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 «Системы обработки информации и управления»

Отчет по рубежному контролю №1

«Технологии разведочного анализа и обработки данных»

по дисциплине «Технологии машинного обучения»

Вариант №26

Выполнил:
студент группы ИУ5Ц-84Б
Клеша Е.С.
подпись, дата

Проверил:
к.т.н., доц., Ю.Е. Гапанюк
подпись, дата

2024 г.

СОДЕРЖАНИЕ ОТЧЕТА

1. Примечания:

Если в Вашем наборе данных отсутствуют данные, необходимые для решения задачи, создайте их искусственно. Например, если отсутствуют категориальные признаки, создайте категориальный признак на основе числового. Если отсутствуют пропуски, замените на пропуски часть значений в одном или нескольких признаках.

Также Вы можете дополнительно использовать датасеты, содержащие необходимые данные, например использовать дополнительный датасет, содержащий пропуски.

2. Дополнительные требования по группам:

1. Для студентов группы ИУ5-64Б, ИУ5Ц-84Б - для произвольной колонки данных построить график "Скрипичная диаграмма (violin plot)".

Задача №4

Для заданного набора данных постройте основные графики, входящие в этап разведочного анализа данных. В случае наличия пропусков в данных удалите строки или колонки, содержащие пропуски. Какие графики Вы построили и почему? Какие выводы о наборе данных Вы можете сделать на основании построенных графиков?

Наборы данных: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_wine.html#sklearn.datasets.load_wine

3. Листинг

3.1. Импорт библиотек, загрузка данных

```
import sys
sys.path
import pandas as pd
import numpy as np
import seaborn as sns
np.seterr(divide='ignore', invalid='ignore')
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
%matplotlib inline
```

```
wine = load_wine()
df = pd.DataFrame(wine.data, columns=wine.feature_names)
df['TARGET'] = wine.target
```

3.2. Общее описание датасета

Первые пять строк датасета

```
df.head()
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29	5.64	1.04	
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4.38	1.05	
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	2.81	5.68	1.03	
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	2.18	7.80	0.86	
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	1.82	4.32	1.04	

3.3. Общее описание

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 14 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   alcohol                       178 non-null    float64
 1   malic_acid                    178 non-null    float64
 2   ash                           178 non-null    float64
 3   alcalinity_of_ash             178 non-null    float64
 4   magnesium                     178 non-null    float64
 5   total_phenols                 178 non-null    float64
 6   flavanoids                    178 non-null    float64
 7   nonflavanoid_phenols          178 non-null    float64
 8   proanthocyanins               178 non-null    float64
 9   color_intensity               178 non-null    float64
 10  hue                           178 non-null    float64
 11  od280/od315_of_diluted_wines  178 non-null    float64
 12  proline                       178 non-null    float64
 13  TARGET                        178 non-null    int64  
dtypes: float64(13), int64(1)
memory usage: 19.6 KB
```

Проверим количество пустых значений

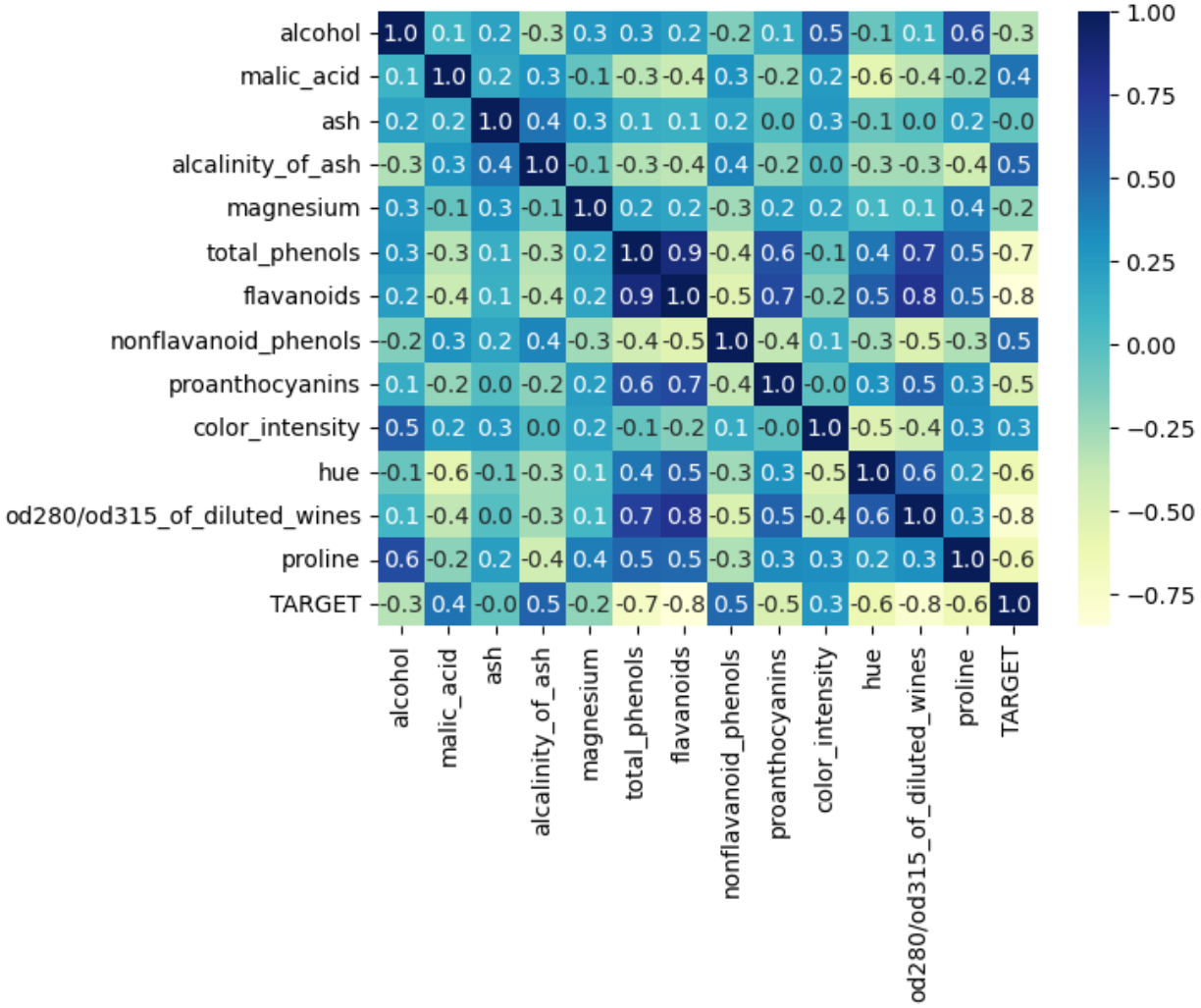
```
for col_empty in df.columns:
    empty_count = df[df[col_empty].isnull()].shape[0]
    print('{} - {}'.format(col_empty, empty_count))
```

```
alcohol - 0
malic_acid - 0
ash - 0
alcalinity_of_ash - 0
magnesium - 0
total_phenols - 0
flavanoids - 0
nonflavanoid_phenols - 0
proanthocyanins - 0
color_intensity - 0
hue - 0
od280/od315_of_diluted_wines - 0
proline - 0
TARGET - 0
```

Пустых значений не обнаружено.

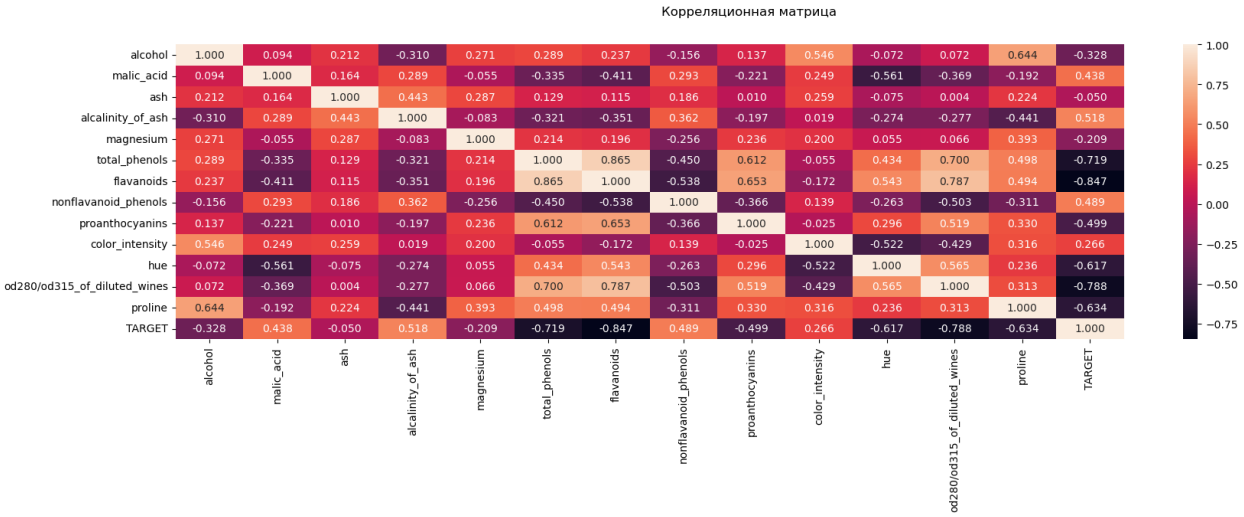
3.4. Корреляция признаков

```
sns.heatmap(df.corr(), cmap='YlGnBu', annot=True, fmt='.1f')
```



Наиболее сильную корреляцию имеют признаки total_phenols и flavanoids. Это связано с тем, что флавониды относятся к классу полифенолов.

```
fig, ax = plt.subplots(1, 1, sharex='col', sharey='row', figsize=(20,5))
fig.suptitle('Корреляционная матрица')
sns.heatmap(df.corr(), ax=ax, annot=True, fmt='.3f')
```



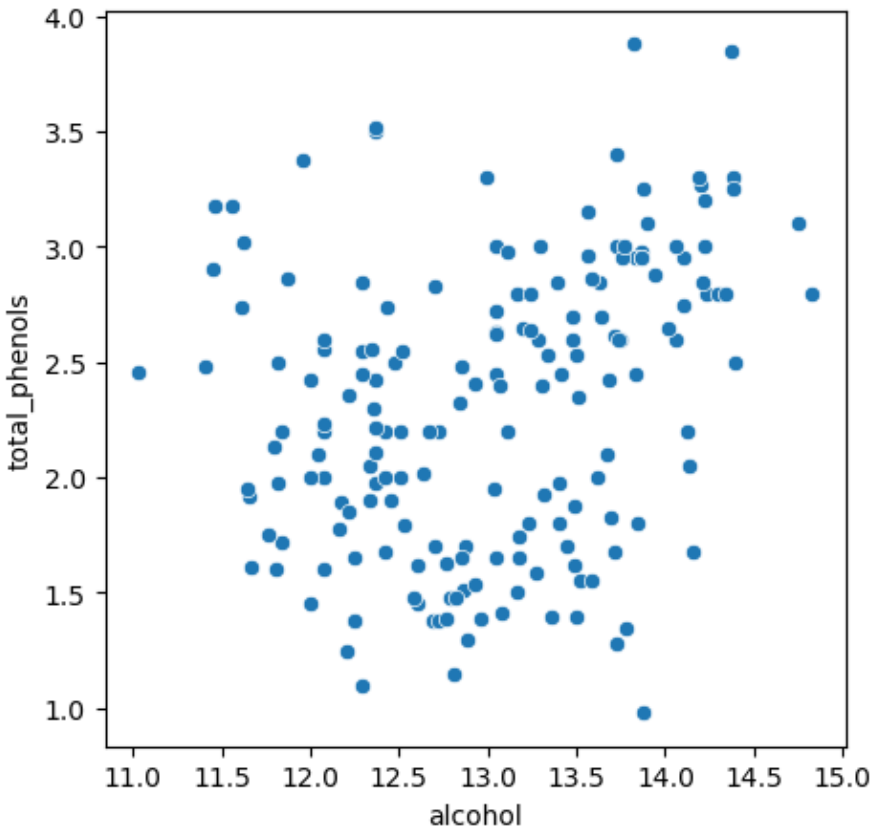
С целевым признаком TARGET сильнее всего коррелируют признаки "flavanoids", "od280/od315_of_diluted_wines", "total_phenols", "hue", "proline". Соответственно, их стоит учитывать для более информативного построения модели машинного обучения.

3.5. Визуальное исследование датасета

Диаграмма рассеивания для признаков total_phenols и alcohol

```
fig, ax = plt.subplots(figsize=(5,5))
sns.scatterplot(ax=ax, x='alcohol', y='total_phenols', data=df)
```

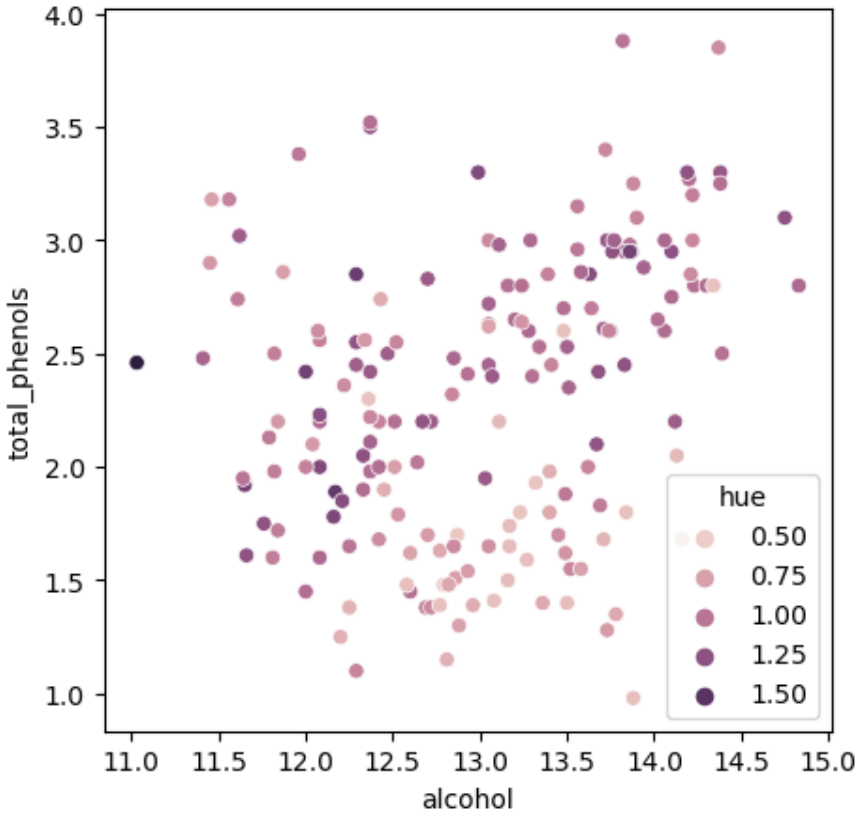
```
<Axes: xlabel='alcohol', ylabel='total_phenols'>
```



Данная диаграмма показывает количество фенолов в каждом проценте вина.

```
1 fig, ax = plt.subplots(figsize=(5,5))
2 sns.scatterplot(ax=ax, x='alcohol', y='total_phenols', data=df, hue='hue')
```

```
<Axes: xlabel='alcohol', ylabel='total_phenols'>
```

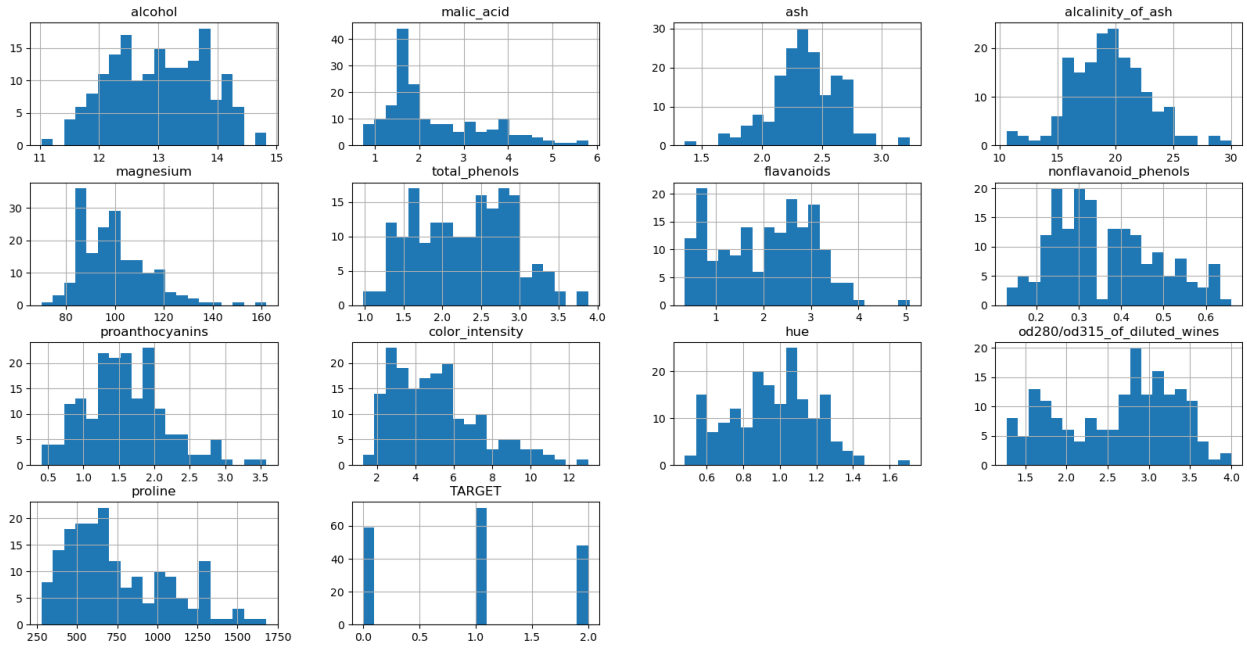


Такая же диаграмма показывает количество фенолов в каждом проценте вина, но еще добавили "hue", т.е. в каждой точке можем рассмотреть оттеннок конкретного вина.

3.6. Гистограмма для всех признаков

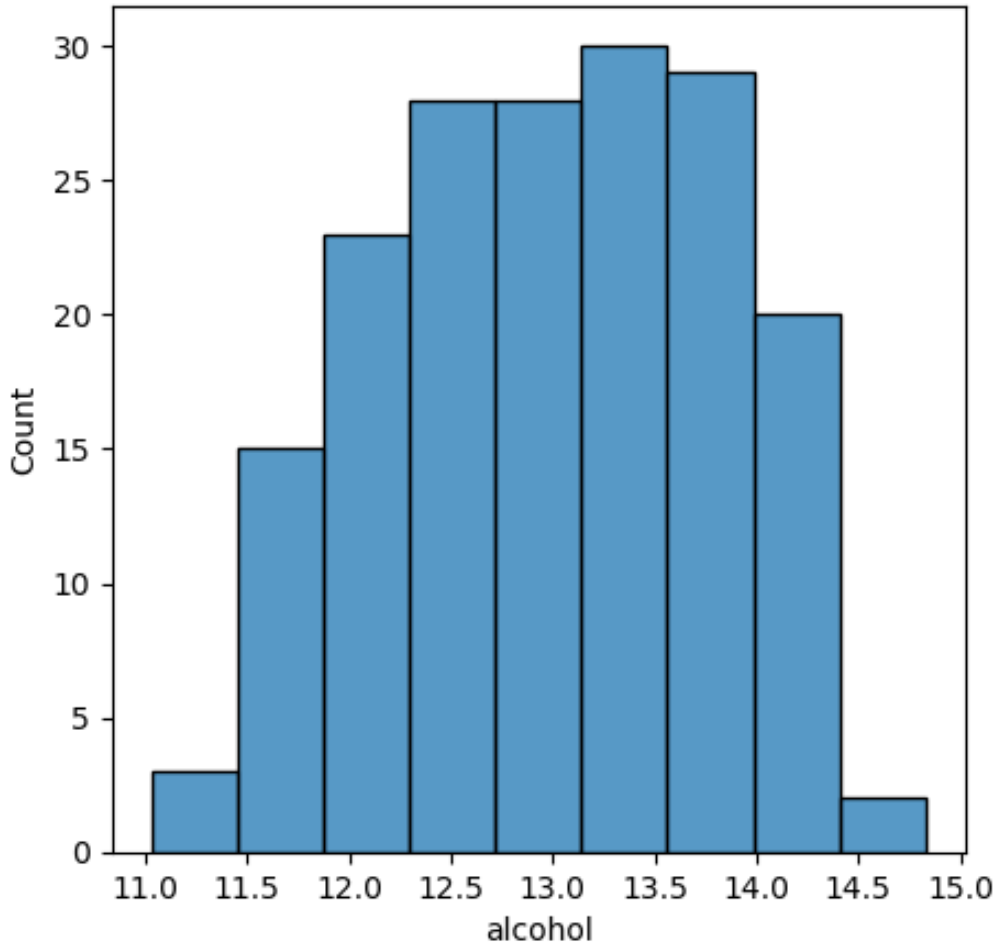
```
1 df.hist(bins=20, figsize = (20,10))
```

```
array([[<Axes: title={'center': 'alcohol'}>,
        <Axes: title={'center': 'malic_acid'}>,
        <Axes: title={'center': 'ash'}>,
        <Axes: title={'center': 'alcalinity_of_ash'}>],
       [<Axes: title={'center': 'magnesium'}>,
        <Axes: title={'center': 'total_phenols'}>,
        <Axes: title={'center': 'flavanoids'}>,
        <Axes: title={'center': 'nonflavanoid_phenols'}>],
       [<Axes: title={'center': 'proanthocyanins'}>,
        <Axes: title={'center': 'color_intensity'}>,
        <Axes: title={'center': 'hue'}>,
        <Axes: title={'center': 'od280/od315_of_diluted_wines'}>],
       [<Axes: title={'center': 'proline'}>,
        <Axes: title={'center': 'TARGET'}>, <Axes: >, <Axes: >]],
      dtype=object)
```




```
fig, ax = plt.subplots(figsize=(5,5))
sns.histplot(df['alcohol'])
```

```
<Axes: xlabel='alcohol', ylabel='Count'>
```

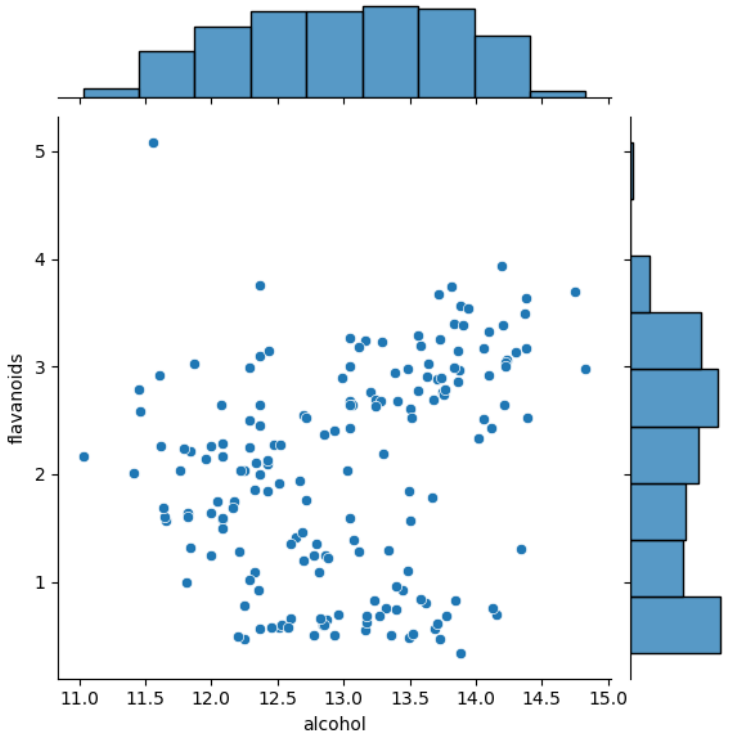


Данная гистограмма показывает наибольшее количество процента алкоголя в вине.

3.7. Jointplot

```
1 sns.jointplot(x='alcohol', y='flavanoids', data=df)
```

```
<seaborn.axisgrid.JointGrid at 0x12ed63a10>
```



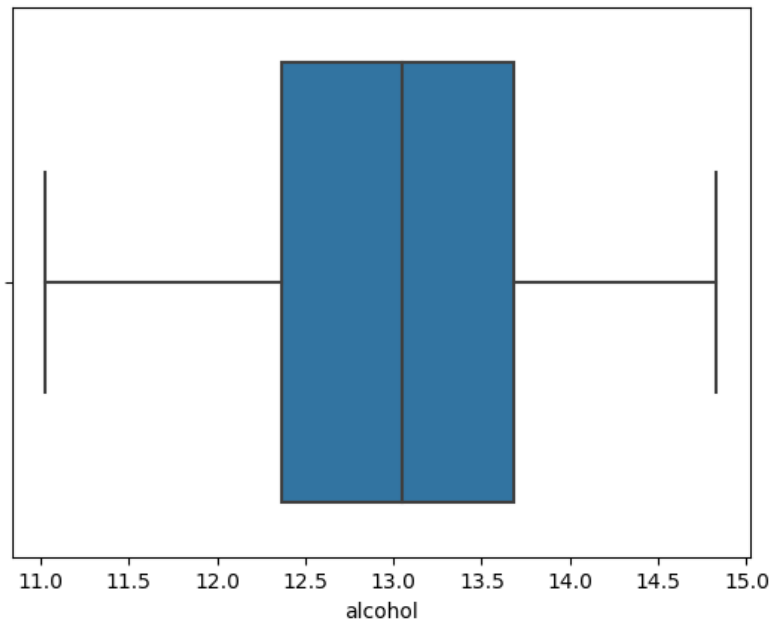
Комбинация гистограмм и диаграмм рассеивания.

3.8. “Ящик с усами”

3.8.1. По оси абсцисс

```
1 sns.boxplot(x=df['alcohol'])
```

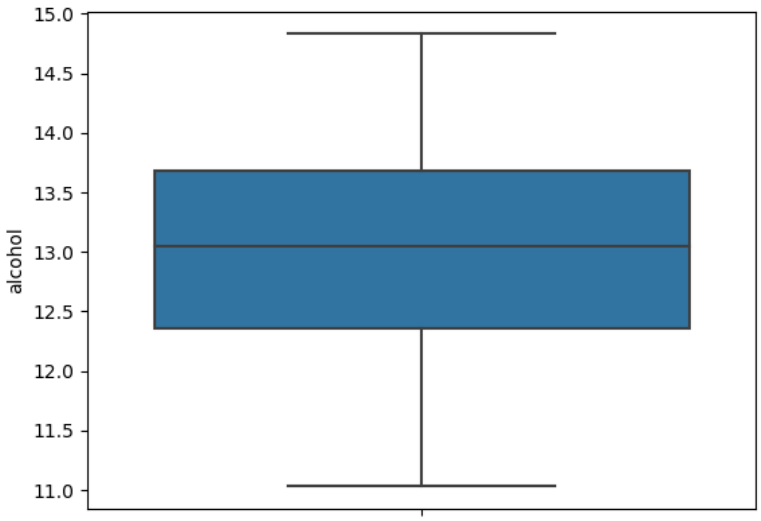
```
<Axes: xlabel='alcohol'>
```



3.8.2. По оси ординат

```
1 sns.boxplot(y=df['alcohol'])
```

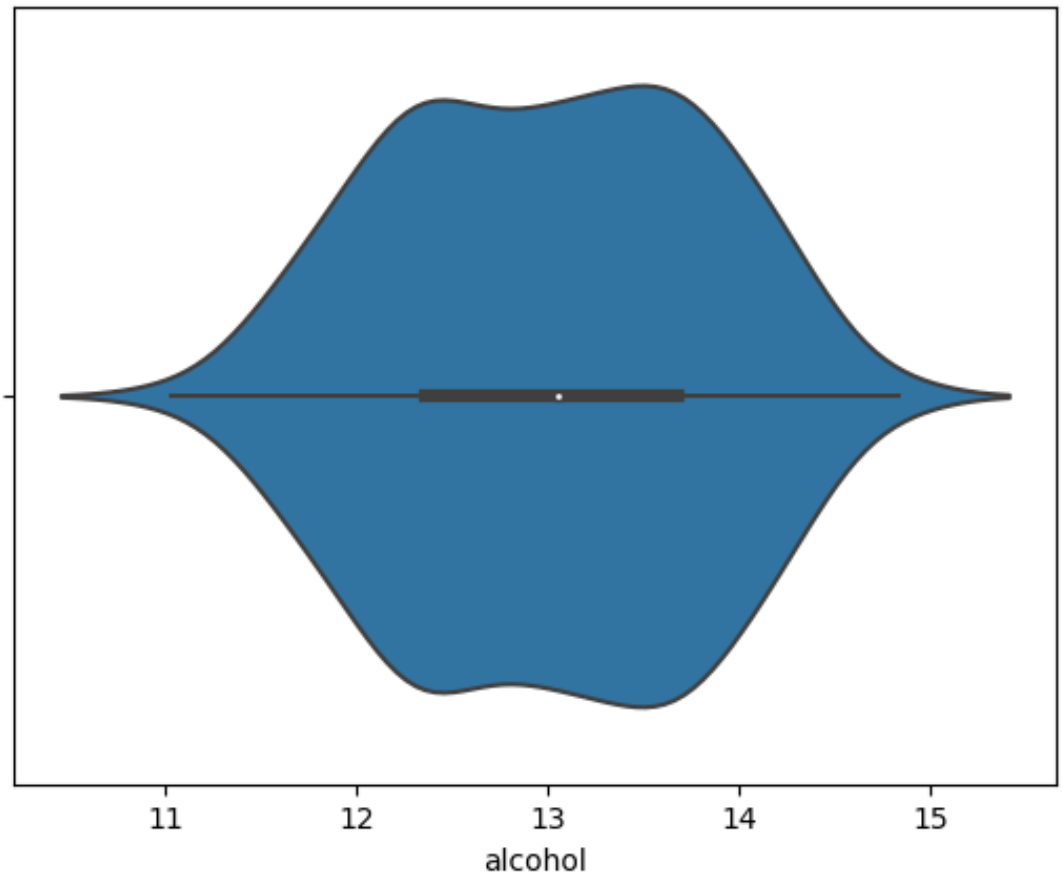
```
<Axes: ylabel='alcohol'>
```



3.9. Скрипичная диаграмма

```
1 sns.violinplot(x=df['alcohol'])
```

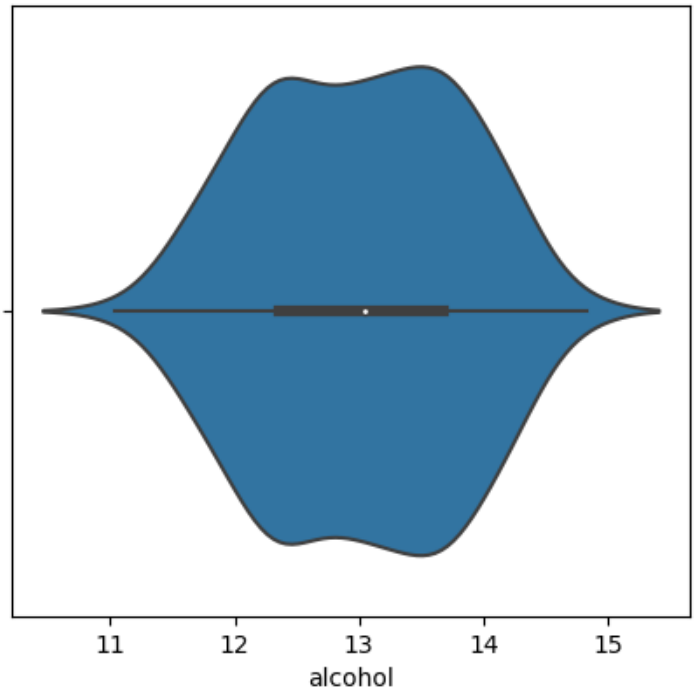
```
<Axes: xlabel='alcohol'>
```

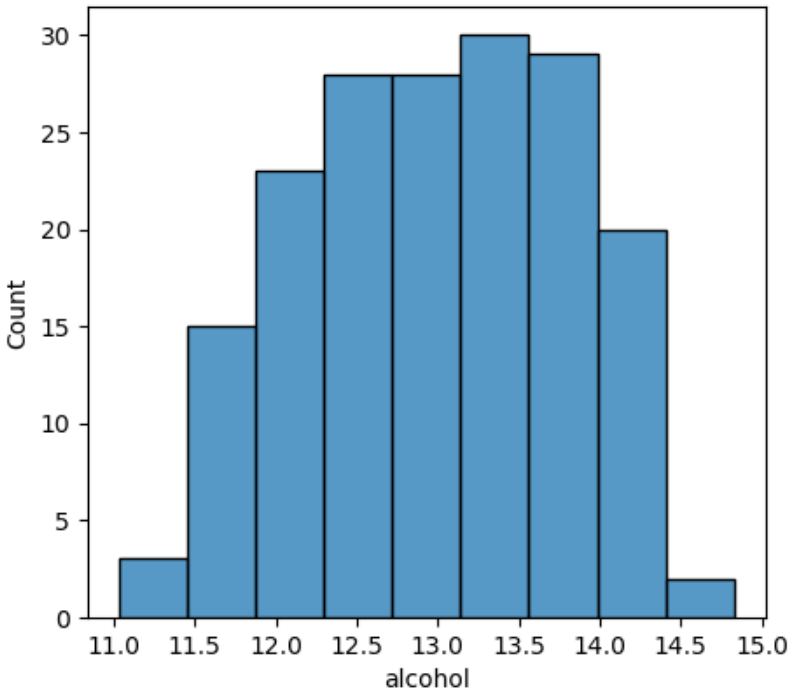


Скрипичная диаграмма показывает распределение плотности по краям диаграммы.

```
fig, ax = plt.subplots(2, 1, figsize=(5,5))
sns.violinplot(ax=ax[0], x=df['alcohol'])
sns.histplot(df['alcohol'])
```

```
<Axes: xlabel='alcohol', ylabel='Count'>
```





Из приведенных графиков видно, что скрипичная диаграмма действительно показывает распределение плотности.