

```
In [44]: # First, we'll import pandas, a data processing and CSV file I/O library
import pandas as pd

# We'll also import seaborn, a Python graphing library
import warnings # current version of seaborn generates a bunch of warnings that we'll
warnings.filterwarnings("ignore")
import seaborn as sns
import matplotlib.pyplot as plt
sns.set(style="white", color_codes=True)

# Next, we'll load the Iris flower dataset, which is in the "../input/" directory
iris = pd.read_csv("Iris.csv") # the iris dataset is now a Pandas DataFrame

# Let's see what's in the iris data - Jupyter notebooks print the result of the last
iris.head()
```

```
Out[44]:
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

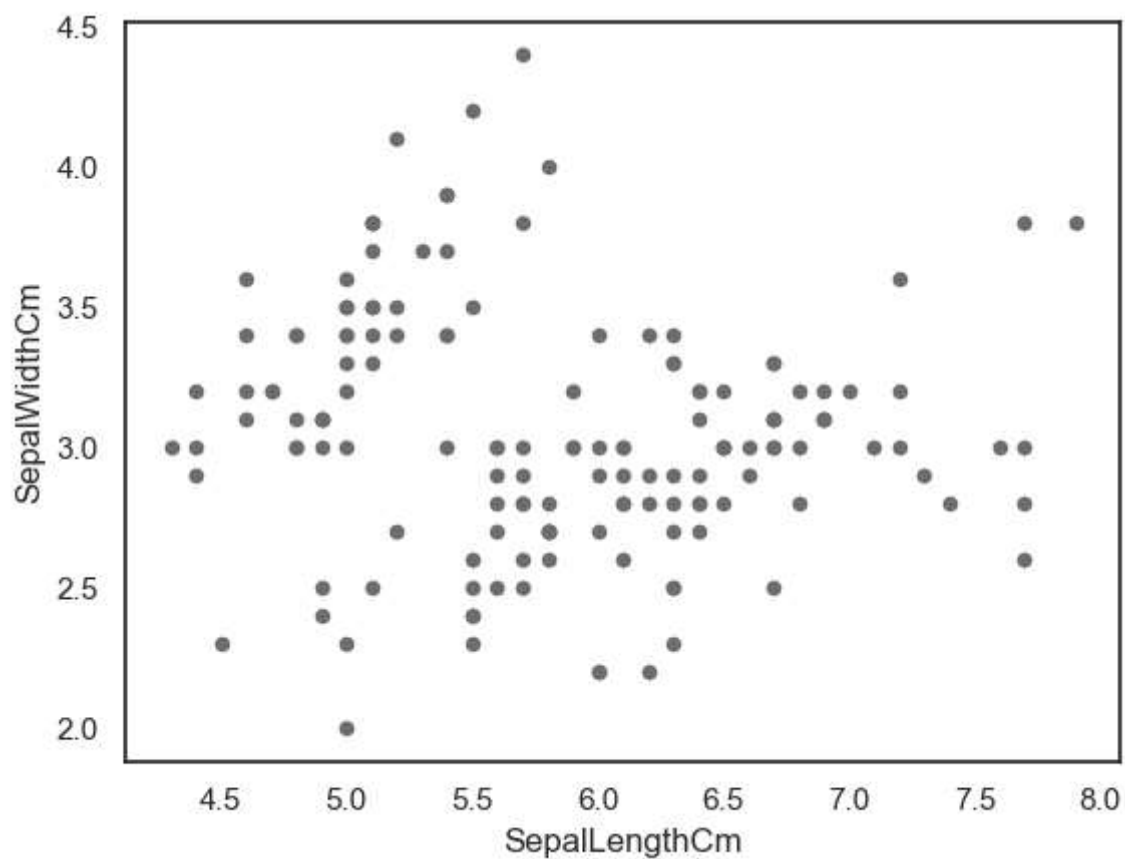
```
In [45]: # Let's see how many examples we have of each species
iris["Species"].value_counts()
```

```
Out[45]: Iris-setosa      50
Iris-versicolor    50
Iris-virginica      50
Name: Species, dtype: int64
```

```
In [46]: # The first way we can plot things is using the .plot extension from Pandas dataframes
# We'll use this to make a scatterplot of the Iris features.
iris.plot(kind="scatter", x="SepalLengthCm", y="SepalWidthCm")
```

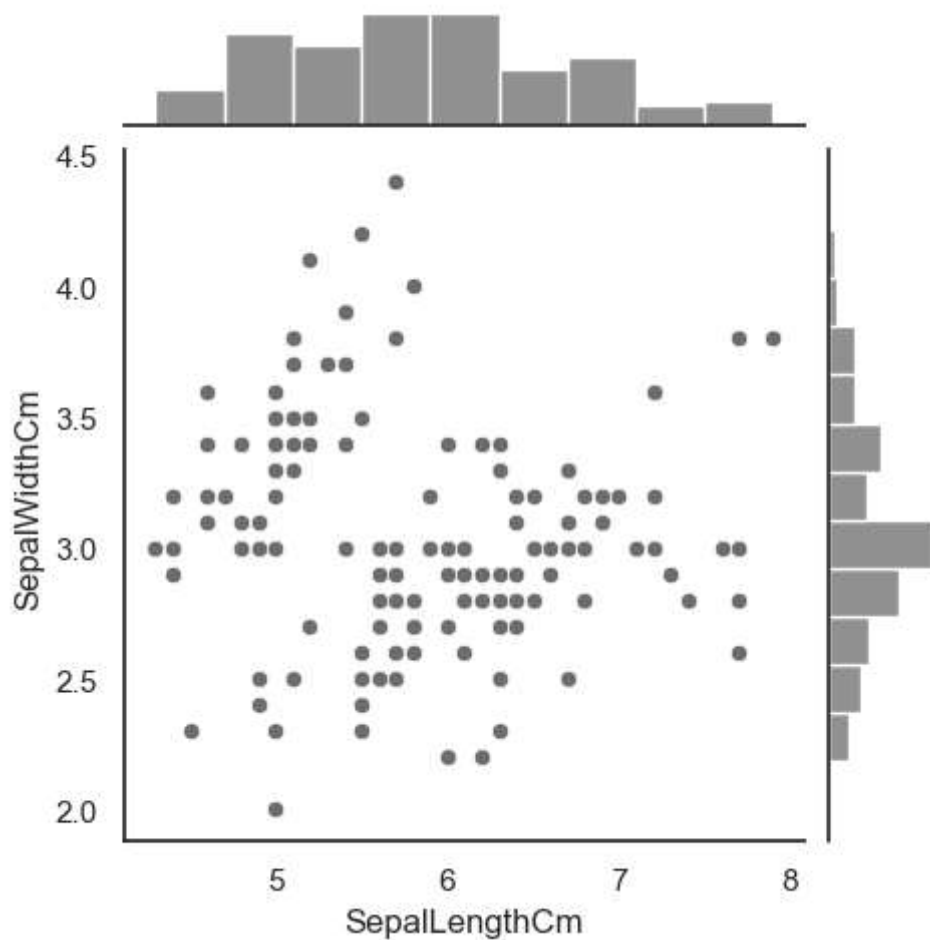
*\*c\* argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length matches with *\*x\** & *\*y\**. Please use the *\*color\** keyword-argument or provide a 2D array with a single row if you intend to specify the same RGB or RGBA value for all points.*

```
Out[46]: <AxesSubplot:xlabel='SepalLengthCm', ylabel='SepalWidthCm'>
```



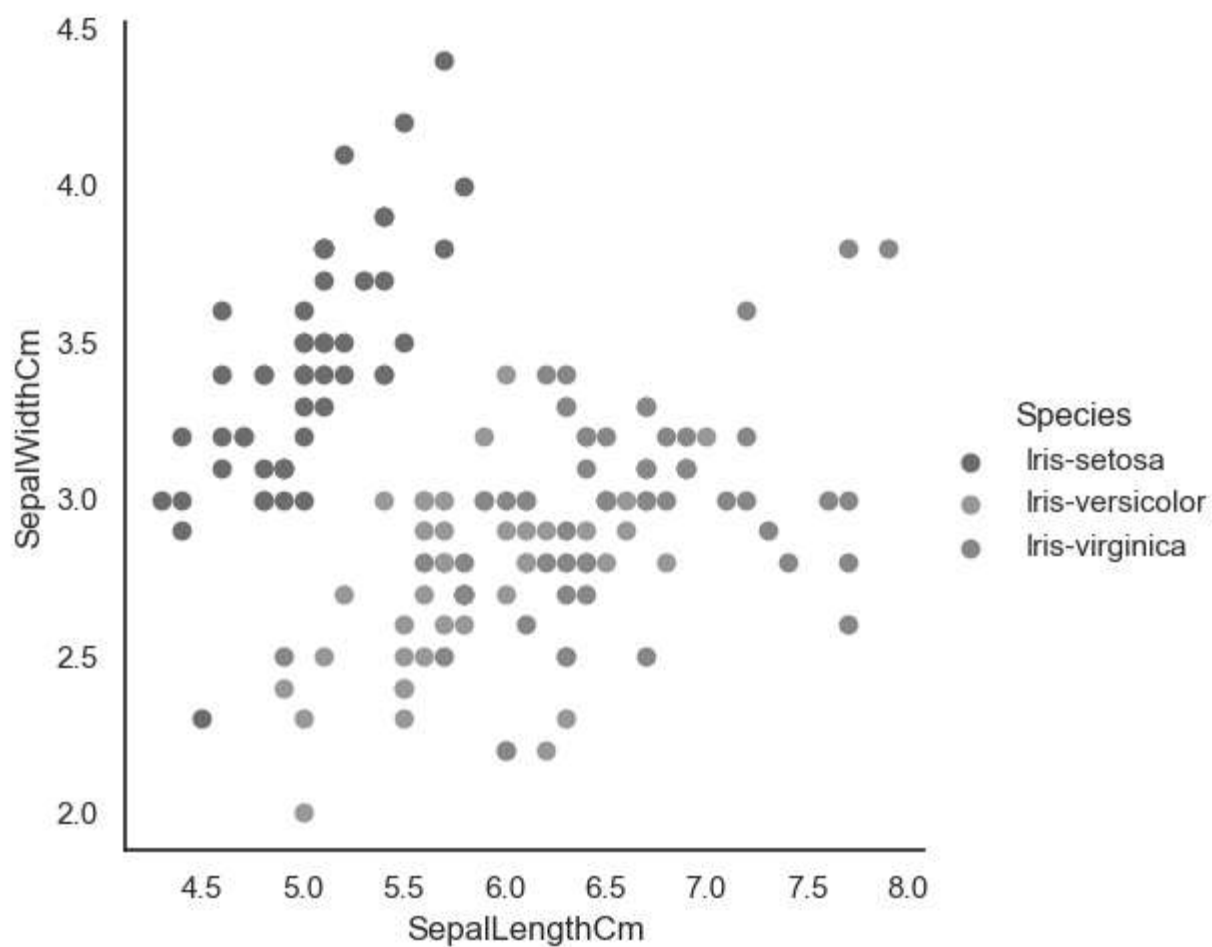
```
In [47]: # We can also use the seaborn library to make a similar plot
# A seaborn jointplot shows bivariate scatterplots and univariate histograms in the same figure
sns.jointplot(x="SepalLengthCm", y="SepalWidthCm", data=iris, size=5)
```

```
Out[47]: <seaborn.axisgrid.JointGrid at 0x1682aa6bdc0>
```



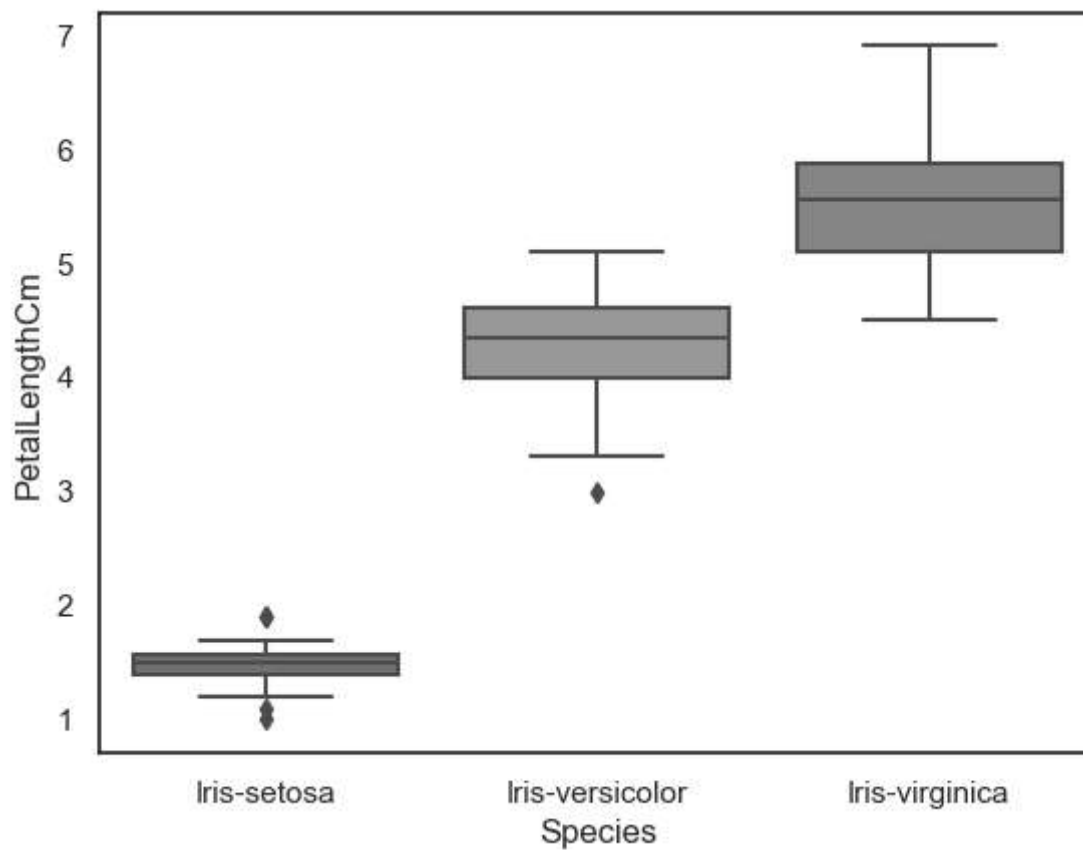
```
In [48]: # One piece of information missing in the plots above is what species each plant is
# We'll use seaborn's FacetGrid to color the scatterplot by species
sns.FacetGrid(iris, hue="Species", size=5) \
    .map(plt.scatter, "SepalLengthCm", "SepalWidthCm") \
    .add_legend()
```

```
Out[48]: <seaborn.axisgrid.FacetGrid at 0x1682abe0100>
```

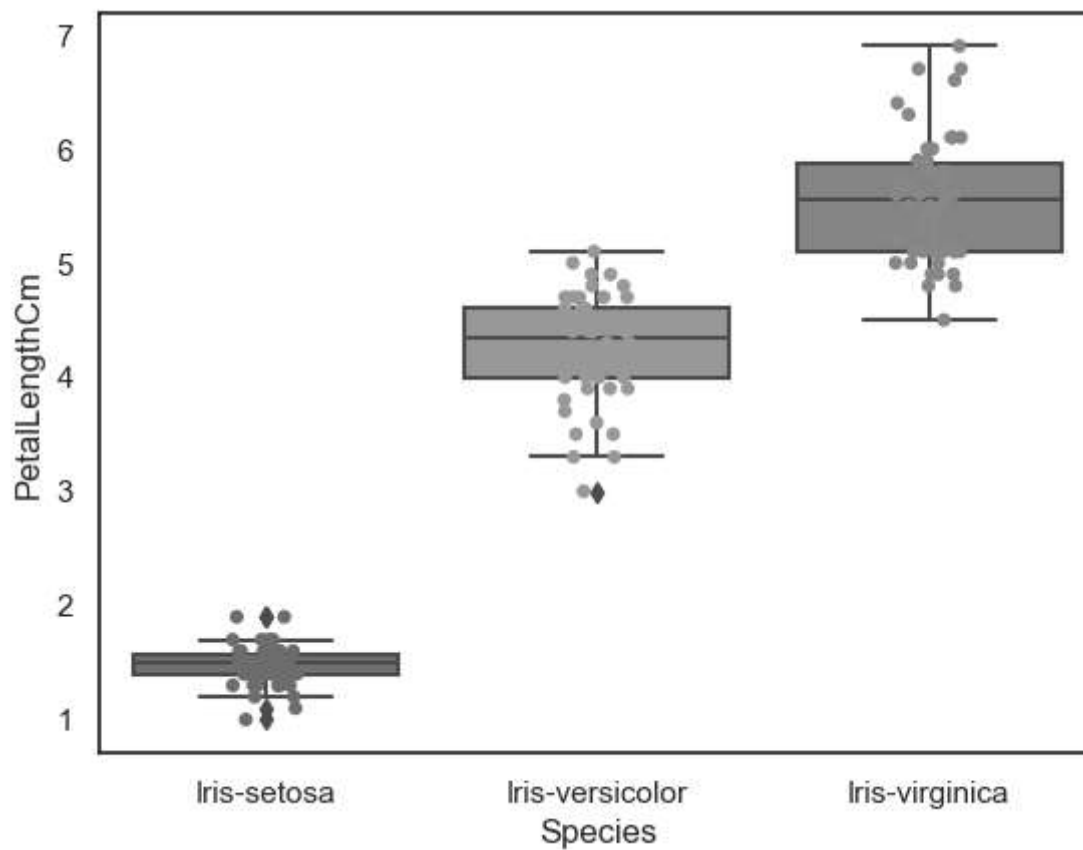


```
In [49]: # We can look at an individual feature in Seaborn through a boxplot  
sns.boxplot(x="Species", y="PetalLengthCm", data=iris)
```

```
Out[49]: <AxesSubplot:xlabel='Species', ylabel='PetalLengthCm'>
```

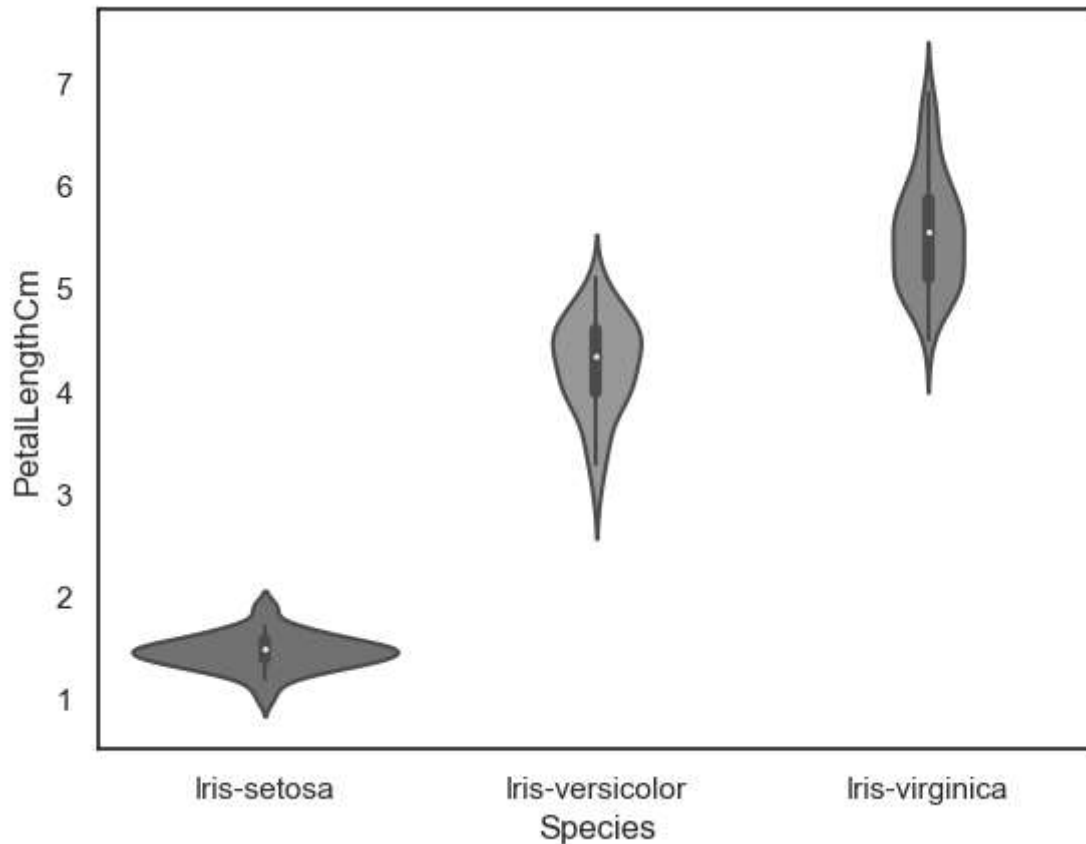


```
In [50]: ax = sns.boxplot(x="Species", y="PetalLengthCm", data=iris)
ax = sns.stripplot(x="Species", y="PetalLengthCm", data=iris, jitter=True, edgecolor='')
```



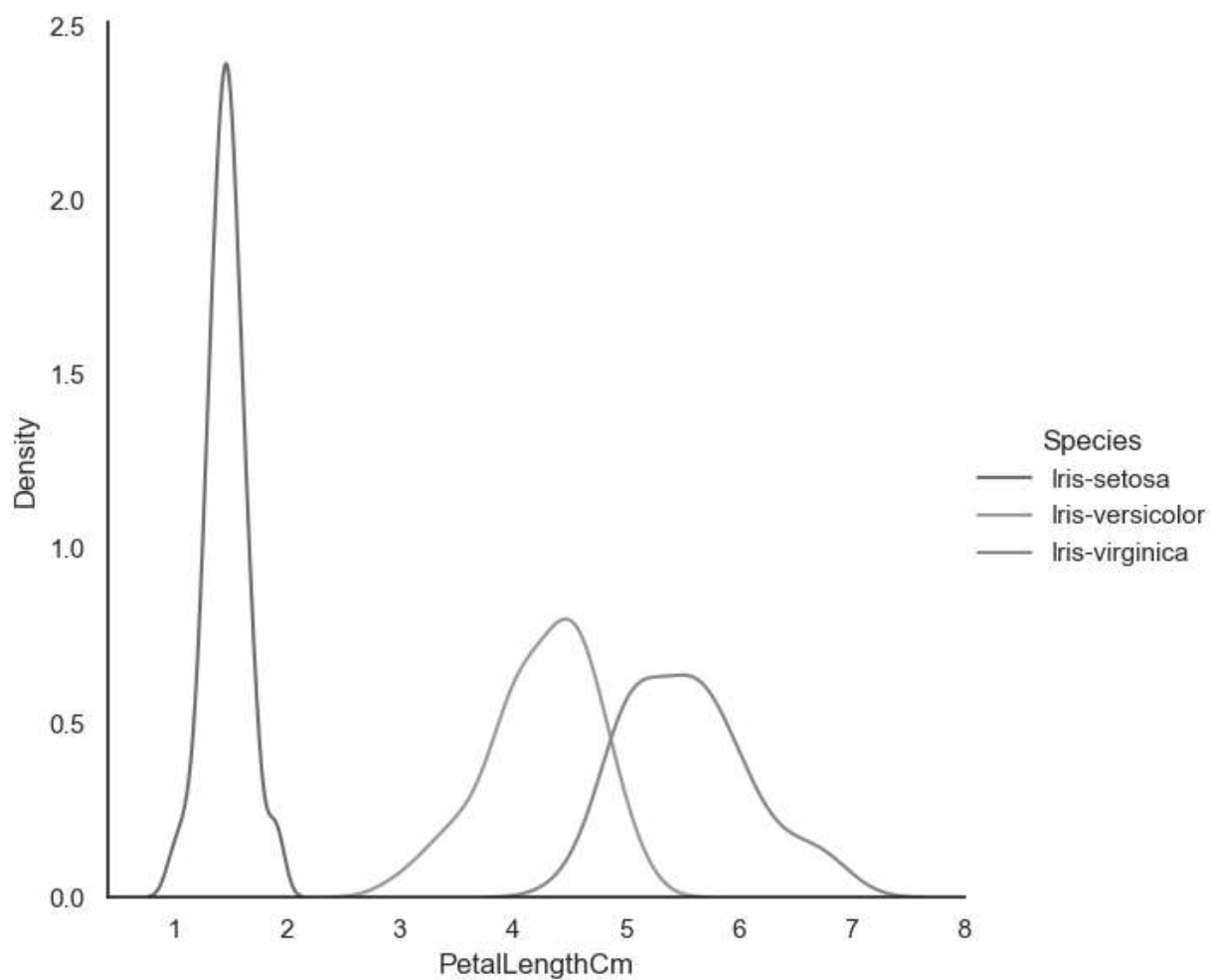
```
In [51]: # A violin plot combines the benefits of the previous two plots and simplifies them  
# Denser regions of the data are fatter, and sparser thinner in a violin plot  
sns.violinplot(x="Species", y="PetalLengthCm", data=iris, size=6)
```

```
Out[51]: <AxesSubplot:xlabel='Species', ylabel='PetalLengthCm'>
```



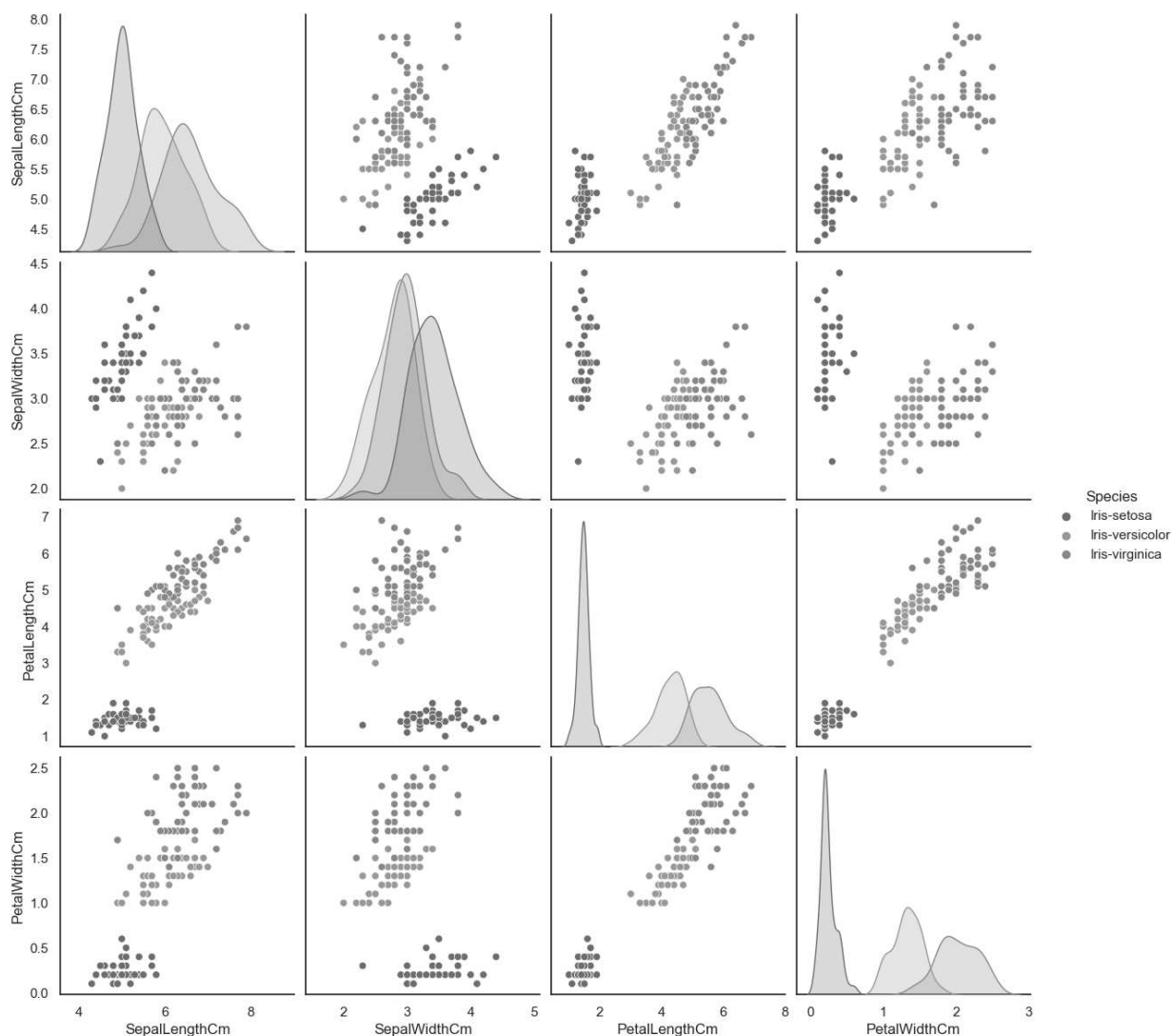
```
In [52]: # A final seaborn plot useful for looking at univariate relations is the kdeplot,  
# which creates and visualizes a kernel density estimate of the underlying feature  
sns.FacetGrid(iris, hue="Species", size=6) \  
    .map(sns.kdeplot, "PetalLengthCm") \  
    .add_legend()
```

```
Out[52]: <seaborn.axisgrid.FacetGrid at 0x1682adf3220>
```



```
In [53]: # Another useful seaborn plot is the pairplot, which shows the bivariate relation
# between each pair of features
#
# From the pairplot, we'll see that the Iris-setosa species is separataed from the oth
# two across all feature combinations
sns.pairplot(iris.drop("Id", axis=1), hue="Species", size=3)
```

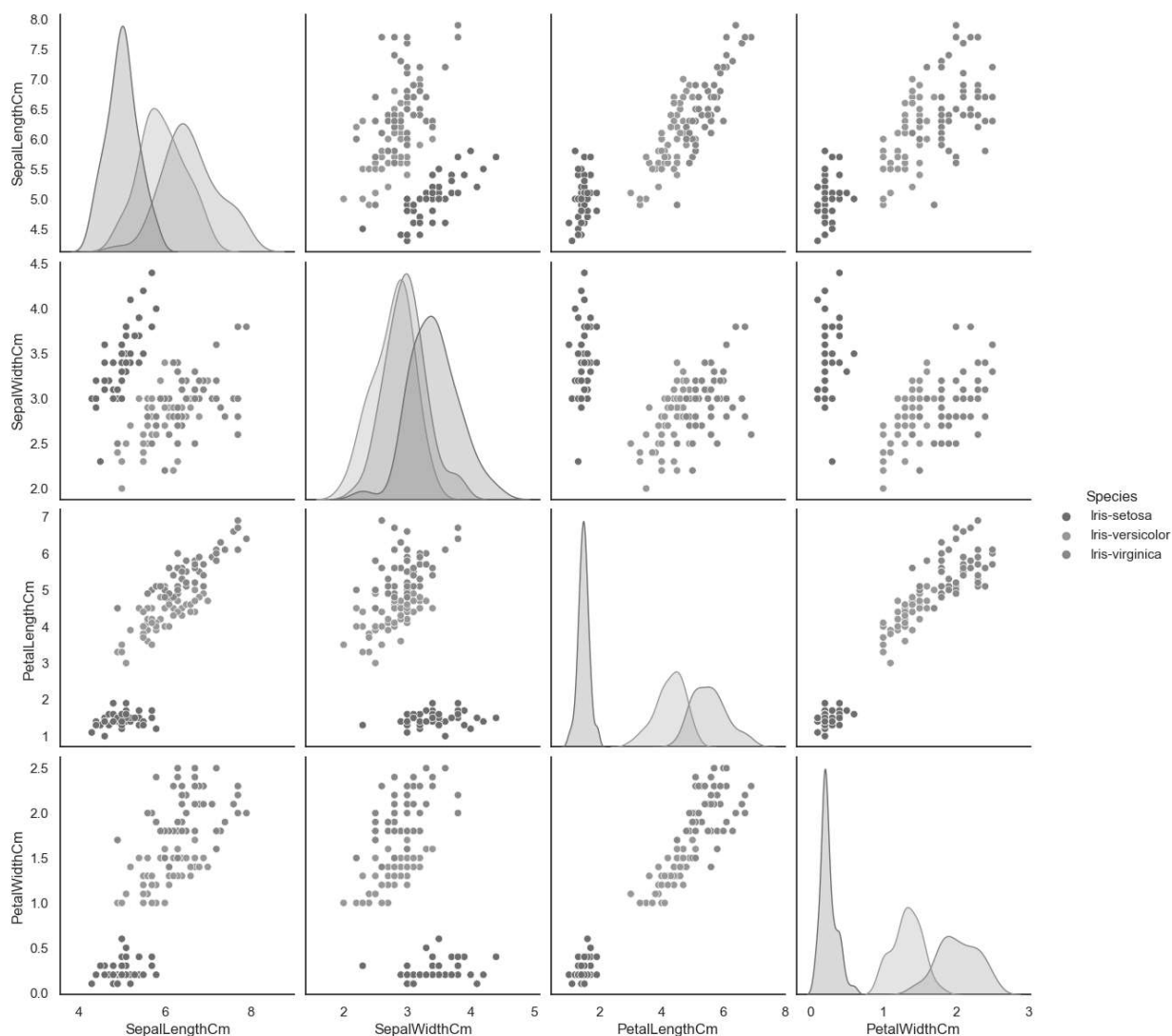
```
Out[53]: <seaborn.axisgrid.PairGrid at 0x1682b005610>
```



```
In [54]: # The diagonal elements in a pairplot show the histogram by default
# We can update these elements to show other things, such as a kde
sns.pairplot(iris.drop("Id", axis=1), hue="Species", size=3, diag_kind="kde")
```

```
Out[54]: <seaborn.axisgrid.PairGrid at 0x1682cfb4790>
```





```
In [55]: # Now that we've covered seaborn, let's go back to some of the ones we can make with Pandas
# We can quickly make a boxplot with Pandas on each feature split out by species
iris.drop("Id", axis=1).boxplot(by="Species", figsize=(12, 6))
```

```
Out[55]: array([[<AxesSubplot:title={'center':'PetalLengthCm'}, xlabel='[Species]'],
      [<AxesSubplot:title={'center':'PetalWidthCm'}, xlabel='[Species]'],
      [<AxesSubplot:title={'center':'SepalLengthCm'}, xlabel='[Species]'],
      [<AxesSubplot:title={'center':'SepalWidthCm'}, xlabel='[Species]']],
      dtype=object)
```

