



**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 вычислительная техника»  
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Технология машинного обучения»

Отчет по лабораторной работе №1  
«Разведочный анализ данных. Исследование и визуализация данных.»

Выполнил:  
студент группы ИУ5-63Б

Баркалова И.В.

Подпись и дата:

Проверил:  
преподаватель каф.  
ИУ5

Гапанюк Ю.Е.

Подпись и дата:

Москва, 2022 г.

## **Цель лабораторной работы:**

Изучение различных методов визуализация данных.

## **Описание задания:**

- Выбрать набор данных (датасет)
- Для первой лабораторной работы рекомендуется использовать датасет без пропусков в данных

Для лабораторных работ не рекомендуется выбирать датасеты большого размера.

- Создать ноутбук, который содержит следующие разделы:
  1. Текстовое описание выбранного Вами набора данных.
  2. Основные характеристики датасета.
  3. Визуальное исследование датасета.

## ТЕКСТОВОЕ ОПИСАНИЕ НАБОРА ДАННЫХ

В качестве набора данных мы будем использовать набор данных по вину. Рассмотрим такой DataSet для того, чтобы исследовать - какую опасность для нас несет алкоголь. Из данных получим сравнение фенолов и флаваноидов, а также щелочи и яблочной кислоты. Благодаря этому выясним, какое безопасное содержание данных примесей для нашего организма, а также узнаем, как они зависят друг от друга.

Датасет состоит из следующих значений: 1) alcohol - крепость вина 2) malic\_acid - количество яблочной кислоты 3) ash - количество золы 4) alcalinity\_of\_ash - щелочность 5) magnesium - количество магния 6) total\_phenols - количество фенолов 7) flavanoids - количество флаваноидов 8) nonflavanoid\_phenols - количество нефлавоноидных фенолов 9) proanthocyanins - количество проантоцианинов 10) color\_intensity - насыщенность цвета 11) hue - оттенок 12) od280/od315\_of\_diluted\_wines - количество разбавленных винных ферментов 13) proline - количество пролина

## ИМПОРТ БИБЛИОТЕК

Импортируем библиотеки с помощью команды import. Загрузим файлы датасета в помощью библиотеки Pandas.

```
In [4]: #Импорт библиотек
import numpy as np
import pandas as pd
from sklearn.datasets import *
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

```
In [5]: #Преобразование формата в DataFrame - выгрузка датасета про вино
wine = load_wine()
```

```
In [6]: type(wine)
```

```
Out[6]: sklearn.utils.Bunch
```

## ОСНОВНЫЕ ХАРАКТЕРИСТИКИ DATASET

```
In [7]: # Датасет возвращается в виде словаря со следующими ключами
for x in wine:
    print(x)
```

```
data
target
frame
target_names
DESCR
feature_names
```

```
In [8]: wine['target_names']
```

```
Out[8]: array(['class_0', 'class_1', 'class_2'], dtype='<U7')
```

```
In [9]: wine['feature_names']
```

```
Out[9]: ['alcohol',
'malic_acid',
'ash',
'alcalinity_of_ash',
'magnesium',
'total_phenols',
'flavanoids',
'nonflavanoid_phenols',
'proanthocyanins',
'color_intensity',
'hue',
'od280/od315_of_diluted_wines',
'proline']
```

```
In [10]: # Размерность данных
wine['data'].shape
```

In [10]: `# Размерность данных`  
`wine['data'].shape`

Out[10]: (178, 13)

In [11]: `# Размерность целевого признака`  
`wine['target'].shape`

Out[11]: (178,)

In [12]: `# Преобразование в Pandas DataFrame`  
`data1 = pd.DataFrame(data= np.c_[wine['data'], wine['target']],`  
`columns= wine['feature_names'] + ['target'])`

In [13]: `data1`

Out[13]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_dilut
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	
...	...	...	...	...	...	...	...	...	...	...	...	...
173	13.71	5.65	2.45	20.5	95.0	1.68	0.61	0.52	1.06	7.70	0.64	
174	13.40	3.91	2.48	23.0	102.0	1.80	0.75	0.43	1.41	7.30	0.70	
175	13.27	4.28	2.26	20.0	120.0	1.59	0.69	0.43	1.35	10.20	0.59	
176	13.17	2.59	2.37	20.0	120.0	1.65	0.68	0.53	1.46	9.30	0.60	
177	14.13	4.10	2.74	24.5	96.0	2.05	0.76	0.56	1.35	9.20	0.61	

178 rows x 14 columns

In [14]: `# Количество строк в DataSet`  
`total_count = data1.shape[0]`  
`print('Всего строк: {}'.format(total_count))`

Всего строк: 178

In [15]: `# Список колонок`  
`data1.columns`

Out[15]: Index(['alcohol', 'malic\_acid', 'ash', 'alcalinity\_of\_ash', 'magnesium',  
          'total\_phenols', 'flavanoids', 'nonflavanoid\_phenols',  
          'proanthocyanins', 'color\_intensity', 'hue',  
          'od280/od315\_of\_diluted\_wines', 'proline', 'target'],  
      dtype='object')

In [16]: `# Список типов данных всех переменных в DataSet  
data1.dtypes`

Out[16]: `alcohol float64  
malic_acid float64  
ash float64  
alcalinity_of_ash float64  
magnesium float64  
total_phenols float64  
flavanoids float64  
nonflavanoid_phenols float64  
proanthocyanins float64  
color_intensity float64  
hue float64  
od280/od315_of_diluted_wines float64  
proline float64  
target float64  
dtype: object`

In [17]: `# Проверка на пустые значения  
for col in data1.columns:  
 # Количество пустых значений - все значения заполнены  
 temp_null_count = data1[data1[col].isnull()].shape[0]  
 print('{} - {}'.format(col, temp_null_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`

In [18]: `# Основные статистические характеристики набора данных  
data1.describe()`

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue
count	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000
mean	13.000618	2.336348	2.366517	19.494944	99.741573	2.295112	2.029270	0.361854	1.590899	5.058090	0.957449
std	0.811827	1.117146	0.274344	3.339564	14.282484	0.625851	0.998859	0.124453	0.572359	2.318286	0.228572
min	11.030000	0.740000	1.360000	10.600000	70.000000	0.980000	0.340000	0.130000	0.410000	1.280000	0.480000
25%	12.362500	1.602500	2.210000	17.200000	88.000000	1.742500	1.205000	0.270000	1.250000	3.220000	0.782500
50%	13.050000	1.865000	2.360000	19.500000	98.000000	2.355000	2.135000	0.340000	1.555000	4.690000	0.965000
75%	13.677500	3.082500	2.557500	21.500000	107.000000	2.800000	2.875000	0.437500	1.950000	6.200000	1.120000
max	14.830000	5.800000	3.230000	30.000000	162.000000	3.880000	5.080000	0.660000	3.580000	13.000000	1.710000

In [19]: `# Определим уникальные значения для целевого признака  
data1['magnesium'].unique()`

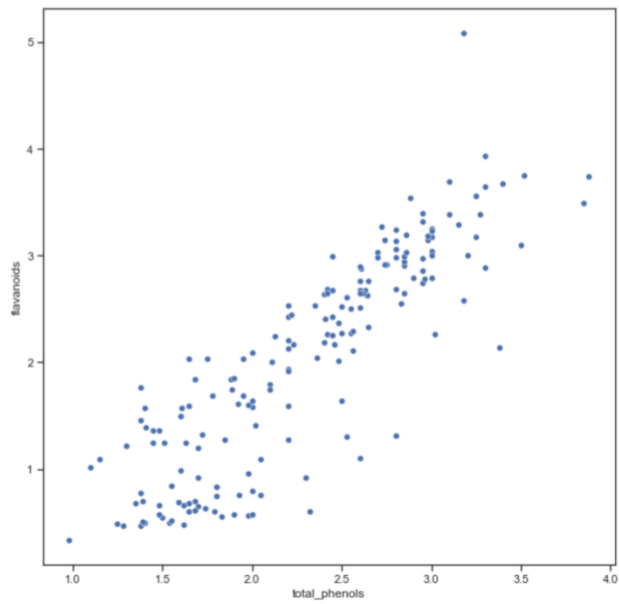
Out[19]: `array([127., 100., 101., 113., 118., 112., 96., 121., 97., 98., 105.,  
 95., 89., 91., 102., 120., 115., 108., 116., 126., 124., 93.,  
 94., 107., 106., 104., 132., 110., 128., 117., 90., 103., 111.,  
 92., 88., 87., 78., 151., 86., 139., 136., 85., 99., 84.,  
 70., 81., 80., 162., 134., 119., 82., 122., 123.])`

ВИЗУАЛИЗАЦИЯ DATASET

Для визуального исследования могут быть использованы различные виды диаграмм.

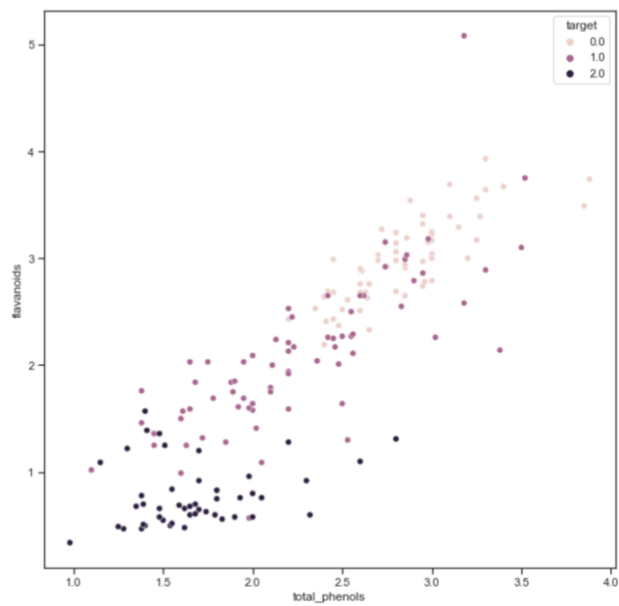
```
In [20]: # Визуализация
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='total_phenols', y='flavanoids', data=data1)
```

```
Out[20]: <AxesSubplot: xlabel='total_phenols', ylabel='flavanoids'>
```



```
In [126]: # Визуализация с группировкой по целевому признаку
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='total_phenols', y='flavanoids', data=data1, hue='magnesium')
```

```
Out[126]: <AxesSubplot: xlabel='total_phenols', ylabel='flavanoids'>
```

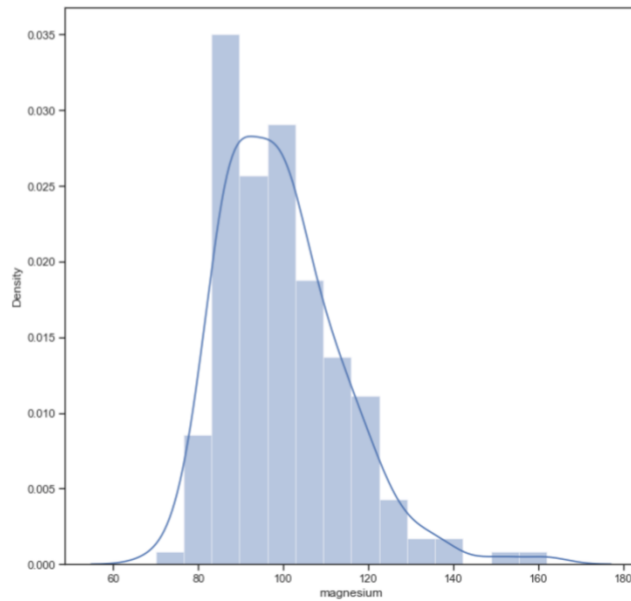


total\_phenols

```
In [21]: # Гистограмма позволяет оценить плотность вероятности распределения данных
fig, ax = plt.subplots(figsize=(10,10))
sns.distplot(data['magnesium'])
```

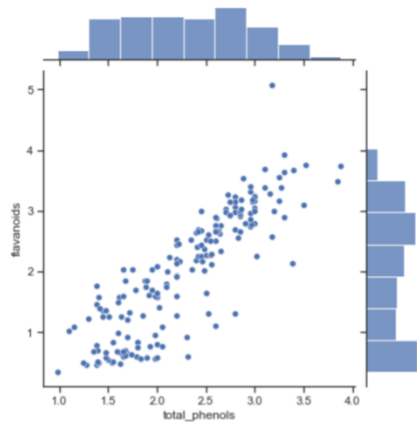
/Users/imac/opt/anaconda3/lib/python3.9/site-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).  
warnings.warn(msg, FutureWarning)

```
Out[21]: <AxesSubplot: xlabel='magnesium', ylabel='Density'>
```



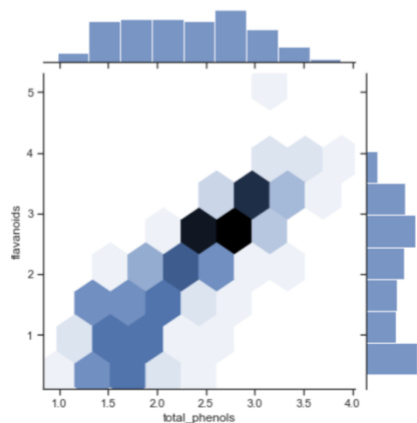
```
In [22]: # Комбинация гистограмм и диаграмм рассеивания.
sns.jointplot(x='total_phenols', y='flavanoids', data=data)
```

```
Out[22]: <seaborn.axisgrid.JointGrid at 0x7fc2fb8ed190>
```



```
In [23]: sns.jointplot(x='total_phenols', y='flavanoids', data=data, kind='hex')
```

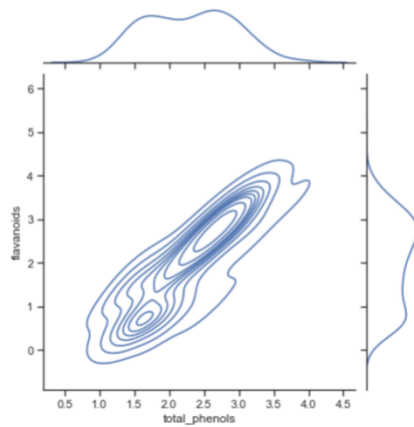
```
Out[23]: <seaborn.axisgrid.JointGrid at 0x7fc2fb8ed280>
```



```
total_phenols

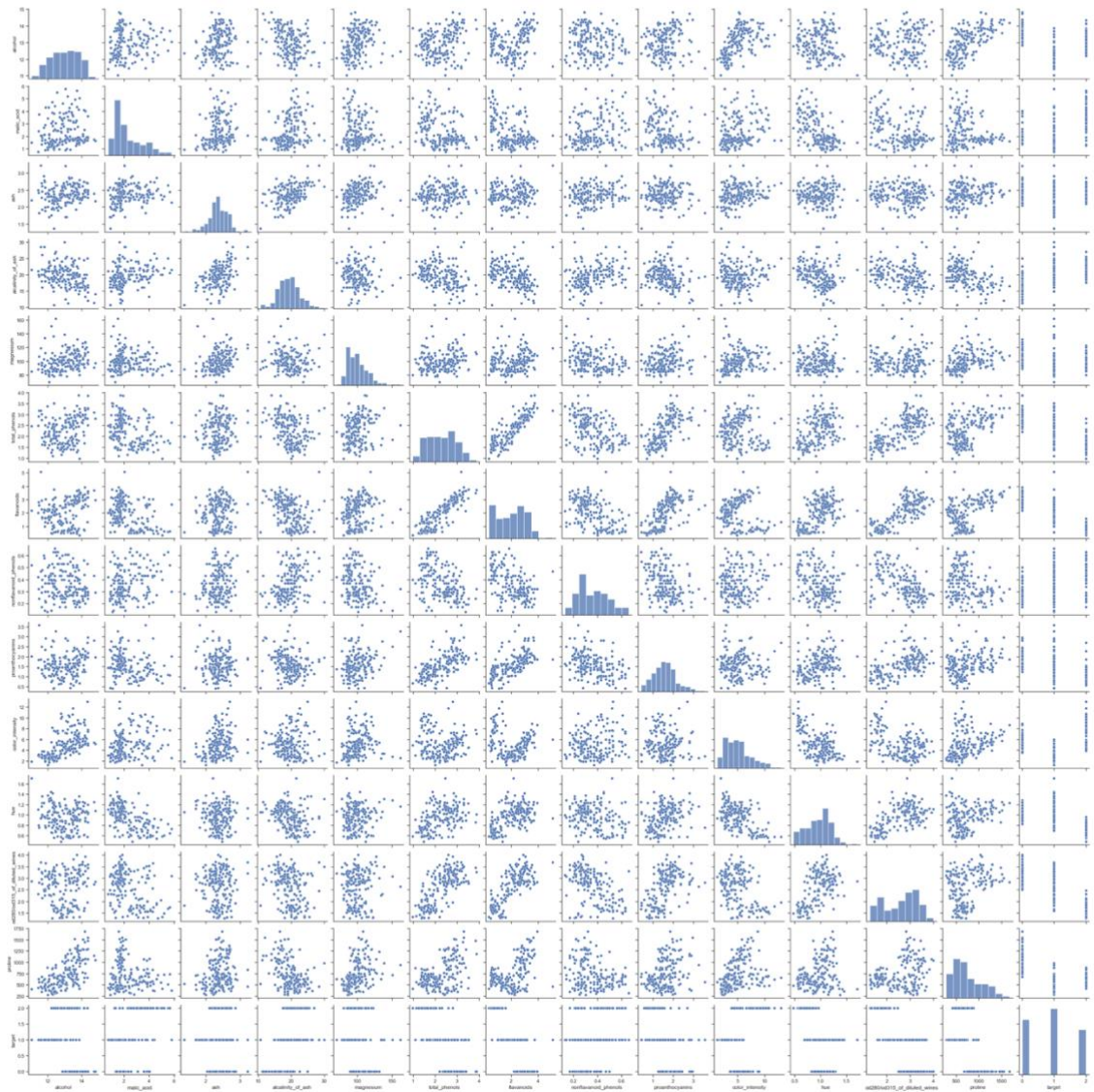
In [24]: sns.jointplot(x='total_phenols', y='flavanoids', data=datal, kind="kde")

Out[24]: <seaborn.axisgrid.JointGrid at 0x7fc9032e0>
```



```
In [25]: # Комбинация гистограмм и диаграмм рассеивания для всего набора данных.
sns.pairplot(datal)

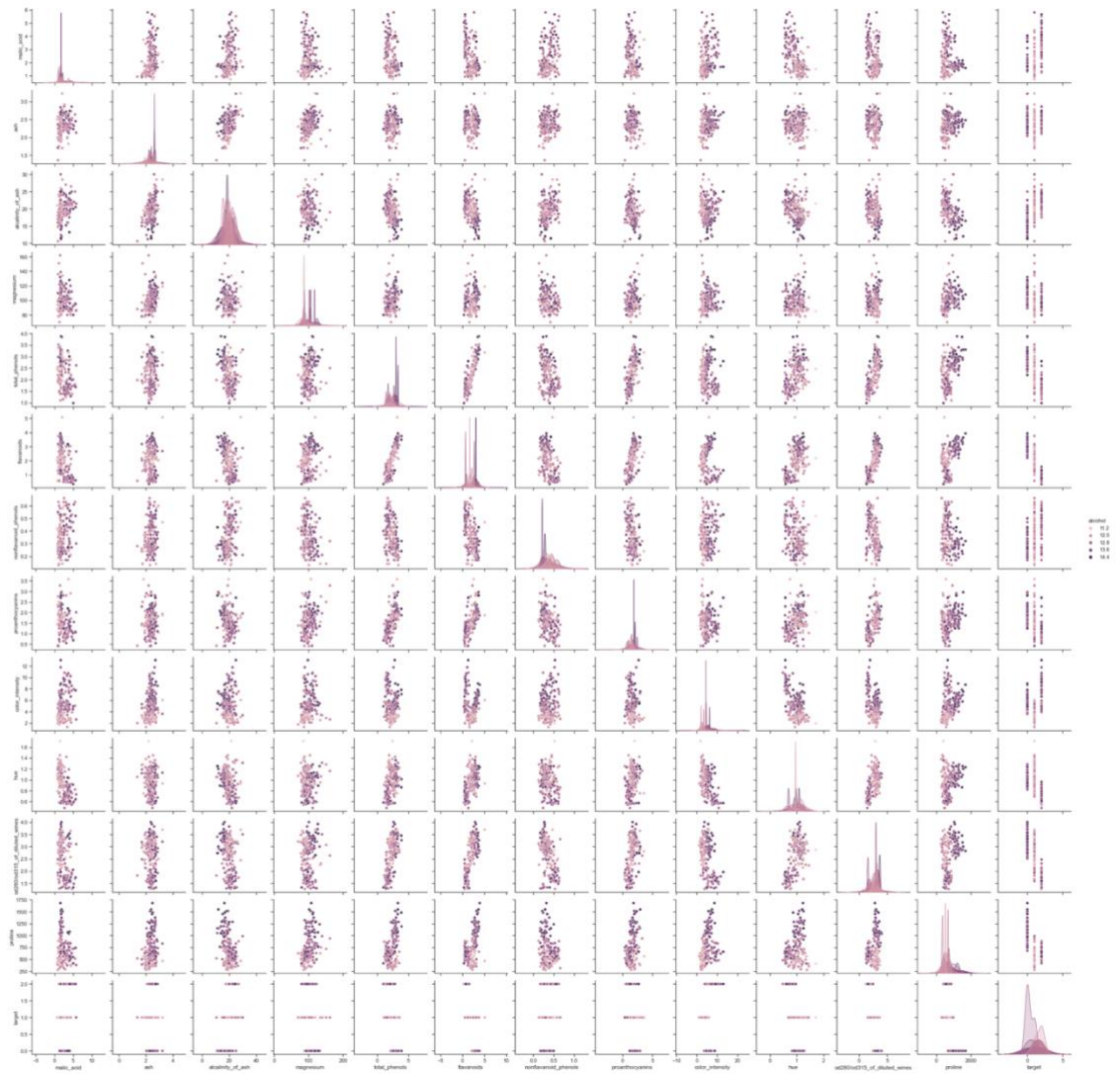
Out[25]: <seaborn.axisgrid.PairGrid at 0x7fc2fcad35e0>
```





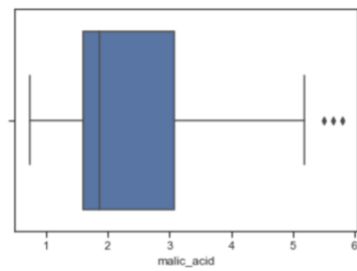
```
In [26]: # С помощью параметра "hue" возможна группировка по значениям какого-либо признака.
sns.pairplot(data1, hue="alcohol")
```

```
Out[26]: <seaborn.axisgrid.PairGrid at 0x7fc2e680afd0>
```



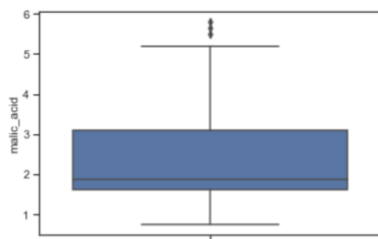
```
In [27]: # Отображает одномерное распределение вероятности - Ящик с усами
sns.boxplot(x=datal['malic_acid'])
```

Out[27]: <AxesSubplot:xlabel='malic\_acid'>



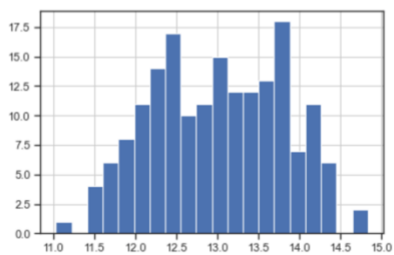
```
In [28]: sns.boxplot(y=datal['malic_acid'])
```

Out[28]: <AxesSubplot:ylabel='malic\_acid'>



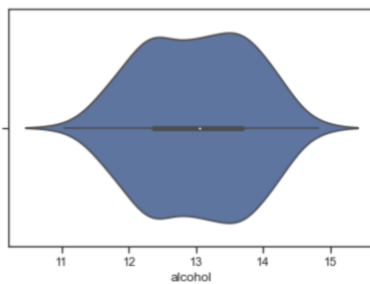
```
In [29]: # Гистограмма по крепости алкоголя
datal['alcohol'].hist(bins=20)
```

Out[29]: <AxesSubplot:>



```
In [30]: # по краям отображаются распределения плотности
sns.violinplot(x=datal['alcohol'])
```

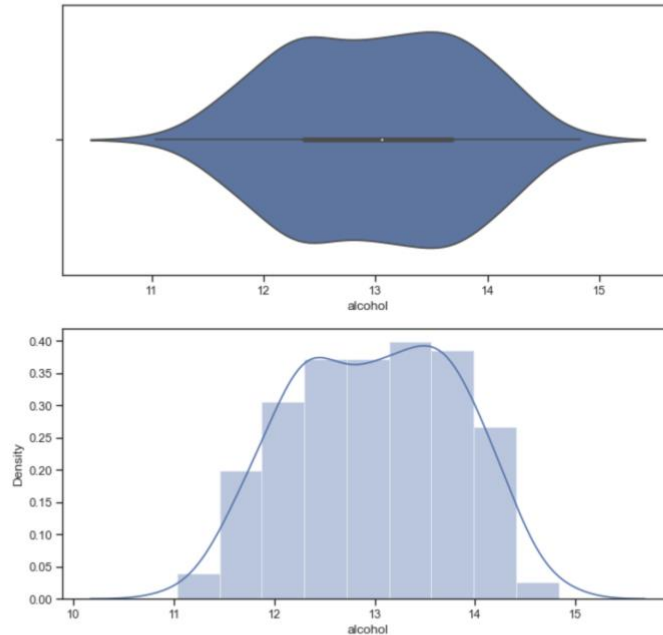
Out[30]: <AxesSubplot:xlabel='alcohol'>



```
In [31]: fig, ax = plt.subplots(2, 1, figsize=(10,10))
sns.violinplot(ax=ax[0], x=data1['alcohol'])
sns.distplot(data1['alcohol'], ax=ax[1])
```

/Users/imac/opt/anaconda3/lib/python3.9/site-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).  
warnings.warn(msg, FutureWarning)

Out [31]: <AxesSubplot:xlabel='alcohol', ylabel='Density'>



#### ИНФОРМАЦИЯ О КОРРЕЛЛЯЦИИ ПРИЗНАКОВ

Проверка корреляции признаков позволяет решить две задачи:

- 1) Понять какие признаки (колонки датасета) наиболее сильно коррелируют с целевым признаком (в нашем примере это колонка "Оссурансу"). Именно эти признаки будут наиболее информативными для моделей машинного обучения. Признаки, которые слабо коррелируют с целевым признаком, можно попробовать исключить из построения модели, и тогда это повышает качество модели. Нужно отметить, что некоторые алгоритмы машинного обучения автоматически определяют ценность того или иного признака для построения модели.
- 2) Понять какие нецелевые признаки линейно зависимы между собой. Линейно зависимые признаки, как правило, очень плохо влияют на качество моделей. Поэтому если несколько признаков линейно зависимы, то для построения модели из них выбирают какой-то один признак.

In [32]: # Информация о корреляции признаков  
data1.corr()

Out[32]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins
alcohol	1.000000	0.094397	0.211545	-0.310235	0.270798	0.289101	0.236815	-0.155929	0.136698
malic_acid	0.094397	1.000000	0.164045	0.288500	-0.054575	-0.335167	-0.411007	0.292977	-0.220746
ash	0.211545	0.164045	1.000000	0.443367	0.286587	0.128980	0.115077	0.186230	0.009652
alcalinity_of_ash	-0.310235	0.288500	0.443367	1.000000	-0.083333	-0.321113	-0.351370	0.361922	-0.197327
magnesium	0.270798	-0.054575	0.286587	-0.083333	1.000000	0.214401	0.195784	-0.256294	0.236441
total_phenols	0.289101	-0.335167	0.128980	-0.321113	0.214401	1.000000	0.864564	-0.449935	0.612413
flavanoids	0.236815	-0.411007	0.115077	-0.351370	0.195784	0.864564	1.000000	-0.537900	0.652692
nonflavanoid_phenols	-0.155929	0.292977	0.186230	0.361922	-0.256294	-0.449935	-0.537900	1.000000	-0.365845
proanthocyanins	0.136698	-0.220746	0.009652	-0.197327	0.236441	0.612413	0.652692	-0.365845	1.000000
color_intensity	0.546364	0.248985	0.258887	0.018732	0.199950	-0.055136	-0.172379	0.139057	-0.025250
hue	-0.071747	-0.561296	-0.074667	-0.273955	0.055398	0.433681	0.543479	-0.262640	0.295544
od280/od315_of_diluted_wines	0.072343	-0.368710	0.003911	-0.276769	0.066004	0.699949	0.787194	-0.503270	0.519067
proline	0.643720	-0.192011	0.223626	-0.440597	0.393351	0.498115	0.494193	-0.311385	0.330417
target	-0.328222	0.437776	-0.049643	0.517859	-0.209179	-0.719163	-0.847498	0.489109	-0.499130

In [33]: # Информация о корреляции признаков разными методами  
data1.corr(method='kendall')

Out[33]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins
alcohol	1.000000	0.093844	0.170154	-0.212978	0.250506	0.209099	0.191087	-0.109554	0.133526
malic_acid	0.093844	1.000000	0.158178	0.210119	0.050869	-0.174929	-0.211918	0.175129	-0.168714
ash	0.170154	0.158178	1.000000	0.258352	0.254246	0.089855	0.049474	0.098937	0.018240
alcalinity_of_ash	-0.212978	0.210119	0.258352	1.000000	-0.121005	-0.256669	-0.309865	0.278091	-0.171404
magnesium	0.250506	0.050869	0.254246	-0.121005	1.000000	0.172195	0.161603	-0.158361	0.117871
total_phenols	0.209099	-0.174929	0.089855	-0.256669	0.172195	1.000000	0.701999	-0.310443	0.466517
flavanoids	0.191087	-0.211918	0.049474	-0.309865	0.161603	0.701999	1.000000	-0.378099	0.534615
nonflavanoid_phenols	-0.109554	0.175129	0.098937	0.278091	-0.158361	-0.310443	-0.378099	1.000000	-0.269189
proanthocyanins	0.133526	-0.168714	0.018240	-0.171404	0.117871	0.466517	0.534615	-0.269189	1.000000
color_intensity	0.434353	0.195607	0.187786	-0.057281	0.241781	0.028264	0.028674	0.036065	-0.014962
hue	-0.021717	-0.388707	-0.037234	-0.239210	0.023760	0.289210	0.354372	-0.179755	0.231071
od280/od315_of_diluted_wines	0.061513	-0.162909	-0.006341	-0.226253	0.034307	0.478267	0.520448	-0.363787	0.369104
proline	0.449387	-0.044660	0.171574	-0.313218	0.343016	0.280203	0.263661	-0.174108	0.204172
target	-0.238984	0.247494	-0.038085	0.449402	-0.184992	-0.590404	-0.725255	0.379234	-0.450225

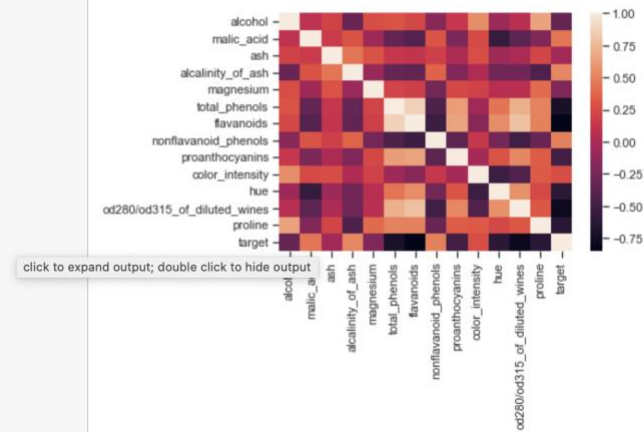
```
In [34]: data1.corr(method='spearman')
```

```
Out [34]:
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins
alcohol	1.000000	0.140430	0.243722	-0.306598	0.365503	0.310920	0.294740	-0.162207	0.192734
malic_acid	0.140430	1.000000	0.230674	0.304069	0.080188	-0.280225	-0.325202	0.255236	-0.244825
ash	0.243722	0.230674	1.000000	0.366374	0.361488	0.132193	0.078796	0.145583	0.024384
alcalinity_of_ash	-0.306598	0.304069	0.366374	1.000000	-0.169558	-0.376657	-0.443770	0.389390	-0.253695
magnesium	0.365503	0.080188	0.361488	-0.169558	1.000000	0.246417	0.233167	-0.236786	0.173647
total_phenols	0.310920	-0.280225	0.132193	-0.376657	0.246417	1.000000	0.879404	-0.448013	0.666689
flavanoids	0.294740	-0.325202	0.078796	-0.443770	0.233167	0.879404	1.000000	-0.543897	0.730322
nonflavanoid_phenols	-0.162207	0.255236	0.145583	0.389390	-0.236786	-0.448013	-0.543897	1.000000	-0.384629
proanthocyanins	0.192734	-0.244825	0.024384	-0.253695	0.173647	0.666689	0.730322	-0.384629	1.000000
color_intensity	0.635425	0.290307	0.283047	-0.073776	0.357029	0.011162	-0.042910	0.059639	-0.030947
hue	-0.024203	-0.560265	-0.050183	-0.352507	0.036095	0.439457	0.535430	-0.267813	0.342795
od280/od315_of_diluted_wines	0.103050	-0.255185	-0.007500	-0.325890	0.056963	0.687207	0.741533	-0.494950	0.554031
proline	0.633580	-0.057466	0.253163	-0.456090	0.507575	0.419470	0.429904	-0.270112	0.308249
target	-0.354167	0.346913	-0.053988	0.569792	-0.250498	-0.726544	-0.854908	0.474205	-0.570648

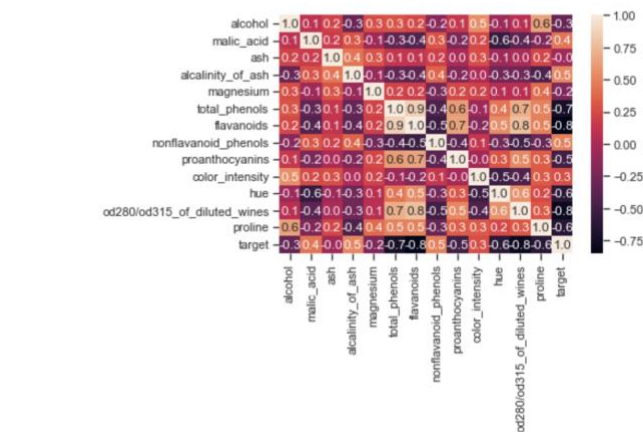
```
In [35]: #Для визуализации корреляционной матрицы будем использовать "тепловую карту",  
#которая показывает степень корреляции различными цветами.  
sns.heatmap(data1.corr())
```

```
Out [35]: <AxesSubplot:>
```



```
In [36]: # Вывод значений в ячейках  
sns.heatmap(data1.corr(), annot=True, fmt='.1f')
```

```
Out [36]: <AxesSubplot:>
```

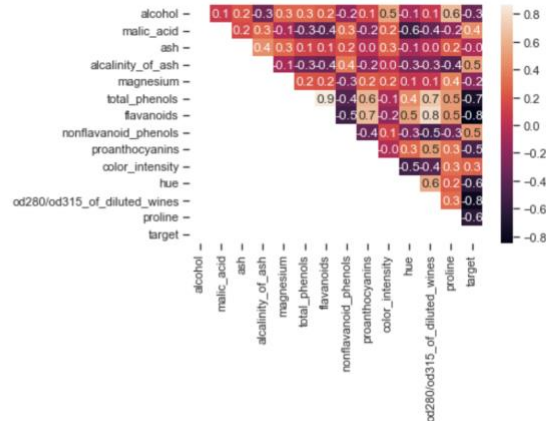




```
In [37]: # Треугольный вариант матрицы
mask = np.zeros_like(data1.corr(), dtype=np.bool)
# чтобы оставить нижнюю часть матрицы
# mask[np.triu_indices_from(mask)] = True
# чтобы оставить верхнюю часть матрицы
mask[np.tril_indices_from(mask)] = True
sns.heatmap(data1.corr(), mask=mask, annot=True, fmt='.1f')

/var/folders/7p/qf20jzcs0857b0yp3vsrlzv00000gp/T/ipykernel_13637/174623946.py:2: DeprecationWarning: `np.bool` is a
deprecated alias for the builtin `bool`. To silence this warning, use `bool` by itself. Doing this will not modify
any behavior and is safe. If you specifically wanted the numpy scalar type, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
    mask = np.zeros_like(data1.corr(), dtype=np.bool)
```

Out [37]: <AxesSubplot:>



```
In [38]: fig, ax = plt.subplots(1, 3, sharex='col', sharey='row', figsize=(15,5))
sns.heatmap(data1.corr(method='pearson'), ax=ax[0], annot=True, fmt='.1f')
sns.heatmap(data1.corr(method='kendall'), ax=ax[1], annot=True, fmt='.1f')
sns.heatmap(data1.corr(method='spearman'), ax=ax[2], annot=True, fmt='.1f')
fig.suptitle('Корреляционные матрицы, построенные различными методами')
ax[0].title.set_text('Pearson')
ax[1].title.set_text('Kendall')
ax[2].title.set_text('Spearman')
```

