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 ИУ5 «Системы обработки информации и управления»

Курс «Технологии машинного обучения»

Отчет по лабораторной работе №1

«Разведочный анализ данных. Исследование и визуализация данных»

Выполнила:

студент группы ИУ5-61Б

Павловская А.А.

21.04.2021

Проверил:

преподаватель каф. ИУ5

Гапанюк Ю.Е.

Москва, 2021 г.

Цель лабораторной работы: изучение различных методов визуализация данных.

Задание:

- Выбрать набор данных (датасет). Вы можете найти список свободно распространяемых датасетов здесь.
- Для первой лабораторной работы рекомендуется использовать датасет без пропусков в данных, например из Scikit-learn.
- Пример преобразования датасетов Scikit-learn в Pandas Dataframe можно посмотреть здесь.

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.
4. Информация о корреляции признаков.

- Сформировать отчет и разместить его в своем репозитории на github.

Набор данных: Wine recognition dataset

Текст программы и экранные формы с примерами выполнения программы (ячейки ноутбука):

ИУ5-61Б Павловская А.А. Лаб1 ТМО

In [1]:

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

In [2]:

```
#data = pd.read_csv('archive/heart.csv')
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

data = make_dataframe(load_wine)
```

In [3]:

```
data.head()
```

Out[3]:

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

In [4]:

```
data.shape
```

Out[4]:

```
(178, 14)
```

In [14]:

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

```
Всего строк: 178
```

In [15]:

```
# Список колонок
data.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 колонок с типами данных
data.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 наличия 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 [18]:

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

Out[18]:

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

In [19]:

```
# Определение уникальных значения для целевого признака
data['target'].unique()
```

Out[19]:

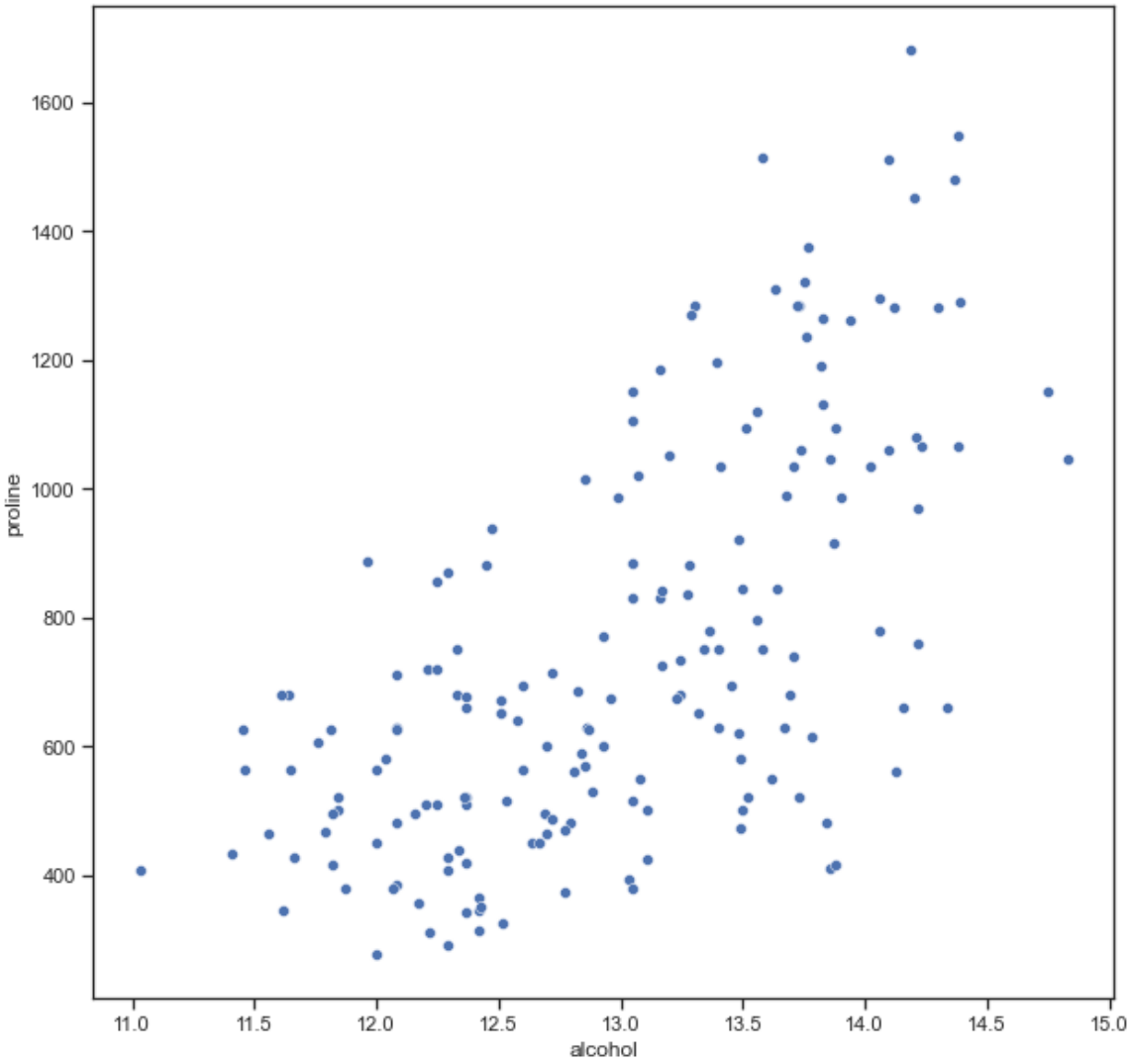
```
array([0., 1., 2.])
```

In [34]:

```
# Диаграмма рассеивания
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='alcohol', y='proline', data=data)
```

Out[34]:

```
<AxesSubplot:xlabel='alcohol', ylabel='proline'>
```

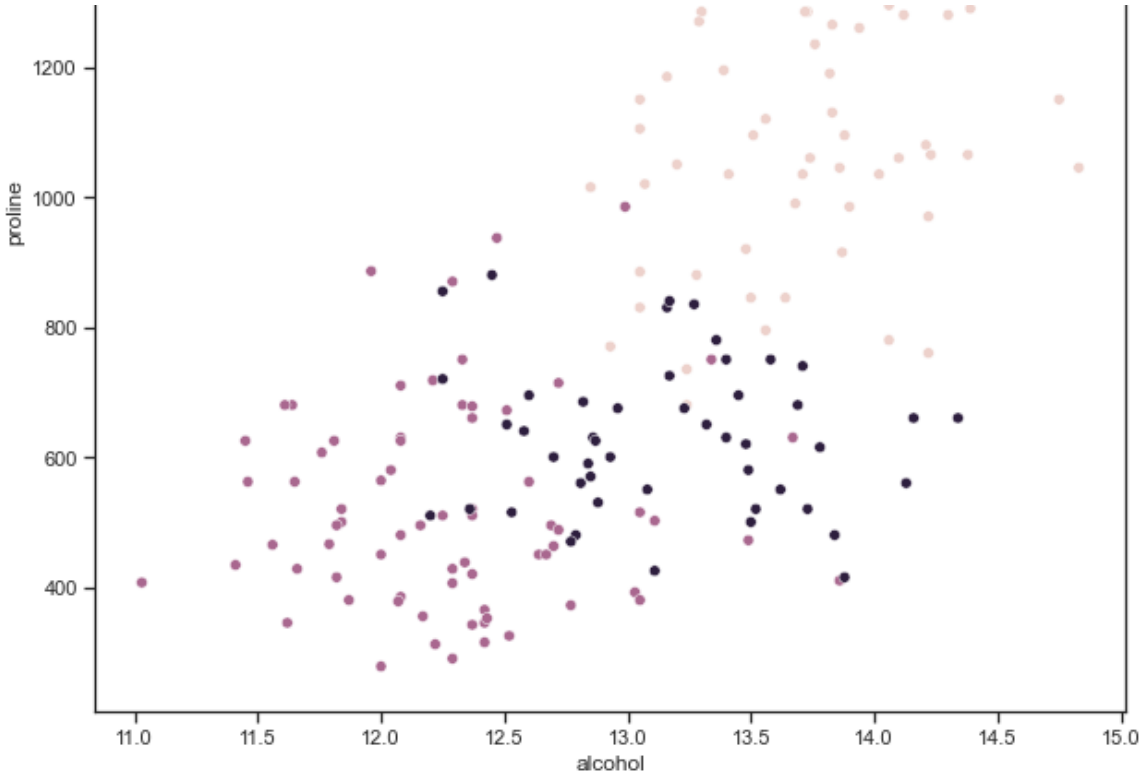


In [35]:

```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='alcohol', y='proline', data=data, hue = 'target')
```

Out[35]:

```
<AxesSubplot:xlabel='alcohol', ylabel='proline'>
```



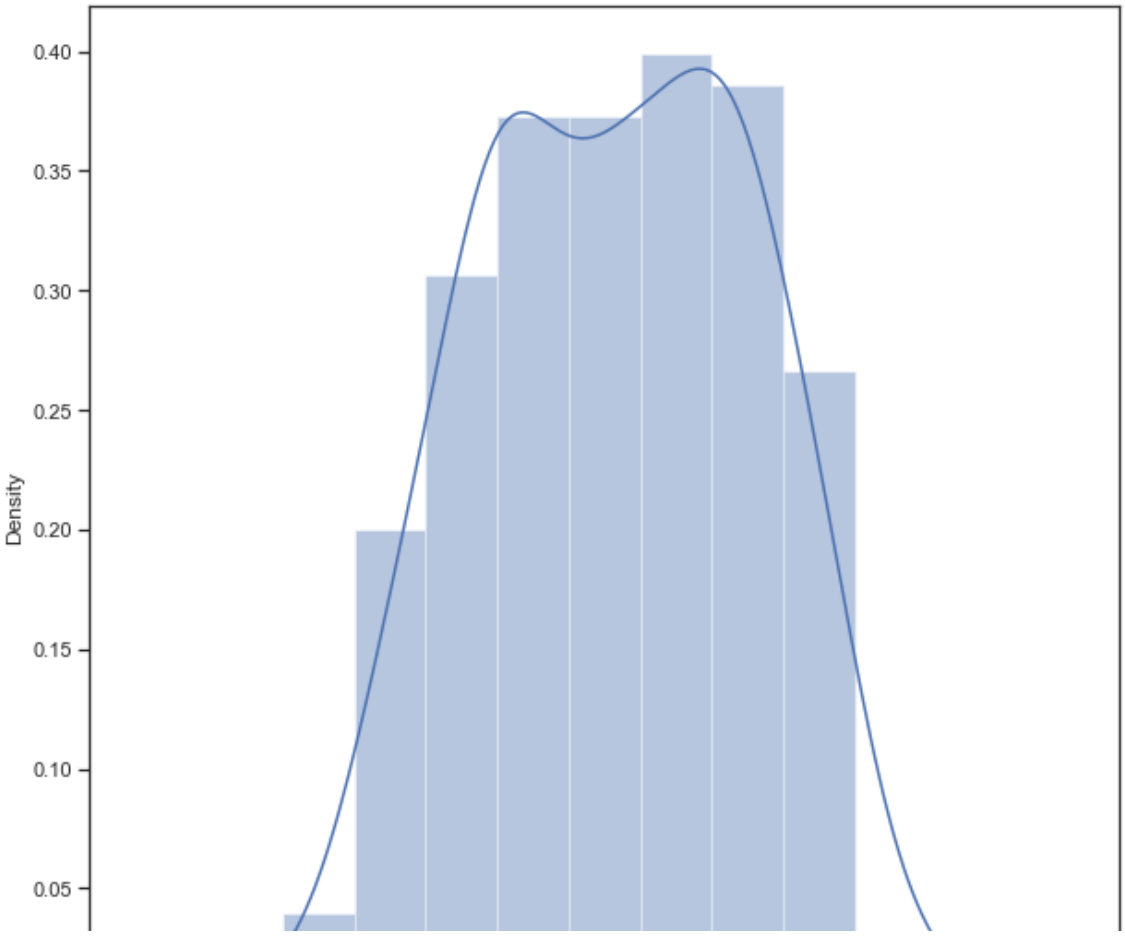
In [22]:

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

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: 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[22]:

```
<AxesSubplot:xlabel='alcohol', ylabel='Density'>
```

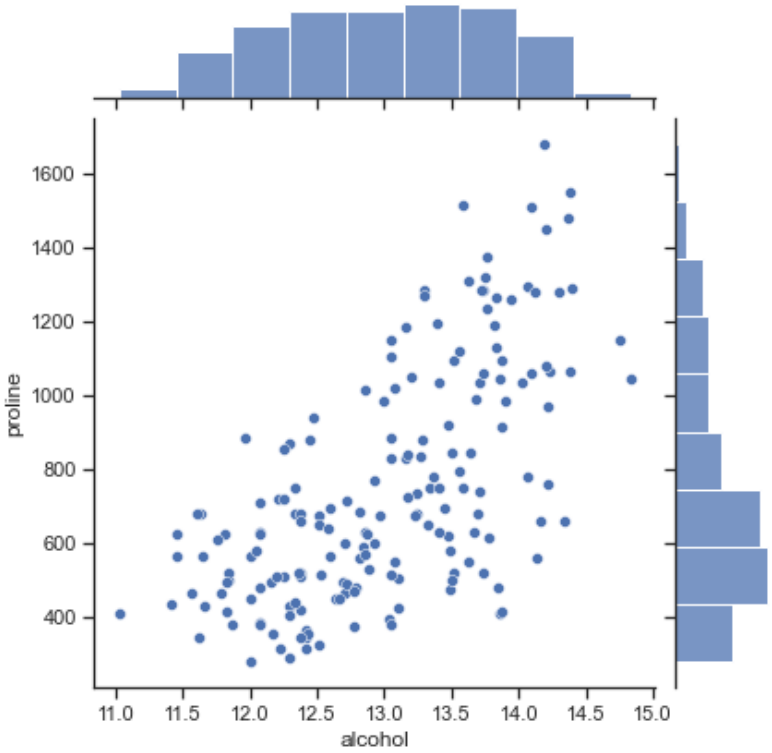


In [36]:

```
# Jointplot
sns.jointplot( x='alcohol', y='proline', data=data)
```

Out[36]:

```
<seaborn.axisgrid.JointGrid at 0x14f75a07400>
```

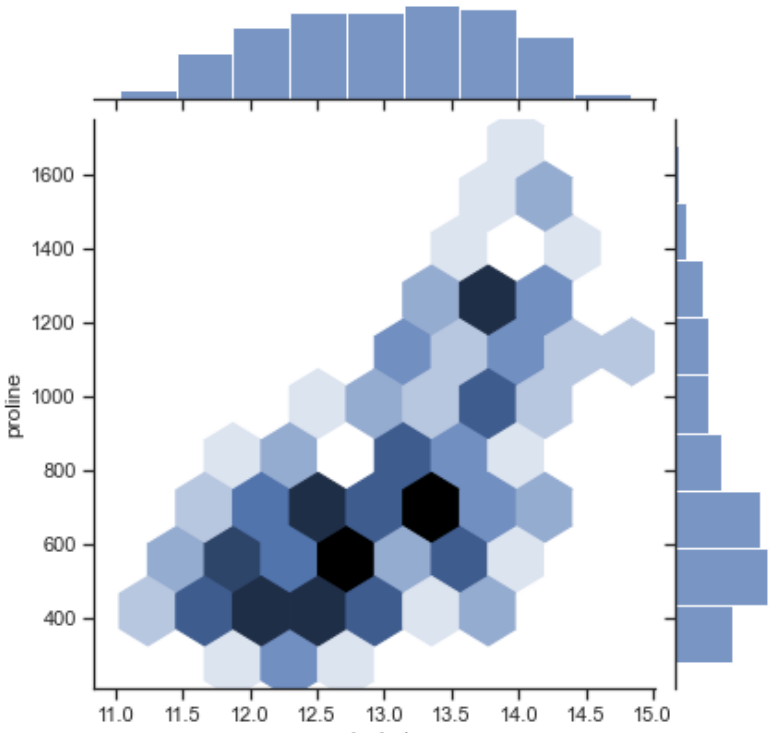


In [37]:

```
sns.jointplot( x='alcohol', y='proline', data=data, kind ="hex")
```

Out[37]:

```
<seaborn.axisgrid.JointGrid at 0x14f772890d0>
```



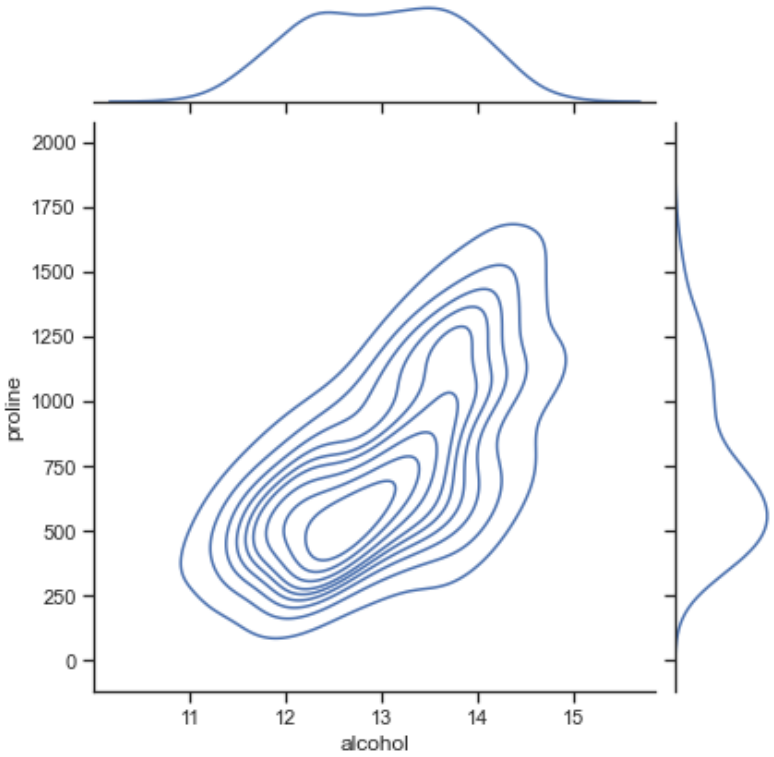
alcohol

In [38]:

```
sns.jointplot(x='alcohol', y='proline', data=data, kind ="kde")
```

Out[38]:

```
<seaborn.axisgrid.JointGrid at 0x14f773aabb0>
```

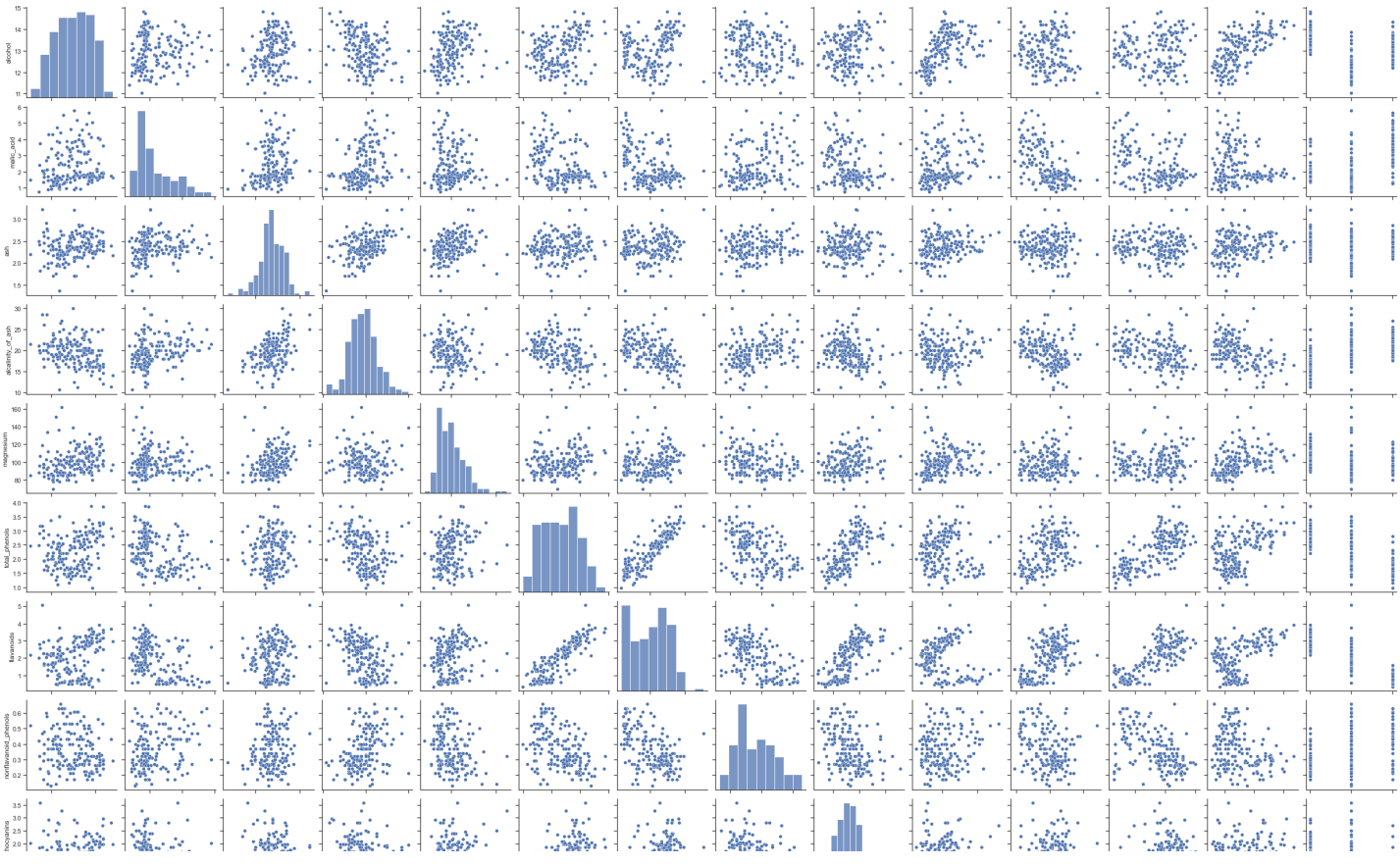


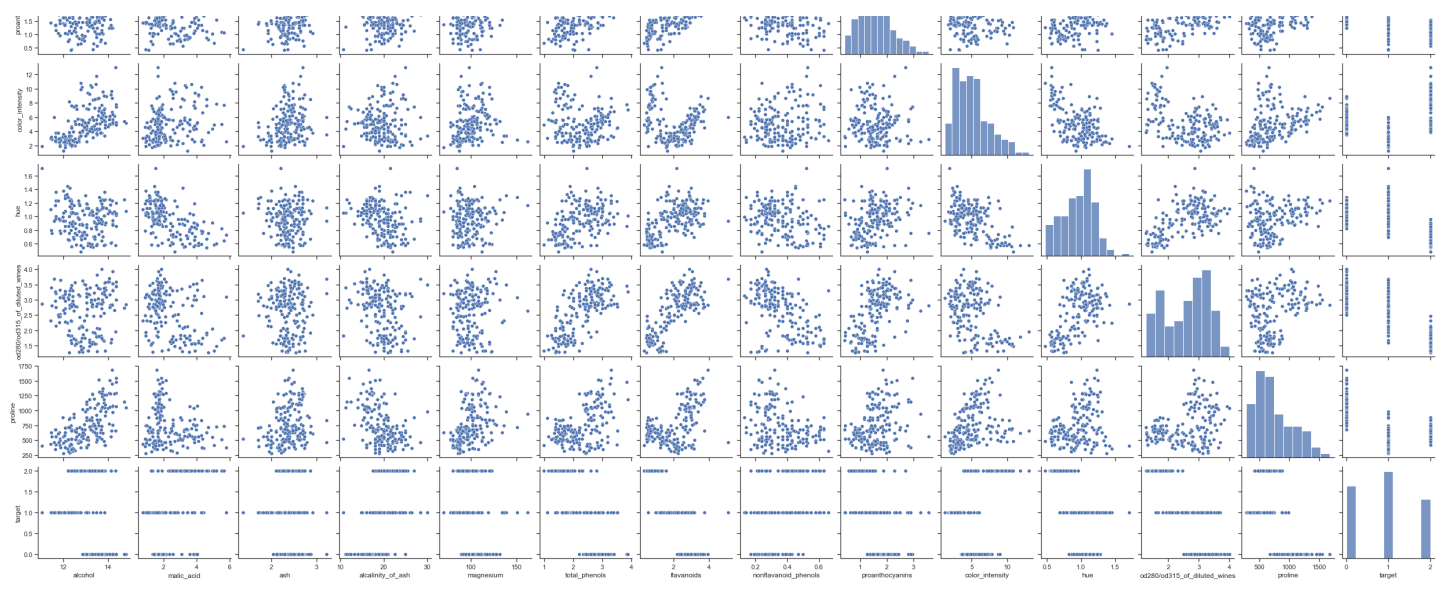
In [33]:

```
# Парные диаграммы
sns.pairplot(data)
```

Out[33]:

```
<seaborn.axisgrid.PairGrid at 0x14f6d748cd0>
```



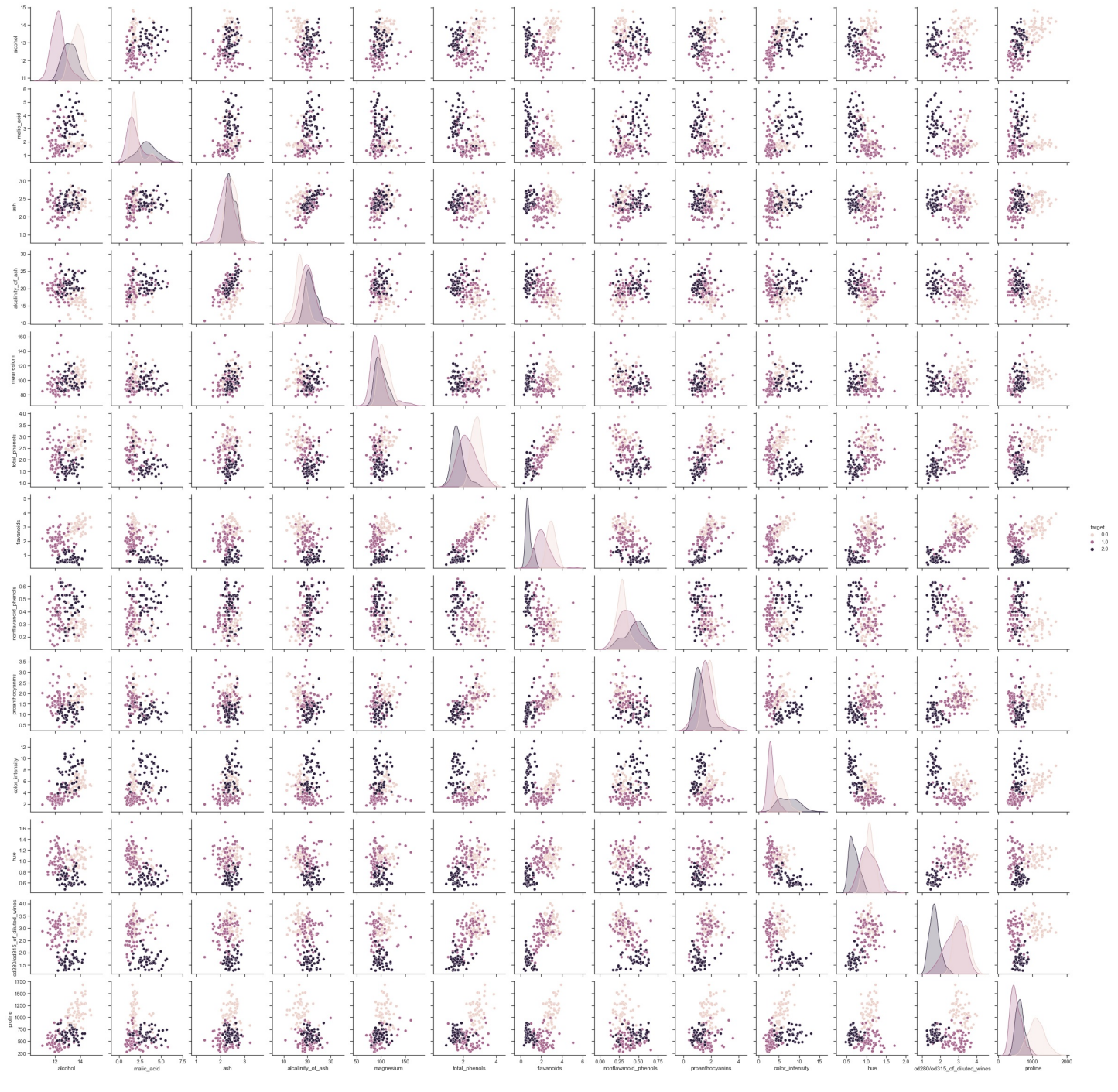


In [39]:

```
sns.pairplot(data, hue="target")
```

Out[39]:

<seaborn.axisgrid.PairGrid at 0x14f778447c0>

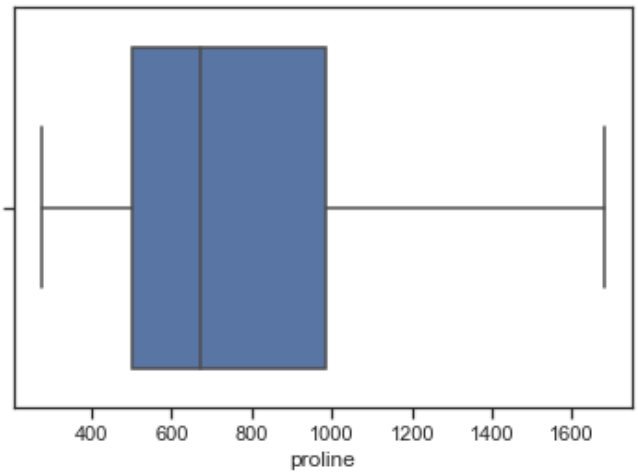


In [40]:

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

Out[40]:

```
<AxesSubplot:xlabel='proline'>
```

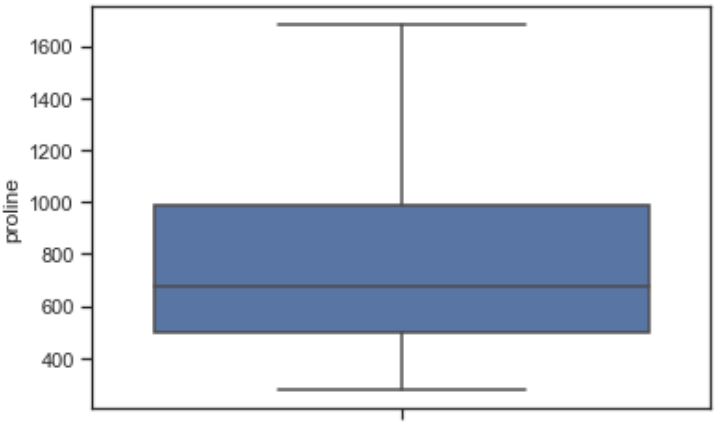


In [41]:

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

Out[41]:

```
<AxesSubplot:ylabel='proline'>
```

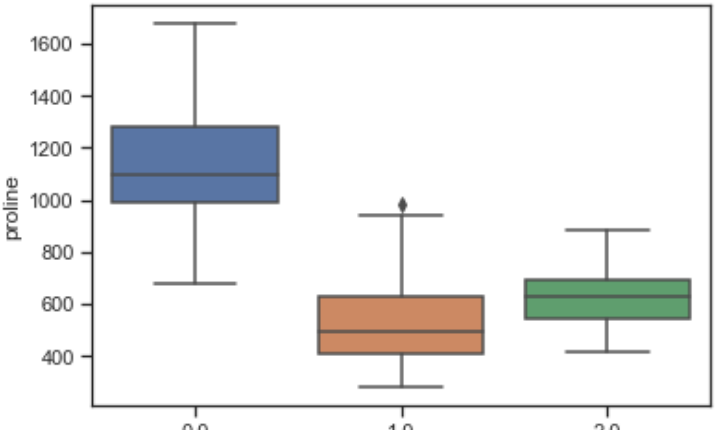


In [43]:

```
# Распределение параметра proline сгруппированные по target.
sns.boxplot(x='target', y='proline', data=data)
```

Out[43]:

```
<AxesSubplot:xlabel='target', ylabel='proline'>
```

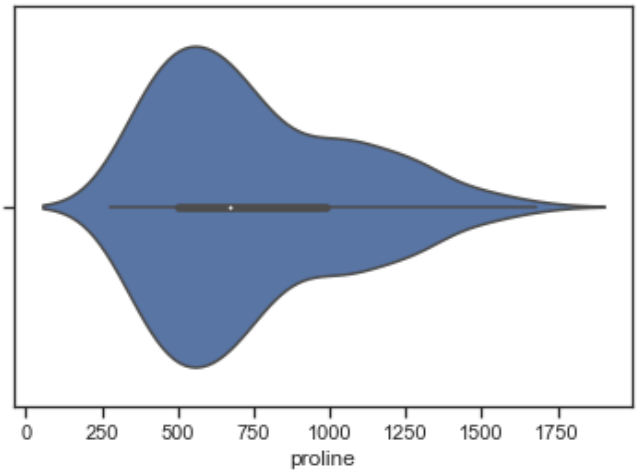


In [44]:

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

Out[44]:

```
<AxesSubplot:xlabel='proline'>
```



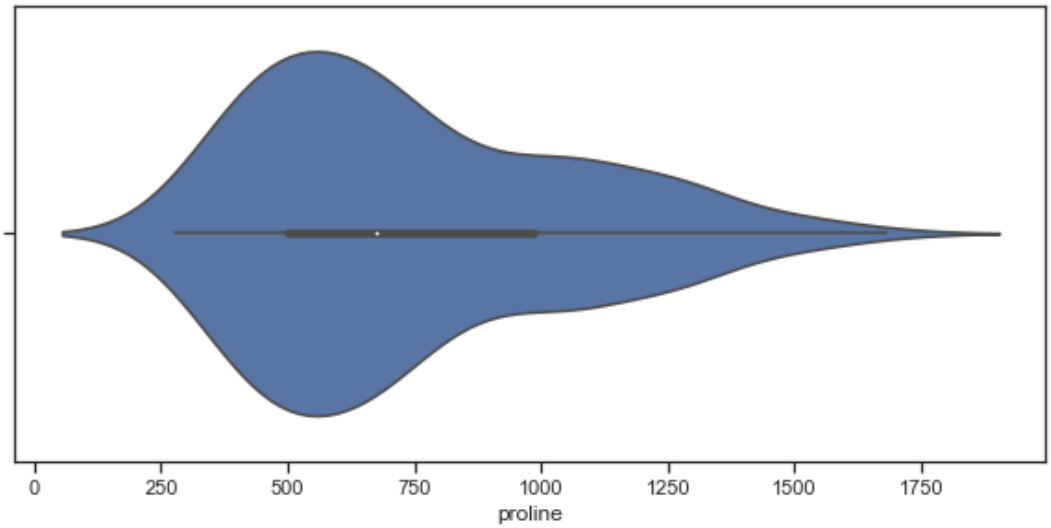
In [45]:

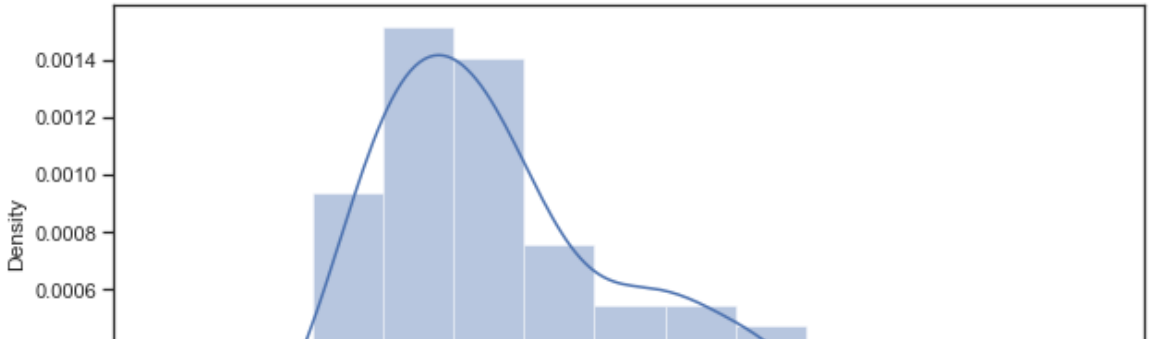
```
fig, ax = plt.subplots(2, 1, figsize=(10,10))
sns.violinplot(ax=ax[0], x=data['proline'])
sns.distplot(data['proline'], ax=ax[1])
```

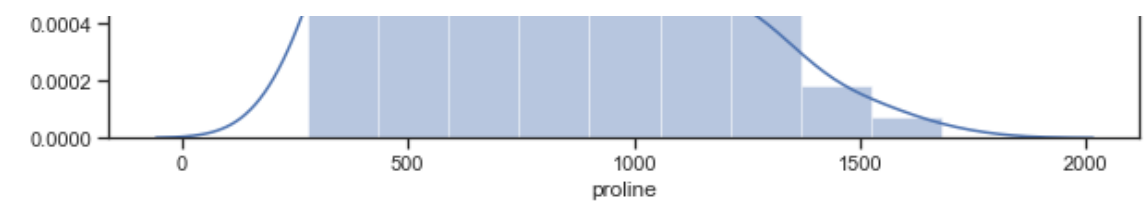
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[45]:

```
<AxesSubplot:xlabel='proline', ylabel='Density'>
```





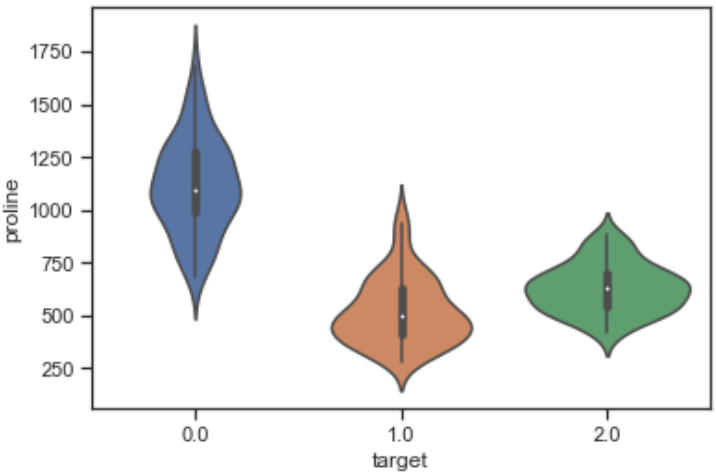


In [46]:

```
# Распределение параметра proline сгруппированные по ср.
sns.violinplot(x='target', y='proline', data=data)
```

Out[46]:

```
<AxesSubplot:xlabel='target', ylabel='proline'>
```

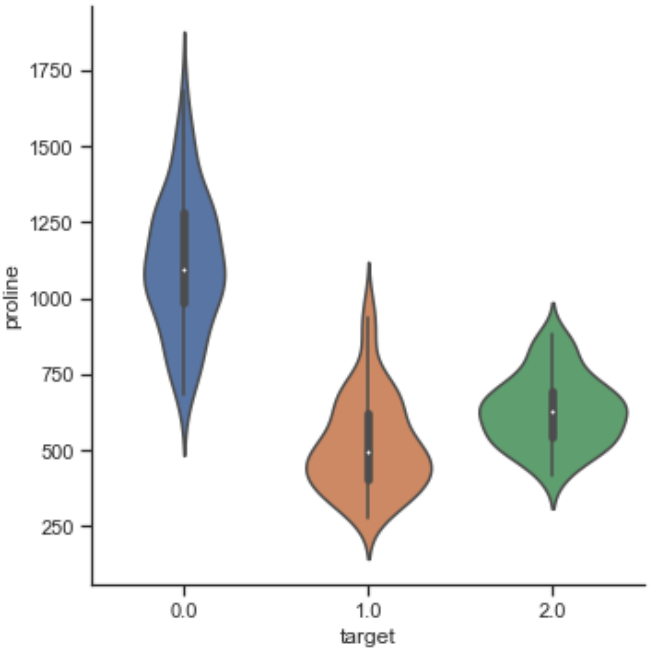


In [47]:

```
sns.catplot(y='proline', x='target', data=data, kind="violin", split=True)
```

Out[47]:

```
<seaborn.axisgrid.FacetGrid at 0x14f00c06760>
```



In [48]:

```
# Информация о корреляции признаков
data.corr()
```

Out[48]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflav

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

In [49]:

```
data.corr(method='pearson')
```

Out[49]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflav
alcohol	1.000000	0.094397	0.211545	-0.310235	0.270798	0.289101	0.236815	
malic_acid	0.094397	1.000000	0.164045	0.288500	-0.054575	-0.335167	-0.411007	
ash	0.211545	0.164045	1.000000	0.443367	0.286587	0.128980	0.115077	
alcalinity_of_ash	-0.310235	0.288500	0.443367	1.000000	-0.083333	-0.321113	-0.351370	
magnesium	0.270798	-0.054575	0.286587	-0.083333	1.000000	0.214401	0.195784	
total_phenols	0.289101	-0.335167	0.128980	-0.321113	0.214401	1.000000	0.864564	
flavanoids	0.236815	-0.411007	0.115077	-0.351370	0.195784	0.864564	1.000000	
nonflavanoid_phenols	-0.155929	0.292977	0.186230	0.361922	-0.256294	-0.449935	-0.537900	
proanthocyanins	0.136698	-0.220746	0.009652	-0.197327	0.236441	0.612413	0.652692	
color_intensity	0.546364	0.248985	0.258887	0.018732	0.199950	-0.055136	-0.172379	
hue	-0.071747	-0.561296	-0.074667	-0.273955	0.055398	0.433681	0.543479	

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflav
od280/od315_of_diluted_wines	0.072343	-0.368710	0.003911	-0.276769	0.066004	0.699949	0.787194	
proline	0.643720	-0.192011	0.223626	-0.440597	0.393351	0.498115	0.494193	
target	-0.328222	0.437776	-0.049643	0.517859	-0.209179	-0.719163	-0.847498	

In [50]:

```
data.corr(method='kendall')
```

Out[50]:

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

In [51]:

```
data.corr(method='spearman')
```

Out[51]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflav
alcohol	1.000000	0.140430	0.243722	-0.306598	0.365503	0.310920	0.294740	
malic_acid	0.140430	1.000000	0.230674	0.304069	0.080188	-0.280225	-0.325202	
ash	0.243722	0.230674	1.000000	0.366374	0.361488	0.132193	0.078796	

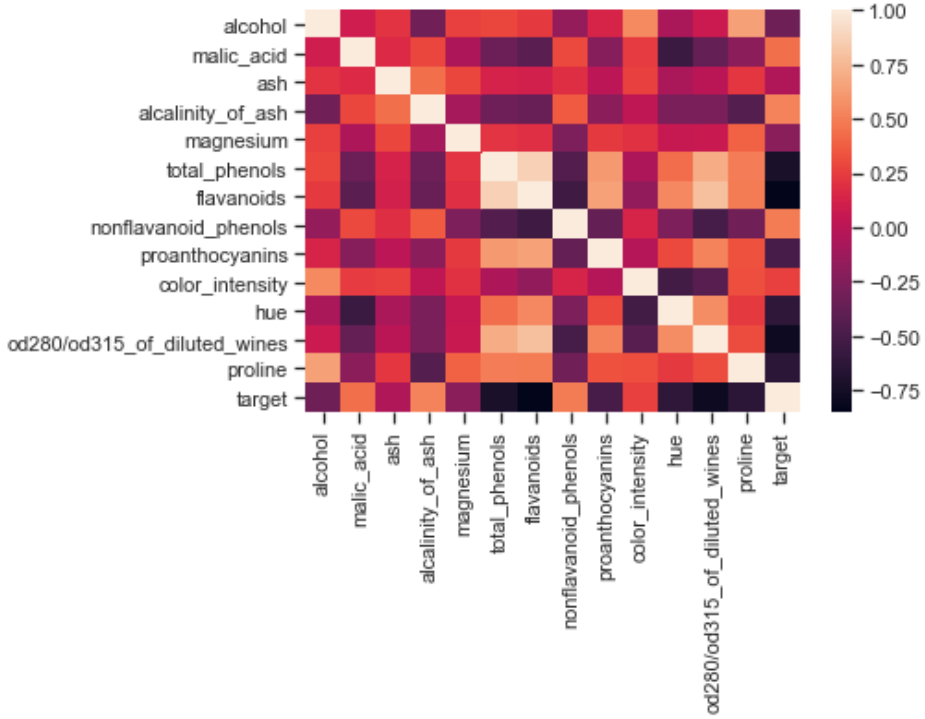
	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflav
alcalinity_of_ash	-0.306398	0.304069	0.366374	1.000000	-0.169558	0.376657	0.443770	
magnesium	0.365503	0.080188	0.361488	-0.169558	1.000000	0.246417	0.233167	
total_phenols	0.310920	-0.280225	0.132193	-0.376657	0.246417	1.000000	0.879404	
flavanoids	0.294740	-0.325202	0.078796	-0.443770	0.233167	0.879404	1.000000	
nonflavanoid_phenols	-0.162207	0.255236	0.145583	0.389390	-0.236786	-0.448013	-0.543897	
proanthocyanins	0.192734	-0.244825	0.024384	-0.253695	0.173647	0.666689	0.730322	
color_intensity	0.635425	0.290307	0.283047	-0.073776	0.357029	0.011162	-0.042910	
hue	-0.024203	-0.560265	-0.050183	-0.352507	0.036095	0.439457	0.535430	
od280/od315_of_diluted_wines	0.103050	-0.255185	-0.007500	-0.325890	0.056963	0.687207	0.741533	
proline	0.633580	-0.057466	0.253163	-0.456090	0.507575	0.419470	0.429904	
target	-0.354167	0.346913	-0.053988	0.569792	-0.250498	-0.726544	-0.854908	

In [52]:

```
sns.heatmap(data.corr())
```

Out[52]:

<AxesSubplot:>

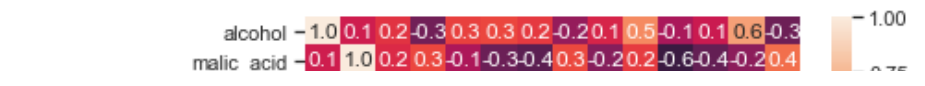


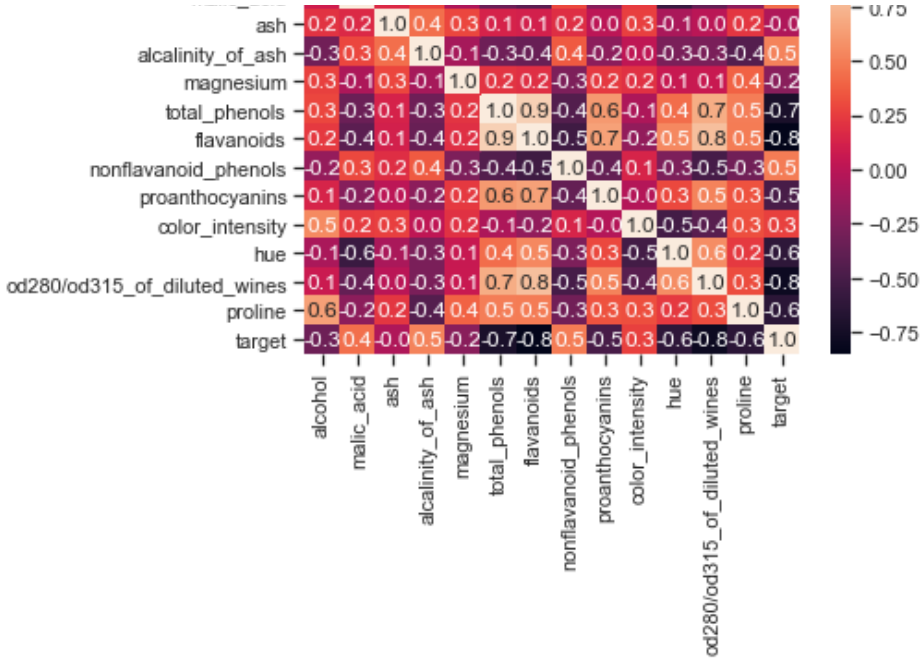
In [53]:

```
# Вывод значений в ячейках
sns.heatmap(data.corr(), annot=True, fmt='.1f')
```

Out[53]:

<AxesSubplot:>



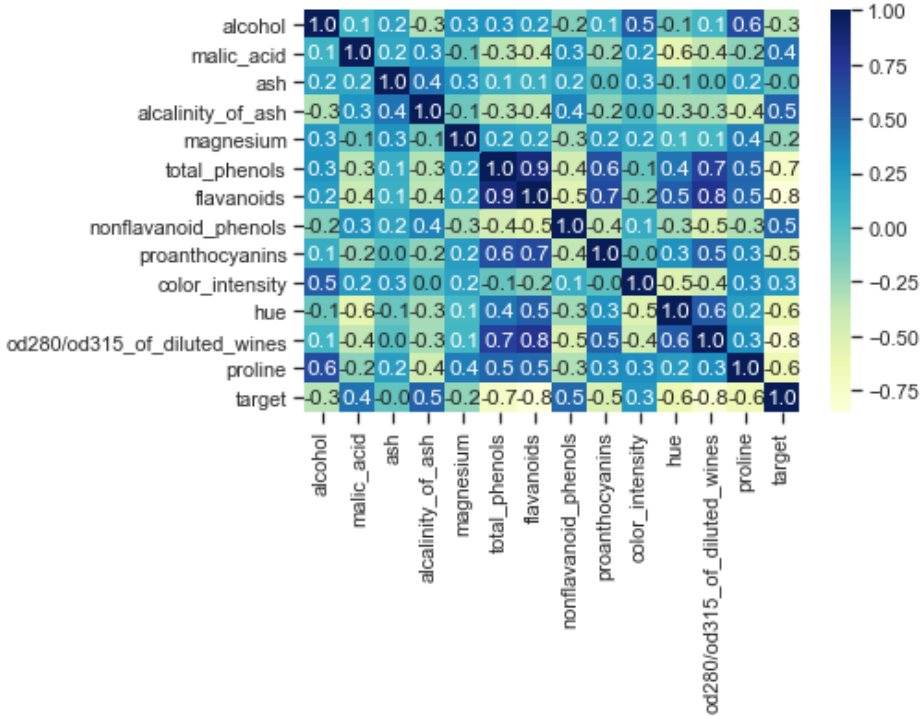


In [54]:

```
# Изменение цветовой гаммы
sns.heatmap(data.corr(), cmap='YlGnBu', annot=True, fmt='.1f')
```

Out[54]:

<AxesSubplot:>

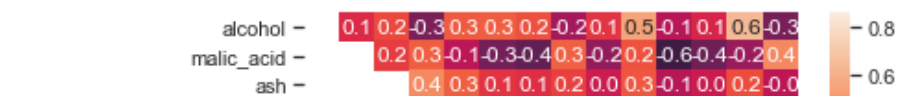


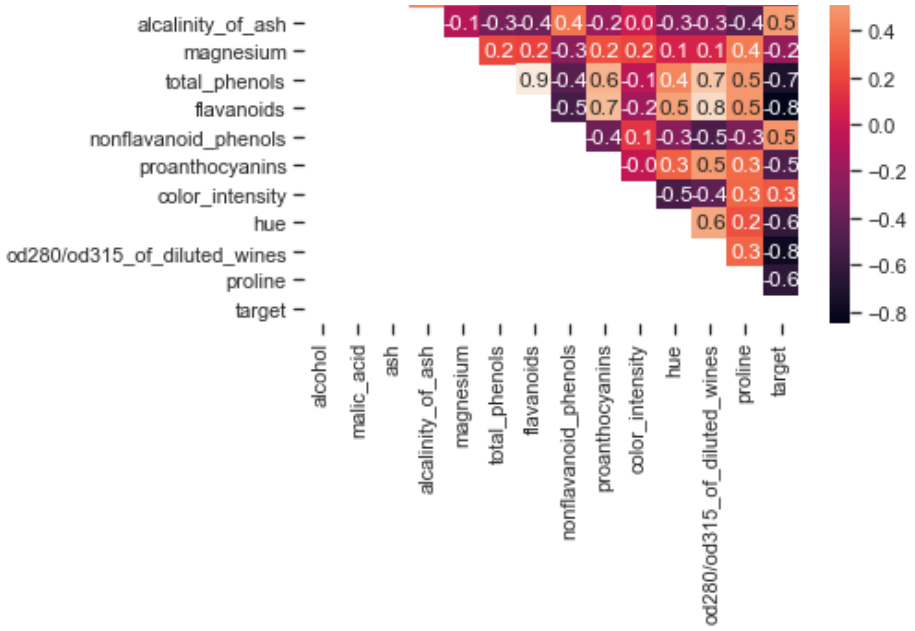
In [55]:

```
# 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='.1f')
```

Out[55]:

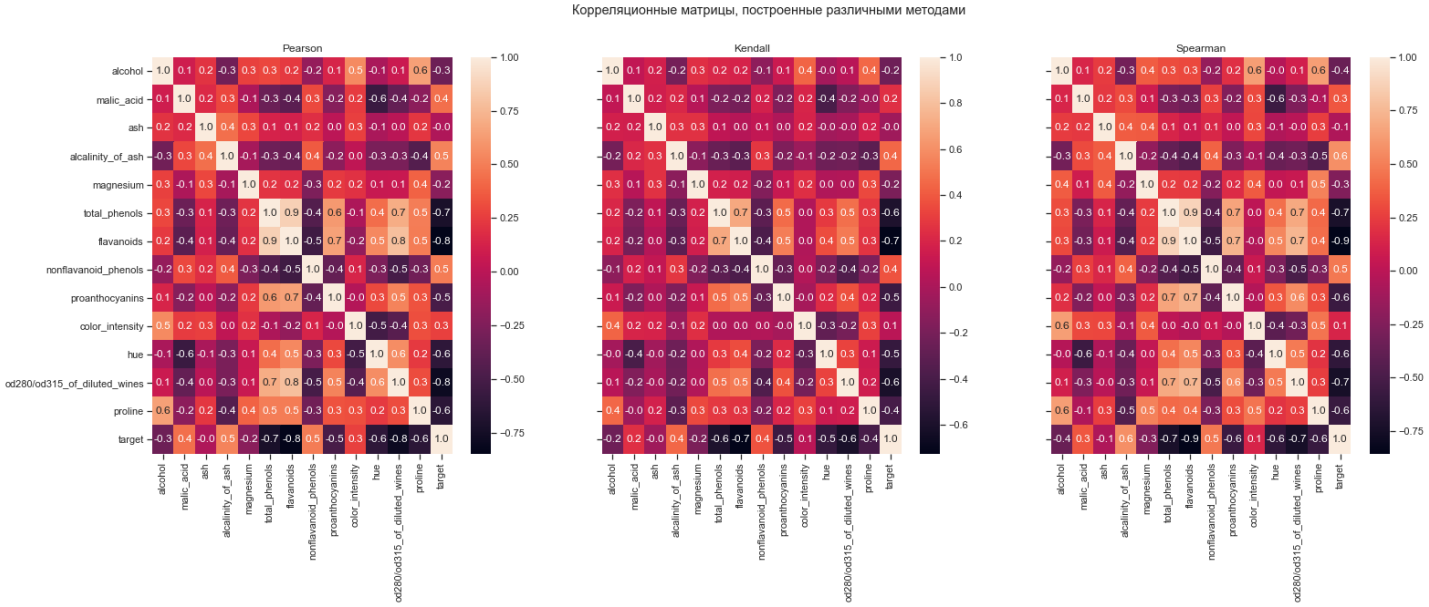
<AxesSubplot:>





In [8]:

```
fig, ax = plt.subplots(1, 3, sharex='col', sharey='row', figsize=(25,8))
sns.heatmap(data.corr(method='pearson'), ax=ax[0], annot=True, fmt='.1f')
sns.heatmap(data.corr(method='kendall'), ax=ax[1], annot=True, fmt='.1f')
sns.heatmap(data.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')
```



In [ ]: