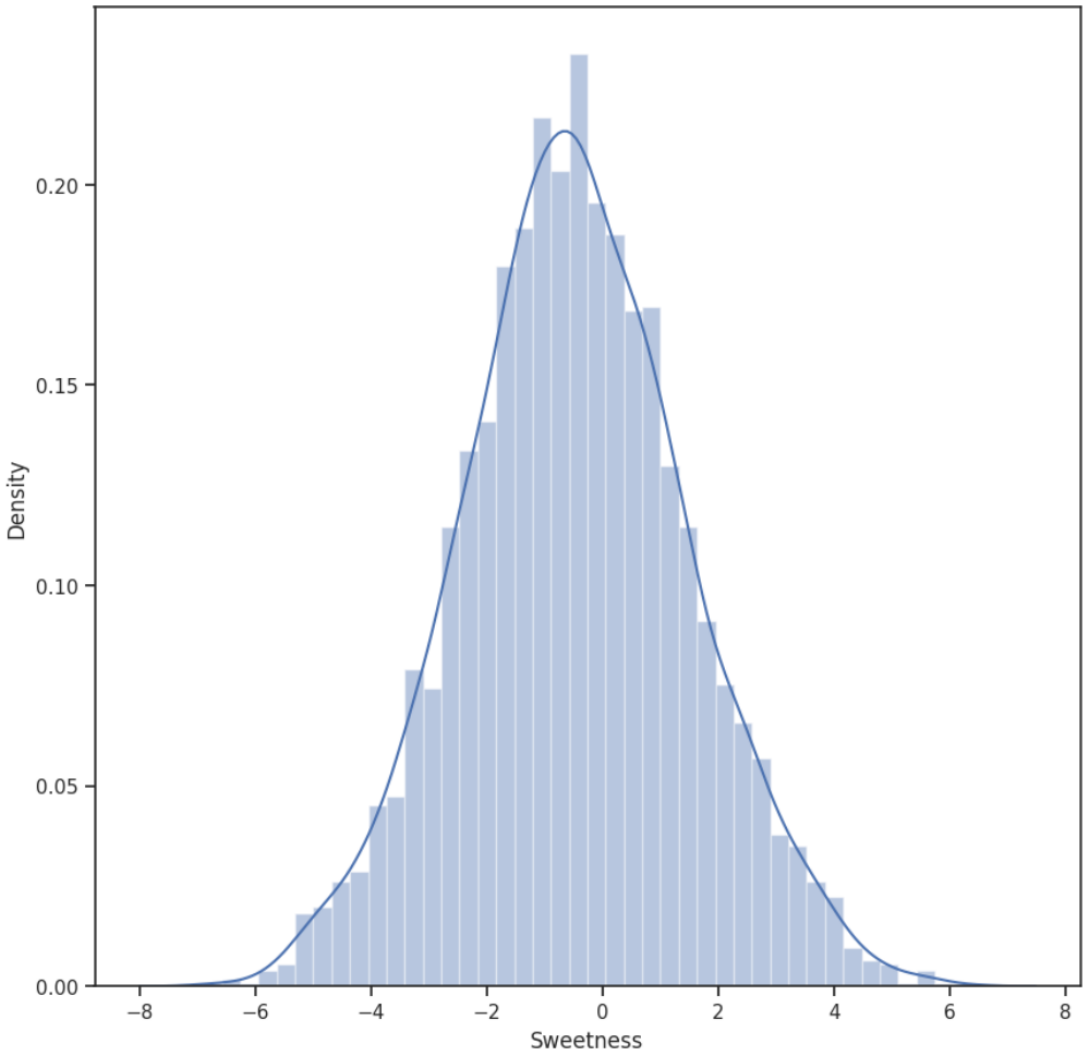
[52]
```
#Диаграмма рассеяния
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='Ripeness', y='Juiciness', data=data)
```

```
<Axes: xlabel='Ripeness', ylabel='Juiciness'>
```



```
#гистограмма
fig, ax = plt.subplots(figsize=(10,10))
sns.distplot(data['Sweetness'])
```

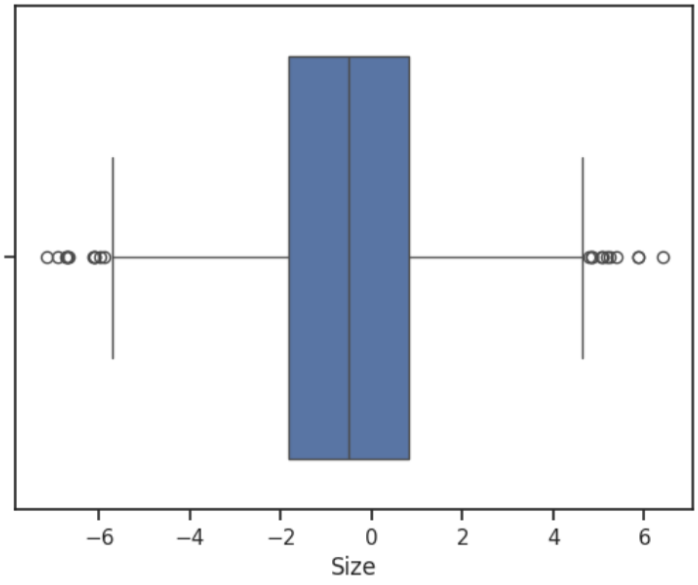
```
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['Sweetness'])
<Axes: xlabel='Sweetness', ylabel='Density'>
```



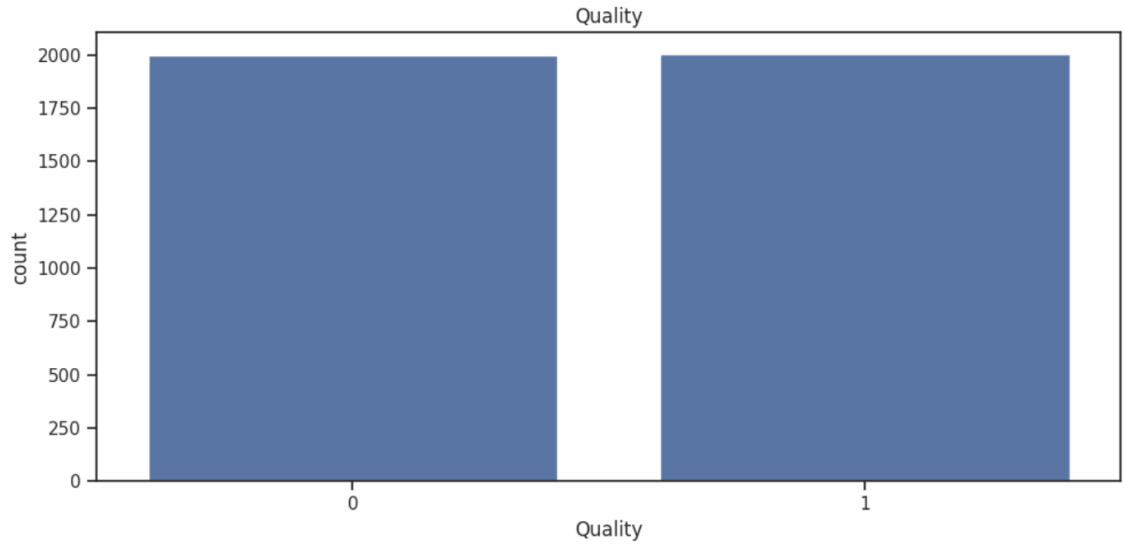
0 сек.

```
#Ящик с усами
sns.boxplot(x=data['Size'])
```

```
<Axes: xlabel='Size'>
```

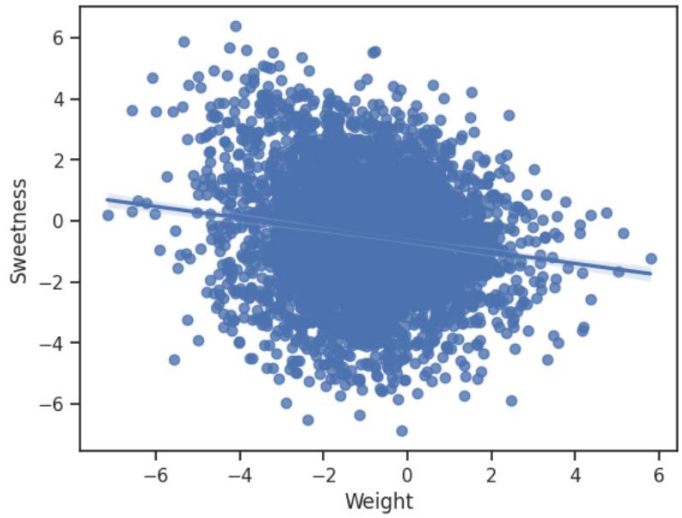


```
plt.figure(figsize=(10, 5))
sns.countplot(x=data['Quality'],data=data)
plt.title('Quality')
plt.tight_layout()
plt.show()
```



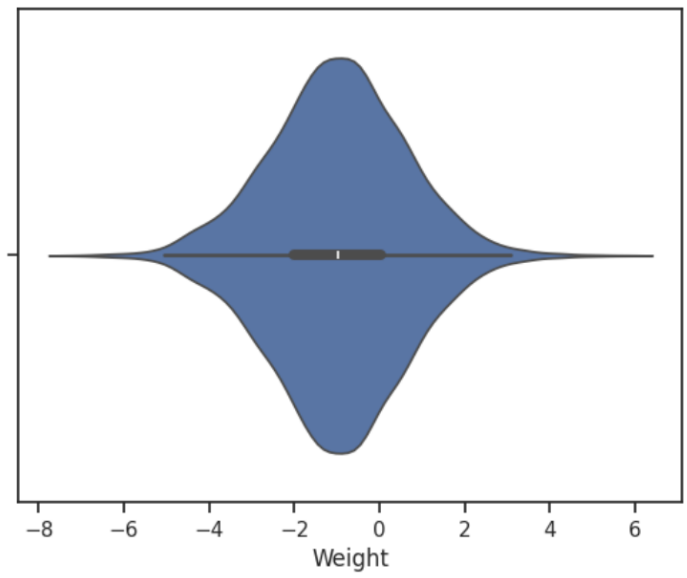

```
[56] sns.regplot(data=data,x="Weight",y="Sweetness")
```

```
<Axes: xlabel='Weight', ylabel='Sweetness'>
```

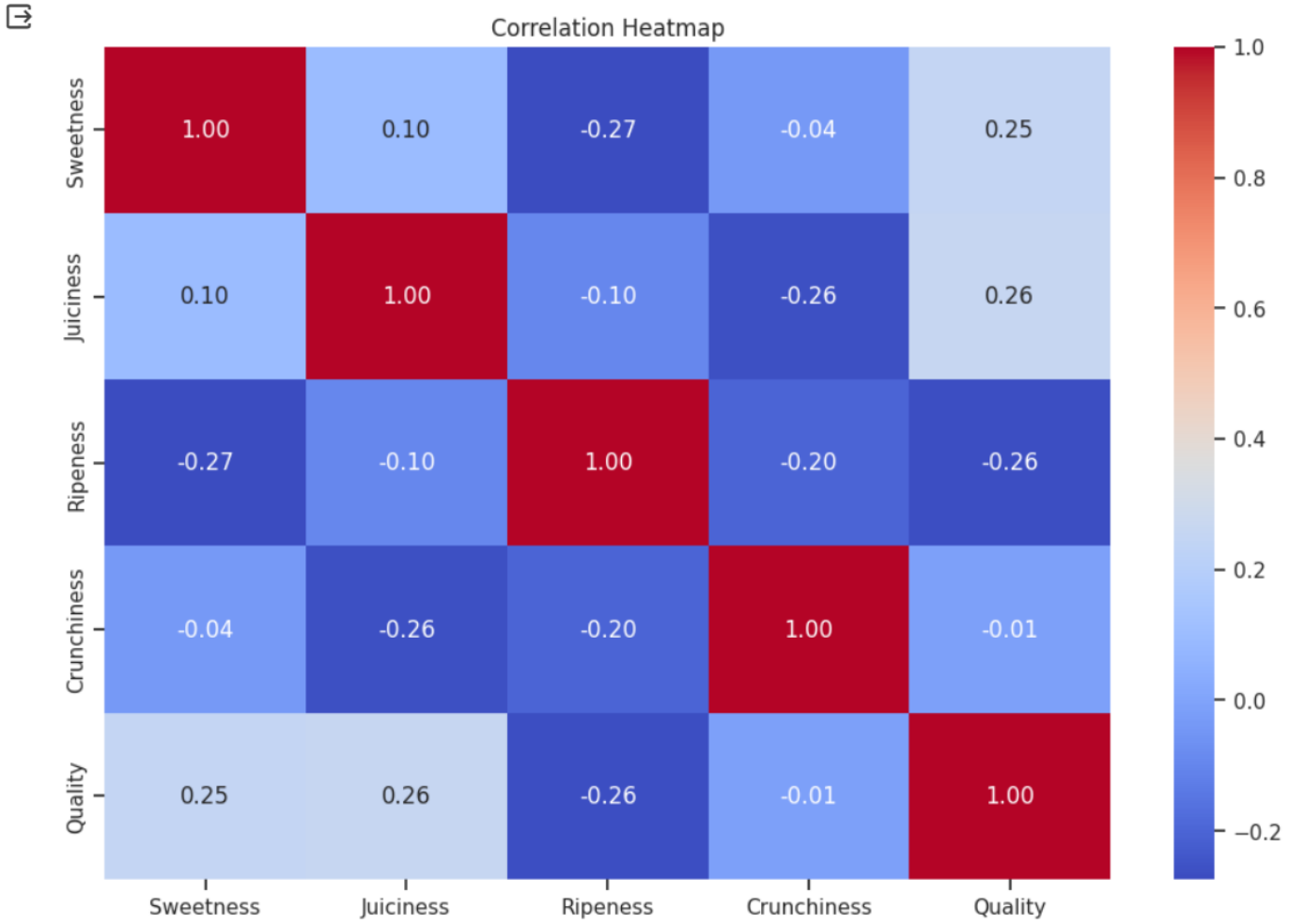



```
#Violin plot
sns.violinplot(x=data['Weight'])
```

```
<Axes: xlabel='Weight'>
```



```
#Разделим датасет на целевой признак и остальные. В предположении, что целевой признак находится в столбце "Quality".

# Выбор признаков для анализа
features = ['Sweetness','Juiciness','Ripeness', 'Crunchiness']  # Замените на реальные названия признаков

# Создание матрицы корреляции между признаками и целевым признаком
correlation_matrix = data[features + ['Quality']].corr()

# Построение тепловой карты
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()
```



На основе полученных результатов можем сделать вывод о том, что качество яблока напрямую зависит от его зрелости. Также имеется прямая зависимость между зрелостью яблока и его хрупкостью. Сладость же и сочность почти не зависят друг от друга.