In [1]:
```
import numpy as np
import pandas as pd

from ipywidgets import interact, interactive, IntSlider, Layout, interact_manual
import ipywidgets as widgets
from IPython.display import display
from numpy.random import randint
import matplotlib.pyplot as plt

from sklearn.linear_model import LinearRegression
from sklearn.linear_model import LogisticRegression
from sklearn import datasets
```

Linear Regressions

In [2]:
```
N_samples = 25
x_scale = 1
y_scale = 1
```

In [3]:
```
x = np.random.randint(0, 50, N_samples)
y = np.random.randint(0, 50, N_samples)
```

In [4]:

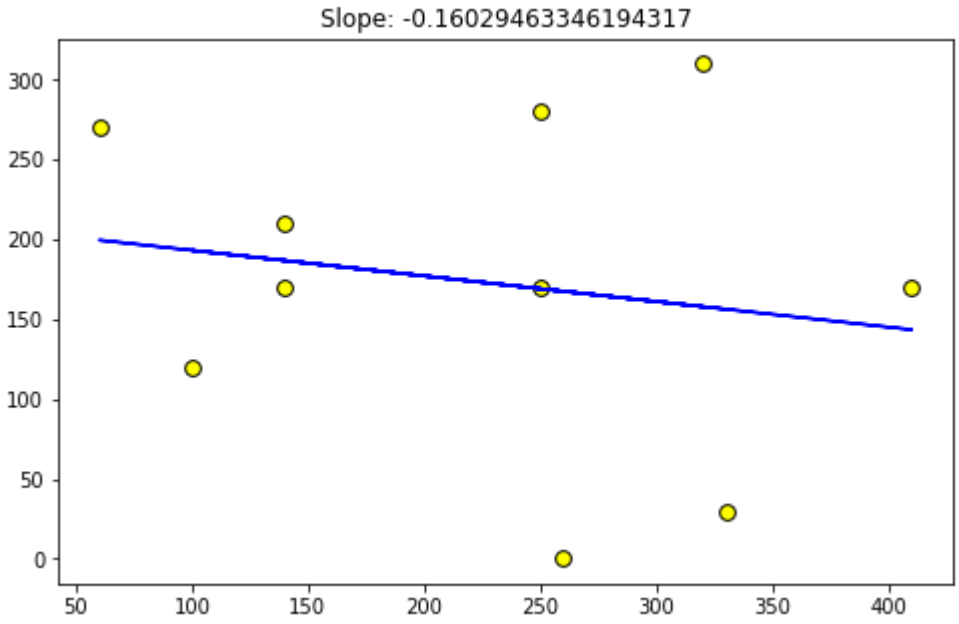
```
def my_reg_func(N_samples, x_scale, y_scale):
    x = np.random.randint(0, 50, N_samples)
    y = np.random.randint(0, 50, N_samples)

    x = x * x_scale
    y = y * y_scale

    model = LinearRegression().fit(x.reshape(-1, 1), y)
    slope = model.coef_[0] # Takes the first element of the array
    intercept = model.intercept_

    line_val = [slope * i + intercept for i in x]

    plt.figure(figsize=(8,5))
    plt.scatter(x,y,edgecolors='k',c='yellow',s=60)
    plt.plot(x, line_val, 'b')
    plt.title("Slope: " + str(slope))
    plt.show()
```

In [5]:
```
p1=interactive(my_reg_func, N_samples={'Low (10 samples)':10, 'Medium (50 samples)':50,
                                       'High (200 samples)':200}, x_scale=(1, 20, 0.05),
               y_scale=(1, 20, 0.05))
display(p1)
```

N_samples: Low (10 samples)

x_scale: 10.00

y_scale: 10.00



In [6]:
```
iris = datasets.load_iris()
```

In [7]:
```
df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = pd.Series(iris.target)
df['label'] = df.target.replace(dict(enumerate(iris.target_names)))
df.head()
```

Out[7]:

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

Correlation

In [8]:
```
df[["sepal length (cm)", "petal width (cm)"]].corr()
df[["sepal length (cm)", "target"]].corr()
```

Out[8]:

	sepal length (cm)	target
sepal length (cm)	1.000000	0.782561
target	0.782561	1.000000

In [ ]:
```

```