CMPT 423/820 Question 7

- Your Name
- Your student number
- Your NSID

In this question, we'll get do more with Pandas DataFrames, methods, indexing. We'll also introduce the Seaborn package.

In [1]:
```
# this is the conventional import
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Task 1

Use Pandas function `read_csv()` to load the `iris.cvs` dataset.

In [2]:
```
dataframe = pd.read_csv('iris.csv',
                        header=None,
                        names=['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm','Species'],
                        index_col=False)
```

Task 2

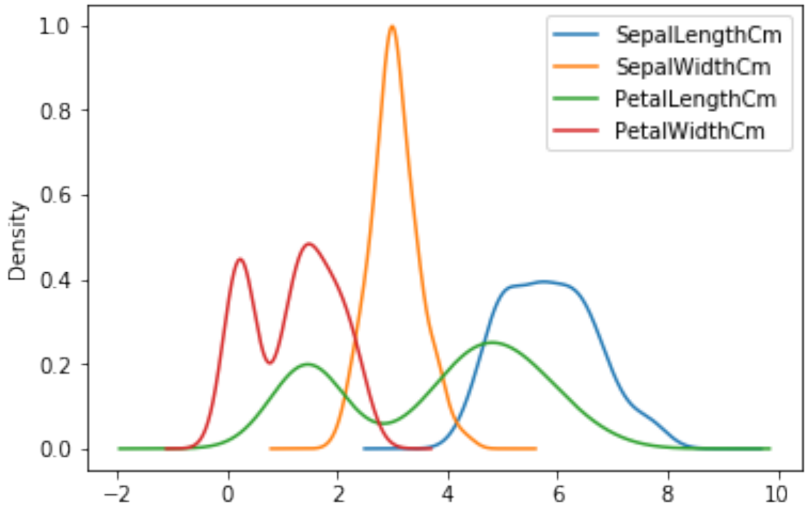
There's a method called `density()` that uses MatPlotLib to create a slightly different kind of histogram (recall Question 6). Behind the scenes, it's applying a statistical method called *density estimation* to create the plot; density estimation is something we will see (and have seen already!) in the course. Pandas calls functions in MatPlotLib to create the plot; you could use MatPlotLib to enhance the plot, adding a title, or axis labels, etc.

In the cell below, use the `density()` method to display a plot for the dataframe.

Hint: Jupyter is aware of Pandas and MatPlotLib, and by default, Jupyter can display any plot if the command to create it is the last thing in a cell. But it is common to call MatPlotLib function `plt.show()` if you want to force a plot to be shown.

In [3]:

```
dataframe.plot.density()
plt.show()
```



Notes:

- Features that have more than one peak suggest that something is going on to cause the different peaks.
- We cannot discount the features with only one peak. The scale of the data might be hiding something interesting.
- These visualization tools are useful to help us get intuitions about the data, but not to draw concllusions about the data.

Task 3 ¶

The density plots from Task 2 are visualizations of the distribution of the four columns of data. In effect, they are visualizations of four marginal probability distributions $P(X_i|Y)$, where the Xi is the column feature, and Y is the label.

1. Using Pandas create 3 new DataFrames, by separating the data into three subsets:

   - all the rows for `versicolor`
   - all the rows for `setosa`
   - all the rows for `virginica`

   Do this using Boolean array indexing; this technique works in Numpy, and Pandas.
2. Then plot the density for each subset individually, using `density()` as above.

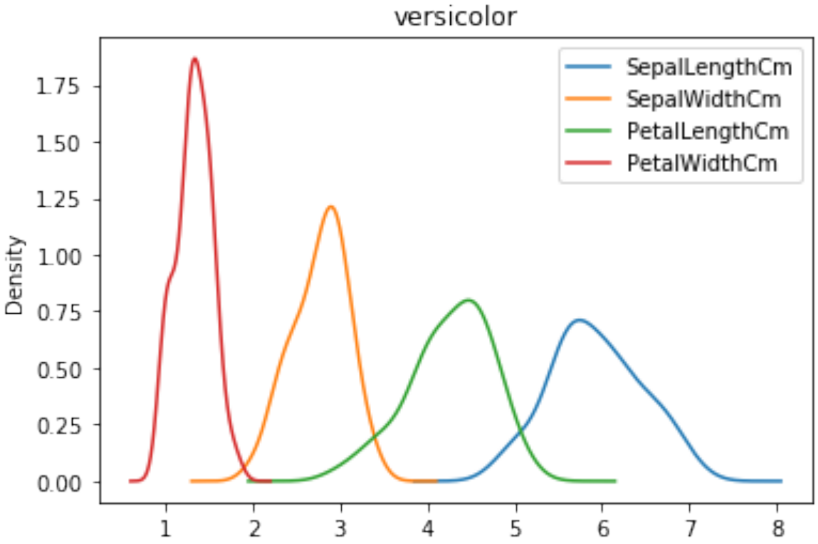
Hint:

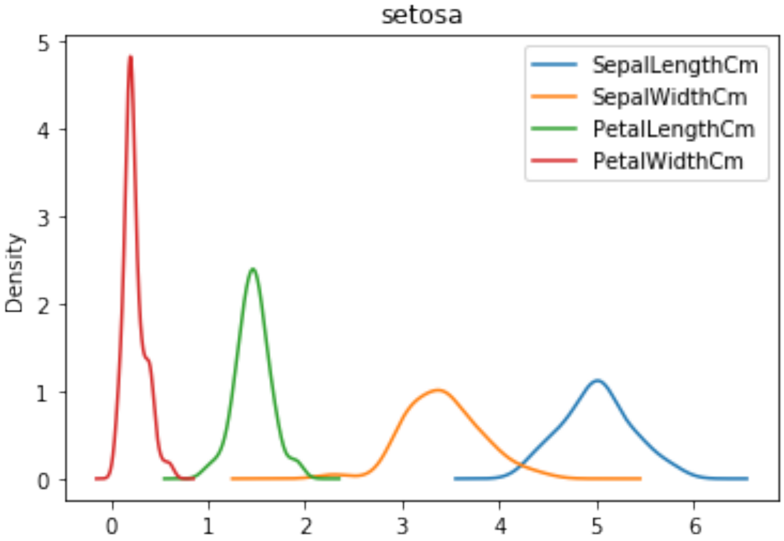
```
sub1 = < expression to select all the rows whose label is 'versicolor' >
...
```

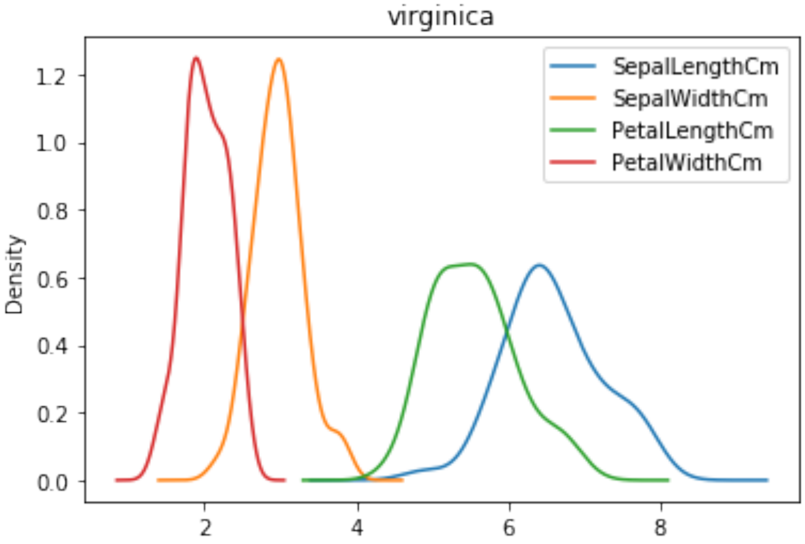
Use Boolean array indexing to select the rows you want in your new dataframe.

In [4]:

```
sub1 = dataframe[dataframe.Species == 'versicolor']
sub1.plot.density()
plt.title('versicolor')

sub2 = dataframe[dataframe.Species == 'setosa']
sub2.plot.density()
plt.title('setosa')

sub3 = dataframe[dataframe.Species == 'virginica']
sub3.plot.density()
plt.title('virginica')

plt.show()
```


versicolor
Density
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
SepalLengthCm
SepalWidthCm
PetalLengthCm
PetalWidthCm


setosa
Density
5
4
3
2
1
0
0
1
2
3
4
5
6
SepalLengthCm
SepalWidthCm
PetalLengthCm
PetalWidthCm


virginica
Density
1.2
1.0
0.8
0.6
0.4
0.2
0.0
2
4
6
8
SepalLengthCm
SepalWidthCm
PetalLengthCm
PetalWidthCm

Task 4

These densities are visualizations for conditional probability distributions, conditioned on the label (i.e., $P(X_i|setosa)$ for all i). We can see that one of these species is really quite distinct from the others (consider the order of the 4 peaks).

In the above plots, the four numerical columns are visualized for each separate species. We see columns 1-4 for each species.

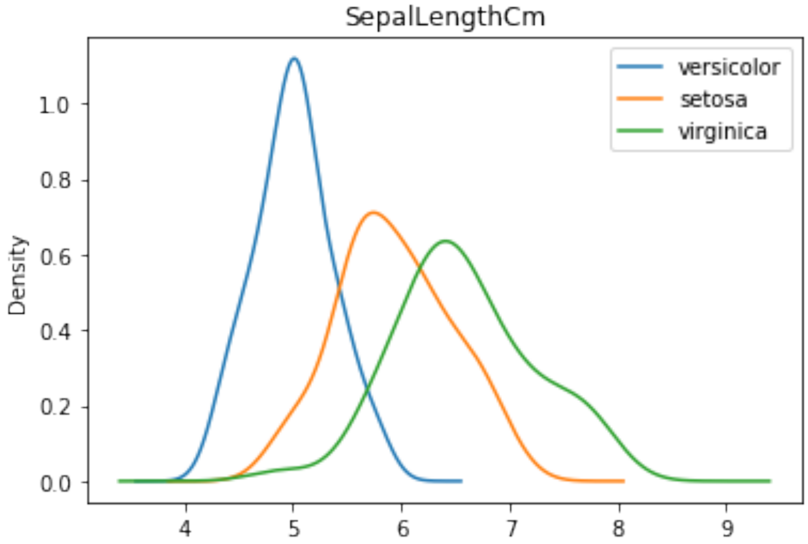
It might be more useful to plot the density for X_1 for all three species in a single plot, to compare more easily the differences in the distribution of that column for a single species.

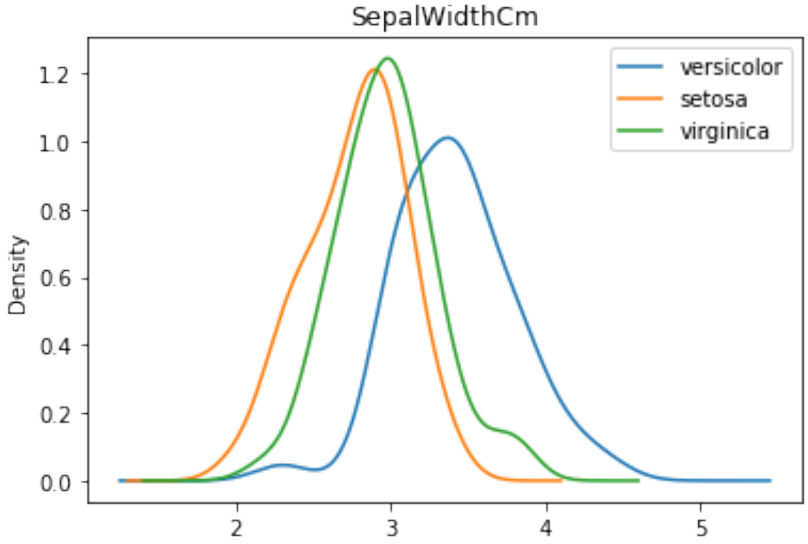
In the cell below, plot the density of X_1 for all three labels. Then do similar plots for the other 3 numerical columns.

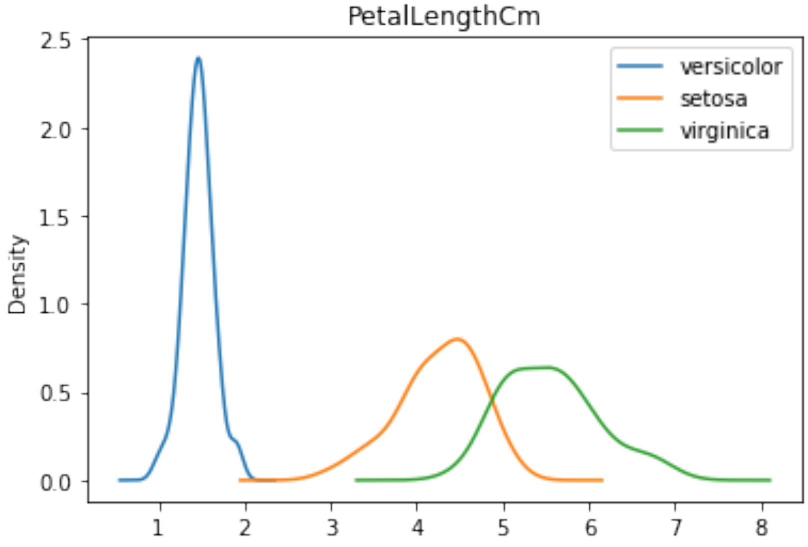
Hint: There are many ways you could do this. This exercise is to familiarize you with Pandas, and there's no single answer that the marker is looking for. Remember to use Boolean array indexing! One useful Pandas method is called `concat()` . Also, you'll want to create dataframes with the data you want to visualize, using `density()` .

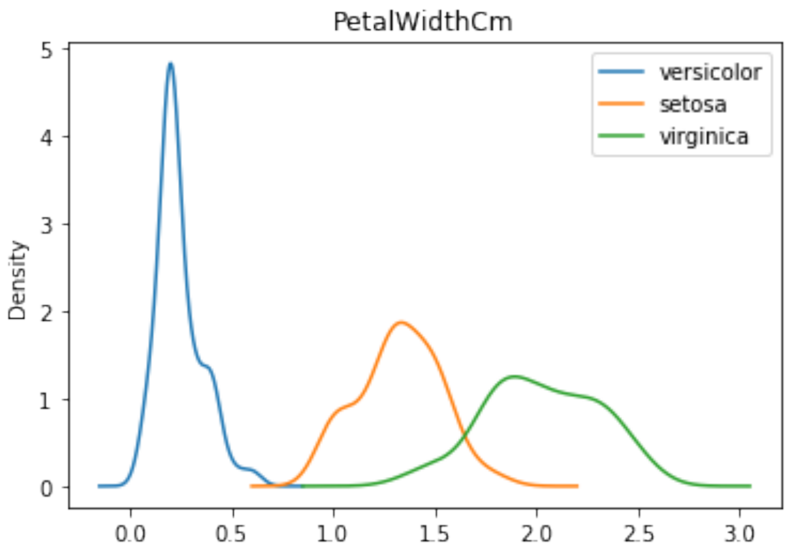
In [19]:

```
names=['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm']

column = names[0]
sub1 = dataframe[[column,'Species']]
sub11 = sub1[sub1.Species == 'versicolor'][[column]]
sub12 = sub1[sub1.Species == 'setosa'][[column]]
sub13 = sub1[sub1.Species == 'virginica'][[column]]

sub2 = pd.concat([sub12, sub11, sub13], axis=1)
sub2.columns = ['versicolor', 'setosa', 'virginica']
sub2.plot.density()
plt.title(column)

column = names[1]
sub1 = dataframe[[column,'Species']]
sub11 = sub1[sub1.Species == 'versicolor'][[column]]
sub12 = sub1[sub1.Species == 'setosa'][[column]]
sub13 = sub1[sub1.Species == 'virginica'][[column]]

sub2 = pd.concat([sub12, sub11, sub13], axis=1)

sub2.columns = ['versicolor', 'setosa', 'virginica']
sub2.plot.density()
plt.title(column)

column = names[2]
sub1 = dataframe[[column,'Species']]
sub11 = sub1[sub1.Species == 'versicolor'][[column]]
sub12 = sub1[sub1.Species == 'setosa'][[column]]
sub13 = sub1[sub1.Species == 'virginica'][[column]]

sub2 = pd.concat([sub12, sub11, sub13], axis=1)

sub2.columns = ['versicolor', 'setosa', 'virginica']
sub2.plot.density()
plt.title(column)

column = names[3]
sub1 = dataframe[[column,'Species']]
sub11 = sub1[sub1.Species == 'versicolor'][[column]]
sub12 = sub1[sub1.Species == 'setosa'][[column]]
sub13 = sub1[sub1.Species == 'virginica'][[column]]

sub2 = pd.concat([sub12, sub11, sub13], axis=1)

sub2.columns = ['versicolor', 'setosa', 'virginica']
sub2.plot.density()
plt.title(column)

plt.show()
```


SepalLengthCm
Density
1.0
0.8
0.6
0.4
0.2
0.0
4
5
6
7
8
9
versicolor
setosa
virginica


SepalWidthCm
Density
1.2
1.0
0.8
0.6
0.4
0.2
0.0
2
3
4
5
versicolor
setosa
virginica


PetalLengthCm
Density
2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
6
7
8
versicolor
setosa
virginica



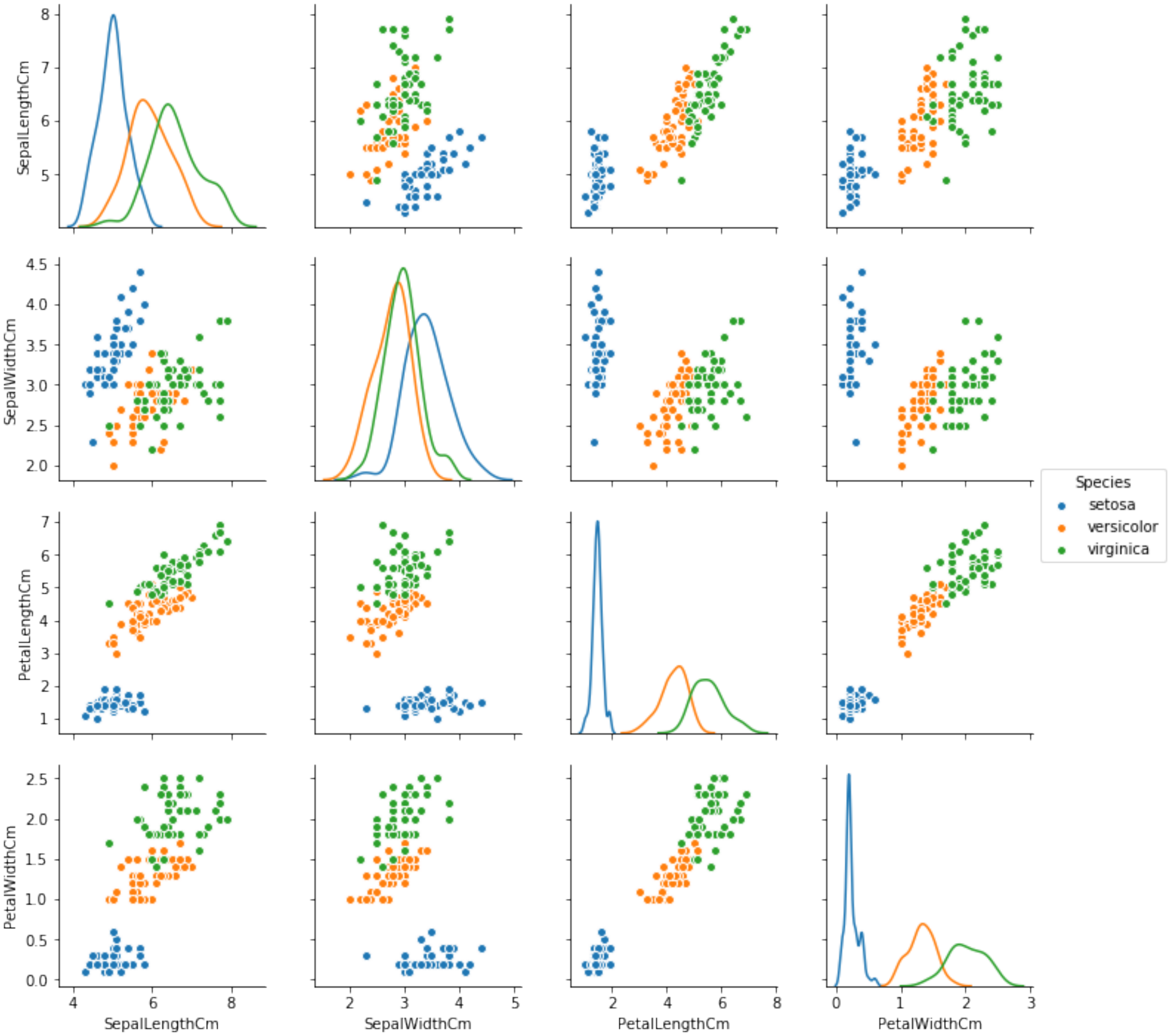
Task 5

Task 4 was a lot of work. Let's use Seaborn's `pairplot()` to visualize the data.

Hint: Explore the Seaborn documentation, and see if you can get density graphs to appear on the diagonal. Then compare the work you did in Task 4.

In [18]:

```
sns.pairplot(dataframe, hue="Species", diag_kind='kde')
plt.show()
```


SepalLengthCm
SepalWidthCm
PetalLengthCm
PetalWidthCm
Species
setosa
versicolor
virginica
8
7
6
5
4.5
4.0
3.5
3.0
2.5
2.0
1
2
3
4
5
0.0
0.5
1.0
1.5
2.0
2.5
4
6
8
2
3
4
5
2
4
6
8
0
1
2
3

Task 6

Explain what you see in the above visualization. Make a few comments about the different kinds of plots, and any general trends or patterns you can see. This is not a test! I just want you to see what the visualization might say.

> First, Seaborn did a lot of the work we tried to do in Pandas jsut above, and in Question 5.
> Second, the scatter plots tell us that the samples from the *setosa* class are quite distinct from the other two classes; on almost every scatter plot above, the setosa points are well-separated from the other two classes. This is good news for almost any classifier! This can also be seen in the density graphs along the diagonal, and in Task 4 above. The other two classes have data samples closer together in most dimensions. When a density plot overlaps 2 or more classes a lot, then classification will be harder. The most promising feature or attribute is the 4th column (which I named *PetalWidth*). It seems to give the best separation. THe least promising feature or attribute is the 2nd column (*SepalWidth*). There's a lot of overlap.
> These comments are not to be taken too strongly, since the scatter plot only tells us how pairs of attributes work together. In combinations of more than 2, there may be separations that we haven't visualized.

What to hand in

Your version of this notebook named A1Q7.pdf, containing completed work above, and your name and student number at the top.

Evaluation:

- 1 mark. For Task 1, you used `read_csv()` to load a datafile into the notebook.
- 1 mark. For Task 2, you used `density()` to display density estimation plots for the dataframe.
- 3 marks. For Task 3, you used Boolean array indexing to create separate DataFrames (one for each label value) and then used `density()` to display density estimation for each dataframe.
- 4 marks. For Task 4, you plotted the density estimation for each of the columns in the original DataFrame, and you have 3 densities allowing you to compare the distribution for each label in one plot. You have 4 such plots.
- 1 mark. For Task 5, you used Seaborn's `pairplot()` to display visualization of the original dataframe.
- 2 marks. For Task 6, you commented on the two kinds of plots, and what they represent. You also made some observations about some patterns in the data.

Grading:

- **Task 1**: 1 mark.
  - Required: Used `read_csv()`
- **Task 2**: 1 mark.
  - Required: Used `density()`
- **Task 3**: 3 marks.
  - Required: Used Boolean array indexing.
  - Required: Plotted 3 density plots using Pandas, one for each class label.
- **Task 4**: 4 marks.
  - Required: Separated the columns.
  - Required: Plotted densities for each column showing densities of the different class labels.
- **Task 5**: 1 mark. For Task 5, you used Seaborn's `pairplot()` to display visualization of the original dataframe.
  - Required: Used Seaborn's `pairplot()`
  - Optional: The density on the diagonal. This can be easy or hard, depending on which version of Seaborn you have, and which version of the documentation you look at.
- **Task 6**: 2 marks.
  - Required: Some comment on the one class of samples that seems distinct.
  - Optional: A comment about the density plots, if they appear. Seaborn can do them for us, but we shouldn't be shy about doing it ourselves if we have to.
  - Option: any other comments are fine.
- **Deductions**:
  - Deduct the marks for any required item missing.