Задание: 1) Выбрать набор данных (датасет). Вы можете найти список свободно распространяемых датасетов здесь. 2) Для лабораторных работ не рекомендуется выбирать датасеты большого размера. 3) Создать ноутбук, который содержит следующие разделы: 4) Текстовое описание выбранного Вами набора данных. 5) Основные характеристики датасета. 6) Визуальное исследование датасета. 7) Информация о корреляции признаков. 8) Сформировать отчет и разместить его в своем репозитории на github.

In [49]:

```
import numpy as np #библиотека для работы с многомерными массивами данных
 и математическими операциями над ними
import pandas as pd #библиотека для анализа и обработки данных
from sklearn.datasets import load_iris #берём датасет
import matplotlib.pyplot as plt #простое рисование графиков
import seaborn as sns #удобные дефолтные настройки графиков из matpotlib

%matplotlib inline
#для сохранения в ноутбуке вывода моих графиков
```

In [50]:

```
# Загрузите данные, если их еще нет на диске, и загрузите их в виде массив
ов
df = load_iris()

# Смотрим названия целей (всего 3)
for name in df.target_names:
    print(name)

#sepal - чашелистик
#petal - лепесток
```

```
setosa
versicolor
virginica
```

Создаем датафрейм на основе необходимых полей. Задаём поле Данные и Колонки. Выведем на экран, чтобы посмотреть на

получившуюся таблицу, которую будем анализировать дальше.

In [51]:

```
df = pd.DataFrame(data = np.c_[df['data'], df['target']], columns = df['fe
ature_names'] + ['target'])
df.head()
```

Out[51]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0.0
1	4.9	3.0	1.4	0.2	0.0
2	4.7	3.2	1.3	0.2	0.0
3	4.6	3.1	1.5	0.2	0.0
4	5.0	3.6	1.4	0.2	0.0

Посмотрим на типы данных. Получили, что все данные - представитали типа float64, а пустых ячеек нет ни в одной строке. Это упрощает работу, поскольку не нужно обрабатывать уникальные поля.

In [52]:

```
df.dtypes
```

Out[52]:

```
sepal length (cm)    float64
sepal width (cm)     float64
petal length (cm)    float64
petal width (cm)     float64
target               float64
dtype: object
```

In [53]:

```
df[df.isnull().any(axis=1)]
```

Out[53]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target

In [54]:

```
df.describe() #Кол-во; Среднее; и т.д. (кстати, что?)
```

Out[54]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [55]:

```
#sns.pairplot(data=df)
```

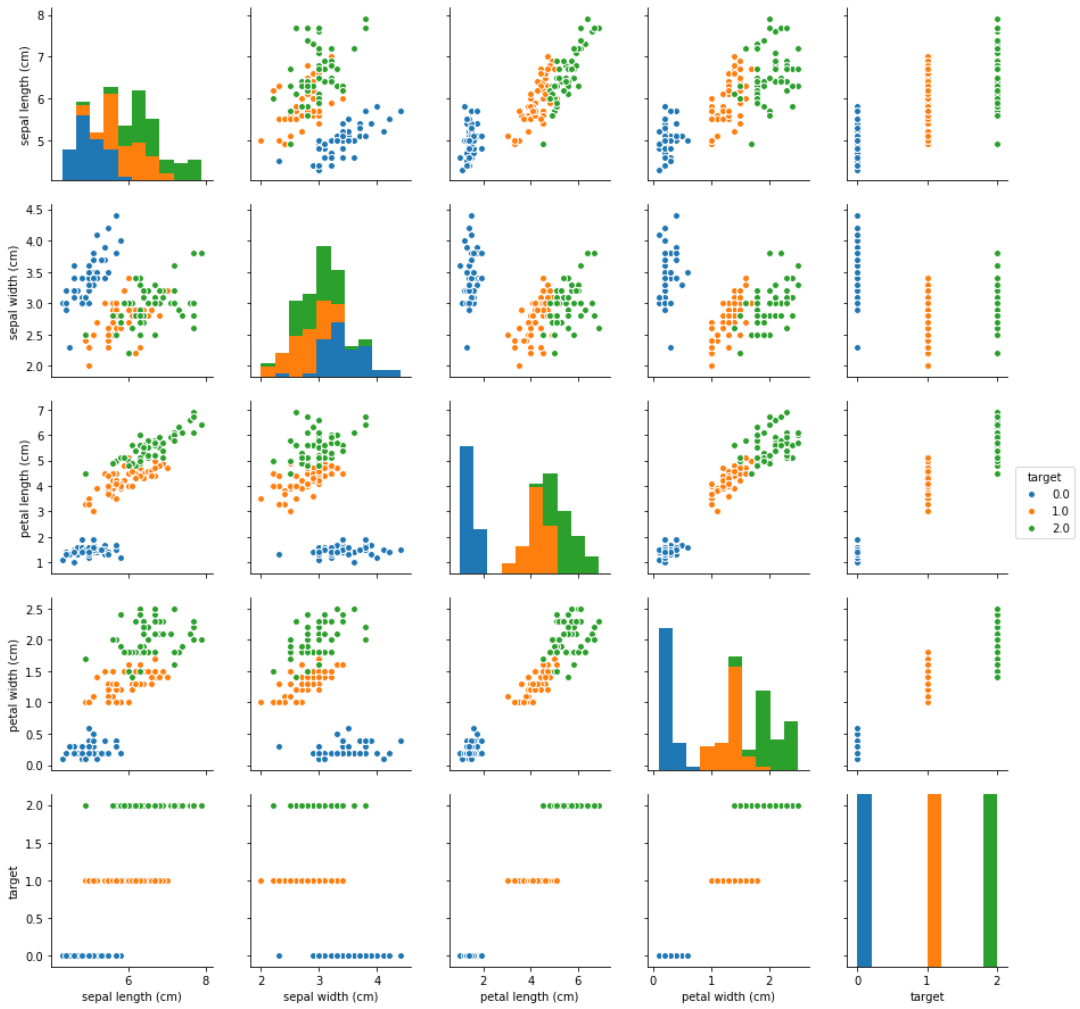
Наша цель содержит 3 класса, поэтому удобно использовать параметр hue, который покрасит наши графики по нашим target-ам.

In [56]:

```
sns.pairplot(data=df, hue='target')
```

Out[56]:

```
<seaborn.axisgrid.PairGrid at 0x2563f5df2e8>
```



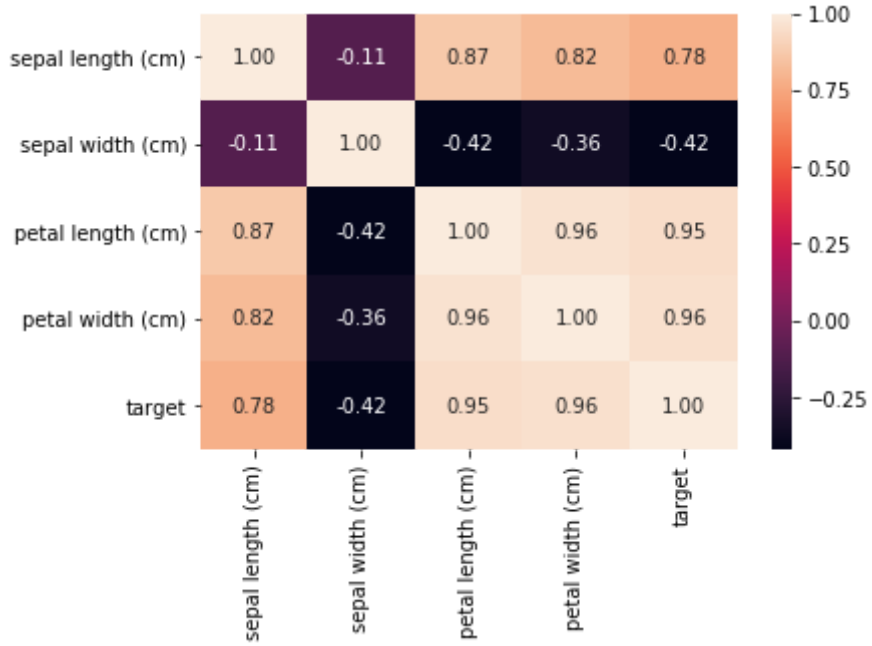
А теперь осталась только Корреляция:

In [57]:

```
sns.heatmap(df.corr(method='pearson'), annot=True, fmt='.2f')
```

Out[57]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x256411929e8>
```



In [58]:

```
#sns.jointplot(x="sepal length (cm)", y="target", data=df, kind="kde")
```

In [59]:

```
#sns.jointplot(x="petal length (cm)", y="target", data=df, kind="kde")
```

In [60]:

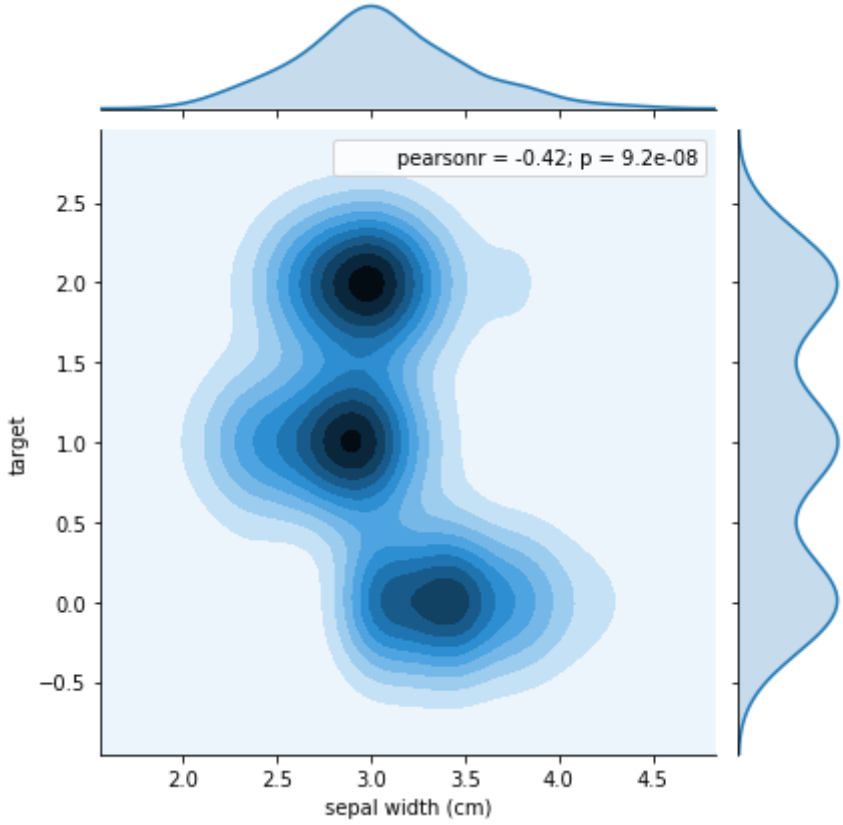
```
#sns.jointplot(x="petal width (cm)", y="target", data=df, kind="kde")
```

In [61]:

```
sns.jointplot(x="sepal width (cm)", y="target", data=df, kind="kde")
# видим, что плохо коррелируется для 1 и 2 target
```

Out[61]:

```
<seaborn.axisgrid.JointGrid at 0x25641a77eb8>
```



In [62]:

```
df = df.drop(['sepal width (cm)'], axis=1) # удалим мешающую фичу
sns.heatmap(df.corr(method='pearson'), annot=True, fmt='.2f')
```

Out[62]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25641b7bf28>
```