Data Importing

In [2]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import sys
import matplotlib.pyplot as plt
sns.set(color_codes=True)
%matplotlib inline
```

In [6]:

```
df = pd.read_csv ('iris.csv')
```

In [7]:

```
df.head()
```

Out[7]:

	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 [10]:

```
col_name = ['Id','sepal length', 'sepal width', 'petal length', 'petal width', 'class']
```

In [12]:

```
df.columns = col_name
```

In [13]:

```
df.head()
```

Out[13]:

	Id	sepal length	sepal width	petal length	petal width	class
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

IRIS Data from Seaborn

In [14]:

```
iris = sns.load_dataset('iris')
iris.head()
```

Out[14]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [15]:

```
df.describe()
```

Out[15]:

	Id	sepal length	sepal width	petal length	petal width
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

In [16]:

```
iris.describe()
```

Out[16]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
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 [17]:

```
print(iris.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
None
```

In [18]:

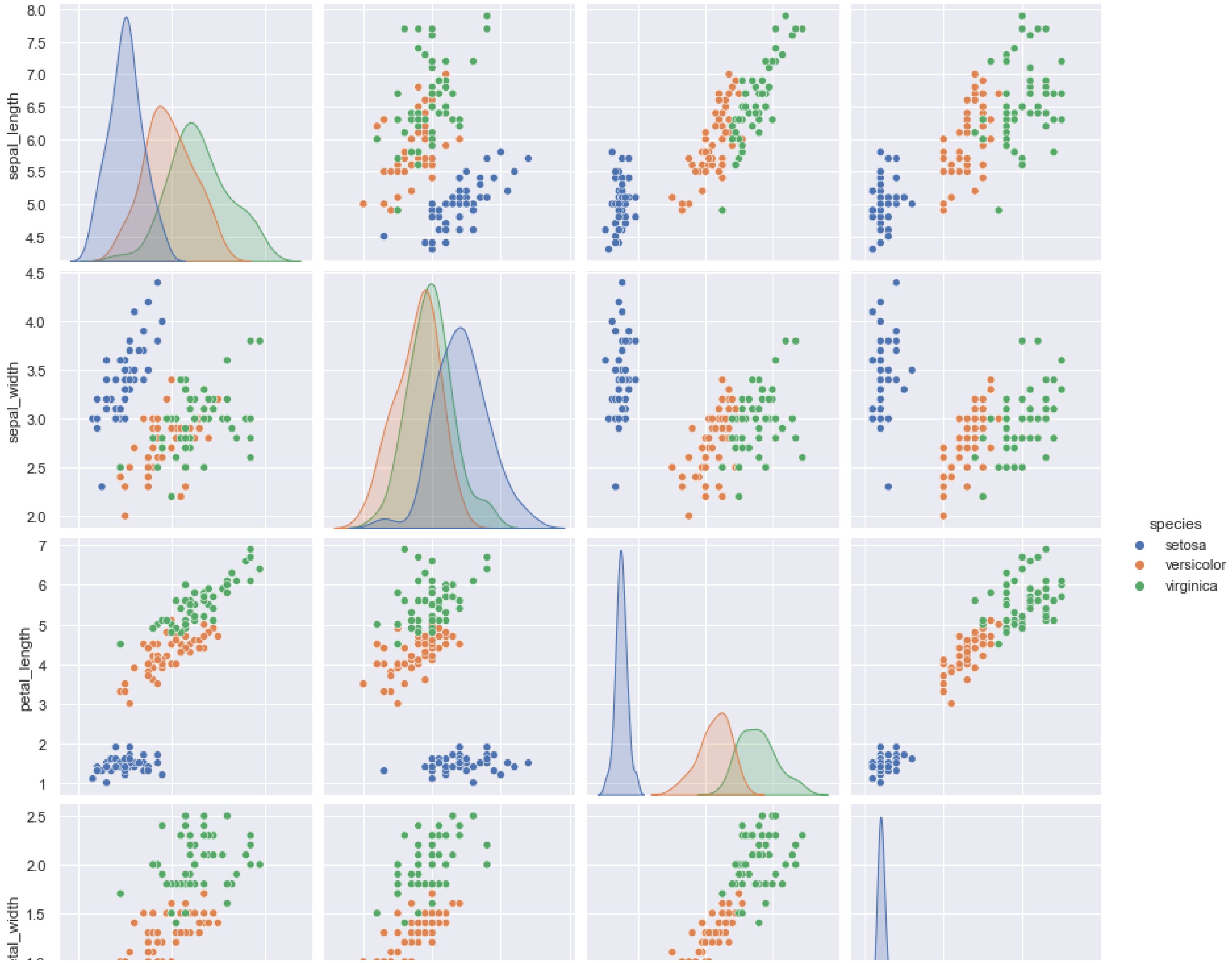
```
print(iris.groupby('species').size())
```

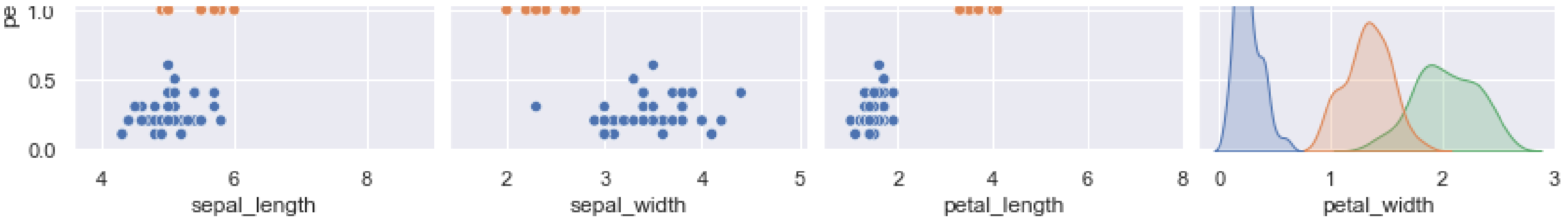
```
species
setosa        50
versicolor    50
virginica     50
dtype: int64
```

Visualisation

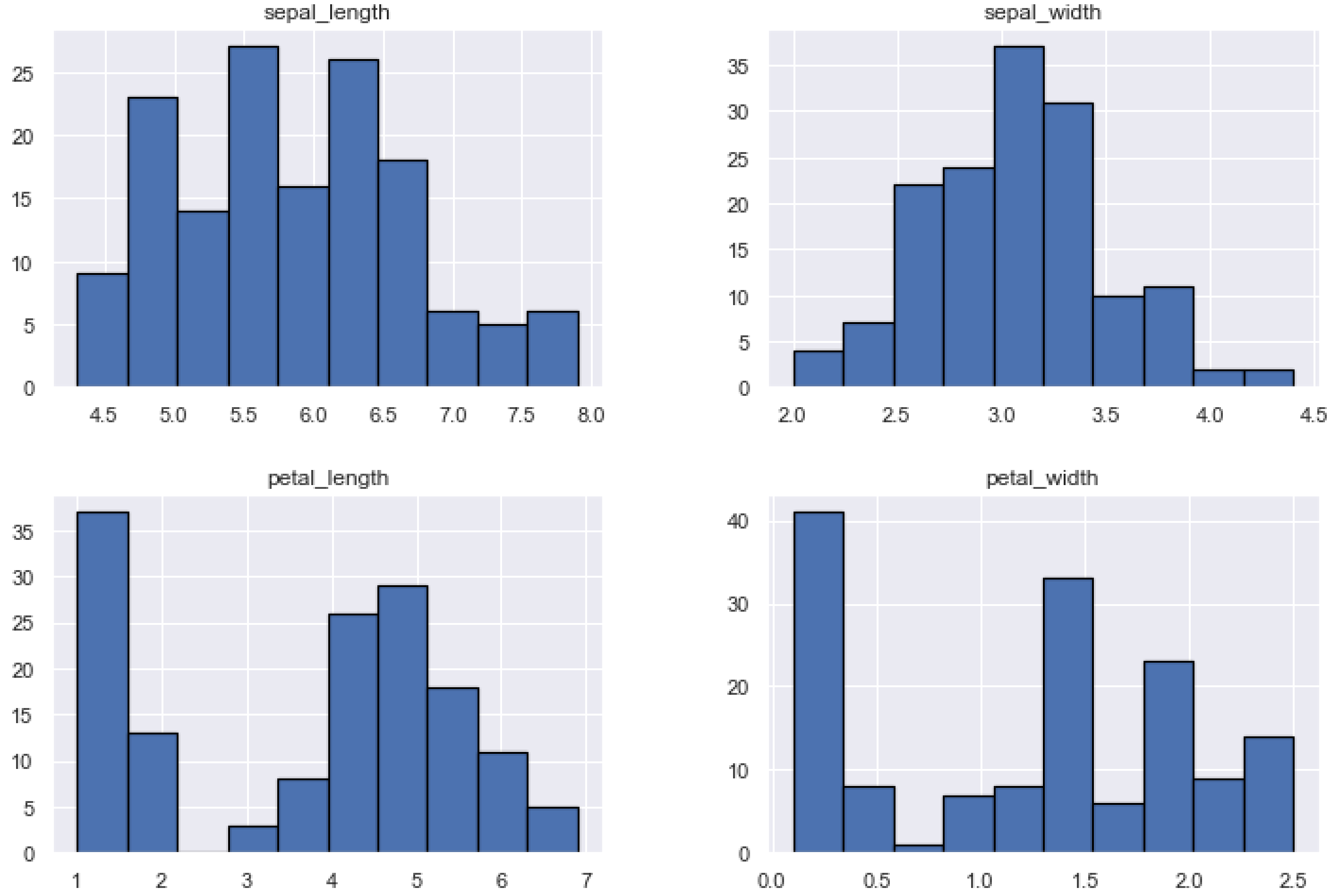
In [20]:

```
sns.pairplot(iris, hue='species', height=3, aspect=1);
```


8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
sepal_length
4.5
4.0
3.5
3.0
2.5
2.0
sepal_width
7
6
5
4
3
2
1
petal_length
2.5
2.0
1.5
tal_width
species
● setosa
● versicolor
● virginica

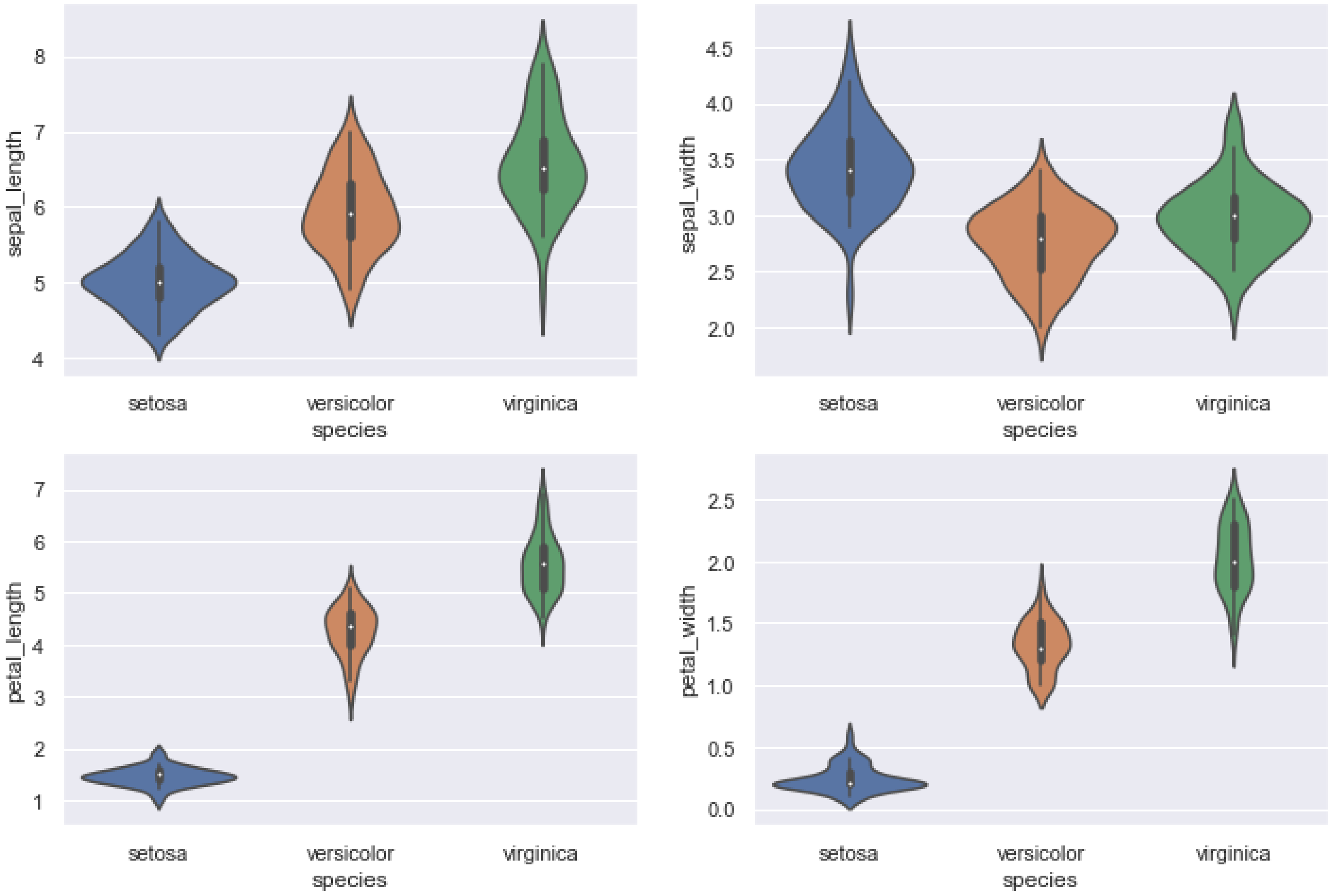


In [21]:
```
iris.hist(edgecolor='black', linewidth=1.2, figsize=(12,8));
plt.show();
```

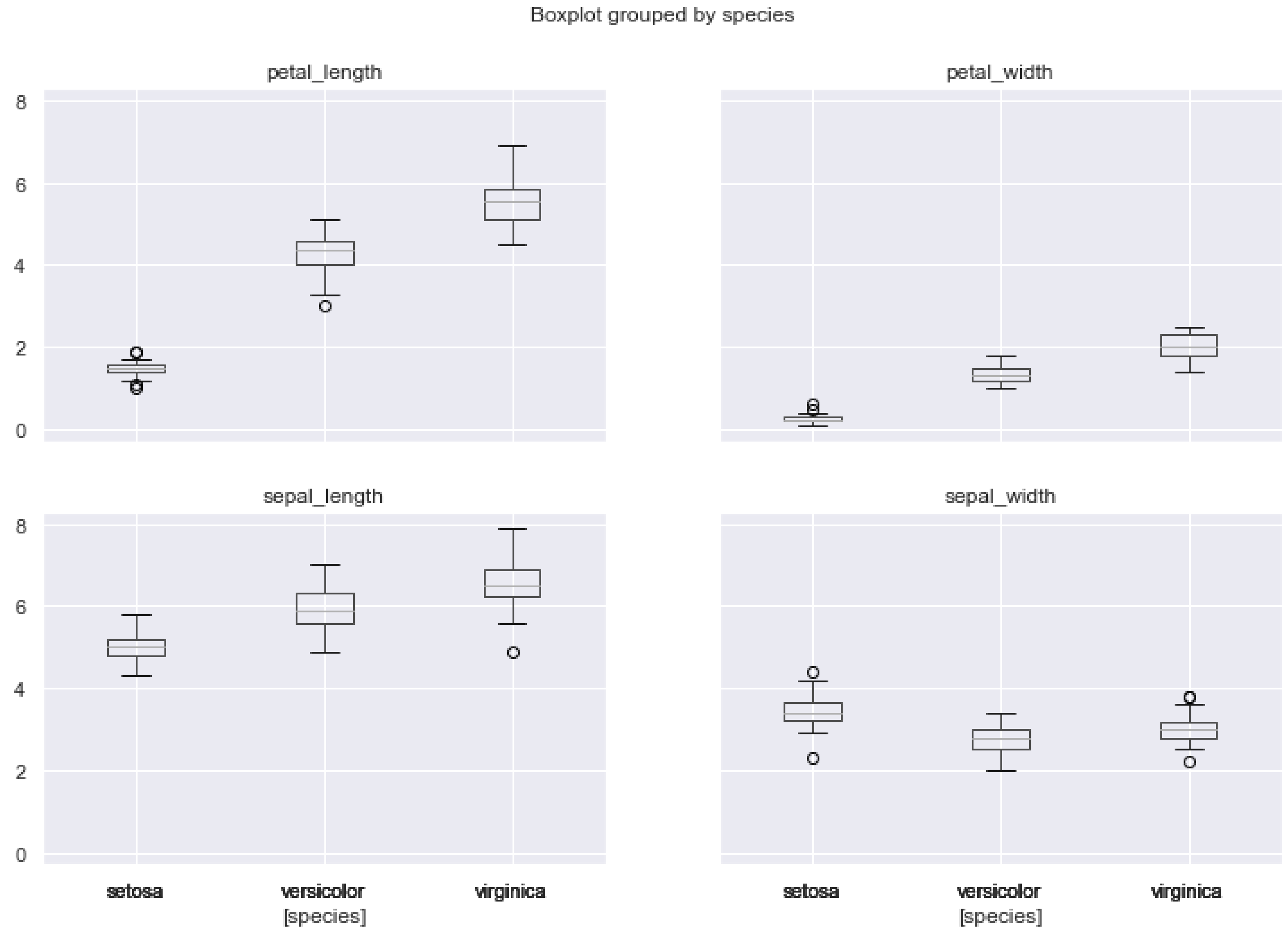


In [22]:
```
plt.figure(figsize=(12,8));
plt.subplot(2,2,1)
```

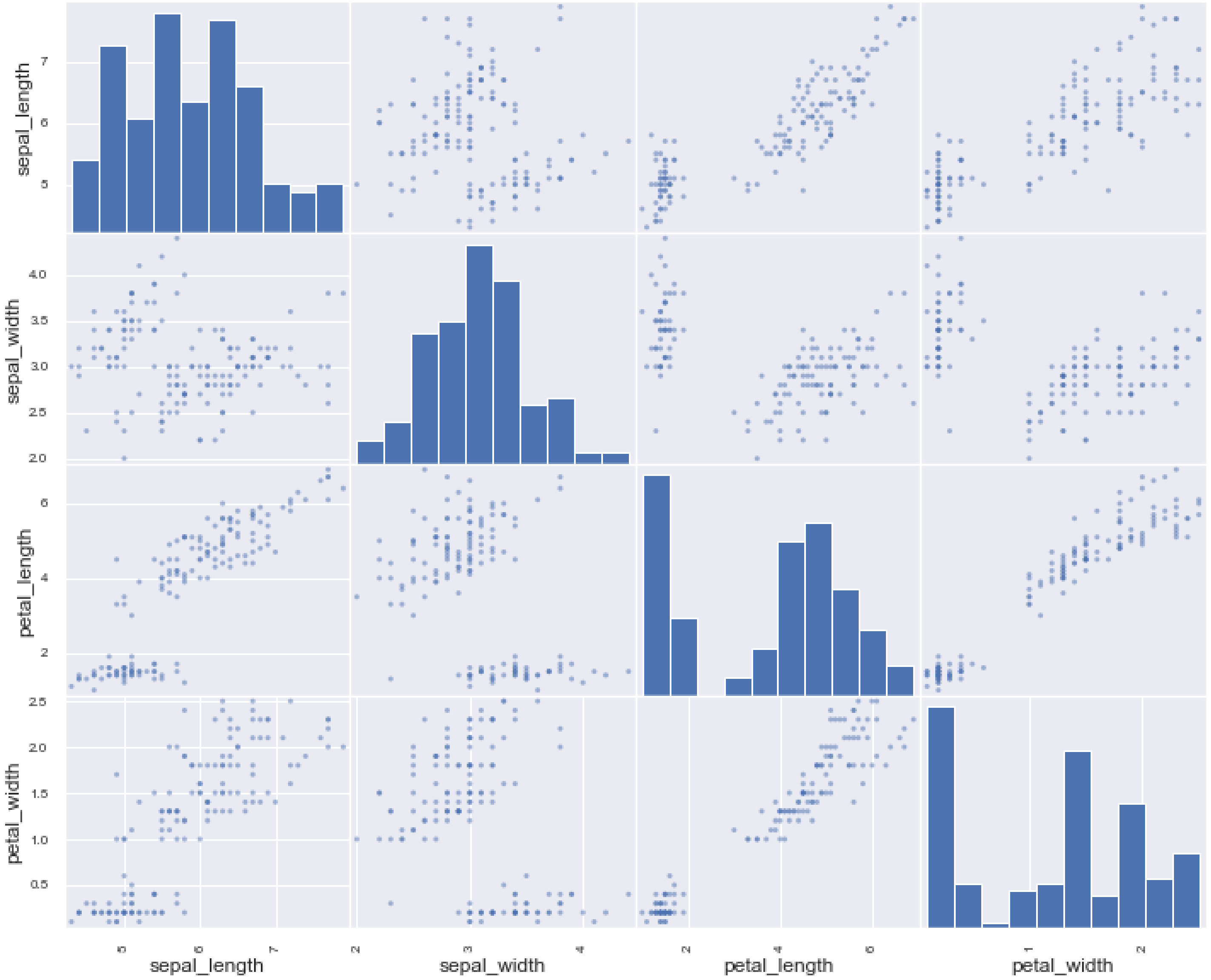
```
sns.violinplot(x='species', y='sepal_length', data=iris)
plt.subplot(2,2,2)
sns.violinplot(x='species', y='sepal_width', data=iris)
plt.subplot(2,2,3)
sns.violinplot(x='species', y='petal_length', data=iris)
plt.subplot(2,2,4)
sns.violinplot(x='species', y='petal_width', data=iris);
```



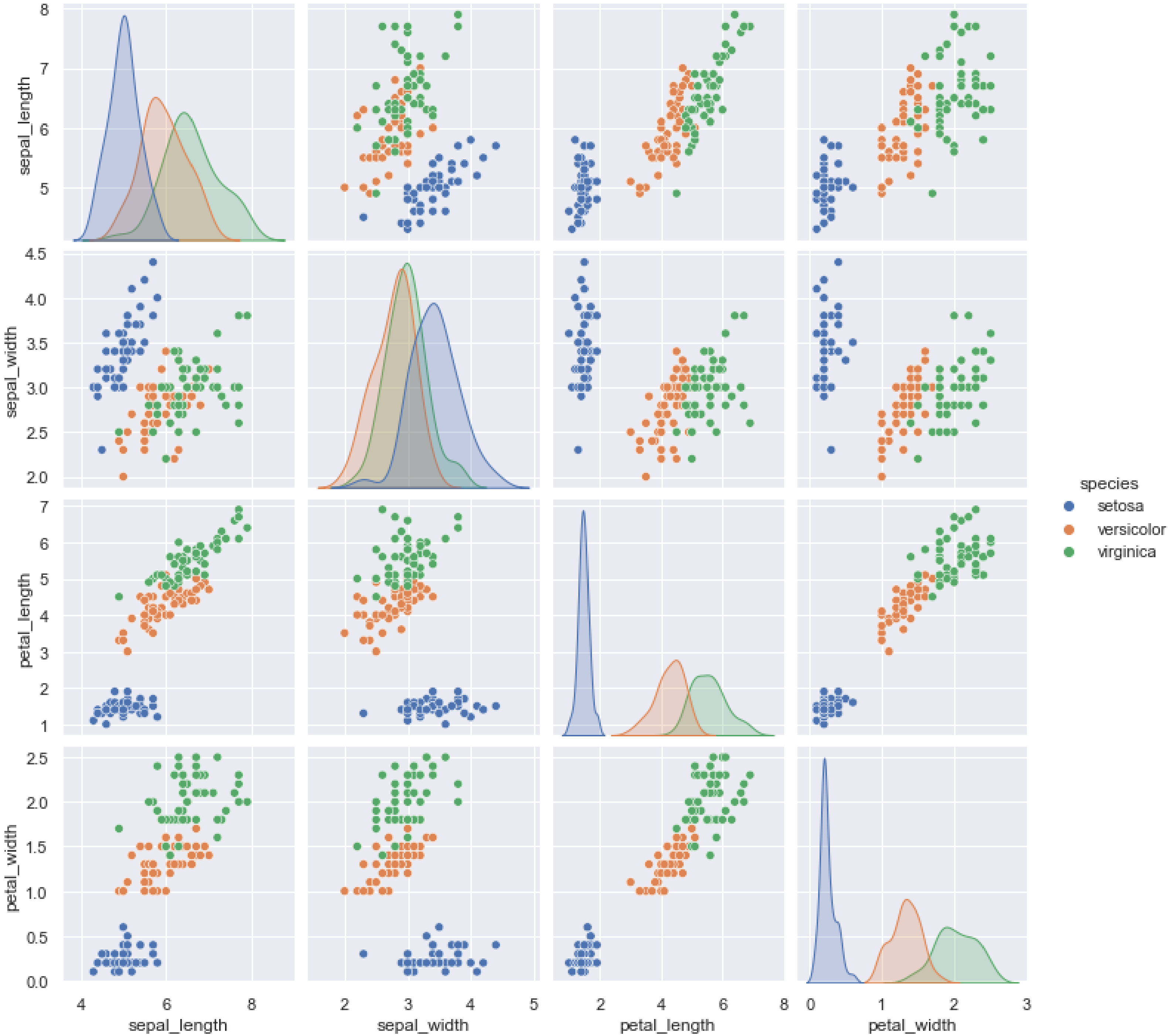
In [23]:
```
iris.boxplot(by='species', figsize=(12,8));
```



In [24]:
```
pd.plotting.scatter_matrix(iris, figsize=(12,10))
plt.show()
```



```
In [25]: sns.pairplot(iris, hue="species", diag_kind="kde");
```


sepal_length
sepal_width
petal_length
petal_width
8
7
6
5
4.5
4.0
3.5
3.0
2.5
2.0
7
6
5
4
3
2
1
2.5
2.0
1.5
1.0
0.5
0.0
4
6
8
2
3
4
5
2
4
6
8
0
1
2
3
species
● setosa
● versicolor
● virginica

Scikit-learn

In [26]:
```
iris.head()
```

Out[26]:

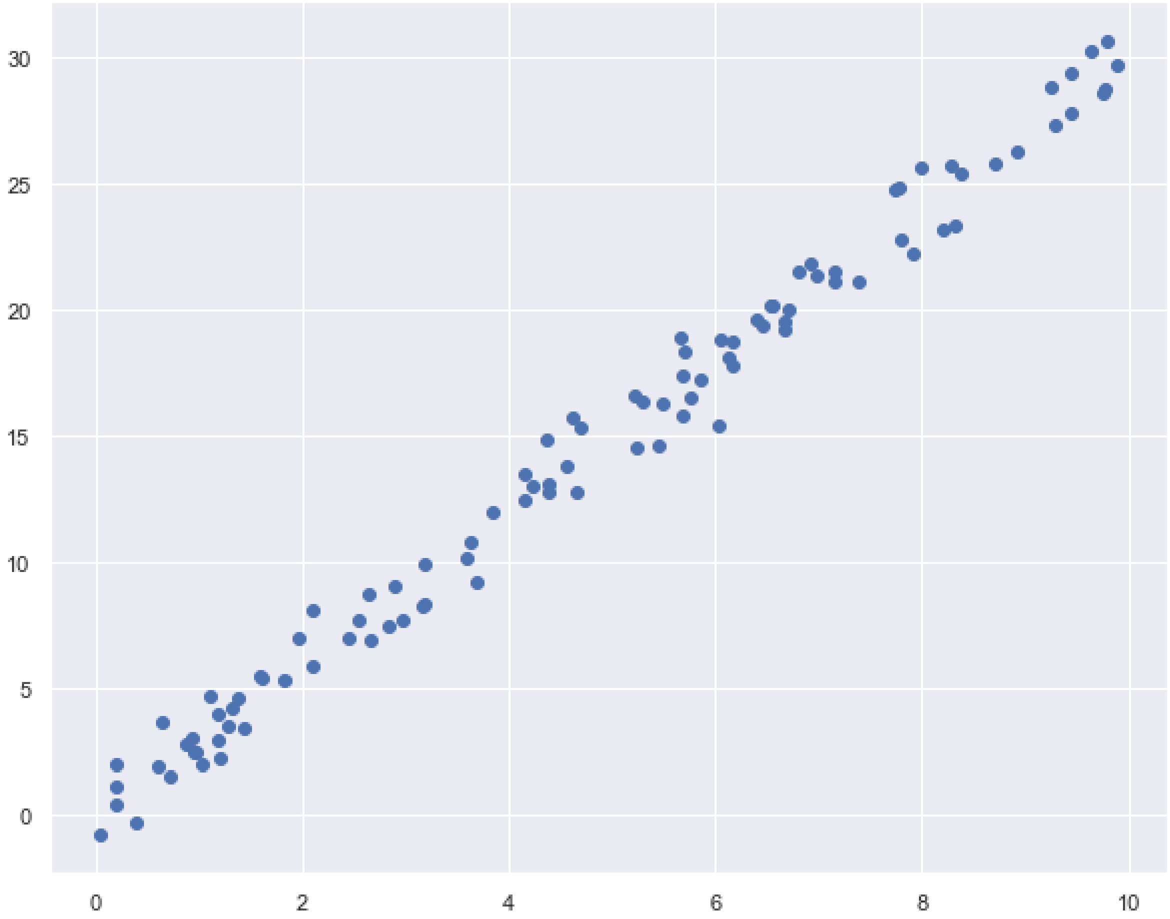
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Supervised Learning: Simple Linear Regression

In [27]:
```
generate_random = np.random.RandomState(0)
x = 10 * generate_random.rand(100)
```

In [28]:
```
y = 3 * x + np.random.randn(100)
```

In [29]:
```
plt.figure(figsize = (10, 8))
plt.scatter(x, y);
```



In [30]:
```
from sklearn.linear_model import LinearRegression
```

In [31]:
```
model = LinearRegression(fit_intercept=True)
```

In [32]:
```
model
```

Out[32]:
```
LinearRegression()
```

In [33]:
```
X = x.reshape(-1, 1)
X.shape
```

Out[33]:
```
(100, 1)
```

In [34]:

```
model.fit(X, y)
```

Out[34]:

```
LinearRegression()
```

In [35]:

```
model.coef_
```

Out[35]:

```
array([3.01298656])
```

In [36]:

```
model.intercept_
```

Out[36]:

```
-0.06468632687944265
```

In [37]:

```
x_fit = np.linspace(-1, 11)
```

In [38]:

```
X_fit = x_fit.reshape(-1,1)
```

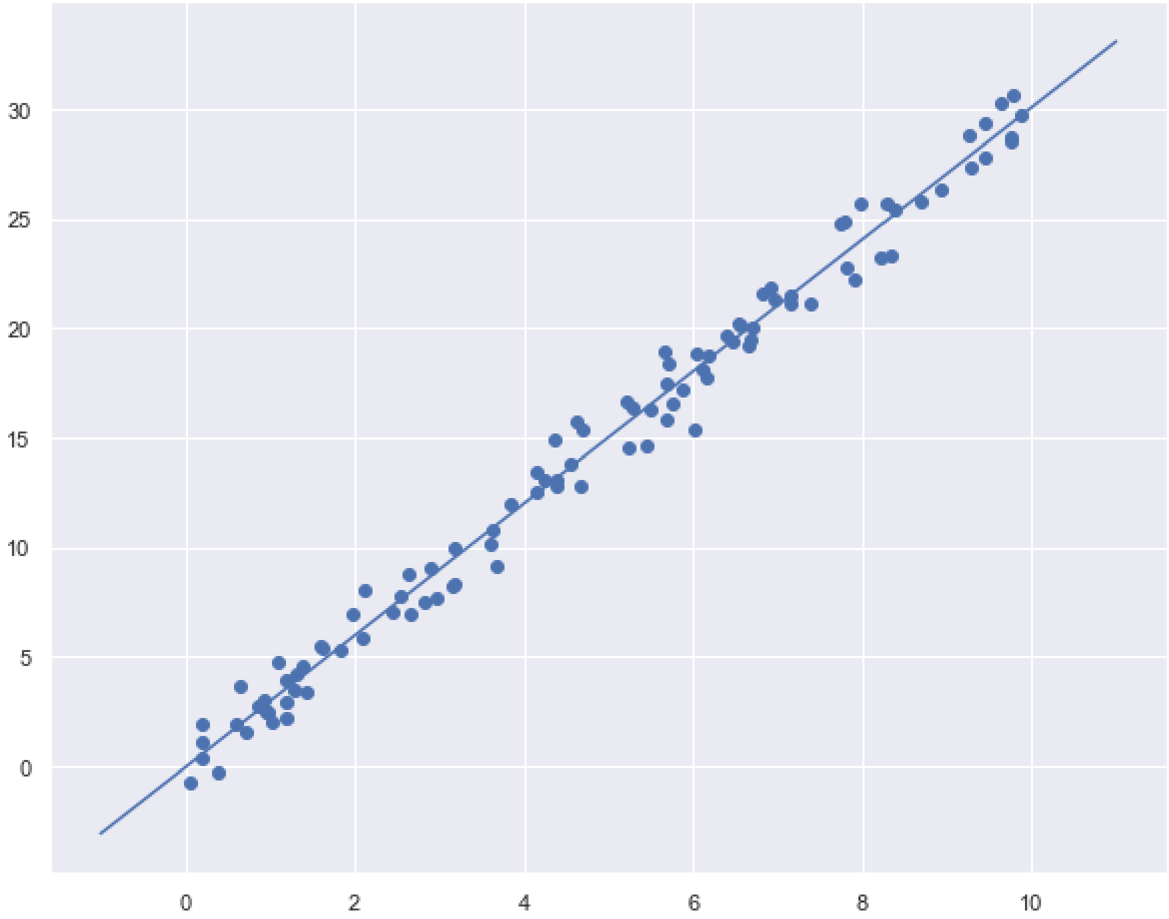
In [39]:

```
y_fit = model.predict(X_fit)
```

Visualise

In [40]:

```
plt.figure(figsize = (10, 8))
plt.scatter(x, y)
plt.plot(x_fit, y_fit);
```


30
25
20
15
10
5
0
0
2
4
6
8
10