Министерство образования и науки 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)

МЕТОДЫ МАШИННОГО ОБУЧЕНИЯ

Отчёт по лабораторной работе № 1

«Разведочный анализ данных. Исследование и визуализация данных.»

Выполнила:
студентка группы ИУ5 – 23М Галичий Д. А.

Преподаватель: Гапанюк Ю. Е.

2020г.

Текстовое описание набора данных

В качестве набора данных будем использовать набор данных, содержащий информацию о вине.

Эти данные являются результатами химического анализа вин, выращенных в одном регионе Италии, но полученных из трех различных сортов. В результате анализа было определено количество 13 компонентов, содержащихся в каждом из трех видов вин.

Файл содержит следующие колонки:

1) Alcohol
2) Malic acid
3) Ash
4) Alcalinity of ash
5) Magnesium
6) Total phenols
7) Flavanoids
8) Nonflavanoid phenols
9) Proanthocyanins
10) Color intensity
11) Hue
12) OD280/OD315 of diluted wines
13) Proline

Класс записан в первом столбце (три варианта), признаки — в столбцах со второго по последний.

Импорт библиотек

In [9]:

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

Загрузка данных

In [10]:

```
data = pd.read_csv('data/wine.csv', sep=",")
```

Основные характеристики датасета

In [12]:

```
# Первые 5 строк датасета
data.head()
```

Out[12]:

	Wine	Alcohol	Malic.acid	Ash	Acl	Mg	Phenols	Flavanoids	Nonflavanoid.phenols	Pr
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	

In [15]:

```
# Строки, колонки - количество
data.shape
```

Out[15]:

```
(178, 14)
```

In [16]:

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

```
Всего строк: 178
```

In [18]:

```
# Список колонок
data.columns
```

Out[18]:

```
Index(['Wine', 'Alcohol', 'Malic.acid', 'Ash', 'Acl', 'Mg', 'Phenols',
       'Flavanoids', 'Nonflavanoid.phenols', 'Proanth', 'Color.int', 'Hu
e',
       'OD', 'Proline'],
      dtype='object')
```

In [20]:

```
# Список колонок с типами данных
data.dtypes
```

Out[20]:

```
Wine                      int64
Alcohol                 float64
Malic.acid              float64
Ash                     float64
Acl                     float64
Mg                        int64
Phenols                 float64
Flavanoids              float64
Nonflavanoid.phenols    float64
Proanth                 float64
Color.int               float64
Hue                     float64
OD                      float64
Proline                   int64
dtype: object
```

In [21]:

```
# 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))
```

```
Wine - 0
Alcohol - 0
Malic.acid - 0
Ash - 0
Acl - 0
Mg - 0
Phenols - 0
Flavanoids - 0
Nonflavanoid.phenols - 0
Proanth - 0
Color.int - 0
Hue - 0
OD - 0
Proline - 0
```

In [23]:

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

Out[23]:

	Wine	Alcohol	Malic.acid	Ash	Acl	Mg	Phenols
count	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000
mean	1.938202	13.000618	2.336348	2.366517	19.494944	99.741573	2.295112
std	0.775035	0.811827	1.117146	0.274344	3.339564	14.282484	0.625851
min	1.000000	11.030000	0.740000	1.360000	10.600000	70.000000	0.980000
25%	1.000000	12.362500	1.602500	2.210000	17.200000	88.000000	1.742500
50%	2.000000	13.050000	1.865000	2.360000	19.500000	98.000000	2.355000
75%	3.000000	13.677500	3.082500	2.557500	21.500000	107.000000	2.800000
max	3.000000	14.830000	5.800000	3.230000	30.000000	162.000000	3.880000

In [40]:

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

Out[40]:

```
array([1065, 1050, 1185, 1480,  735, 1450, 1290, 1295, 1045, 1510, 1280,
       1320, 1150, 1547, 1310, 1130, 1680,  845,  780,  770, 1035, 1015,
        830, 1195, 1285,  915, 1515,  990, 1235, 1095,  920,  880, 1105,
       1020,  760,  795,  680,  885, 1080,  985, 1060, 1260, 1265, 1190,
       1375, 1120,  970, 1270,  520,  450,  630,  420,  355,  678,  502,
        510,  750,  718,  870,  410,  472,  886,  428,  392,  500,  463,
        278,  714,  515,  495,  562,  625,  480,  290,  345,  937,  660,
        406,  710,  438,  415,  672,  315,  488,  312,  325,  607,  434,
        385,  407,  372,  564,  465,  365,  380,  378,  352,  466,  342,
        580,  530,  560,  600,  650,  695,  720,  590,  550,  855,  425,
        675,  640,  725,  620,  570,  615,  685,  470,  740,  835,  840],
      dtype=int64)
```

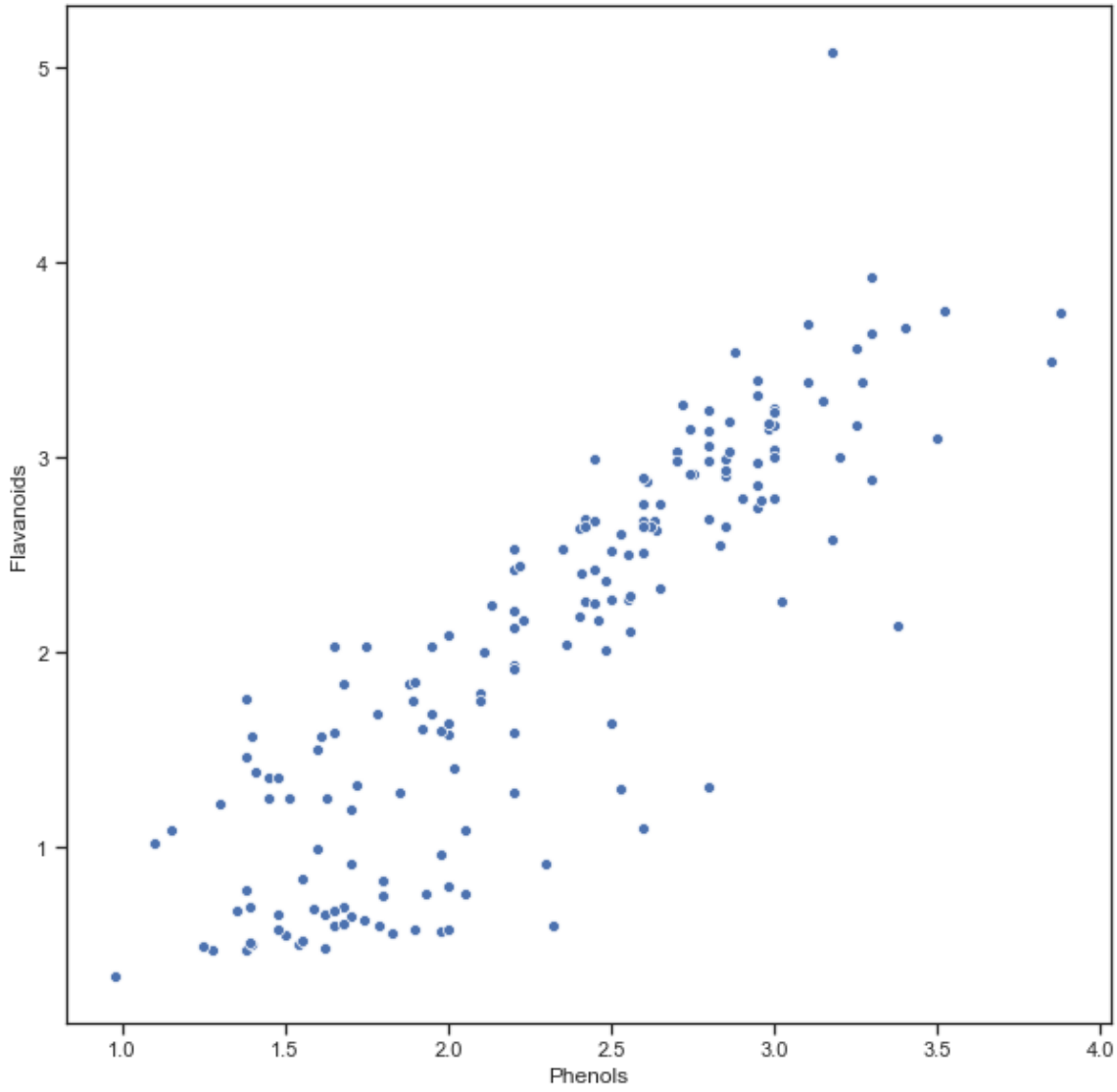
Визуальное исследование датасета

In [39]:

```
# Диаграмма рассеяния
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='Phenols', y='Flavanoids', data=data)
```

Out[39]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x22b9026ac18>
```

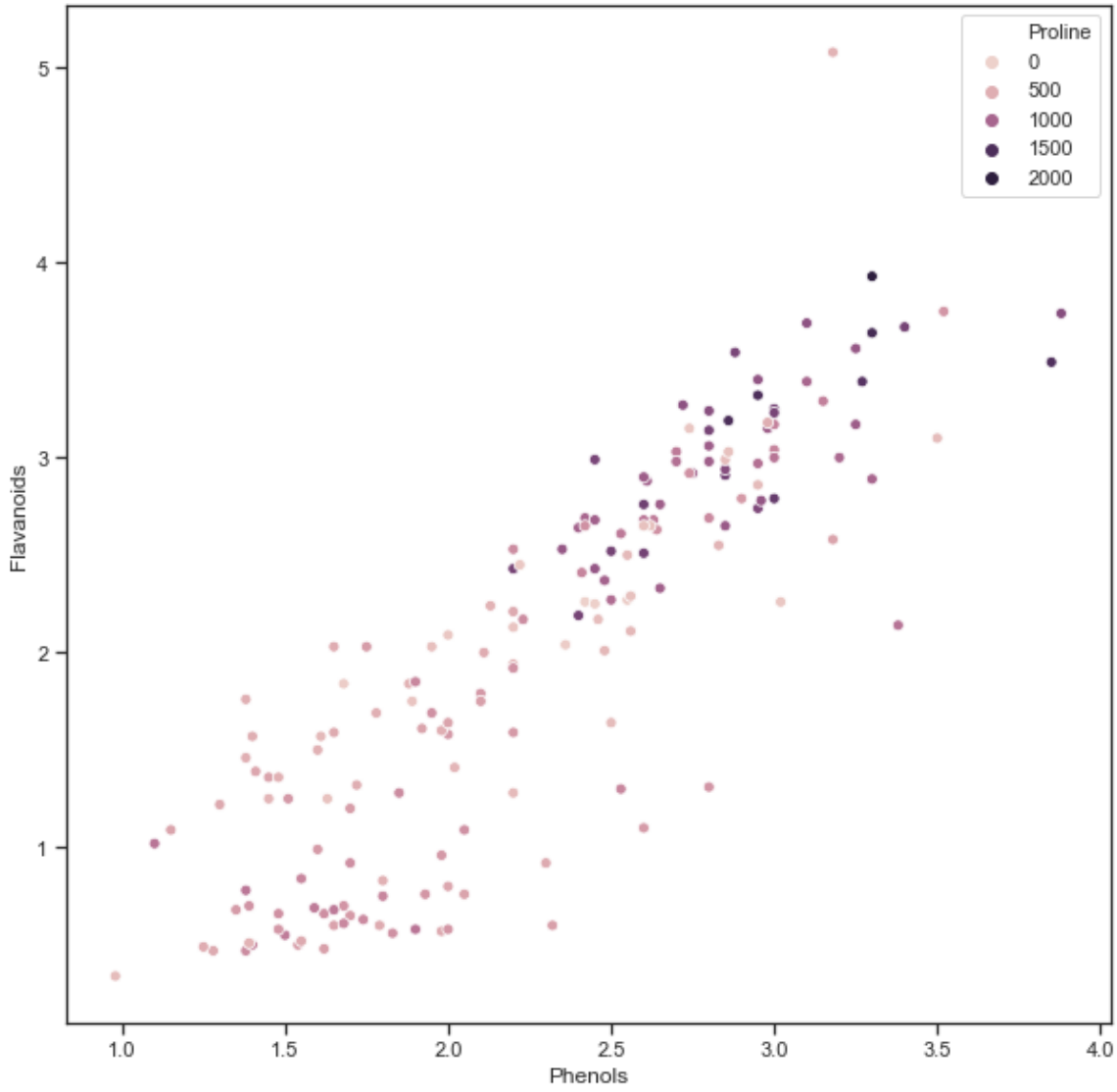


In [41]:

```
# Диаграмма рассеяния с учетом целевого признака
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='Phenols', y='Flavanoids', data=data, hue='Proline')
```

Out[41]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x22b905d2d30>
```



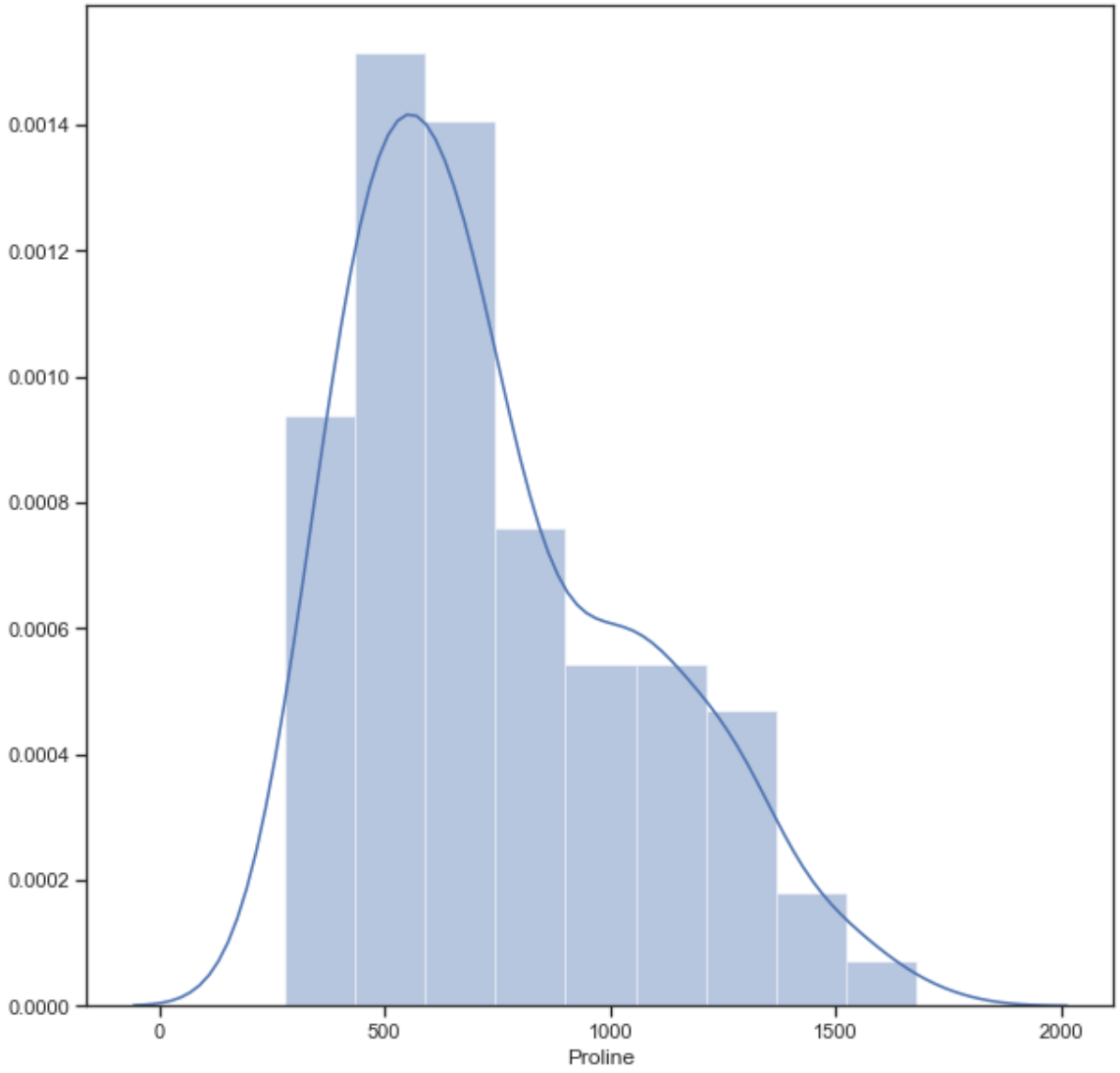
Гистограмма

In [43]:

```
# Плотность вероятности распределения данных
fig, ax = plt.subplots(figsize=(10,10))
sns.distplot(data['Proline'])
```

Out[43]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x22b906df5c0>
```



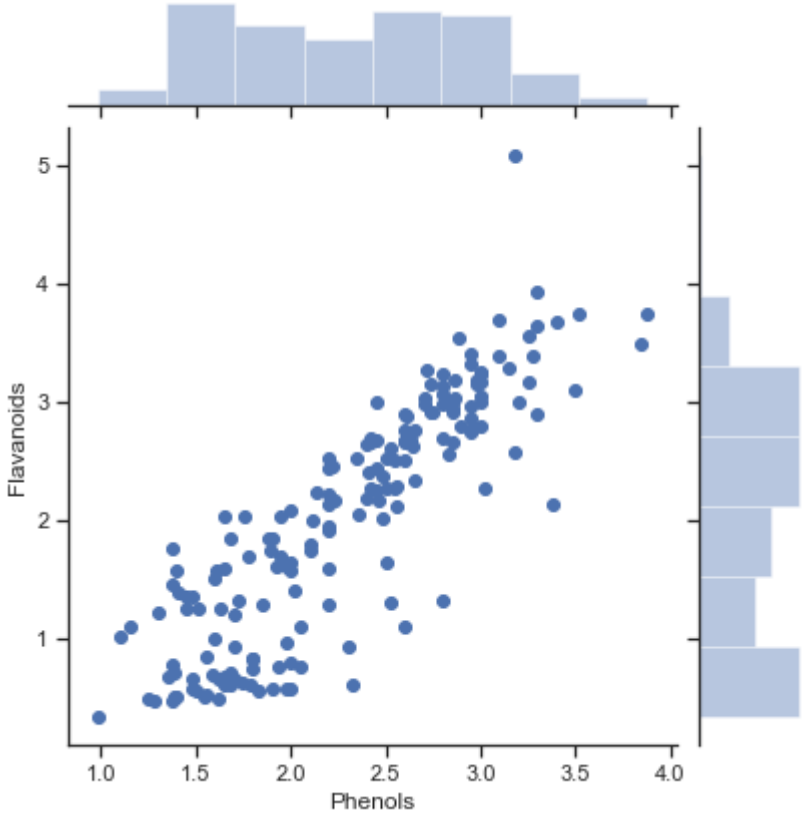
Joinplot

In [45]:

```
# Комбинация гистограмм и диаграмм рассеивания
sns.jointplot(x='Phenols', y='Flavanoids', data=data)
```

Out[45]:

```
<seaborn.axisgrid.JointGrid at 0x22b90766b38>
```

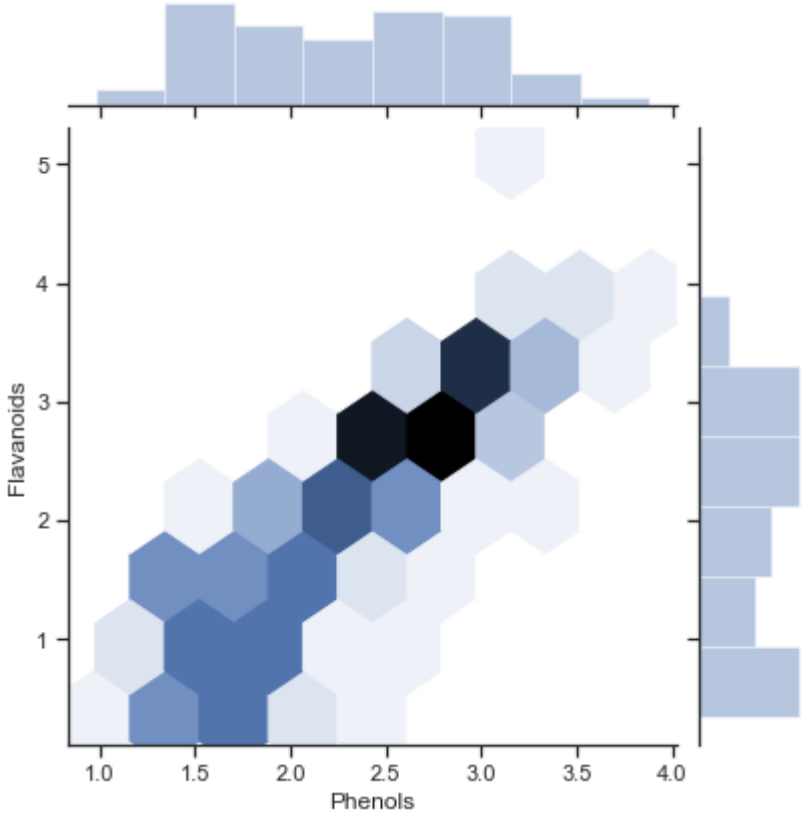


In [46]:

```
sns.jointplot(x='Phenols', y='Flavanoids', data=data, kind="hex")
```

Out[46]:

```
<seaborn.axisgrid.JointGrid at 0x22b90b31390>
```

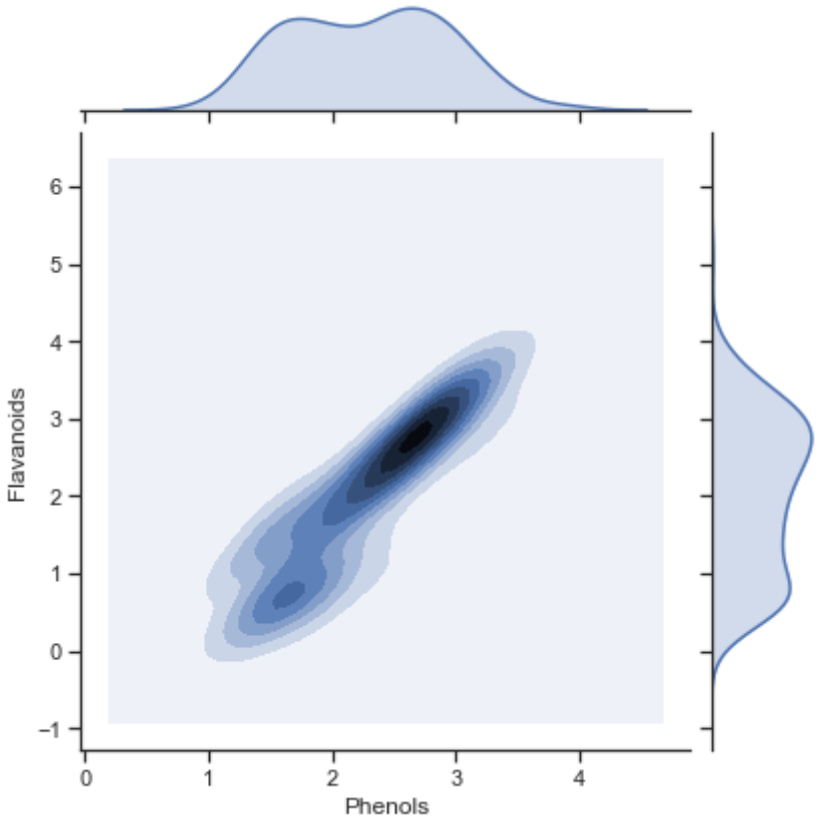


In [47]:

```
sns.jointplot(x='Phenols', y='Flavanoids', data=data, kind="kde")
```

Out[47]:

```
<seaborn.axisgrid.JointGrid at 0x22b910b79e8>
```



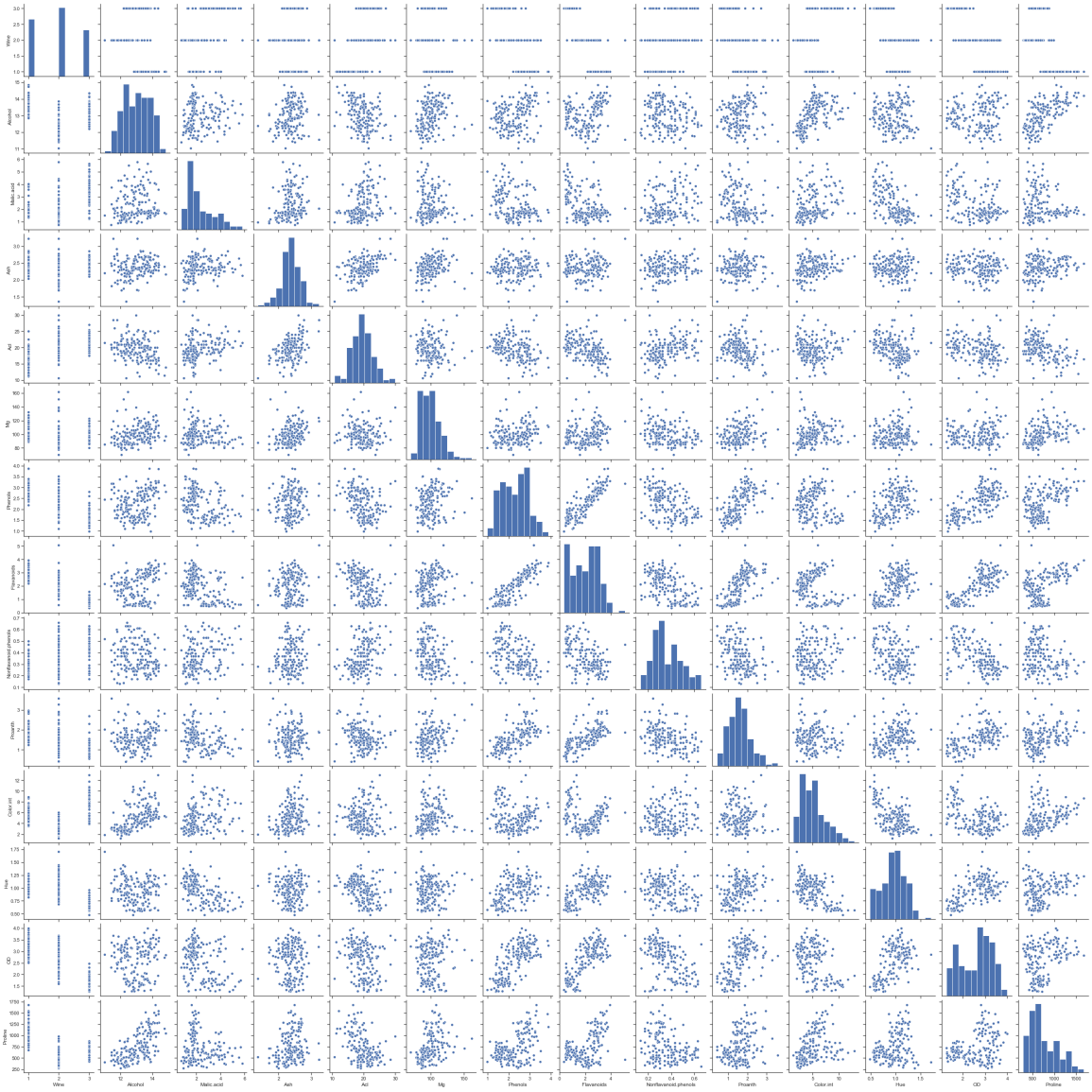
Парные диаграммы

In [49]:

```
# Комбинация гистограмм и диаграмм рассеивания для всего набора данных
sns.pairplot(data)
```

Out[49]:

```
<seaborn.axisgrid.PairGrid at 0x22b999d1cf8>
```



Ящик с усами

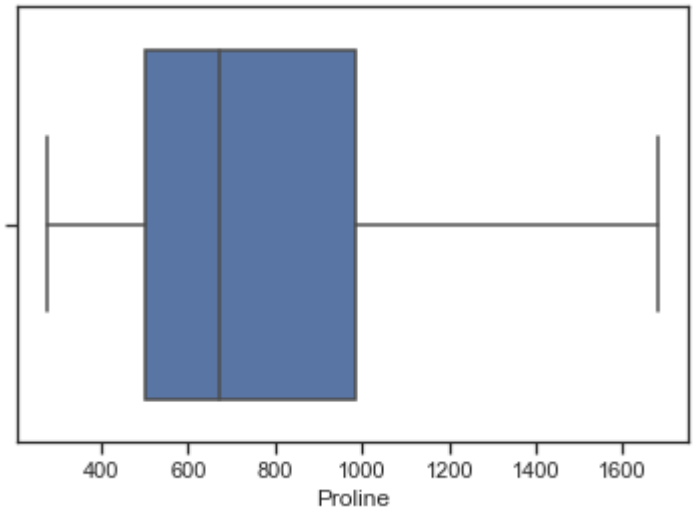
Отображает одномерное распределение вероятности.

In [51]:

```
# Одномерное распределение вероятности
sns.boxplot(x=data['Proline'])
```

Out[51]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x22ba1f86dd8>
```

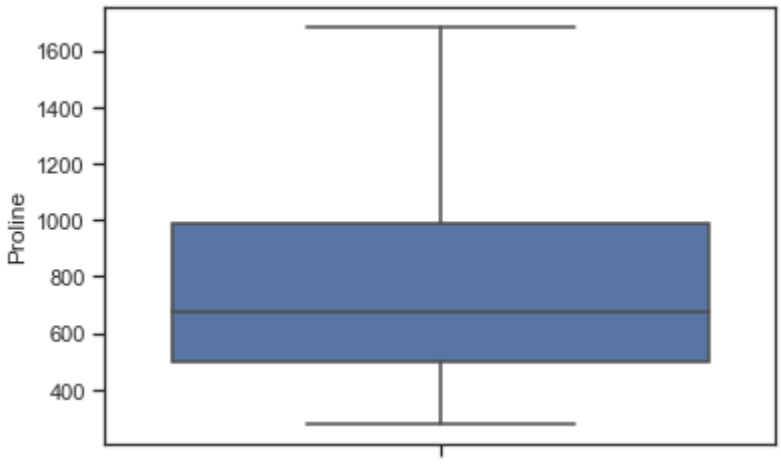


In [54]:

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

Out[54]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x22ba55152b0>
```



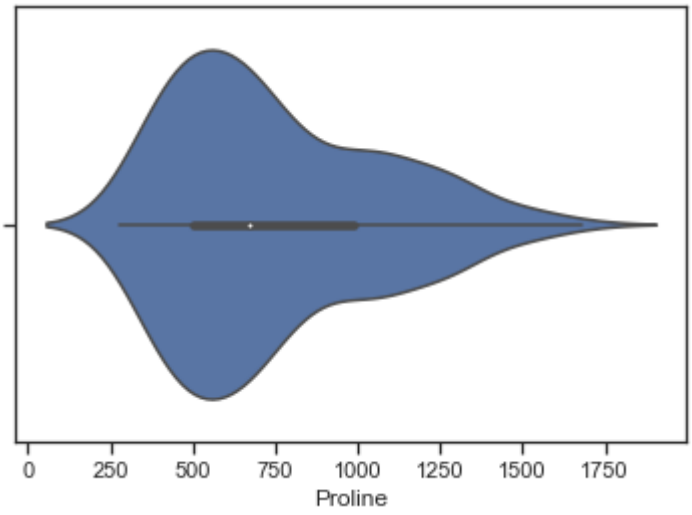
Violin plot

In [56]:

```
# По краям отображаются распределения плотности
sns.violinplot(x=data['Proline'])
```

Out[56]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x22ba2c6a5f8>
```

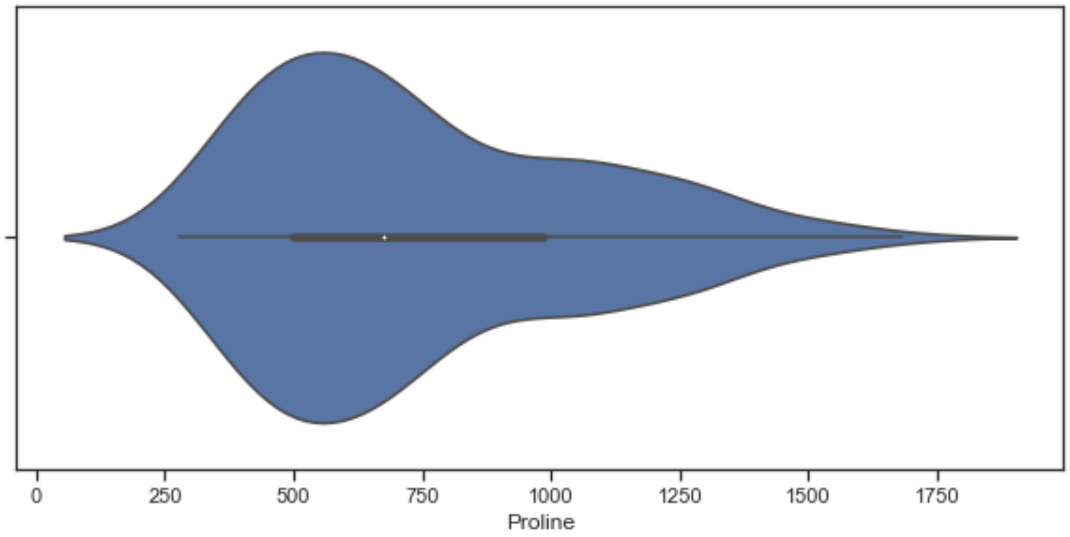


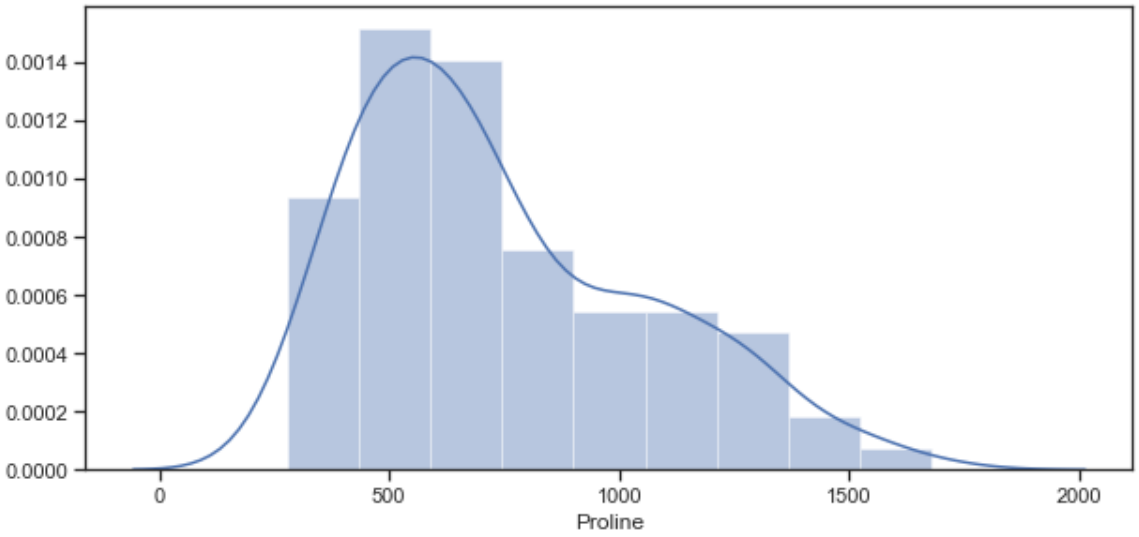
In [57]:

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

Out[57]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x22ba5661eb8>
```


0
250
500
750
1000
1250
1500
1750
Proline


0.0014
0.0012
0.0010
0.0008
0.0006
0.0004
0.0002
0.0000
0
500
1000
1500
2000
Proline

Информация о корреляции признаков

In [58]:

```
# Корреляционная матрица
data.corr()
```

Out[58]:

	Wine	Alcohol	Malic.acid	Ash	Acl	Mg	Phe
Wine	1.000000	-0.328222	0.437776	-0.049643	0.517859	-0.209179	-0.719
Alcohol	-0.328222	1.000000	0.094397	0.211545	-0.310235	0.270798	0.289
Malic.acid	0.437776	0.094397	1.000000	0.164045	0.288500	-0.054575	-0.335
Ash	-0.049643	0.211545	0.164045	1.000000	0.443367	0.286587	0.128
Acl	0.517859	-0.310235	0.288500	0.443367	1.000000	-0.083333	-0.32
Mg	-0.209179	0.270798	-0.054575	0.286587	-0.083333	1.000000	0.214
Phenols	-0.719163	0.289101	-0.335167	0.128980	-0.321113	0.214401	1.000
Flavanoids	-0.847498	0.236815	-0.411007	0.115077	-0.351370	0.195784	0.864
Nonflavanoid.phenols	0.489109	-0.155929	0.292977	0.186230	0.361922	-0.256294	-0.449
Proanth	-0.499130	0.136698	-0.220746	0.009652	-0.197327	0.236441	0.612
Color.int	0.265668	0.546364	0.248985	0.258887	0.018732	0.199950	-0.055
Hue	-0.617369	-0.071747	-0.561296	-0.074667	-0.273955	0.055398	0.433
OD	-0.788230	0.072343	-0.368710	0.003911	-0.276769	0.066004	0.699
Proline	-0.633717	0.643720	-0.192011	0.223626	-0.440597	0.393351	0.498

In [60]:

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

Out[60]:

	Wine	Alcohol	Malic.acid	Ash	Acl	Mg	Phe
Wine	1.000000	-0.238984	0.247494	-0.038085	0.449402	-0.184992	-0.59
Alcohol	-0.238984	1.000000	0.093844	0.170154	-0.212978	0.250506	0.20
Malic.acid	0.247494	0.093844	1.000000	0.158178	0.210119	0.050869	-0.17
Ash	-0.038085	0.170154	0.158178	1.000000	0.258352	0.254246	0.08
Acl	0.449402	-0.212978	0.210119	0.258352	1.000000	-0.121005	-0.25
Mg	-0.184992	0.250506	0.050869	0.254246	-0.121005	1.000000	0.17
Phenols	-0.590404	0.209099	-0.174929	0.089855	-0.256669	0.172195	1.00
Flavanoids	-0.725255	0.191087	-0.211918	0.049474	-0.309865	0.161603	0.70
Nonflavanoid.phenols	0.379234	-0.109554	0.175129	0.098937	0.278091	-0.158361	-0.31
Proanth	-0.450225	0.133526	-0.168714	0.018240	-0.171404	0.117871	0.46
Color.int	0.065124	0.434353	0.195607	0.187786	-0.057281	0.241781	0.02
Hue	-0.479229	-0.021717	-0.388707	-0.037234	-0.239210	0.023760	0.28
OD	-0.607572	0.061513	-0.162909	-0.006341	-0.226253	0.034307	0.47
Proline	-0.406260	0.449387	-0.044660	0.171574	-0.313218	0.343016	0.28

На основе корреляционной матрицы можно сделать следующие выводы:

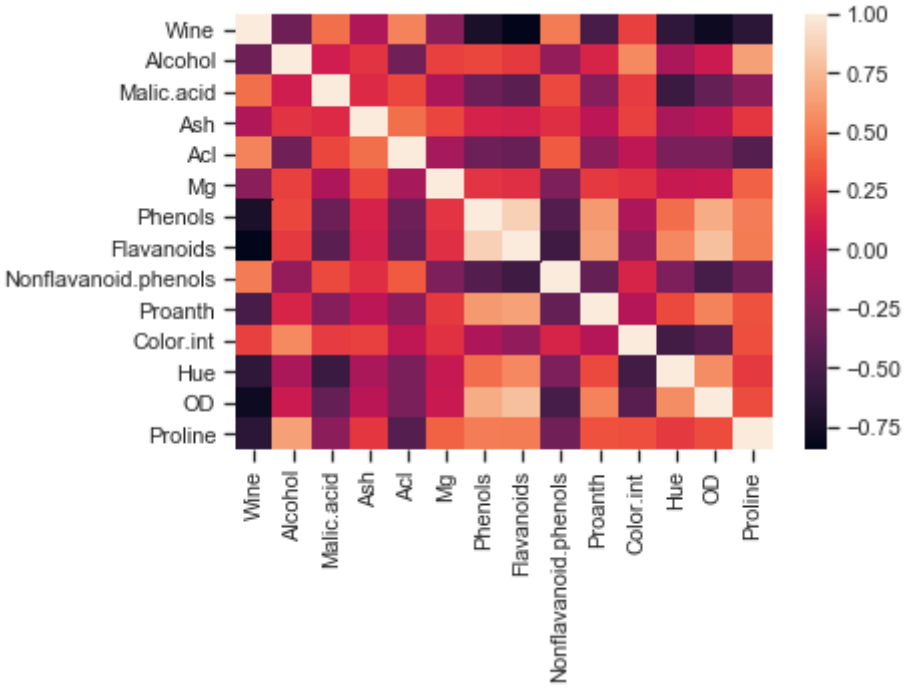
- Целевой признак Proline наиболее сильно коррелирует с Alcohol (0.449) и Wine (0.406). Эти признаки следует оставить в модели.
- Целевой признак отчасти коррелирует с Mg (0.343), Acl (0.313) и Color.int (0.316). Эти признаки стоит также оставить в модели.
- Целевой признак слабо коррелирует с Malic.Acid (0.044), Hue (0.143), OD (0.151) и Ash (0.171). Скорее всего эти признаки стоит исключить из модели, возможно они только ухудшат качество модели.

In [62]:

```
# Тепловая карта
sns.heatmap(data.corr())
```

Out[62]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x22ba56ed550>
```



In [0]:

```
fig, ax = plt.subplots(1, 3, sharex='col', sharey='row', figsize=(15,5))
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')
```