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 Системы обработки информации и управления

Отчёт по лабораторной работе №1

По дисциплине:
«Технологии машинного обучения»

Выполнила:

Студентка группы ИУ5-63Б

Ваксина И. Р.

Проверил:

Гапанюк Ю. Е.

Москва, 2021

Задание

- Выбрать набор данных (желательно без пропусков в данных и небольшого размера)
- Создать ноутбук, который содержит следующие разделы:
  1. Текстовое описание набора данных
  2. Основные характеристики набора данных
  3. Визуальное исследование набора данных
  4. Информацию о корреляции признаков
- Сформировать отчет и разместить его в своем репозитории на github.

Ноутбук с текстом программы и экранными формами с примерами выполнения

Разведочный анализ данных. Исследование и визуализация данных.

1) Текстовое описание набора данных

```
В качестве набора данных мы будем использовать набор данных о характеристиках
ирисов в соответствии с их классификацией - https://scikit-
learn.org/stable/modules/generated/sklearn.datasets.load_iris.html.

Ирисы Фишера состоят из данных о 150 экземплярах ириса, по 50 экземпляров из
трех видов - Ирис щетинистый (Iris Setosa), Ирис виргинский (Iris Virginica) и
Ирис разноцветный (Iris Versicolour). Для каждого экземпляра измерялись 4
характеристики, которые выступают в качестве колонок-атрибутов:

    1) sepal length (cm) - длина наружной доли околоцветника (в сантиметрах)
    2) sepal width (cm) - ширина наружной доли околоцветника (в сантиметрах)
    3) petal length (cm) - длина внутренней доли околоцветника (в сантиметрах)
    4) petal width (cm) - ширина внутренней доли околоцветника (в сантиметрах)
    5) class - вид ириса
```

Импорт библиотек

In [203]:
```
import numpy as np
import pandas as pd
from pandas import DataFrame
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
from sklearn.datasets import load_iris
%matplotlib inline
```

Загрузка данных

In [204]:
```
iris = load_iris()
data = pd.DataFrame(iris.data, columns=iris.feature_names)
data['class'] = iris.target
```

2) Основные характеристики датасета

In [205]:
```
# Первые 5 строк датасета
data.head()
```

Out[205]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	class
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 [206]:
```
# Размер датасета - 150 строк, 5 колонок
data.shape
```

Out[206]:
```
(150, 5)
```

In [207]:
```
total_count = data.shape[0]
print('Всего строк: {}'.format(total_count))
```

```
Всего строк: 150
```

In [208]:
```
# Список колонок
data.columns
```

Out[208]:
```
Index(['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)',
       'petal width (cm)', 'class'],
      dtype='object')
```

In [209]:
```
# Список колонок с типами данных
data.dtypes
```

Out[209]:
```
sepal length (cm)    float64
sepal width (cm)     float64
petal length (cm)    float64
petal width (cm)     float64
class                  int32
dtype: object
```

In [210]:
```
# 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))
```

```
sepal length (cm) - 0
sepal width (cm) - 0
petal length (cm) - 0
petal width (cm) - 0
class - 0
```

In [211]:
```
# Основные статистические характеристки набора данных
data.describe()
```

Out[211]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	class
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 [212]:
```
# Определим уникальные значения для целевого признака
data['class'].unique()
```

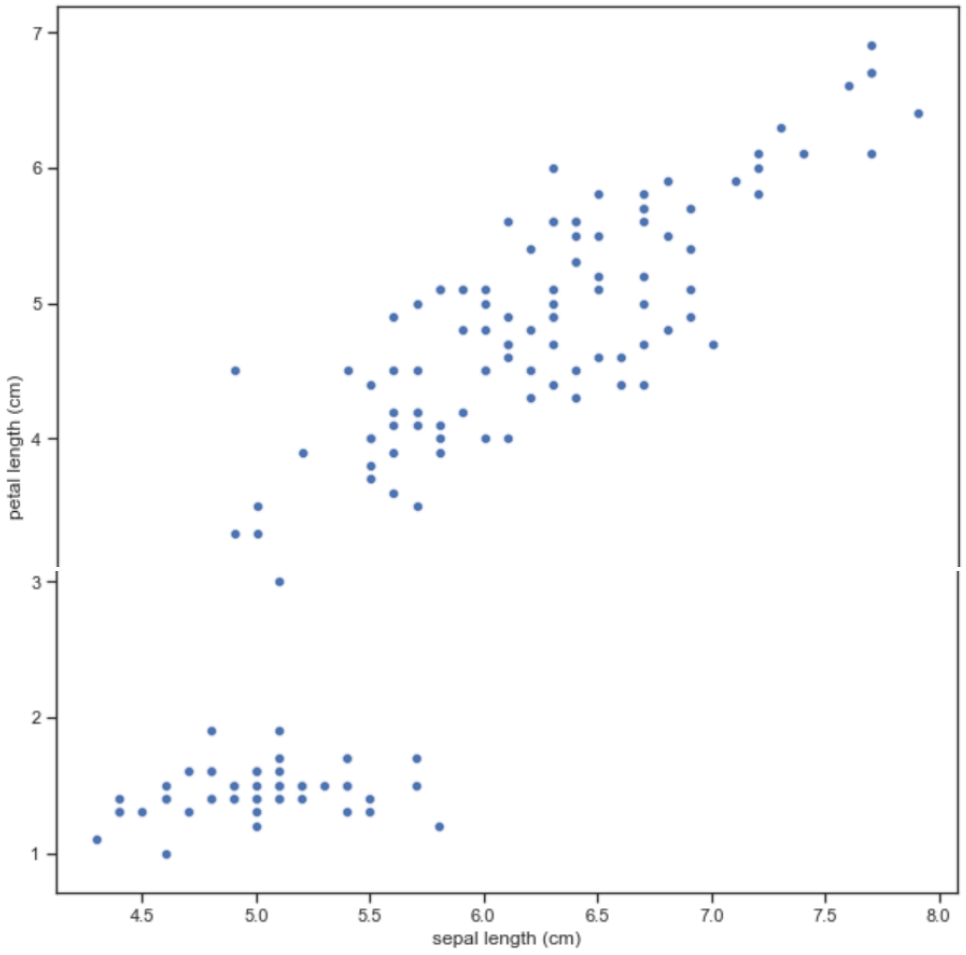
Out[212]:
```
array([0, 1, 2])
```

3) Визуальное исследование датасета

Диаграмма рассеяния

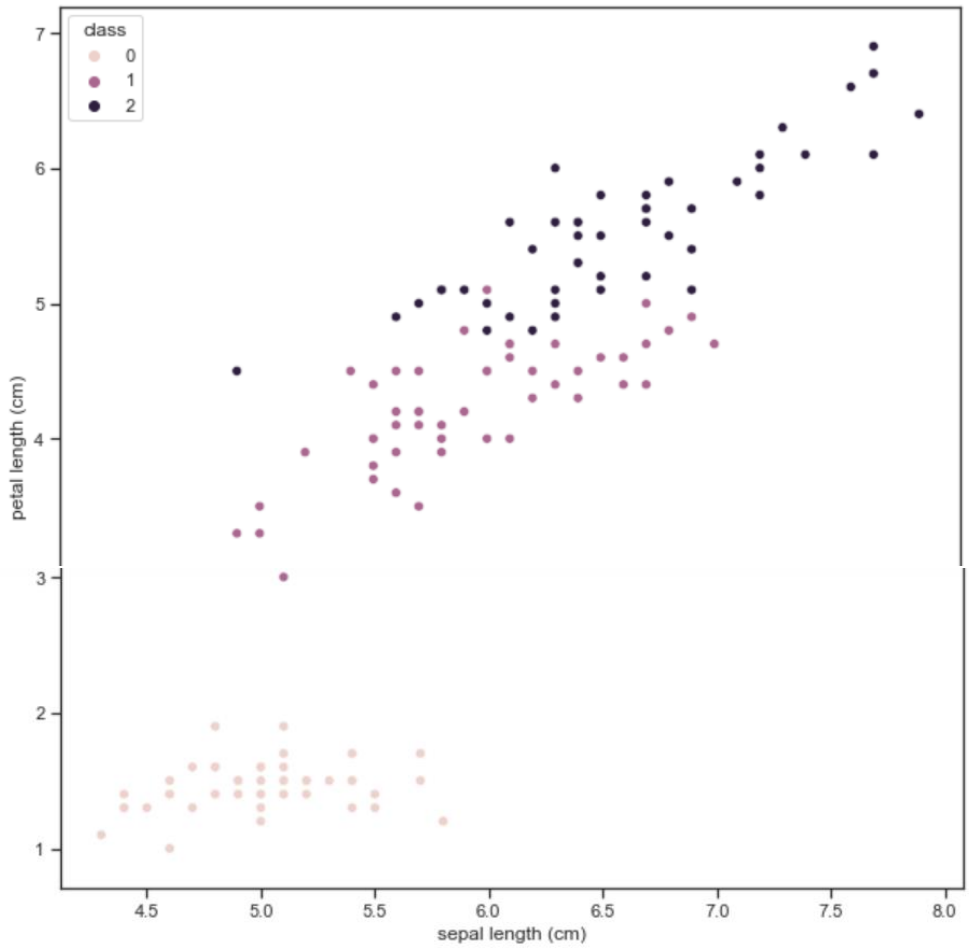
In [213]:
```
# Диаграмма зависимости длины внутренней доли околоцветника от длины наружной до
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='sepal length (cm)', y='petal length (cm)', data=data)
```

Out[213]:
```
<AxesSubplot:xlabel='sepal length (cm)', ylabel='petal length (cm)'>
```



In [214]:
```
# Диаграмма с целевым признаком
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='sepal length (cm)', y='petal length (cm)', data=data,
```

Out[214]:
```
<AxesSubplot:xlabel='sepal length (cm)', ylabel='petal length (cm)'>
```



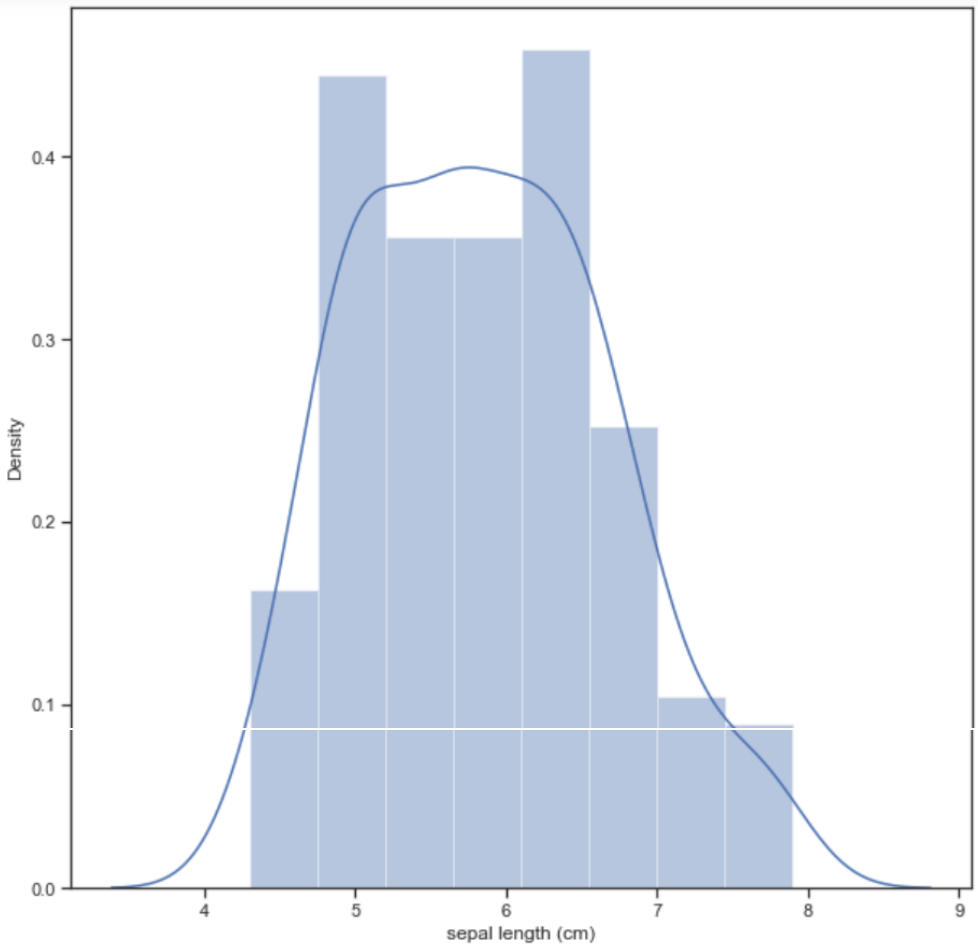
Класс Iris Setosa (class=0) линейно-разделим от двух остальных.

Гистограмма

In [215]:
```
fig, ax = plt.subplots(figsize=(10,10))
sns.distplot(data['sepal length (cm)'])
```

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

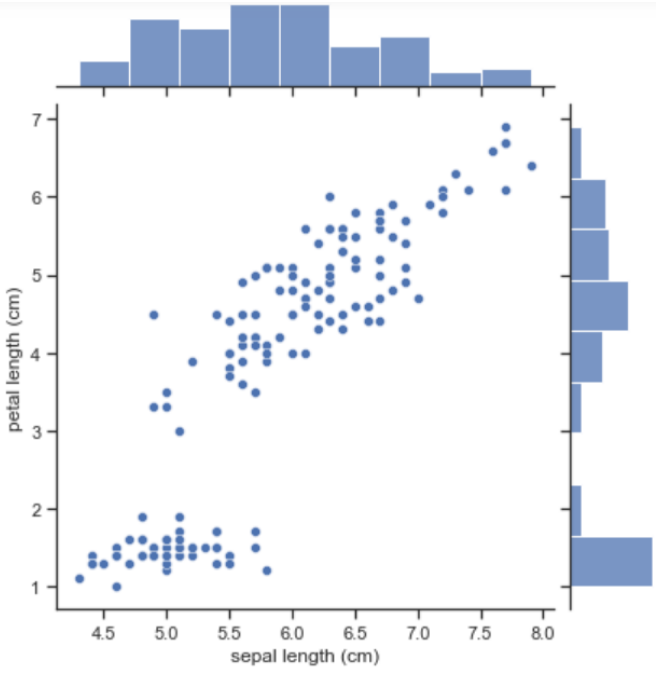
Out[215]:
```
<AxesSubplot:xlabel='sepal length (cm)', ylabel='Density'>
```



Jointplot

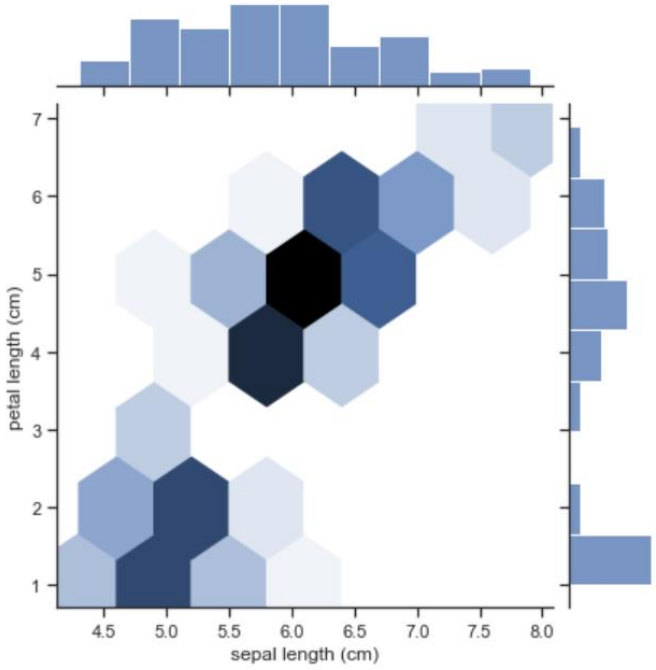
In [216]:
```
sns.jointplot(x='sepal length (cm)', y='petal length (cm)', data=data)
```

Out[216]:
```
<seaborn.axisgrid.JointGrid at 0x16cce699880>
```



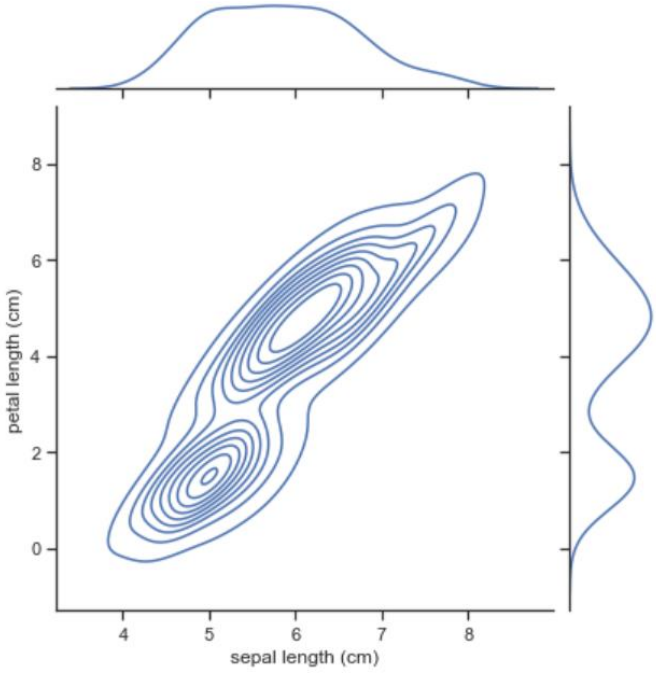
In [217]:
```
sns.jointplot(x='sepal length (cm)', y='petal length (cm)', data=data, kind="hex
```

Out[217]: <seaborn.axisgrid.JointGrid at 0x16cce738640>



In [218]:
```
sns.jointplot(x='sepal length (cm)', y='petal length (cm)', data=data, kind="kde
```

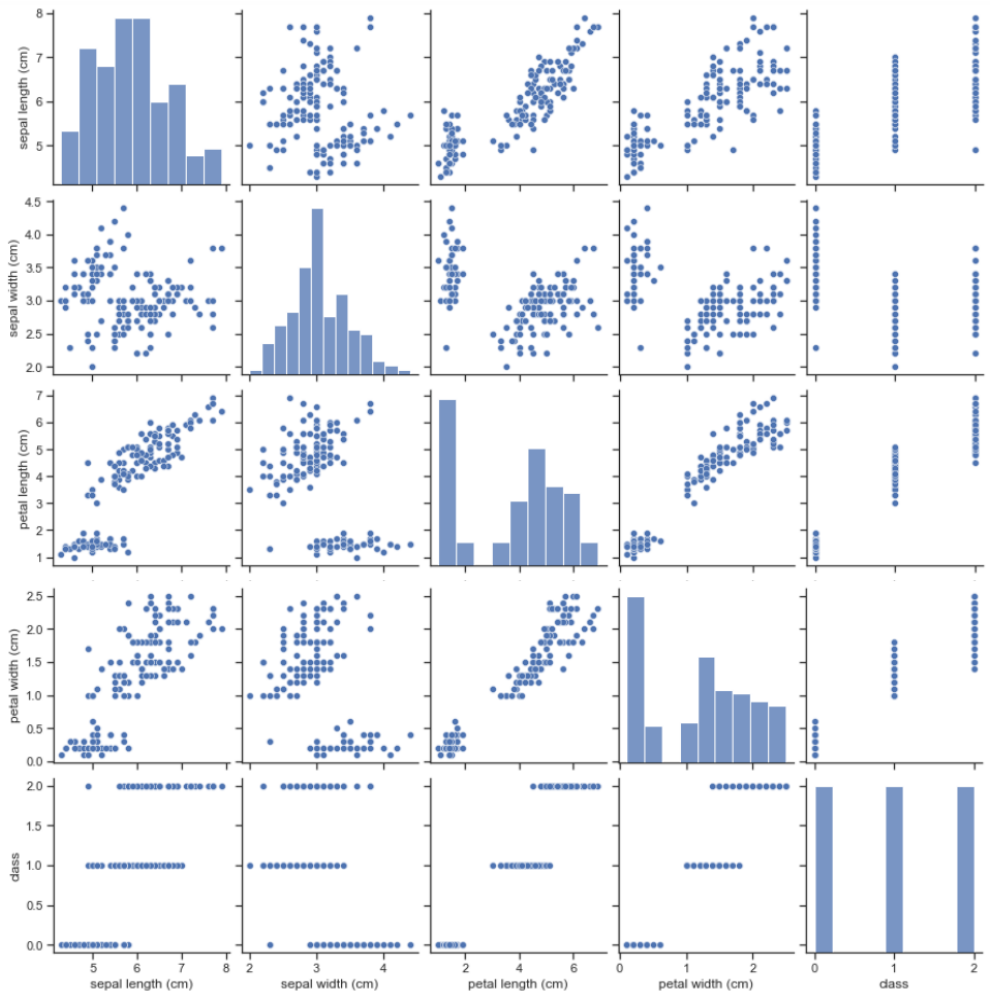
Out[218]: <seaborn.axisgrid.JointGrid at 0x16cce8b7df0>



"Парные диаграммы"

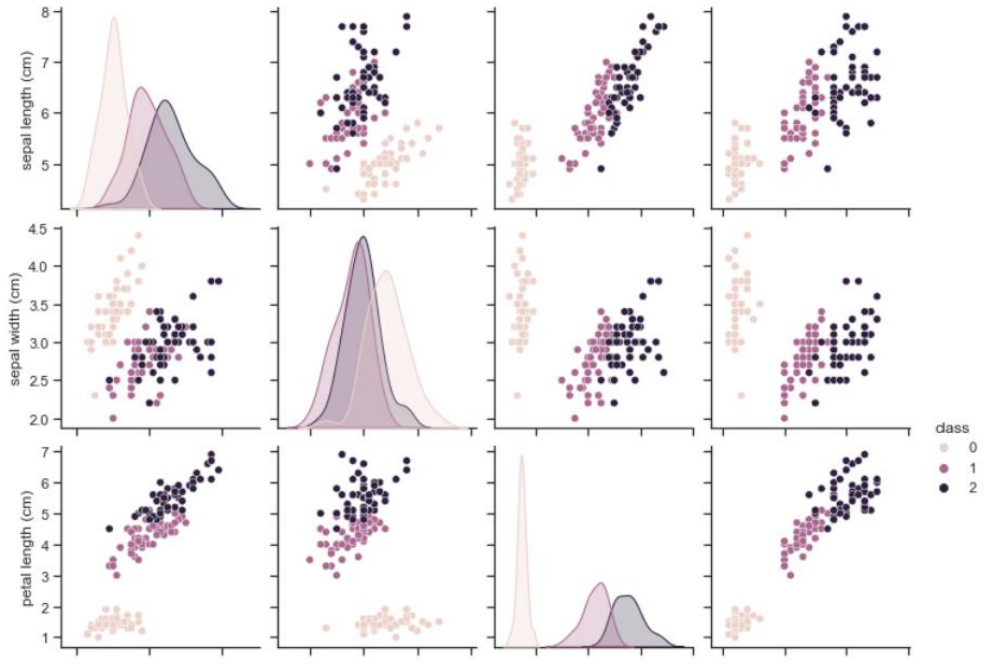
In [219]:
```
sns.pairplot(data)
```

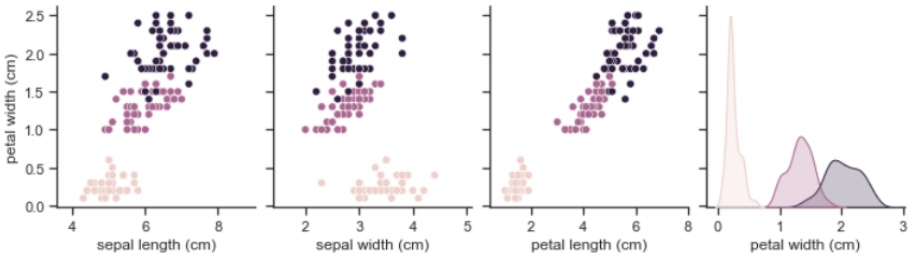
Out[219]: <seaborn.axisgrid.PairGrid at 0x16ccea3fe80>



In [220]:
```
sns.pairplot(data, hue="class")
```

Out[220]:
```
<seaborn.axisgrid.PairGrid at 0x16cd0385250>
```

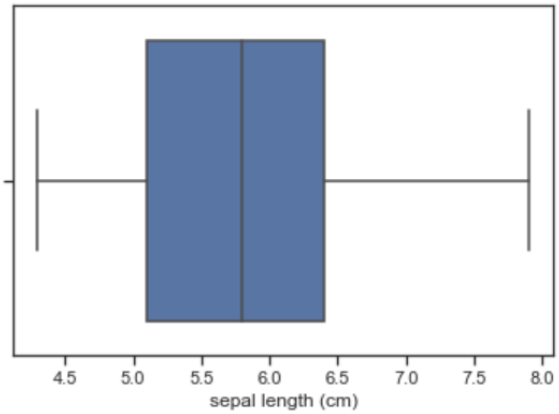




Ящик с усами

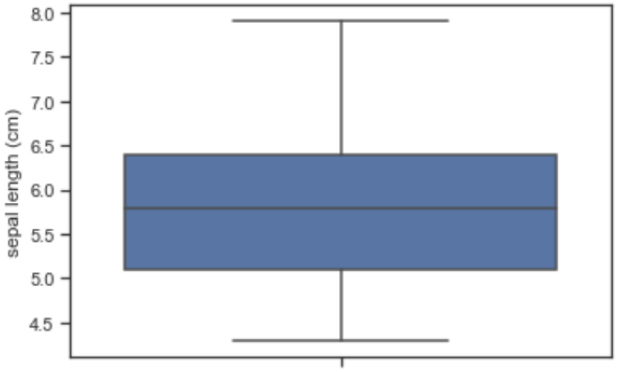
In [221]:
```
# По горизонтали
sns.boxplot(x=data['sepal length (cm)'])
```

Out[221]: <AxesSubplot:xlabel='sepal length (cm)'>



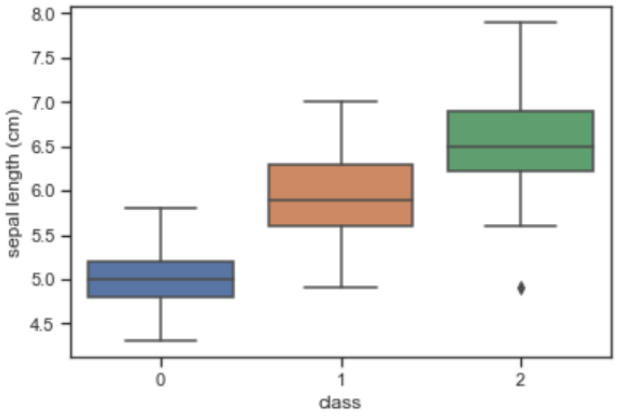
In [222]:
```
# По вертикали
sns.boxplot(y=data['sepal length (cm)'])
```

Out[222]: <AxesSubplot:ylabel='sepal length (cm)'>



In [223]:
```
# Распределение параметра sepal length (cm) сгруппированные по class.
sns.boxplot(x='class', y='sepal length (cm)', data=data)
```

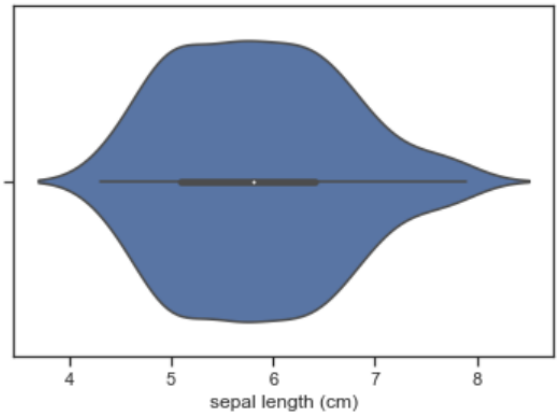
Out[223]: <AxesSubplot:xlabel='class', ylabel='sepal length (cm)'>



Violin plot

In [224]:
```
sns.violinplot(x=data['sepal length (cm)'])
```

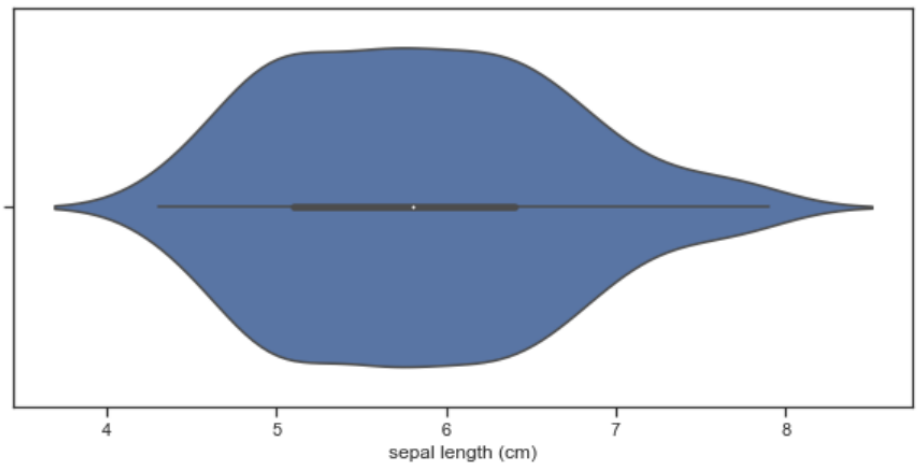
Out[224]:
```
<AxesSubplot:xlabel='sepal length (cm)'>
```

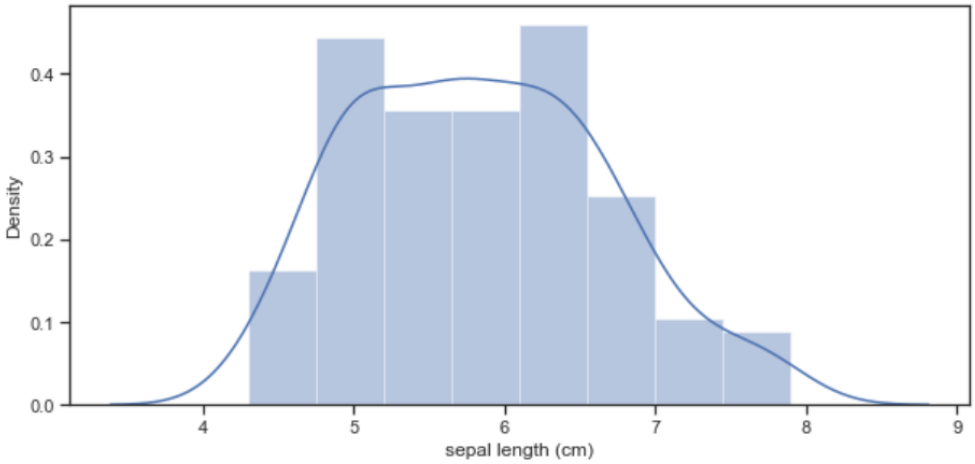


In [225]:
```
fig, ax = plt.subplots(2, 1, figsize=(10,10))
sns.violinplot(ax=ax[0], x=data['sepal length (cm)'])
sns.distplot(data['sepal length (cm)'], ax=ax[1])
```

```
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)
```

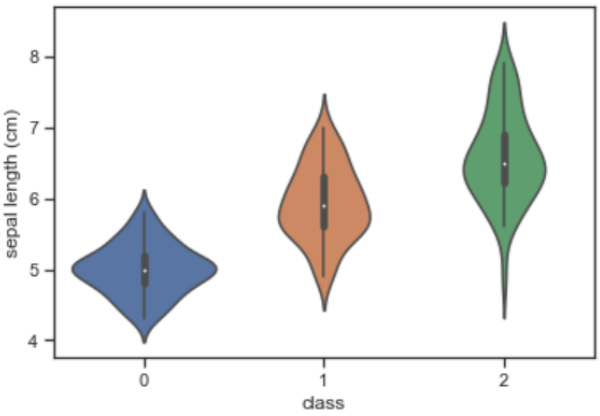
Out[225]:
```
<AxesSubplot:xlabel='sepal length (cm)', ylabel='Density'>
```





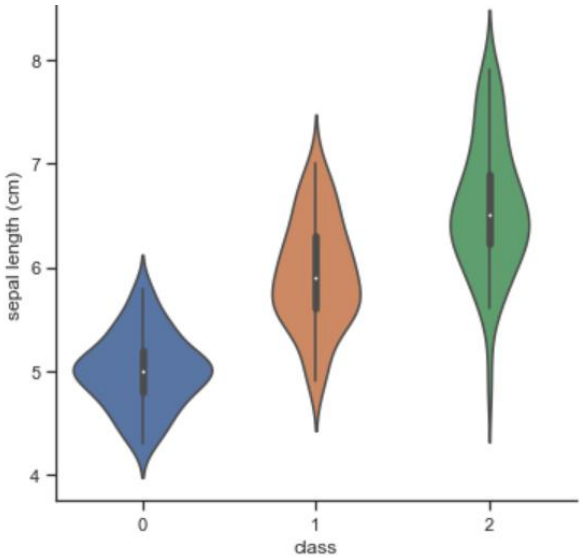
In [226]:
```
# Распределение параметра Humidity сгруппированные по Occupancy.
sns.violinplot(x='class', y='sepal length (cm)', data=data)
```

Out[226]:
```
<AxesSubplot:xlabel='class', ylabel='sepal length (cm)'>
```



In [227]:
```
sns.catplot(y='sepal length (cm)', x='class', data=data, kind="violin", split=Tr
```

Out[227]:
```
<seaborn.axisgrid.FacetGrid at 0x16cce23d400>
```



4) Информация о корреляции признаков

In [228]:
```
data.corr()
```

Out[228]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	class
sepal length (cm)	1.000000	-0.117570	0.871754	0.817941	0.782561
sepal width (cm)	-0.117570	1.000000	-0.428440	-0.366126	-0.426658
petal length (cm)	0.871754	-0.428440	1.000000	0.962865	0.949035
petal width (cm)	0.817941	-0.366126	0.962865	1.000000	0.956547
class	0.782561	-0.426658	0.949035	0.956547	1.000000

```
На основе корреляционной матрицы можно сделать следующие выводы:

Целевой признак наиболее сильно коррелирует с шириной внутренней доли
околоцветника (0.96), длиной внутренней доли околоцветника (0.95) и длиной
наружной доли околоцветника (0.78). Эти признаки обязательно следует оставить в
модели.
Целевой признак слабо коррелирует с шириной наружной доли околоцветника (0.43).
Скорее всего, этот признак стоит исключить из модели, возможно, он только
ухудшит качество модели.
Длина и ширина внутренней доли околоцветника очень сильно коррелируют между
собой (0.96). Поэтому из этих признаков в модели можно оставлять только один.
Также можно сделать вывод, что выбирая из признаков длина и ширина внутренней
доли околоцветника лучше выбрать ширину внутренней доли околоцветника, потому
что она сильнее коррелирована с целевым признаком.
Длина наружной доли околоцветника сильно коррелирует с длиной и шириной
внутренней доли околоцветника (0.87 и 0.82 соответственно), но она слабее
коррелирована с целевым признаком, поэтому ее можно исключить из модели.
```

In [229]:
```
data.corr(method='pearson')
```

Out[229]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	class
sepal length (cm)	1.000000	-0.117570	0.871754	0.817941	0.782561
sepal width (cm)	-0.117570	1.000000	-0.428440	-0.366126	-0.426658
petal length (cm)	0.871754	-0.428440	1.000000	0.962865	0.949035
petal width (cm)	0.817941	-0.366126	0.962865	1.000000	0.956547
class	0.782561	-0.426658	0.949035	0.956547	1.000000

In [230]:
```
data.corr(method='kendall')
```

Out[230]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	class
sepal length (cm)	1.000000	-0.076997	0.718516	0.655309	0.670444
sepal width (cm)	-0.076997	1.000000	-0.185994	-0.157126	-0.337614
petal length (cm)	0.718516	-0.185994	1.000000	0.806891	0.822911
petal width (cm)	0.655309	-0.157126	0.806891	1.000000	0.839687
class	0.670444	-0.337614	0.822911	0.839687	1.000000

In [231]:
```
data.corr(method='spearman')
```

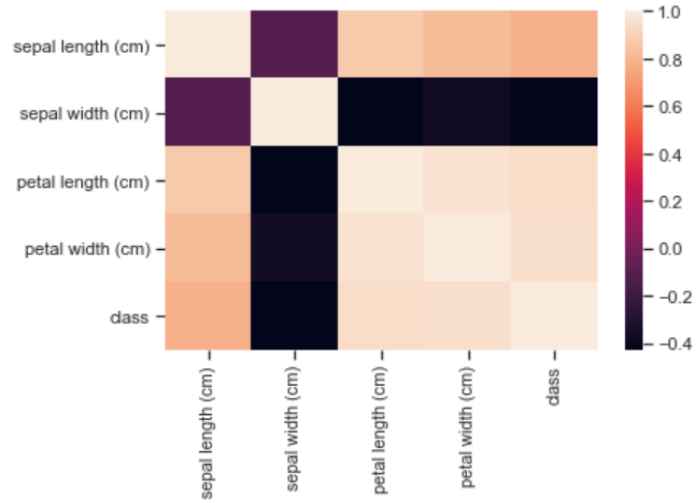
Out[231]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	class
sepal length (cm)	1.000000	-0.166778	0.881898	0.834289	0.798078
sepal width (cm)	-0.166778	1.000000	-0.309635	-0.289032	-0.440290
petal length (cm)	0.881898	-0.309635	1.000000	0.937667	0.935431
petal width (cm)	0.834289	-0.289032	0.937667	1.000000	0.938179
class	0.798078	-0.440290	0.935431	0.938179	1.000000

Тепловая карта

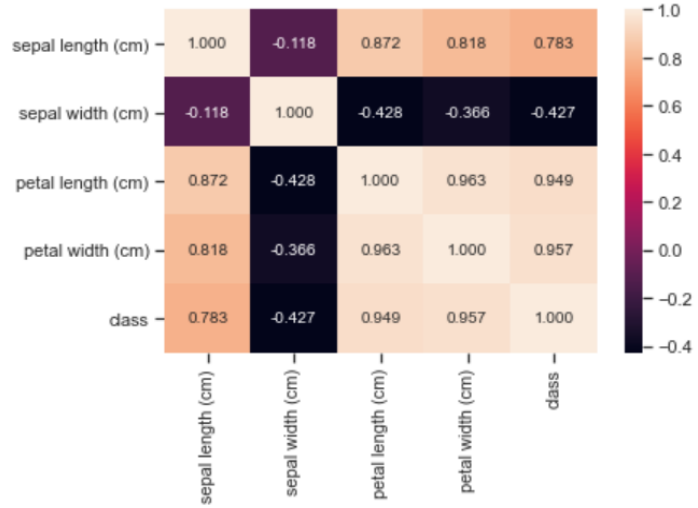
In [232]:
```
sns.heatmap(data.corr())
```

Out[232]: <AxesSubplot:>



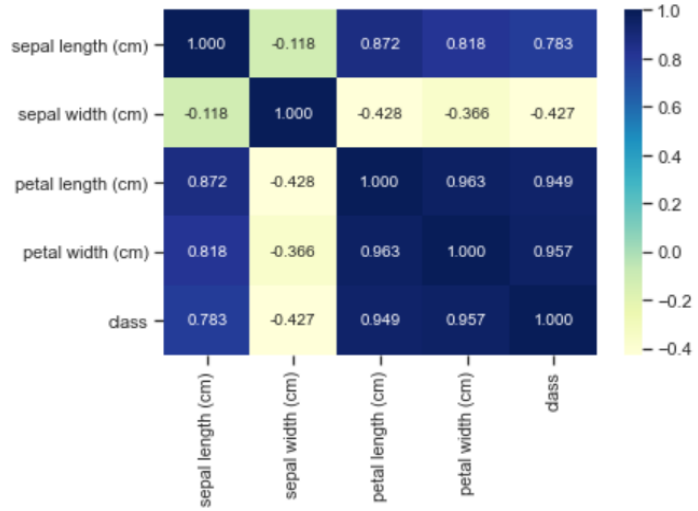
In [233]:
```
# Вывод значений в ячейках
sns.heatmap(data.corr(), annot=True, fmt='.3f')
```

Out[233]: <AxesSubplot:>



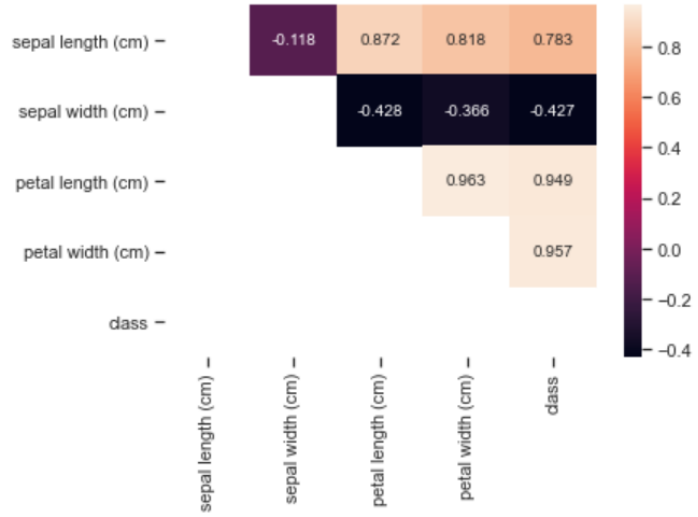
In [234]:
```
# Изменение цветовой гаммы
sns.heatmap(data.corr(), cmap='YlGnBu', annot=True, fmt='.3f')
```

Out[234]: <AxesSubplot:>



In [235]:
```
# 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')
```

Out[235]: <AxesSubplot:>



In [236]:
```
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')
```