Лабортаторная работа N°1 по курсо ТМО

Холодова Карина

ИУ5Ц-82Б

Разведочный анализ данных. Исследование и визуализация данных.

Текстовое описание набора данных

Этот датасет содержит информацию о различных атрибутах набора фруктов - яблоков, позволяющую получить представление об их характеристиках. Набор данных включает такие сведения, как идентификатор фрукта, размер, вес, сладость, хрусткость, сочность, спелость, кислотность и качество.

```
# Установка библиотек numpy, pandas, seaborn, matplotlib для работы с
данными и их визуализации

pip install numpy pandas seaborn matplotlib
```

```
Defaulting to user installation because normal site-packages is not
writeable
Requirement already satisfied: numpy in
/Users/kkholodova/Library/Python/3.9/lib/python/site-packages (1.26.4)
Requirement already satisfied: pandas in
/Users/kkholodova/Library/Python/3.9/lib/python/site-packages (2.2.1)
Requirement already satisfied: seaborn in
/Users/kkholodova/Library/Python/3.9/lib/python/site-packages (0.13.2)
Requirement already satisfied: matplotlib in
/Users/kkholodova/Library/Python/3.9/lib/python/site-packages (3.8.4)
Requirement already satisfied: python-dateutil>=2.8.2 in
/Users/kkholodova/Library/Python/3.9/lib/python/site-packages (from
pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
/Users/kkholodova/Library/Python/3.9/lib/python/site-packages (from
pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in
/Users/kkholodova/Library/Python/3.9/lib/python/site-packages (from
pandas) (2024.1)
Requirement already satisfied: contourpy>=1.0.1 in
/Users/kkholodova/Library/Python/3.9/lib/python/site-packages (from
matplotlib) (1.2.1)
Requirement already satisfied: cycler>=0.10 in
/Users/kkholodova/Library/Python/3.9/lib/python/site-packages (from
matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
```

```
/Users/kkholodova/Library/Python/3.9/lib/python/site-packages (from
matplotlib) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
/Users/kkholodova/Library/Python/3.9/lib/python/site-packages (from
matplotlib) (1.4.5)
Requirement already satisfied: packaging>=20.0 in
/Users/kkholodova/Library/Python/3.9/lib/python/site-packages (from
matplotlib) (24.0)
Requirement already satisfied: pillow>=8 in
/Users/kkholodova/Library/Python/3.9/lib/python/site-packages (from
matplotlib) (10.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/Users/kkholodova/Library/Python/3.9/lib/python/site-packages (from
matplotlib) (3.1.2)
Requirement already satisfied: importlib-resources>=3.2.0 in
/Users/kkholodova/Library/Python/3.9/lib/python/site-packages (from
matplotlib) (6.4.0)
Requirement already satisfied: zipp>=3.1.0 in
/Users/kkholodova/Library/Python/3.9/lib/python/site-packages (from
importlib-resources>=3.2.0->matplotlib) (3.18.1)
Requirement already satisfied: six>=1.5 in
/Library/Developer/CommandLineTools/Library/Frameworks/Python3.framewo
rk/Versions/3.9/lib/python3.9/site-packages (from python-
dateutil>=2.8.2->pandas) (1.15.0)

[notice] A new release of pip is available: 24.0 -> 25.0.1
[notice] To update, run:
/Library/Developer/CommandLineTools/usr/bin/python3 -m pip install --
upgrade pip
Note: you may need to restart the kernel to use updated packages.
```

```
# Установка библиотеки scipy для научных вычислений

pip install scipy
```

```
Defaulting to user installation because normal site-packages is not
writeable
Requirement already satisfied: scipy in
/Users/kkholodova/Library/Python/3.9/lib/python/site-packages (1.13.1)
Requirement already satisfied: numpy<2.3,>=1.22.4 in
/Users/kkholodova/Library/Python/3.9/lib/python/site-packages (from
scipy) (1.26.4)

[notice] A new release of pip is available: 24.0 -> 25.0.1
[notice] To update, run:
/Library/Developer/CommandLineTools/usr/bin/python3 -m pip install --
upgrade pip
Note: you may need to restart the kernel to use updated packages.
```

```
# Импорт необходимых библиотек для анализа данных и визуализации
# Настройка стиля графиков через seaborn

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

# Загрузка датасета из файла apple_quality.csv

df_data = pd.read_csv("apple_quality.csv")

# Вывод случайных 5 строк для предварительного просмотра

df_data.sample(5)
```

```
        A_id      Size    Weight  Sweetness  Crunchiness  Juiciness
Ripeness  \
2685  2685.0 -3.295368 -0.727647   2.064442    -0.763051   0.710398
1.003525
1497  1497.0  3.166010  0.955333  -1.767522     0.058884  -0.264385 -
2.060988
746    746.0 -5.240767 -4.111890   1.299108     0.157965   4.177133
3.069098
26      26.0 -0.300698 -0.513603   0.921006     1.378172   2.274747
0.745336
729    729.0  0.167621  0.310687  -0.670361     1.975892  -1.358239
0.549872

           Acidity Quality
2685  -1.059747757    good
1497  -1.426638729     bad
746    5.136138788     bad
26     -2.93402889    good
729   -3.769068269     bad
```

```
# Проверка размерности датасета

df_data.shape
```

```
(4001, 9)
```

```
# Вывод списка всех столбцов датасета

df_data.columns
```

```
Index(['A_id', 'Size', 'Weight', 'Sweetness', 'Crunchiness',
'Juiciness',
       'Ripeness', 'Acidity', 'Quality'],
      dtype='object')
```

```
# Проверка типов данных для каждого столбца

df_data.dtypes
```

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

```
# Подсчет количества пропущенных значений в каждом столбце

print("Количесво пропусков")
for col in df_data:
    print(f"{col} = {df_data[df_data[col].isnull()].shape[0]}")
```

```
Количесво пропусков
A_id = 1
Size = 1
Weight = 1
Sweetness = 1
Crunchiness = 1
Juiciness = 1
Ripeness = 1
Acidity = 0
Quality = 1
```

```
# Вычисление основных статистик (среднее, стандартное отклонение,
минимум, максимум и т.д.) для числовых столбцов

df_data.describe()
```

```
              A_id         Size       Weight    Sweetness  Crunchiness
\
count  4000.000000  4000.000000  4000.000000  4000.000000  4000.000000

mean   1999.500000    -0.503015    -0.989547    -0.470479     0.985478

std    1154.844867     1.928059     1.602507     1.943441     1.402757

min       0.000000    -7.151703    -7.149848    -6.894485    -6.055058

25%     999.750000    -1.816765    -2.011770    -1.738425     0.062764
```

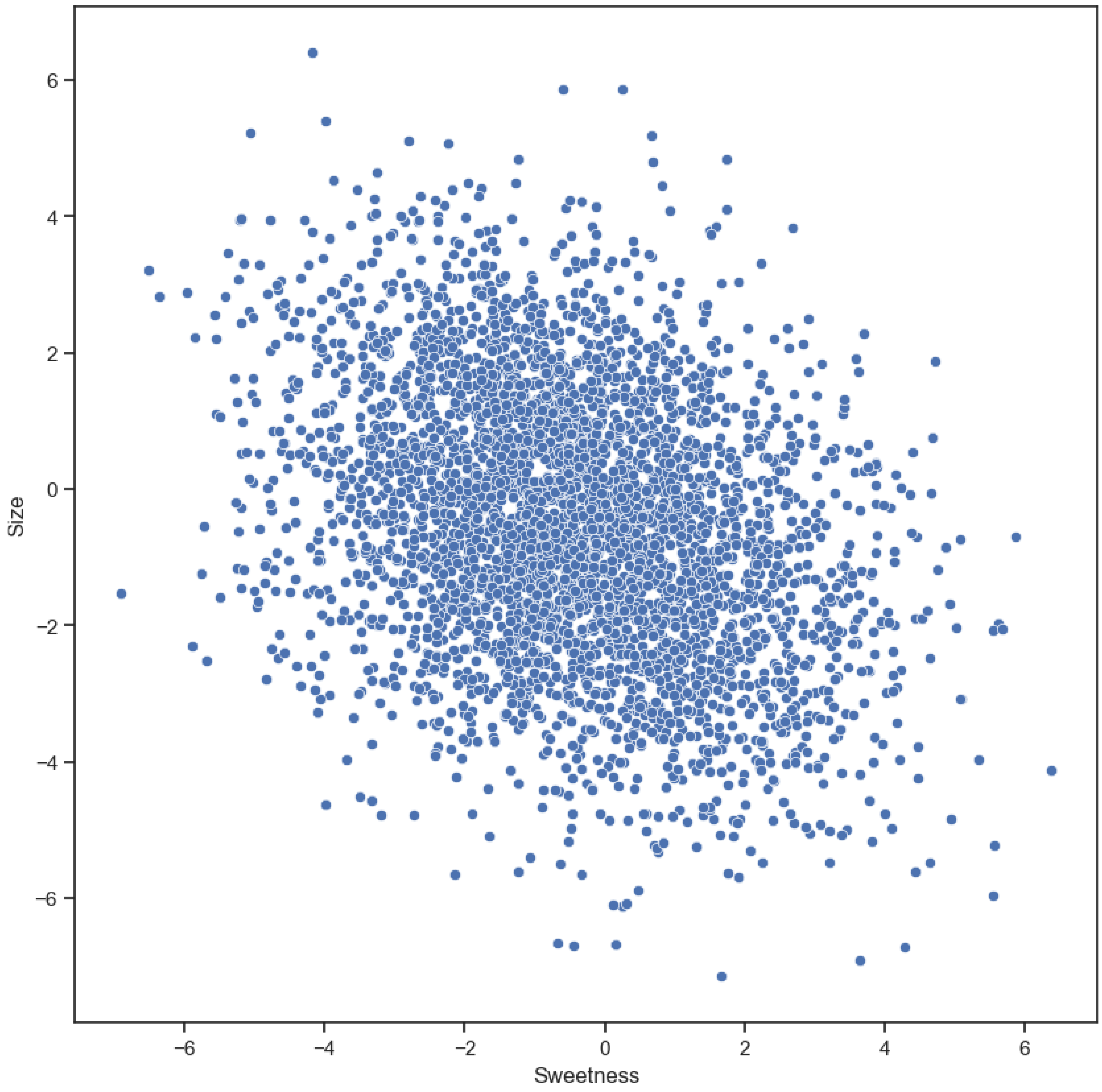
```
50%    1999.500000    -0.513703    -0.984736    -0.504758     0.998249
75%    2999.250000     0.805526     0.030976     0.801922     1.894234
max    3999.000000     6.406367     5.790714     6.374916     7.619852

         Juiciness     Ripeness
count  4000.000000  4000.000000
mean      0.512118     0.498277
std       1.930286     1.874427
min      -5.961897    -5.864599
25%      -0.801286    -0.771677
50%       0.534219     0.503445
75%       1.835976     1.766212
max       7.364403     7.237837
```

```
# Просмотр уникальных значений в столбце Quality

df_data.Quality.unique()
```

```
array(['good', 'bad', nan], dtype=object)
```

```
# Создание точечного графика

fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='Sweetness', y='Size', data=df_data)
```

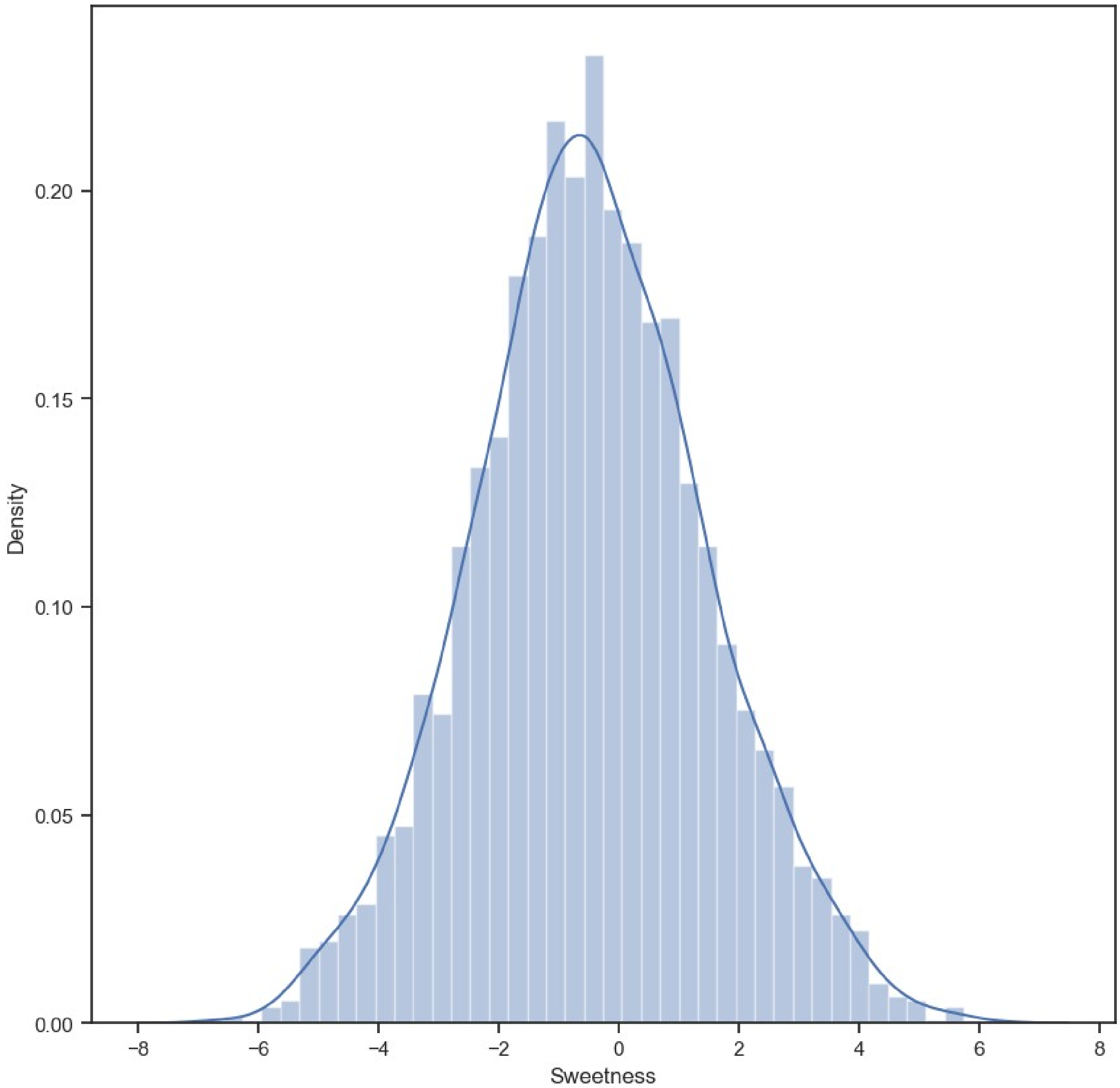
```
<Axes: xlabel='Sweetness', ylabel='Size'>
```



```
# Создание гистограммы

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

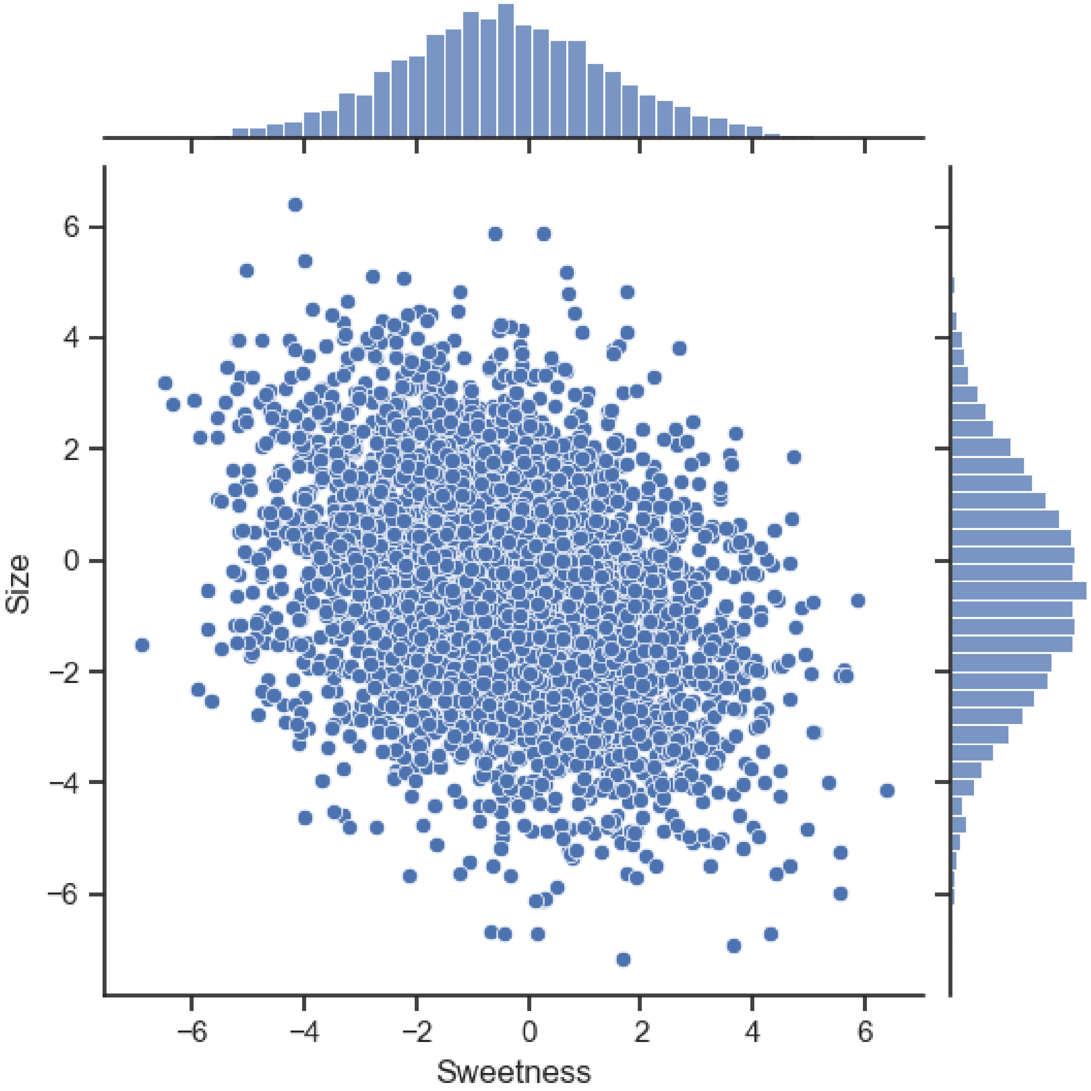
```
/var/folders/8l/5pgwt05s0h5_ftplv2qxvwlm0000gn/T/
ipykernel_44939/3326567540.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

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(df_data['Sweetness'])
```

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

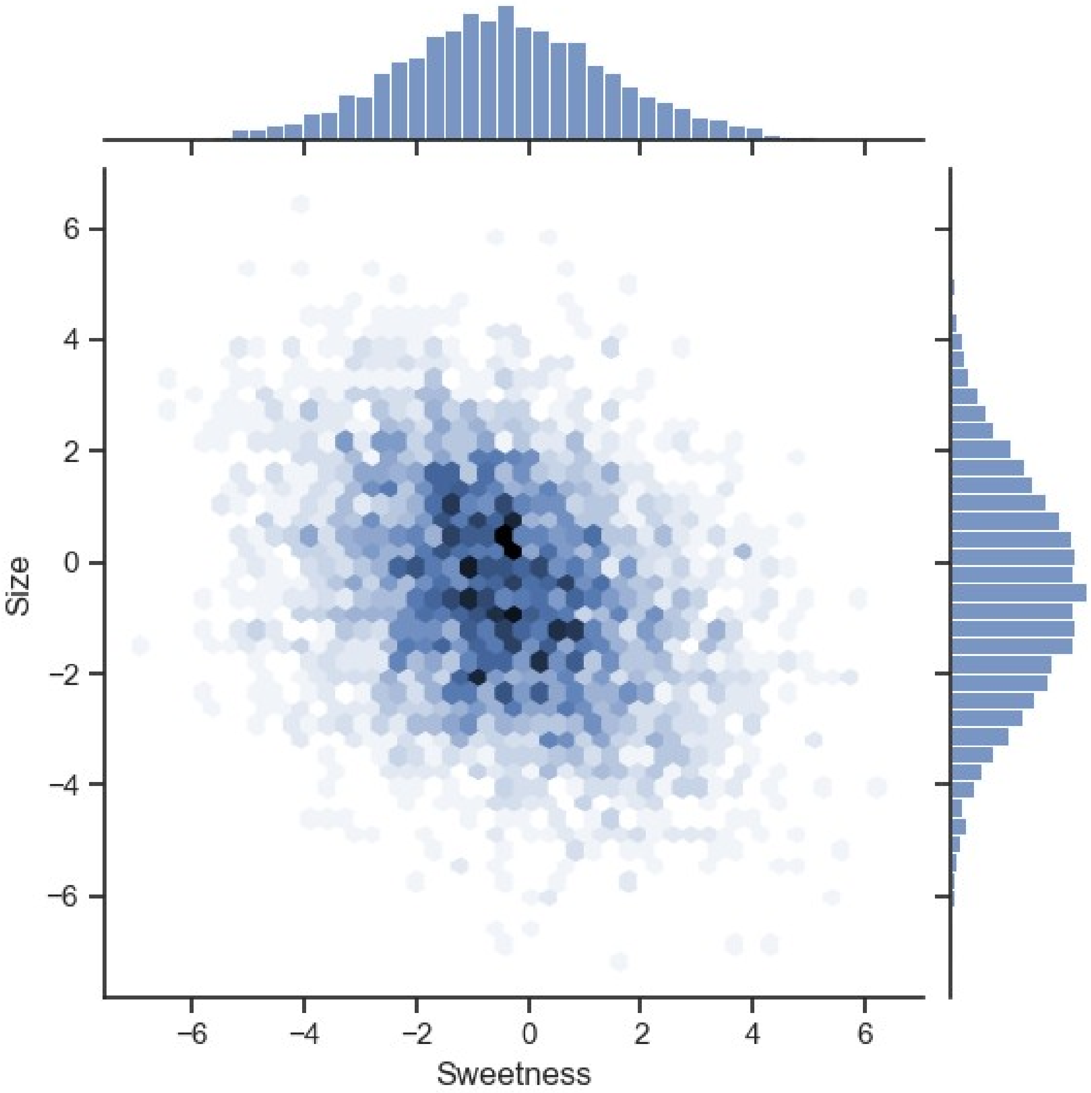


```
# Создание совместного графика

sns.jointplot(x='Sweetness', y='Size', data=df_data)
```

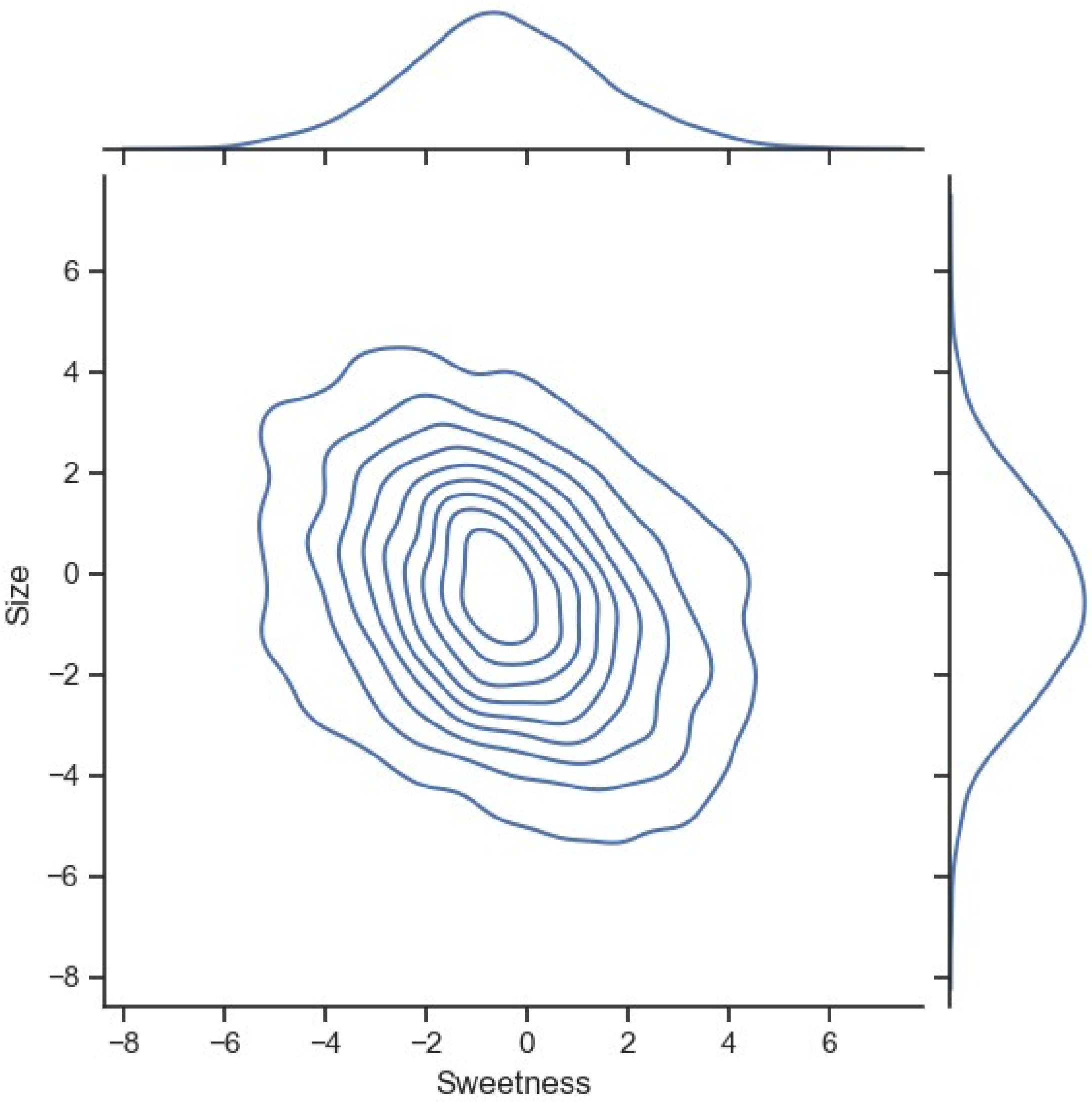
```
<seaborn.axisgrid.JointGrid at 0x1417de130>
```



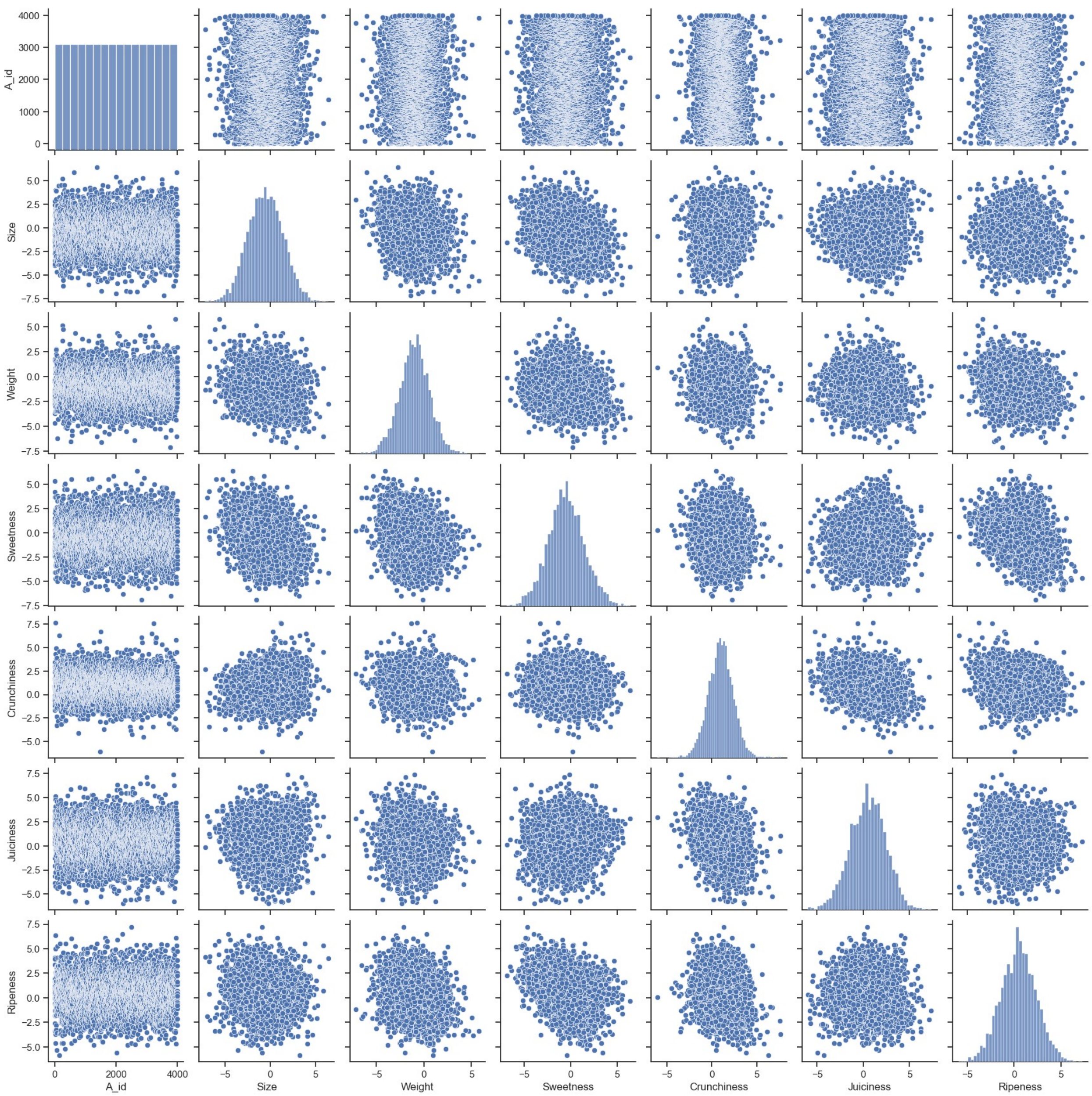
```
# Создание совместного графика

sns.jointplot(x='Sweetness', y='Size', data=df_data, kind="hex")

<seaborn.axisgrid.JointGrid at 0x151036070>
```



```
# Создание совместного графика

sns.jointplot(x='Sweetness', y='Size', data=df_data, kind="kde")

<seaborn.axisgrid.JointGrid at 0x151225970>
```

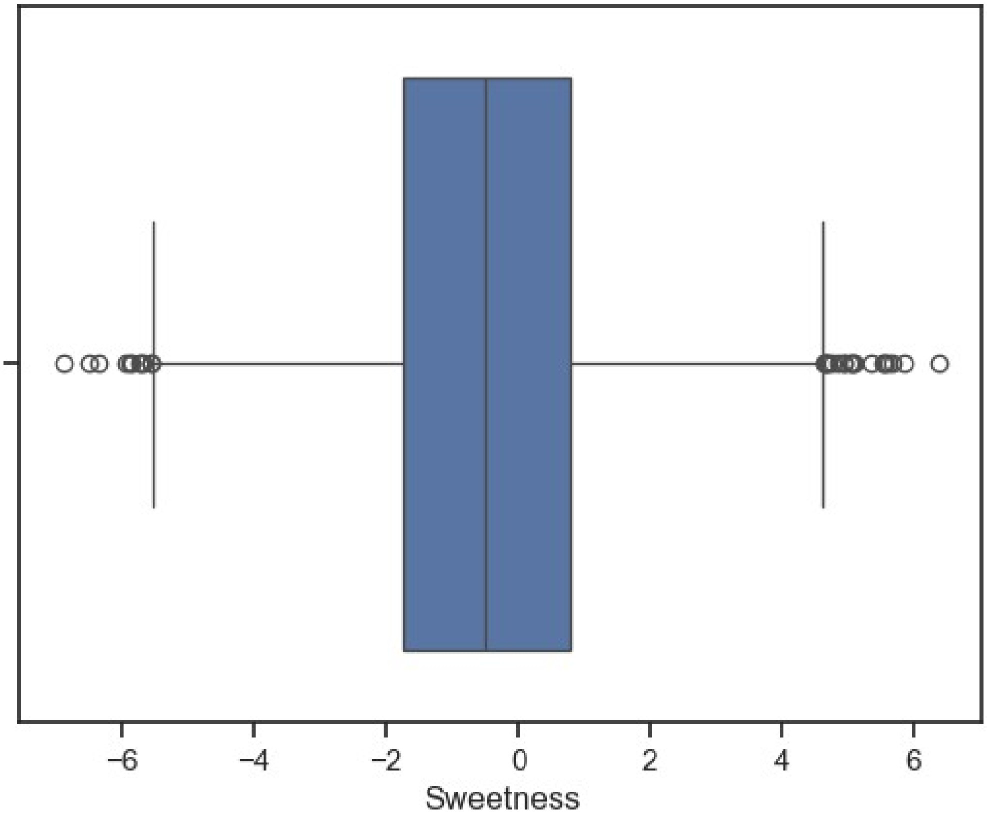


```
# Создание pairplot

sns.pairplot(df_data)

<seaborn.axisgrid.PairGrid at 0x15139bb50>
```



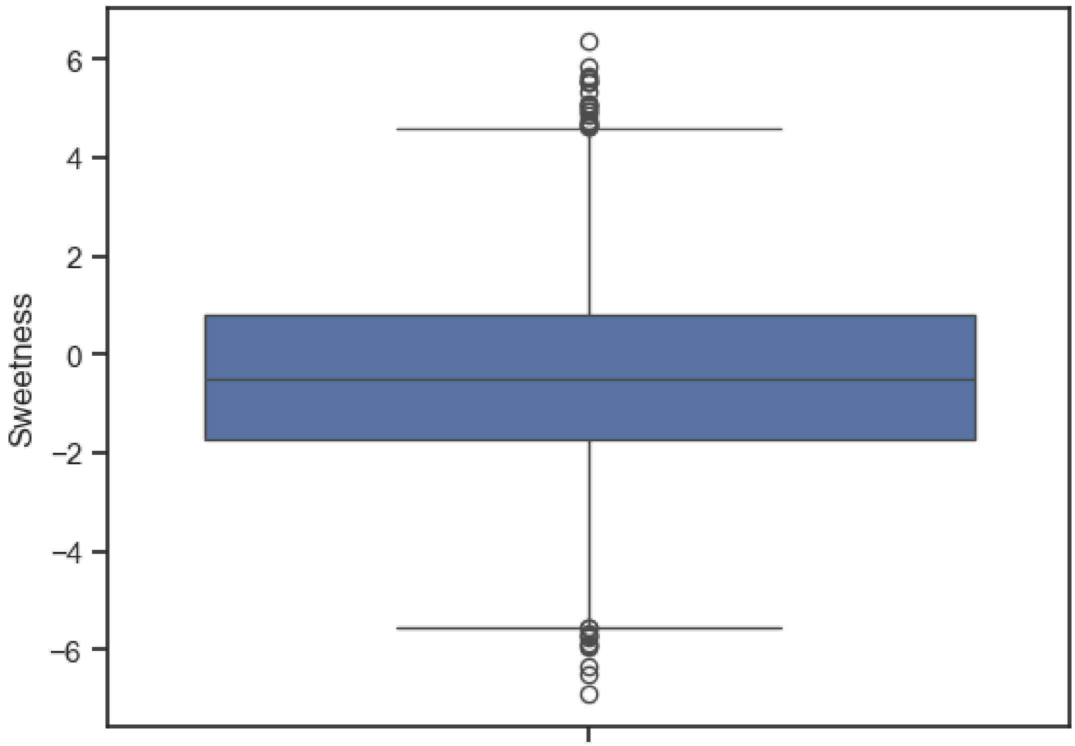
```
sns.boxplot(x=df_data['Sweetness'])
```

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



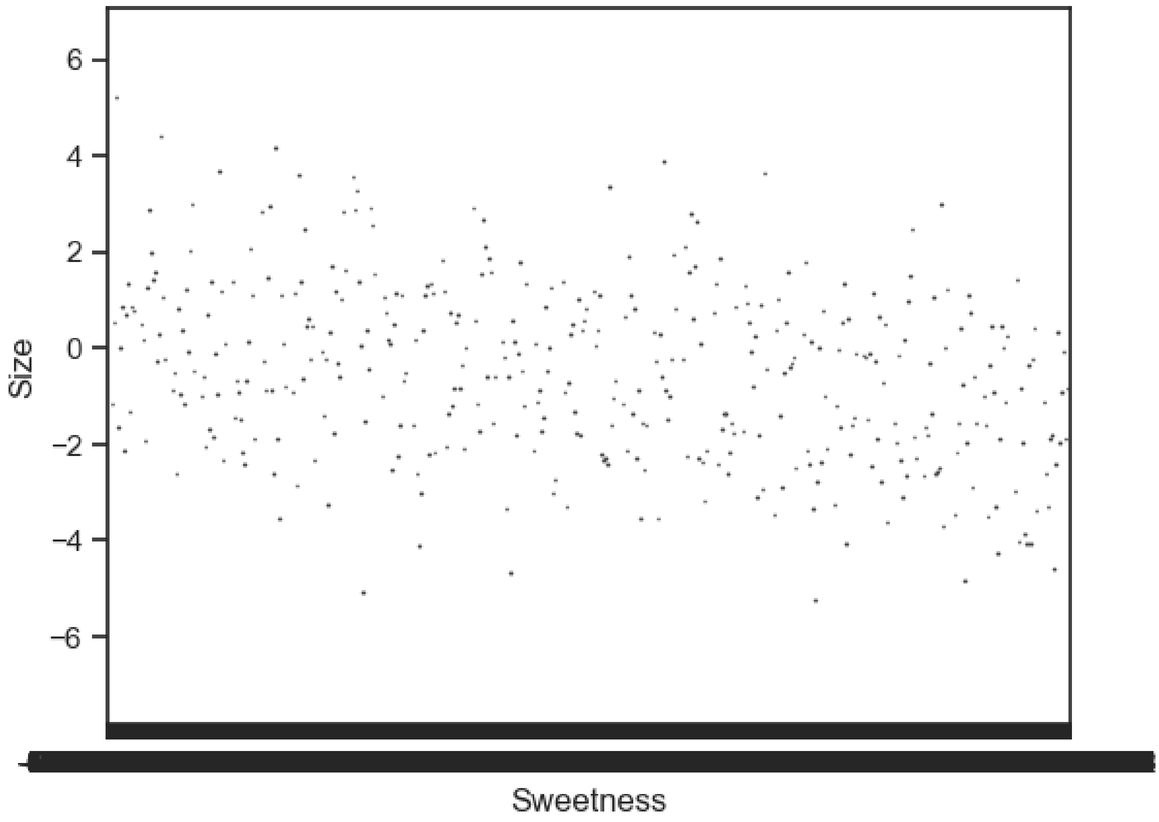
```
sns.boxplot(y=df_data['Sweetness'])
```

```
<Axes: ylabel='Sweetness'>
```



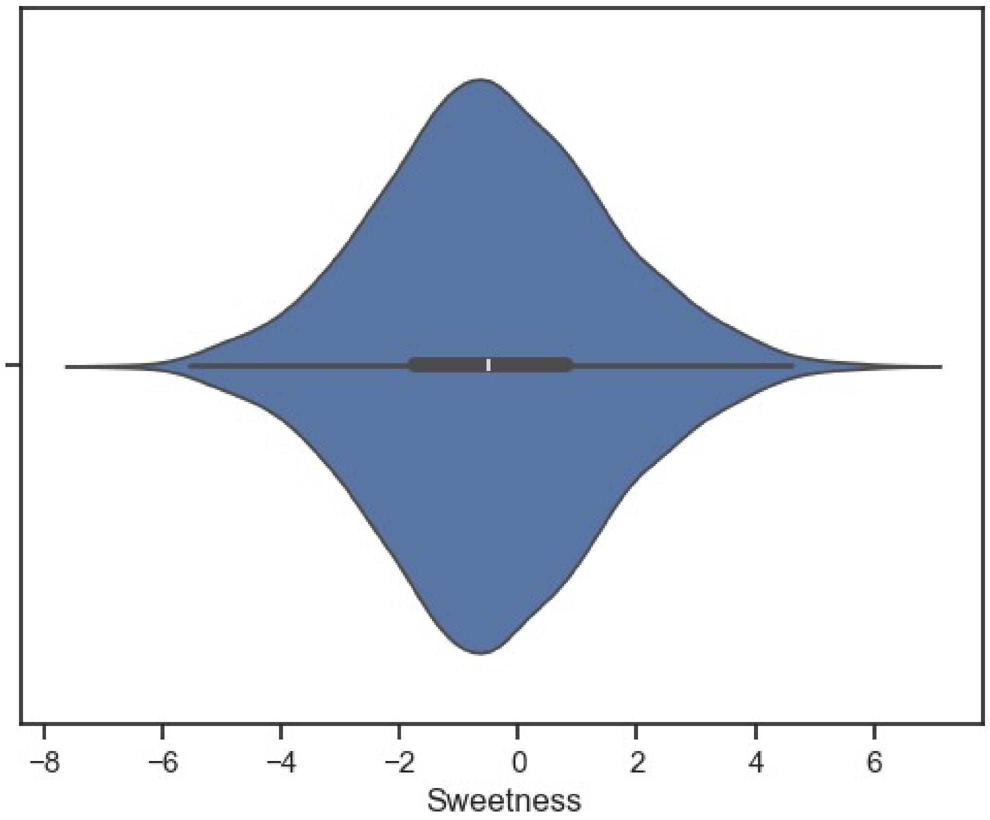
```
sns.boxplot(x='Sweetness', y='Size', data=df_data)
```

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



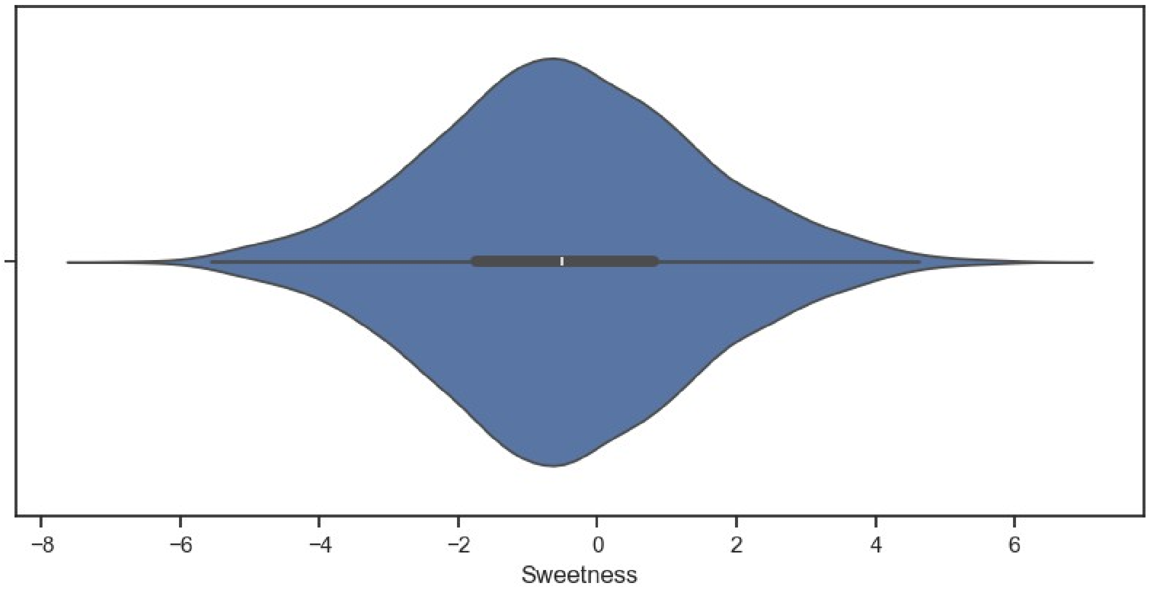
```
sns.violinplot(x=df_data['Sweetness'])
```

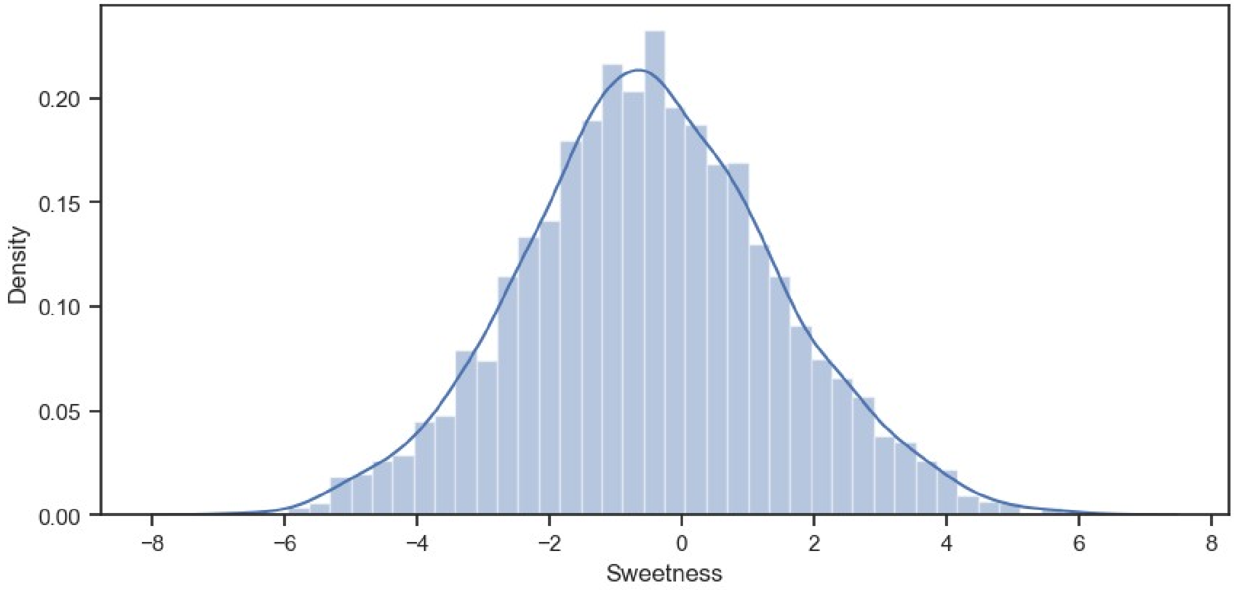
```
<Axes: xlabel='Sweetness'>
```



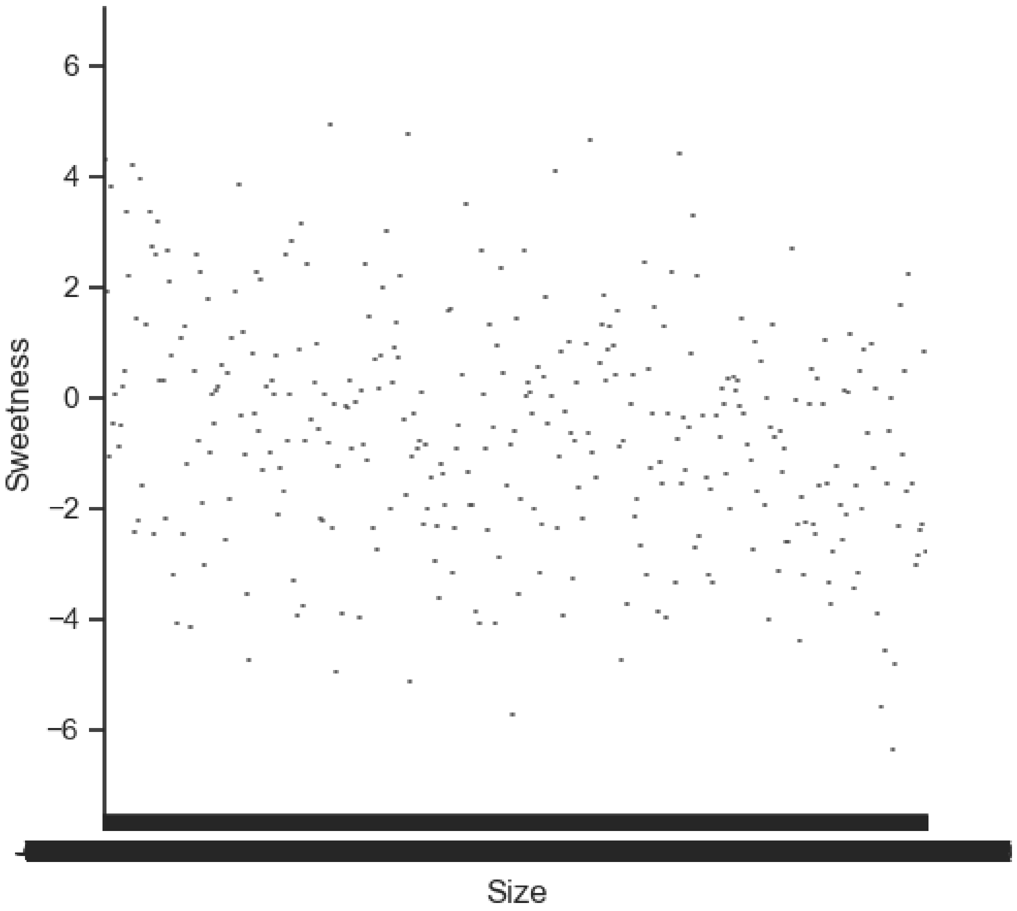
```python
fig, ax = plt.subplots(2, 1, figsize=(10,10))
sns.violinplot(ax=ax[0], x=df_data['Sweetness'])
sns.distplot(df_data['Sweetness'], ax=ax[1])
```

```
/var/folders/8l/5pgwt05s0h5_ftplv2qxvwlm0000gn/T/
ipykernel_44939/2581262117.py:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

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(df_data['Sweetness'], ax=ax[1])

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





```
sns.catplot(y='Sweetness', x='Size', data=df_data, kind="violin",
split=True)

<seaborn.axisgrid.FacetGrid at 0x157d38a30>
```



```
# Вычисляется корреляционная матрица для числовых столбцов (исключая
Quality и Acidity) с помощью метода .corr()

df_data.drop(columns=["Quality", "Acidity"]).corr()

                 A_id      Size    Weight  Sweetness  Crunchiness  
Juiciness  \
A_id         1.000000 -0.028911 -0.005730  -0.002378    -0.013111   
0.006179   
Size        -0.028911  1.000000 -0.170702  -0.324680     0.169868  -
0.018892   
Weight      -0.005730 -0.170702  1.000000  -0.154246    -0.095882  -
0.092263   
Sweetness   -0.002378 -0.324680 -0.154246   1.000000    -0.037552   
0.095882   
Crunchiness -0.013111  0.169868 -0.095882  -0.037552     1.000000  -
0.259607   
Juiciness    0.006179 -0.018892 -0.092263   0.095882    -0.259607   
1.000000   
Ripeness     0.000742 -0.134773 -0.243824  -0.273800    -0.201982  -
0.097144   
```

```
             Ripeness
A_id         0.000742
Size        -0.134773
Weight      -0.243824
Sweetness   -0.273800
Crunchiness -0.201982
Juiciness   -0.097144
Ripeness     1.000000
```

```
df_data.drop(columns=["Quality", "Acidity"]).corr(method='pearson')
```

```
                 A_id      Size    Weight  Sweetness  Crunchiness
Juiciness  \
A_id         1.000000 -0.028911 -0.005730  -0.002378    -0.013111
0.006179
Size        -0.028911  1.000000 -0.170702  -0.324680     0.169868  -
0.018892
Weight      -0.005730 -0.170702  1.000000  -0.154246    -0.095882  -
0.092263
Sweetness   -0.002378 -0.324680 -0.154246   1.000000    -0.037552
0.095882
Crunchiness -0.013111  0.169868 -0.095882  -0.037552     1.000000  -
0.259607
Juiciness    0.006179 -0.018892 -0.092263   0.095882    -0.259607
1.000000
Ripeness     0.000742 -0.134773 -0.243824  -0.273800    -0.201982  -
0.097144

             Ripeness
A_id         0.000742
Size        -0.134773
Weight      -0.243824
Sweetness   -0.273800
Crunchiness -0.201982
Juiciness   -0.097144
Ripeness     1.000000
```

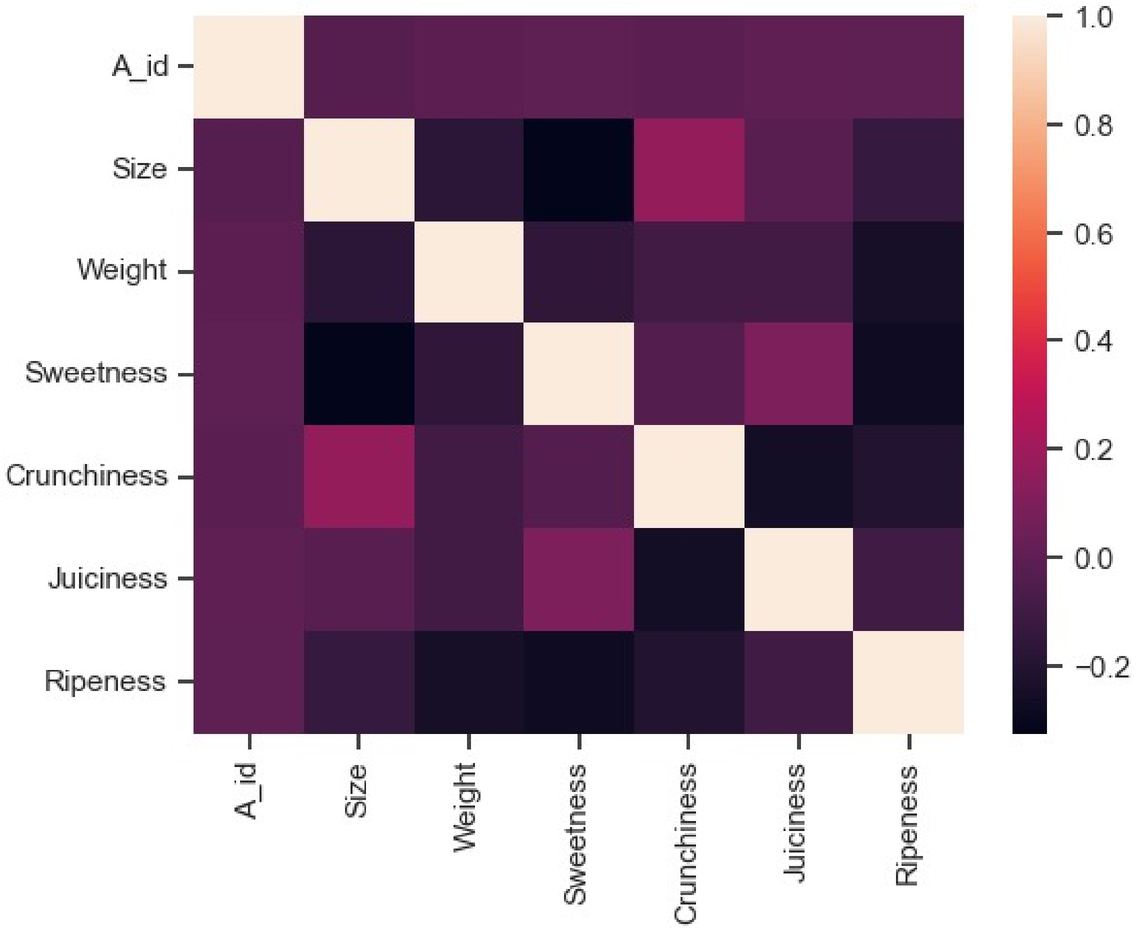
```
df_data.drop(columns=["Quality", "Acidity"]).corr(method='kendall')
```

```
                 A_id      Size    Weight  Sweetness  Crunchiness
Juiciness  \
A_id         1.000000 -0.022124 -0.004756   0.001090    -0.010822
0.002903
Size        -0.022124  1.000000 -0.097221  -0.211004     0.118658  -
0.023001
Weight      -0.004756 -0.097221  1.000000  -0.080836    -0.058782  -
0.060676
Sweetness    0.001090 -0.211004 -0.080836   1.000000    -0.011565
0.065046
Crunchiness -0.010822  0.118658 -0.058782  -0.011565     1.000000  -
```

```
0.161359
Juiciness     0.002903 -0.023001 -0.060676   0.065046    -0.161359
1.000000
Ripeness     -0.003643 -0.101724 -0.166940  -0.171992    -0.125027  -
0.085860

             Ripeness
A_id        -0.003643
Size        -0.101724
Weight      -0.166940
Sweetness   -0.171992
Crunchiness -0.125027
Juiciness   -0.085860
Ripeness     1.000000
```

```
sns.heatmap(df_data.drop(columns=["Quality", "Acidity"]).corr())
```

```
<Axes: >
```


A_id
Size
Weight
Sweetness
Crunchiness
Juiciness
Ripeness
A_id
Size
Weight
Sweetness
Crunchiness
Juiciness
Ripeness