

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
```

```
iris_df = pd.read_csv('iris.data.csv')
iris_df.head()
```

1 to 5 of 5 entries   

index	5.1	3.5	1.4	0.2	Iris-setosa
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa

Show  per page

Like what you see? Visit the [data table notebook](#) to learn more about interactive tables.

Next steps: ☒ [View recommended plots](#)

```
data = load_iris()
data.feature_names
```

```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

```
data.target_names
```

```
array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

```
data.target
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

```
X = data.data
```

```
X.shape
```

```
(150, 4)
```

```
y=data.target
```

```
y.shape
```

```
(150,)
```

```
y = y.reshape(-1, 1)
```

```
y.shape
```

```
(150, 1)
```

```
plt.figure(figsize=(18,8),dpi=100)
```

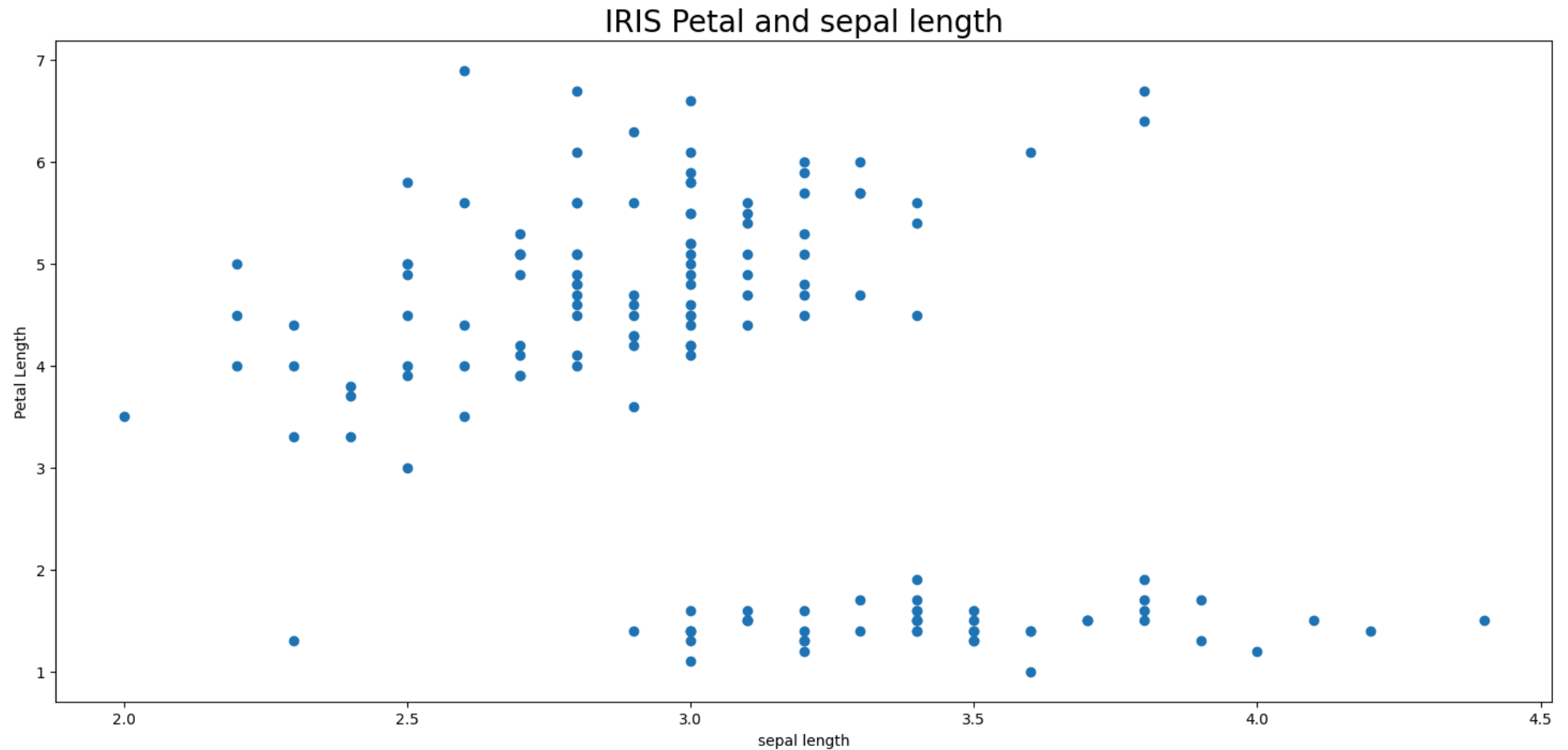
```
plt.scatter(X.T[1],X.T[2])
```

```
plt.title('IRIS Petal and sepal length', fontsize=20)
```

```
plt.ylabel('Petal Length')
```

```
plt.xlabel('sepal length')
```

Text(0.5, 0, 'sepal length')



```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
X_train,X_test,y_train,y_test = train_test_split(X,y, test_size = 0.20)
```

```
lr = LinearRegression()
iris_model = lr.fit(X_train, y_train)
predictions = iris_model.predict(X_test)
```

```
from sklearn.metrics import r2_score    #class will help us to calculate and see the score of our predictions
```

```
r2_score(y_test, predictions)
```

```
0.9001379517220648
```

```
np.sqrt(((predictions - y_test)**2).mean())
```

```
0.24949417513557226
```