Разведочный анализ данных. Исследование и визуализация данных.

1) Текстовое описание набора данных

В качестве набора данных мы будем использовать стандартный набор данных из библиотеки Scikit-learn, wine dataset.

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

In [ ]:
```
def make_dataframe(ds_function):
    ds = ds_function()
    df = pd.DataFrame(data= np.c_[ds['data'], ds['target']],
                      columns= list(ds['feature_names']) + ['target'])
    return df
```

In [ ]:
```
data = make_dataframe(load_wine)
```

2) Основные характеристики датасета

In [ ]:
```
data.head()
```

Out[ ]:

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

In [ ]:
```
# Размер датасета - 178 строк, 13 колонок
data.shape
```

Out[ ]:
```
(178, 14)
```

In [ ]:
```
total_count = data.shape[0]
print('Всего строк: {}'.format(total_count))
```

```
Всего строк: 178
```

In [ ]:
```
# Список колонок
data.columns
```

Out[ ]:
```
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 [ ]:
```
# Список колонок с типами данных
data.dtypes
```

Out[ ]:
```
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 [ ]:
```
# 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))
```

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

Out[ ]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	col
count	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	
std	0.811827	1.117146	0.274344	3.339564	14.282484	0.625851	0.998859	0.124453	0.572359	
min	11.030000	0.740000	1.360000	10.600000	70.000000	0.980000	0.340000	0.130000	0.410000	
25%	12.362500	1.602500	2.210000	17.200000	88.000000	1.742500	1.205000	0.270000	1.250000	
50%	13.050000	1.865000	2.360000	19.500000	98.000000	2.355000	2.135000	0.340000	1.555000	
75%	13.677500	3.082500	2.557500	21.500000	107.000000	2.800000	2.875000	0.437500	1.950000	
max	14.830000	5.800000	3.230000	30.000000	162.000000	3.880000	5.080000	0.660000	3.580000	

In [ ]:
```
# Определим уникальные значения для целевого признака
data['target'].unique()
```

Out[ ]:
```
array([0., 1., 2.])
```

3) Визуальное исследование датасета

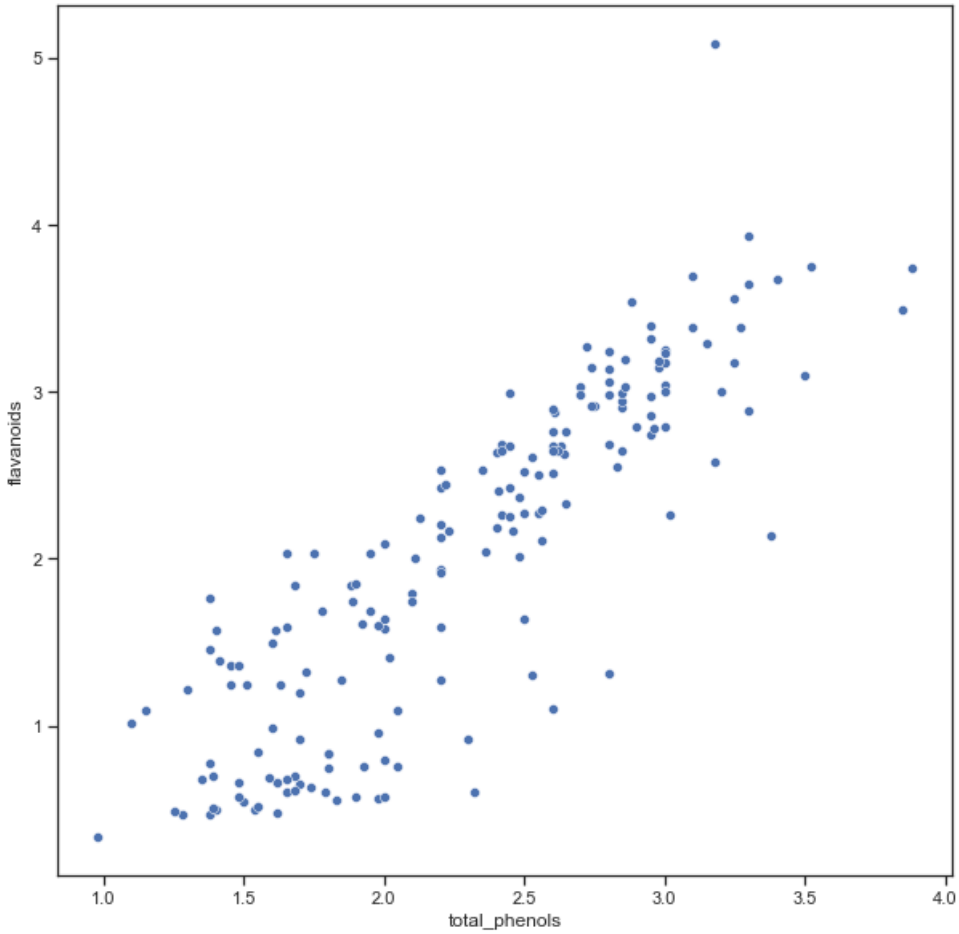
Для визуального исследования могут быть использованы различные виды диаграмм, мы построим только некоторые варианты диаграмм, которые используются достаточно часто.

Диаграмма рассеяния

Позволяет построить распределение двух колонок данных и визуально обнаружить наличие зависимости. Не предполагается, что значения упорядочены (например, по времени).

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

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



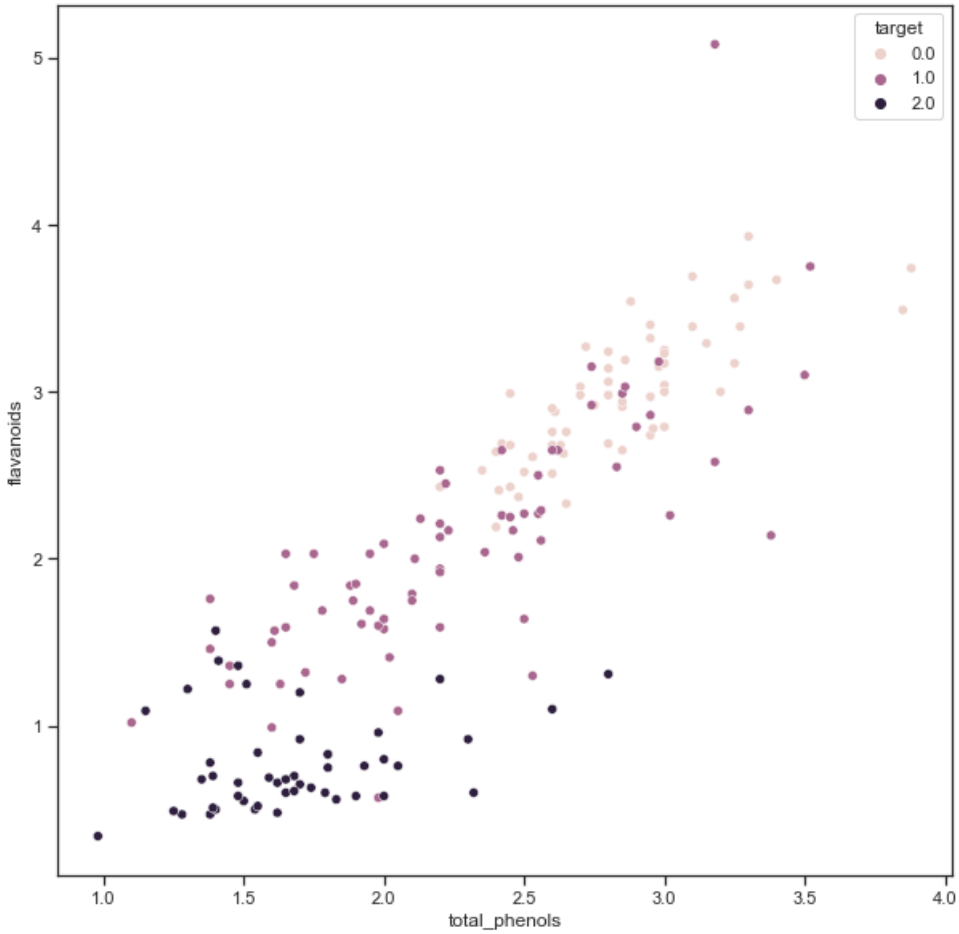
In [ ]:
```
data.head()
```

Out[ ]:

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

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

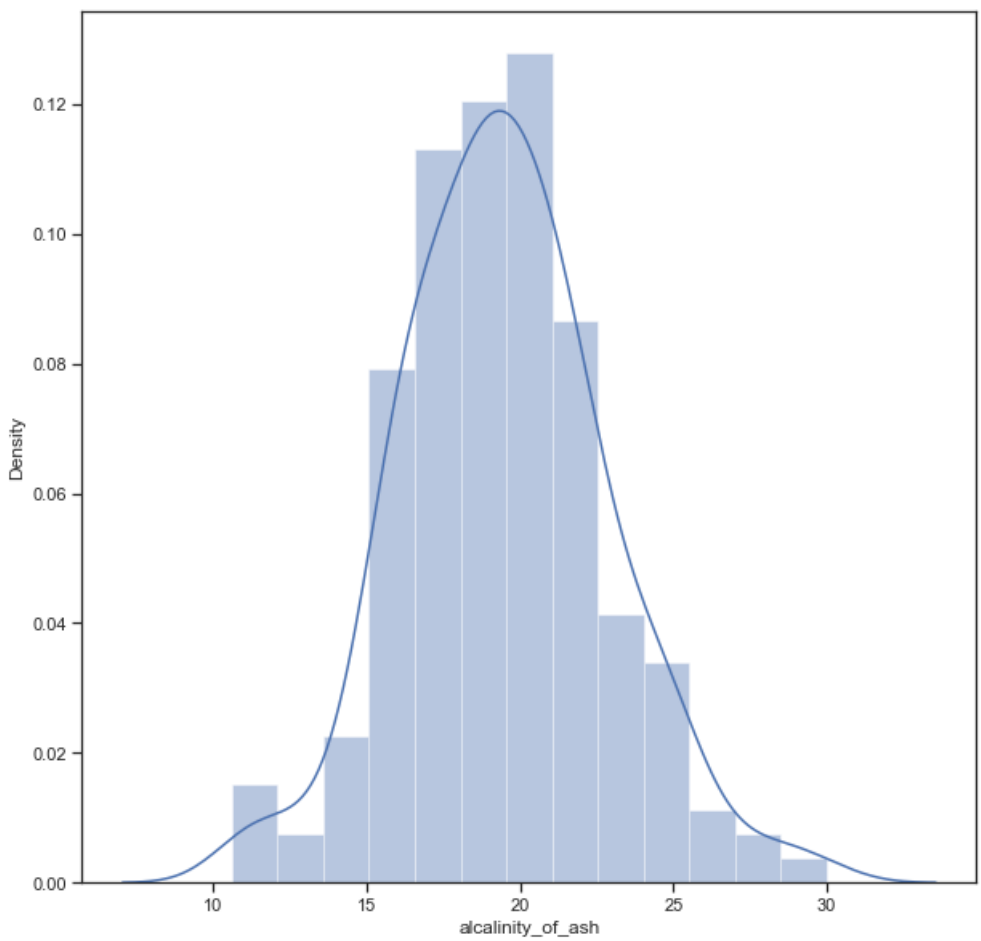
Out[ ]:<AxesSubplot:xlabel='total_phenols', ylabel='flavanoids'>



In [ ]:
```
fig, ax = plt.subplots(figsize=(10,10))
sns.distplot(data['alcalinity_of_ash'])
```

```
c:\Users\danib\Documents\pyton\myvenv\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplo
t` is a deprecated function and will be removed in a future version. Please adapt your code to use either `dis
plot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms)
.
  warnings.warn(msg, FutureWarning)
```

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

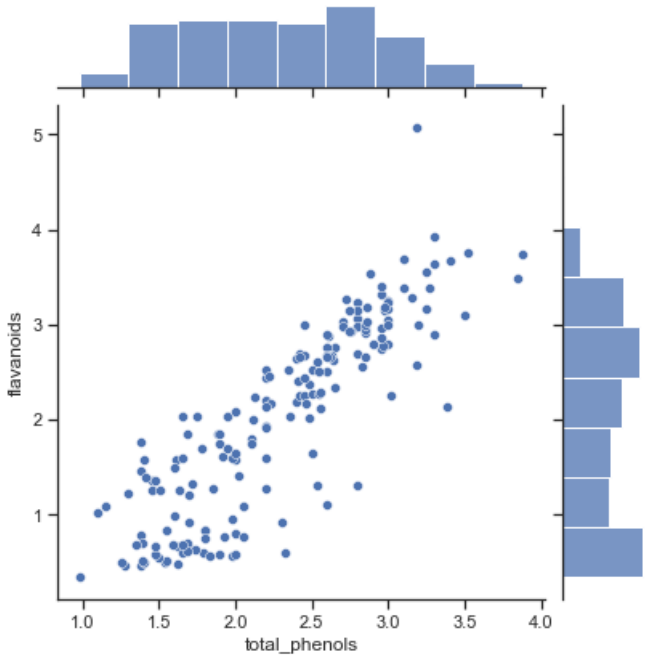


Jointplot

Комбинация гистограмм и диаграмм рассеивания.

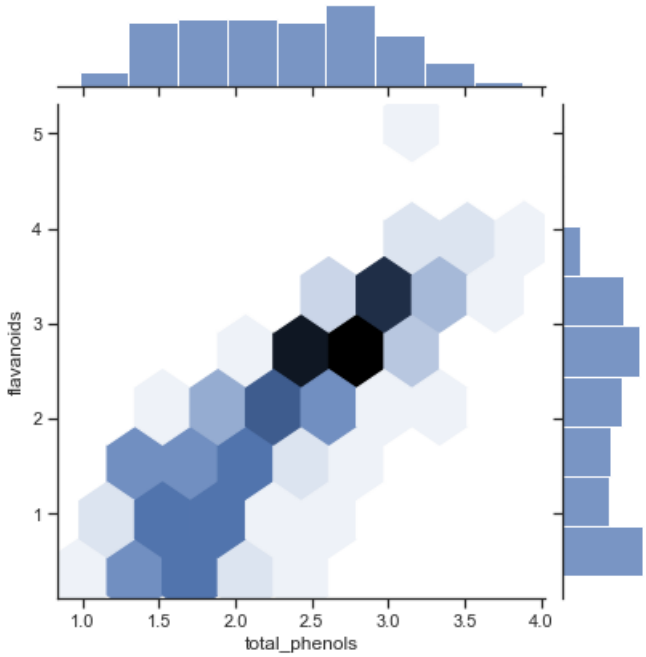
In [ ]:
```
sns.jointplot(ax=ax, x='total_phenols', y='flavanoids',data=data)
```

Out[ ]:<seaborn.axisgrid.JointGrid at 0x18b0454fc40>



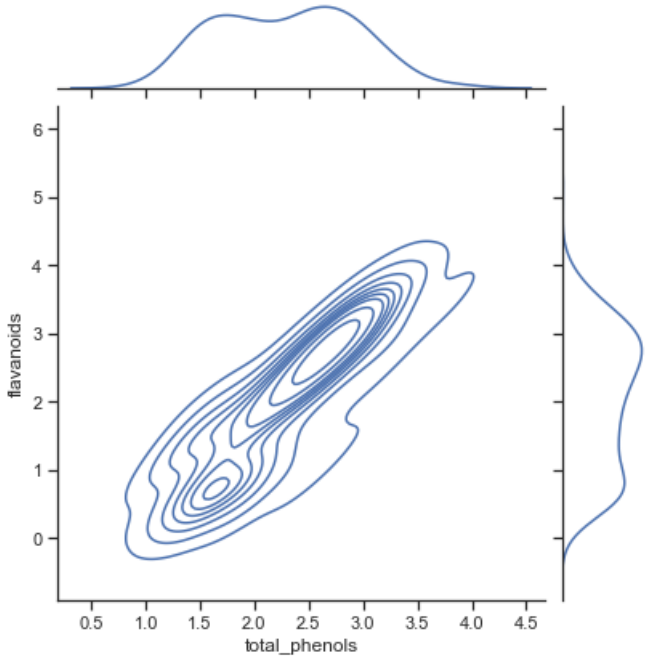
In [ ]:
```
sns.jointplot(x='total_phenols', y='flavanoids', data=data, kind="hex")
```

Out[ ]:<seaborn.axisgrid.JointGrid at 0x18b04cfff70>



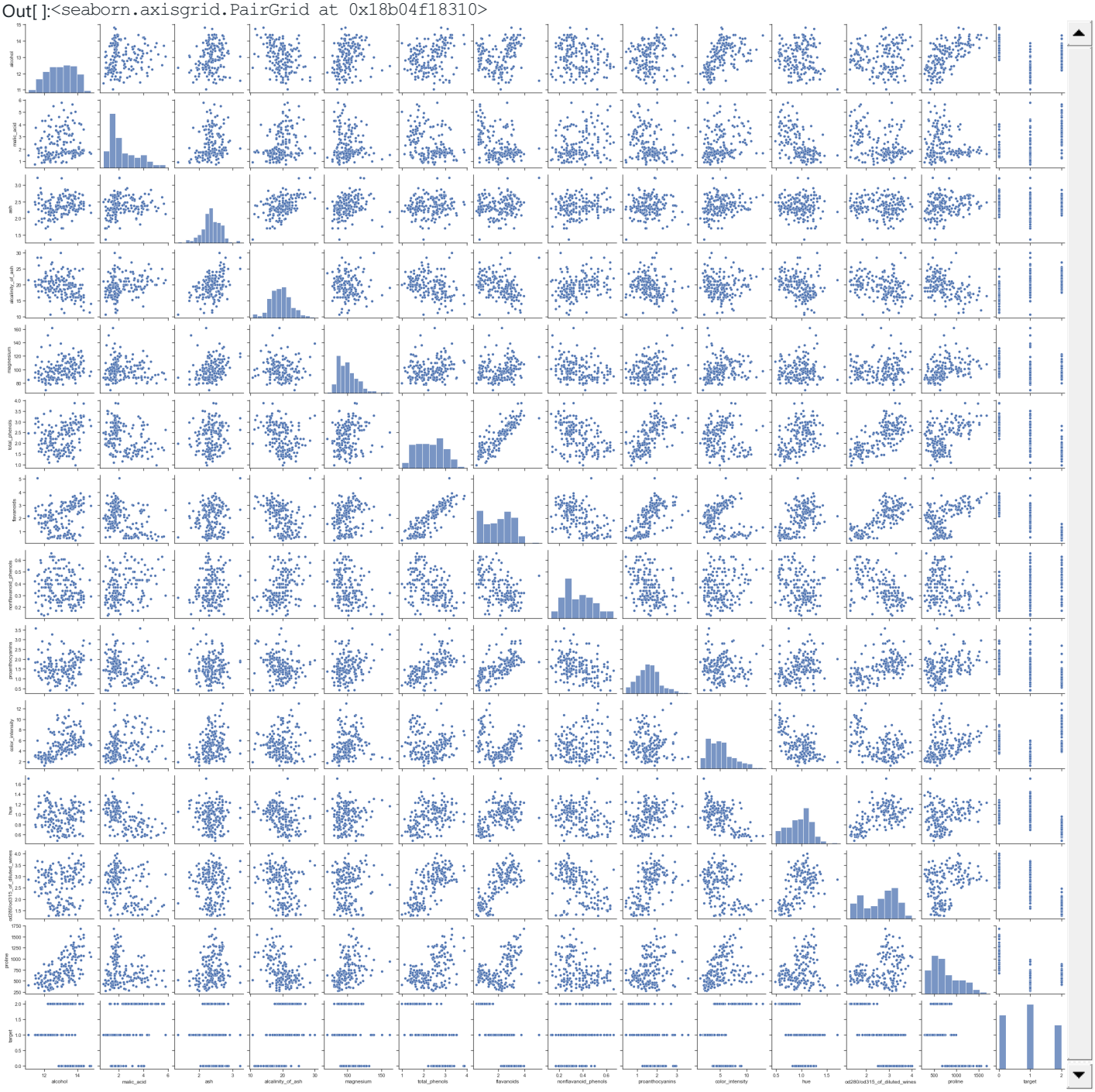
In [ ]:
```
sns.jointplot(x='total_phenols', y='flavanoids', data=data, kind="kde")
```

Out[ ]:<seaborn.axisgrid.JointGrid at 0x18b04498a90>



In [ ]:
```
sns.pairplot(data)
```

Out[ ]:<seaborn.axisgrid.PairGrid at 0x18b04f18310>



In [ ]: `sns.pairplot(data, hue="target")`

Out[ ]:<seaborn.axisgrid.PairGrid at 0x18b83b33400>

Ящик с усами

Отображает одномерное распределение вероятности.

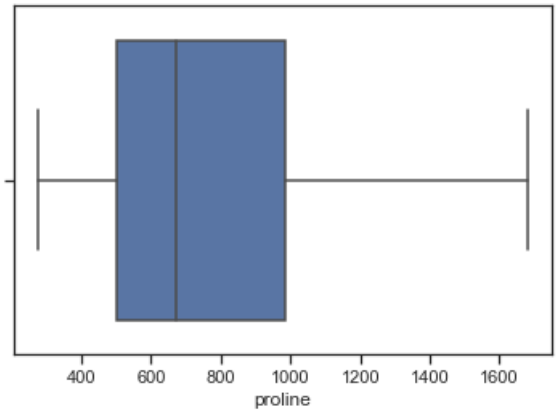
In [ ]: `data.head()`

Out[ ]:

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

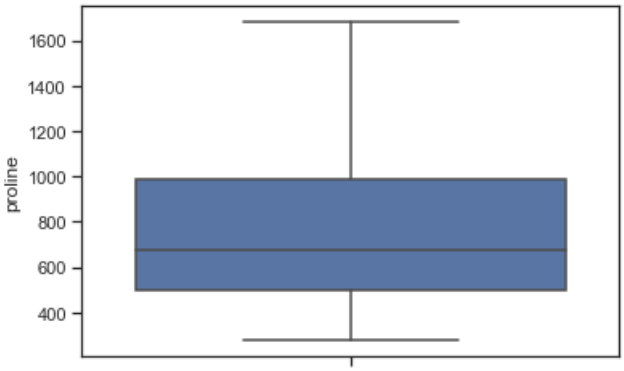
In [ ]: `sns.boxplot(x=data['proline'])`

Out[ ]:<AxesSubplot:xlabel='proline'>



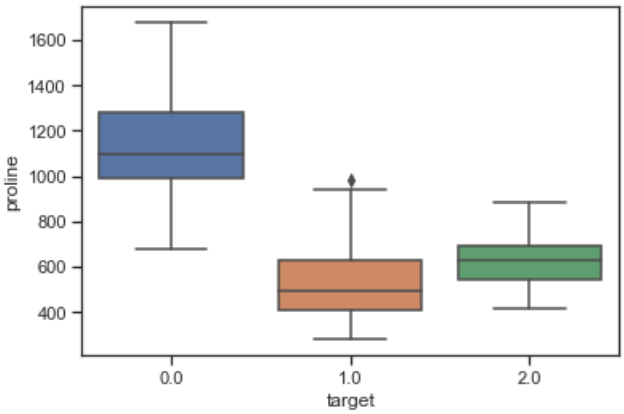
In [ ]:
```
# По вертикали
sns.boxplot(y=data['proline'])
```

Out[ ]:<AxesSubplot:ylabel='proline'>



In [ ]:
```
# Распределение параметра Humidity сгруппированные по Occupancy.
sns.boxplot(x='target', y='proline', data=data)
```

Out[ ]:<AxesSubplot:xlabel='target', ylabel='proline'>



Violin plot

Похоже на предыдущую диаграмму, но по краям отображаются распределения плотности - https://en.wikipedia.org/wiki/Kernel_density_estimation

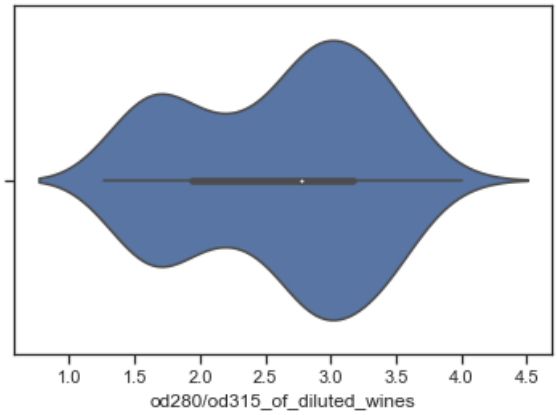
In [ ]:
```
data.head()
```

Out[ ]:

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

In [ ]:
```
sns.violinplot(x=data['od280/od315_of_diluted_wines'])
```

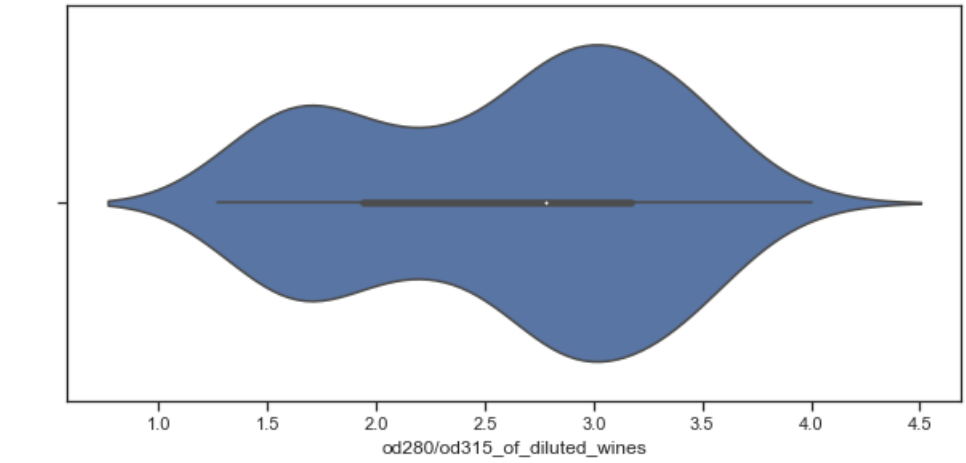
Out[ ]:<AxesSubplot:xlabel='od280/od315_of_diluted_wines'>

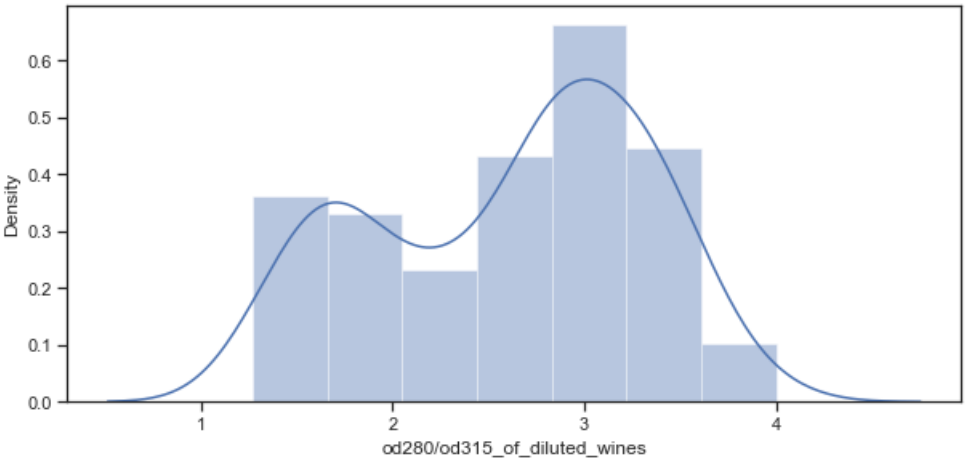


In [ ]:
```
fig, ax = plt.subplots(2, 1, figsize=(10,10))
sns.violinplot(ax=ax[0], x=data['od280/od315_of_diluted_wines'])
sns.distplot(data['od280/od315_of_diluted_wines'], ax=ax[1])
```

```
c:\Users\danib\Documents\pyton\myvenv\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplo
t` is a deprecated function and will be removed in a future version. Please adapt your code to use either `dis
plot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms)
.
  warnings.warn(msg, FutureWarning)
```

Out[ ]:<AxesSubplot:xlabel='od280/od315_of_diluted_wines', ylabel='Density'>

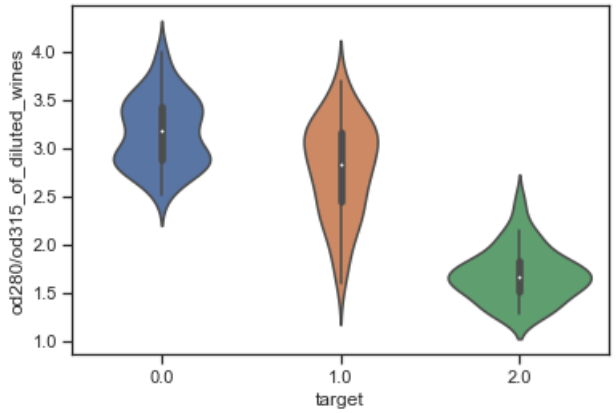




Из приведенных графиков видно, что violinplot действительно показывает распределение плотности.

In [ ]:
```
# Распределение параметра od280/od315_of_diluted_wines сгруппированные по target.
sns.violinplot(x='target', y='od280/od315_of_diluted_wines', data=data)
```

Out[ ]:<AxesSubplot:xlabel='target', ylabel='od280/od315_of_diluted_wines'>



4) Информация о корреляции 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 "Occupancy"). Именно эти признаки будут наиболее информативными для моделей машинного обучения. Признаки, которые слабо коррелируют с целевым признаком, можно попробовать исключить из построения модели, иногда 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 [ ]: `data.corr()`

Out[ ]:

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

Корреляционная матрица содержит коэффициенты корреляции между всеми парами признаков.

Корреляционная матрица симметрична относительно главной диагонали. На главной диагонали расположены единицы (корреляция признака самого с собой).

На основе корреляционной матрицы можно сделать следующие выводы:

- Целевой признак наиболее сильно коррелирует с содержанием флавонидов (0.85), содержанием белка (0.79) и общее количество фенолов (0.72). Эти признаки обязательно следует оставить в модели.
- Целевой признак отчасти коррелирует с кислотностью осадка (0.52), оттенком вина (0.62), содержанием пролина (0.63) и с содержаниемпроантоцианов (0.48), содержанием нефлаваноидных фенолов (0, 49). Эти признаки стоит также оставить в модели.
- Целевой признак слабо коррелирует с содержанием магнезии (0.21), количеством осадка (0,05) и интенсивностью цвета. Скорее всего эти признаки стоит исключить из модели, возможно они только ухудшат качество модели.
- Содержание флавоноидов и количество фенолов очень сильно коррелируют между собой (0.86). Поэтому из этих признаков в модели можно оставлять только один. -Оставить лучше содержание флавонидов, т.к. эта фича лучше коррелирует с целевым признаком

Описание метода corr - https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.corr.html

По умолчанию при построении матрицы используется коэффициент корреляции Пирсона. Возможно также построить корреляционную матрицу на основе коэффициентов корреляции Кендалла и Спирмена. На практике три метода редко дают значимые различия.

In [ ]:
```
data.corr(method='pearson')
```

Out[ ]:

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

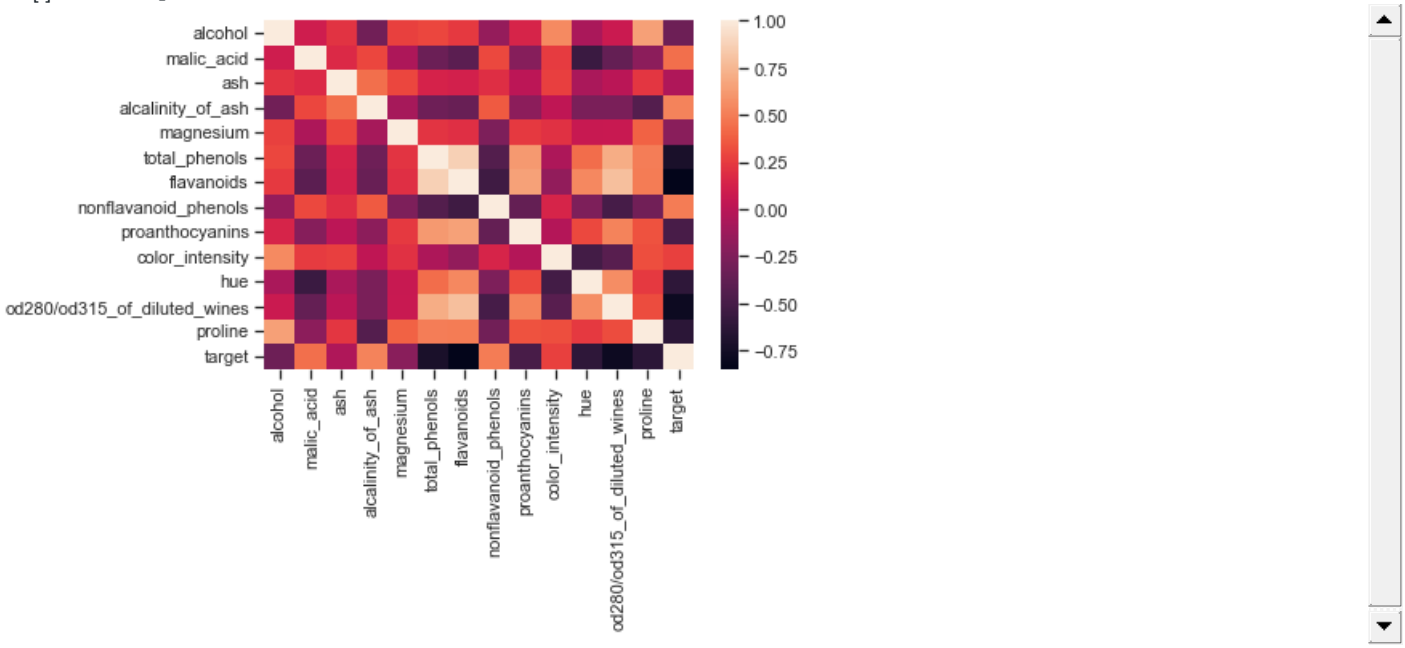
In [ ]:
```
data.corr(method='kendall')
```

Out[ ]:

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

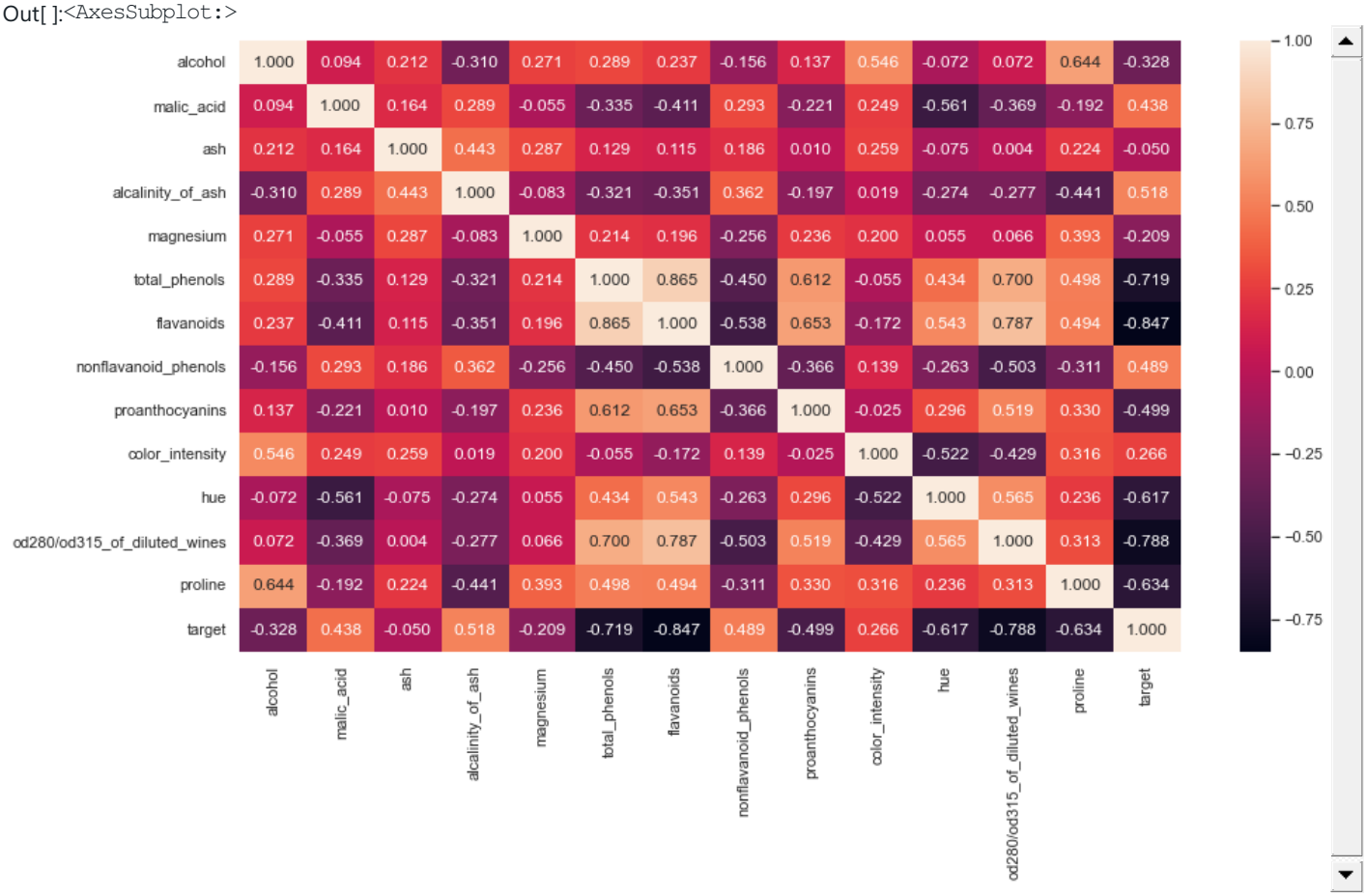
In [ ]:
```
sns.heatmap(data.corr())
```

Out[ ]:<AxesSubplot:>



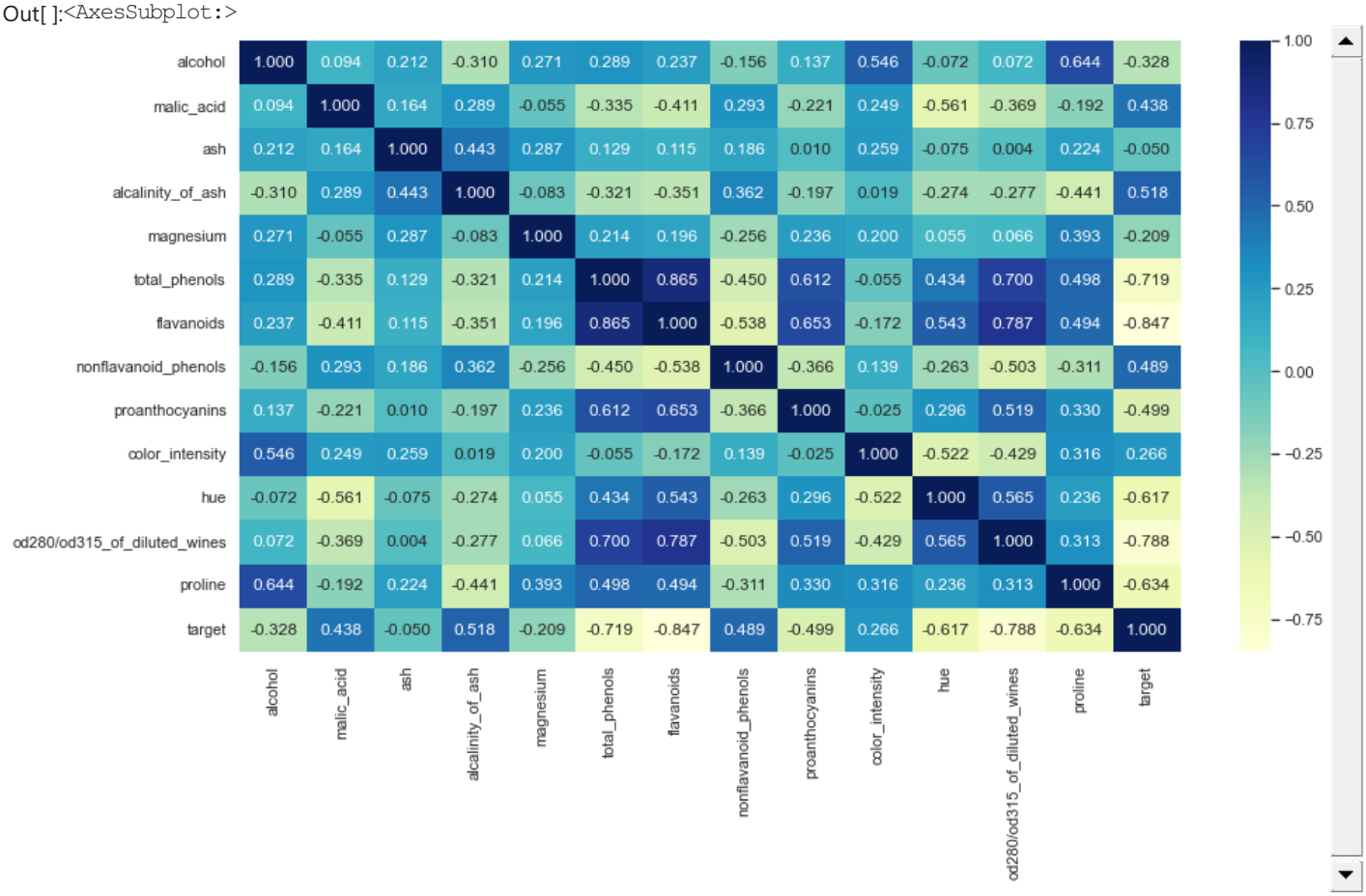
In [ ]:
```
# Вывод значений в ячейках
sns.set(rc = {'figure.figsize':(15,8)})
sns.heatmap(data.corr(), annot=True, fmt='.3f')
```

Out[ ]:<AxesSubplot:>



In [ ]:
```
# Изменение цветовой гаммы
sns.heatmap(data.corr(), cmap='YlGnBu', annot=True, fmt='.3f')
```

Out[ ]:<AxesSubplot:>

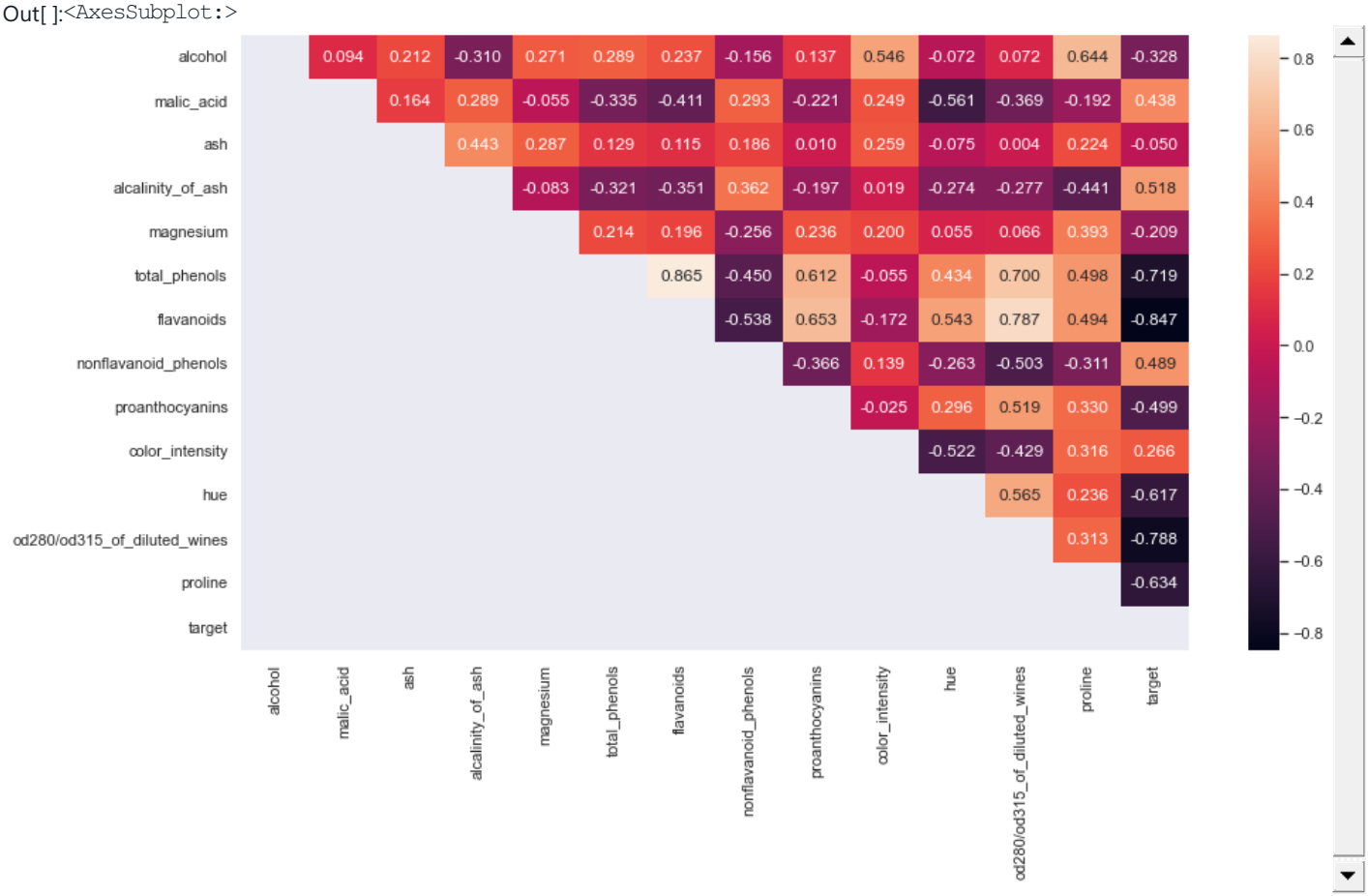


In [ ]:
```
# Треугольный вариант матрицы
mask = np.zeros_like(data.corr(), dtype=np.bool)
# чтобы оставить нижнюю часть матрицы
# mask[np.triu_indices_from(mask)] = True
# чтобы оставить верхнюю часть матрицы
```

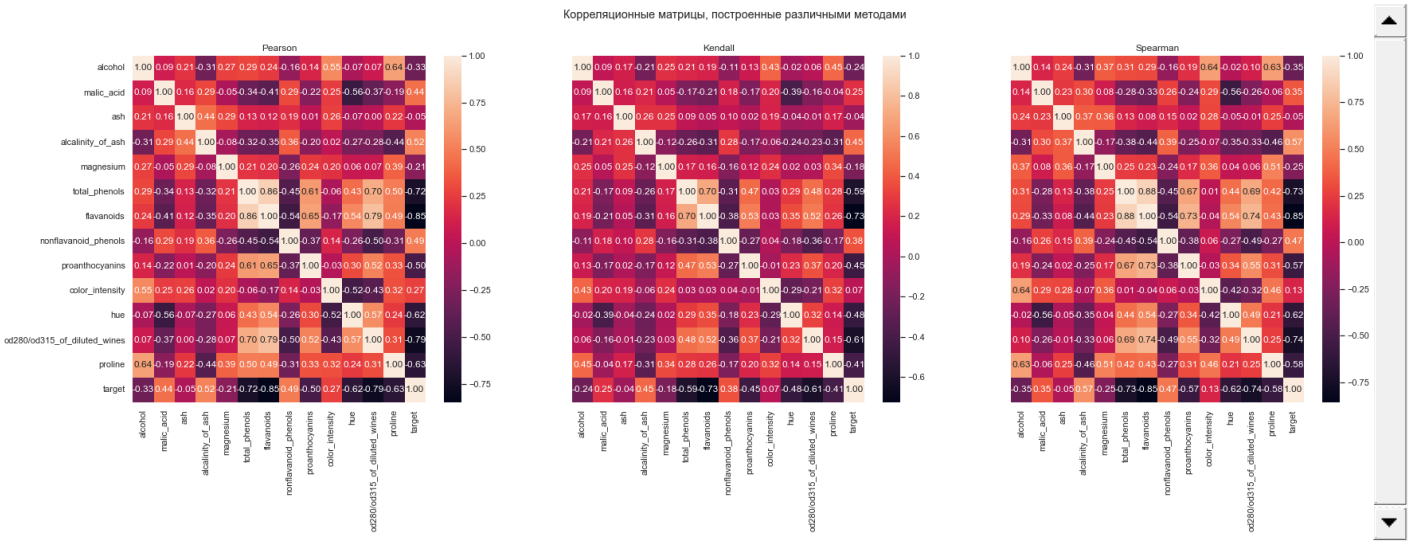
```
    mask[np.tril_indices_from(mask)] = True
    sns.heatmap(data.corr(), mask=mask, annot=True, fmt='.3f')
```

```
C:\Users\danib\AppData\Local\Temp/ipykernel_14872/3444845879.py:2: DeprecationWarning: `np.bool` is a deprecat
ed 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#d
eprecations
  mask = np.zeros_like(data.corr(), dtype=np.bool)
```

Out[ ]:<AxesSubplot:>



In [ ]:
```
fig, ax = plt.subplots(1, 3, sharex='col', sharey='row', figsize=(28,8))
sns.heatmap(data.corr(method='pearson'), ax=ax[0], annot=True, fmt='.2f')
sns.heatmap(data.corr(method='kendall'), ax=ax[1], annot=True, fmt='.2f')
sns.heatmap(data.corr(method='spearman'), ax=ax[2], annot=True, fmt='.2f')
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')
```



Необходимо отметить, что тепловая карта не очень хорошо подходит для определения корреляции нецелевых признаков между собой.

В примере тепловая карта помогает определить значимую корреляцию между признаками Humidity и HumidityRatio, следовательно только один из этих признаков можно включать в модель.

Но в реальной модели могут быть сотни признаков и коррелирующие признаки могут образовывать группы, состояшие более чем из двух признаков. Увидеть такие группы с помощью тепловой карты сложно.

Для решения задачи предлагается новый вариант визуализации - "Солнечная корреляционная карта" Solar correlation map.

К сожалению, данная библиотека пока работает только через файловый интерфейс и не предназначена для встраивания в ноутбук.

Примеры статей с описанием работы библиотеки:

- https://www.oreilly.com/learning/a-new-visualization-to-beautifully-explore-correlations
- https://www.mtab.com/the-puzzle-of-visualizing-correlations/