Лабораторная работа №1

Разведочный анализ данных. Исследование и визуализация данных.

Выполнил: Борисочкин М. И., РТ5-61Б

Текстовое описание набора данных

В качестве набора данных был выбран игрушечный датасет "Ирисы Фишера"(load_iris) из библиотеки scikit-learn.

В данном датасете присутствуют следующие столбцы:

- sepal length — длина чашелистика в см;
- sepal width — ширина чашелистика в см;
- petal length — длина лепестка в см;
- petal width — ширина лепестка в см;
- target — целевой признак. Представляет собой виды ирисов: Iris setosa (0), Iris versicolor (1), Iris virginica (2).

Импорт библиотек

In [15]:
```
from sklearn.datasets import load_iris

import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Загрузка данных

Загрузим выбранный датасет как датафрейм Pandas. Для этого воспользуемся параметром as_frame метода load_iris и полем frame получившегося датасета.

In [16]:
```
iris = load_iris(as_frame=True)
data : pd.DataFrame = iris.frame
```

Основные характеристики датасета

In [17]:
```
# Первые 5 строк датасета
data.head()
```

Out[17]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

In [18]:
```
# Размер датасета
total_rows, total_columns = data.shape[0], data.shape[1]
print('Всего строк: {} \nВсего столбцов: {}'.format(total_rows, total_columns))
```

```
Всего строк: 150
Всего столбцов: 5
```

In [19]:
```
# Тип данных в столбцах
data.dtypes
```

Out[19]:
```
sepal length (cm)    float64
sepal width (cm)     float64
petal length (cm)    float64
petal width (cm)     float64
target                 int32
dtype: object
```

In [20]:
```
# Проверка наличия пустых значений
for column in data.columns:
    temp_null_count = data[data[column].isnull()].shape[0]
    print('{} — {}'.format(column, temp_null_count))
```

```
sepal length (cm) — 0
sepal width (cm) — 0
petal length (cm) — 0
petal width (cm) — 0
target — 0
```

In [21]:
```
# Статистические характеристки набора данных
data.describe()
```

Out[21]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333	1.000000
std	0.828066	0.435866	1.765298	0.762238	0.819232
min	4.300000	2.000000	1.000000	0.100000	0.000000
25%	5.100000	2.800000	1.600000	0.300000	0.000000
50%	5.800000	3.000000	4.350000	1.300000	1.000000
75%	6.400000	3.300000	5.100000	1.800000	2.000000
max	7.900000	4.400000	6.900000	2.500000	2.000000

In [22]:
```
# Уникальные значения для целевого признака
data['target'].unique()
```

Out[22]:
```
array([0, 1, 2])
```

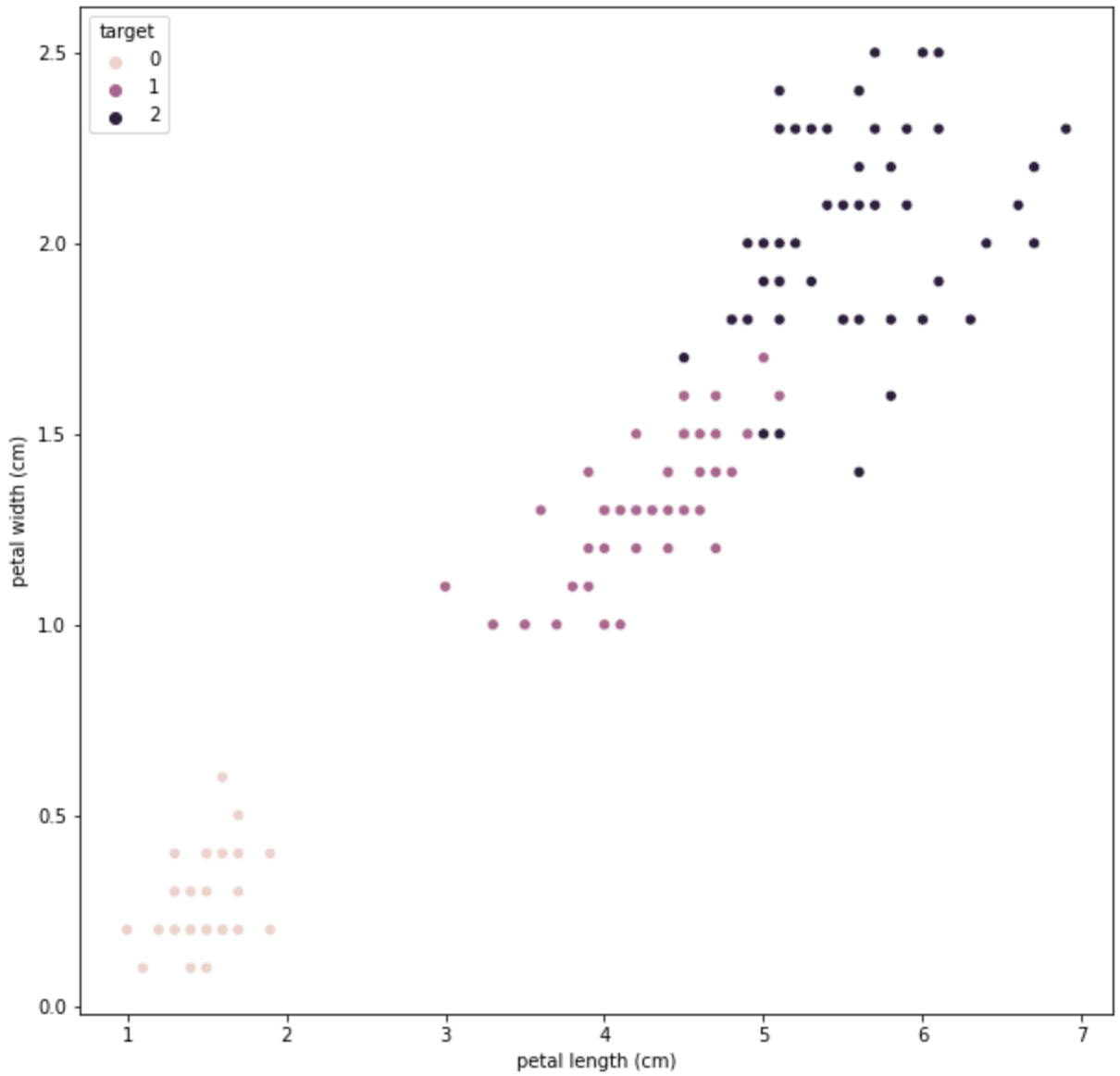
Целевой признак является тернарным и содержит значения 0, 1, 2.

Визуальное исследование датасета

Диаграмма рассеивания

In [23]:
```
fig, ax = plt.subplots(figsize=(10, 10))
sns.scatterplot(ax=ax, x='petal length (cm)', y='petal width (cm)', data=data,
                hue='target')
```

Out[23]:
```
<AxesSubplot:xlabel='petal length (cm)', ylabel='petal width (cm)'>
```

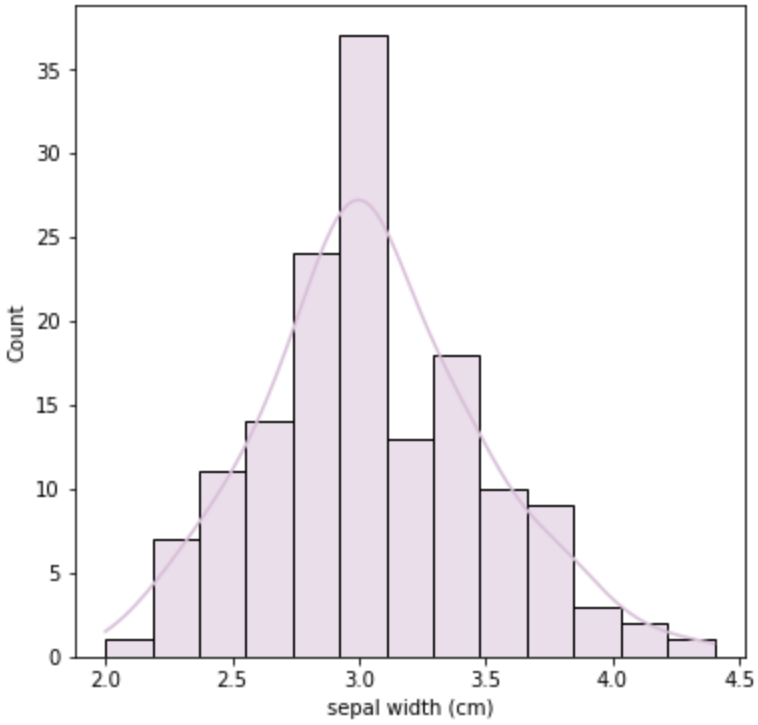


Как видно из диаграммы, ирисы сорта setosa (0) легко отделяются от двух других по длине и ширине лепестка.

Гистограмма распределения ирисов по ширине чашелистика

In [24]:
```
fig, ax = plt.subplots(figsize=(6,6))
sns.histplot(data['sepal width (cm)'], kde=True, color='thistle')
```

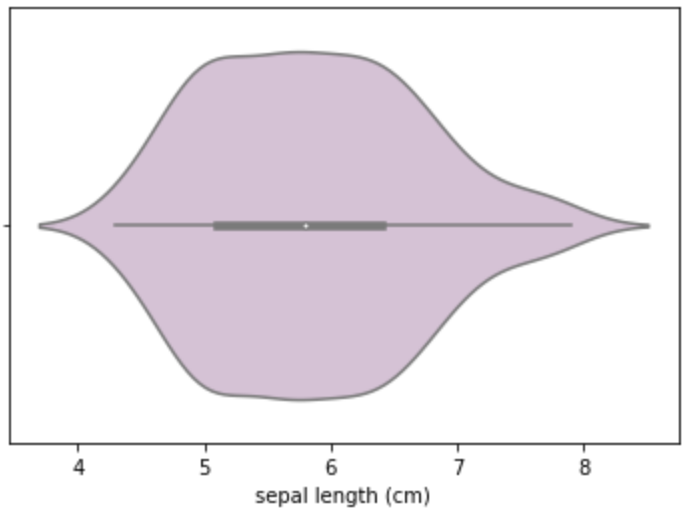
Out[24]:
```
<AxesSubplot:xlabel='sepal width (cm)', ylabel='Count'>
```



Скрипичный график распределения ирисов по длине чашелистика

In [25]:
```
sns.violinplot(x=data['sepal length (cm)'], color='thistle')
```

Out[25]:
```
<AxesSubplot:xlabel='sepal length (cm)'>
```

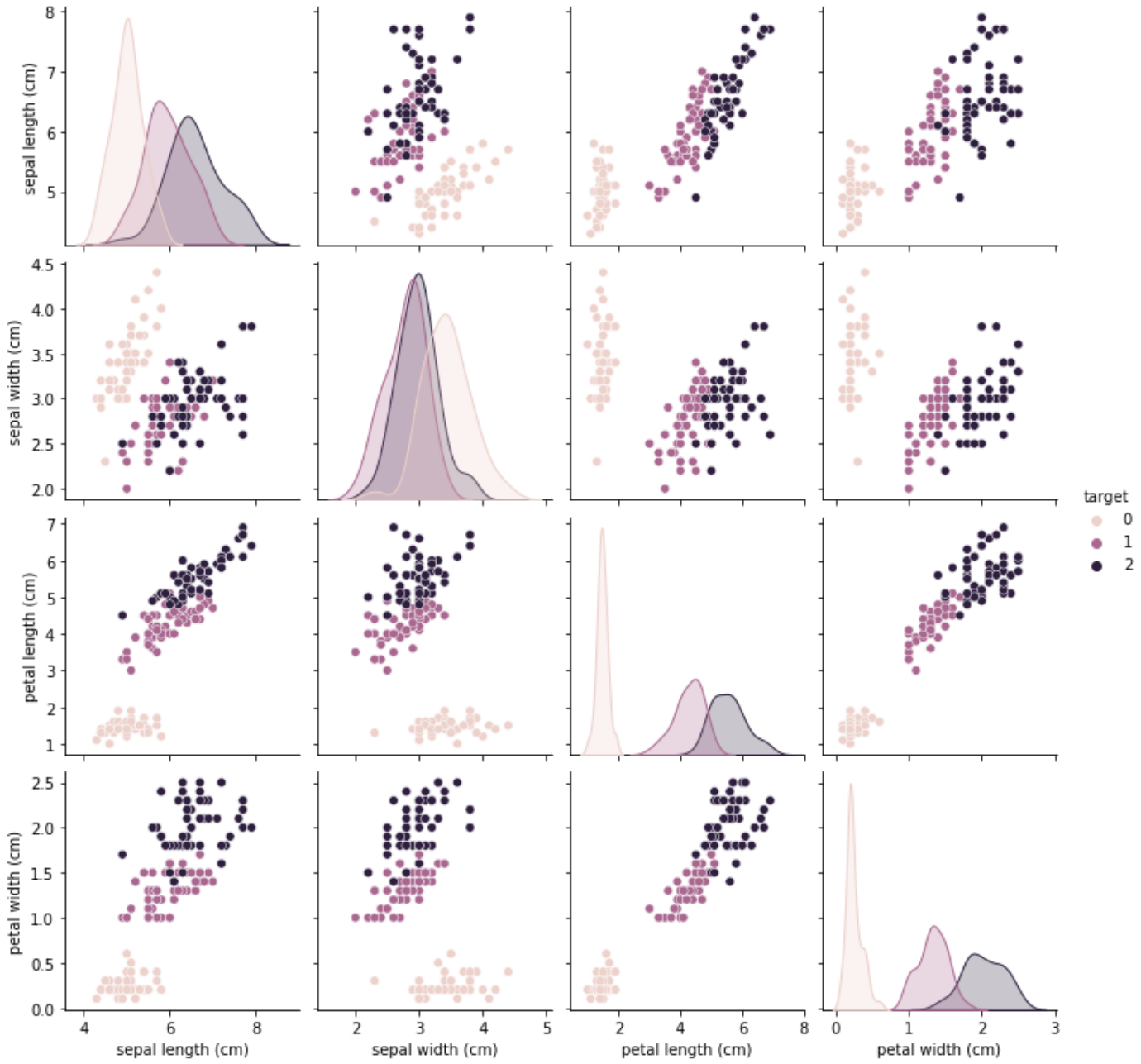


Как мы наблюдаем из графиков выше, распределение длины и ширины чашелистиков ирисов близка к нормальной.

Парные диаграммы

In [26]:
```
sns.pairplot(data, hue='target')
```

Out[26]:
```
<seaborn.axisgrid.PairGrid at 0x268143775b0>
```



Парные диаграммы показавывают нам, что ирисы сорта setosa отличимы от двух других практически по любой паре параметров.

Информация о корреляции признаков

In [27]:
```
mask = np.zeros_like(data.corr(), dtype=bool)
mask[np.tril_indices_from(mask)] = True
sns.heatmap(data.corr(), mask=mask, annot=True, fmt='.3f', cmap='RdPu')
```

Out[27]:
```
<AxesSubplot:>
```

Как мы видим из тепловой карты, длина и ширина лепестка сильно коррелируют с целевым признаком.

Посмотрим на корреляционные матрицы, построенные разными методами.

In [28]:
```
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', cmap='RdPu')
sns.heatmap(data.corr(method='kendall'), ax=ax[1], annot=True, fmt='.2f', cmap='RdPu')
sns.heatmap(data.corr(method='spearman'), ax=ax[2], annot=True, fmt='.2f', cmap='RdPu')
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')
```