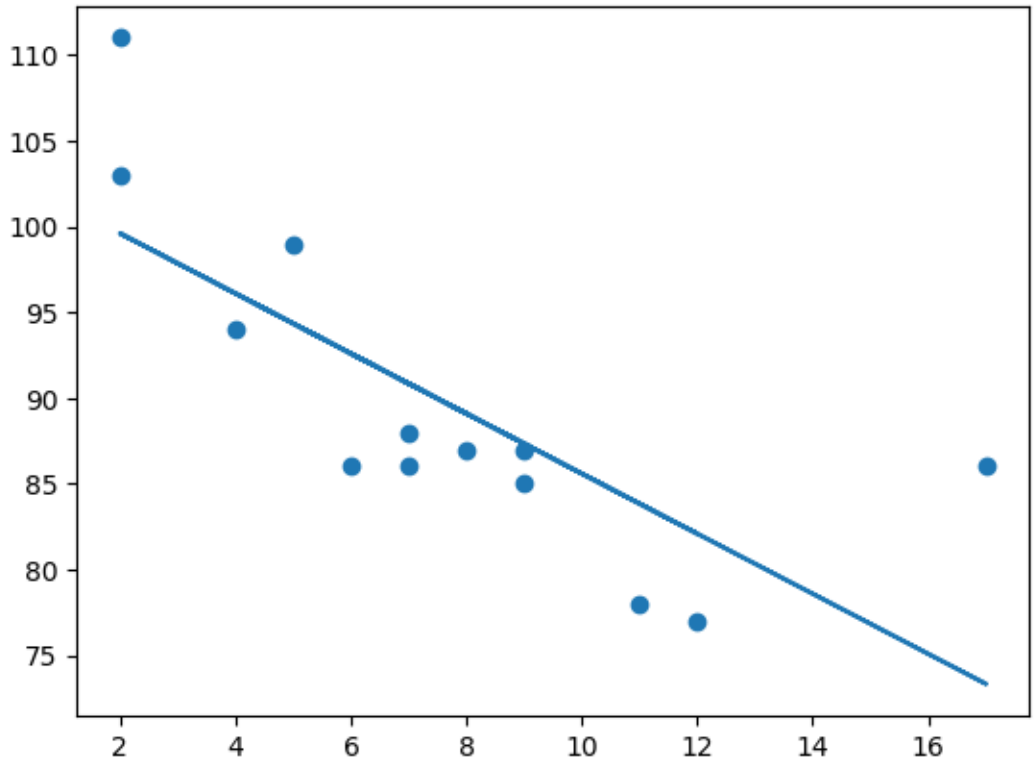
Unit03 Ex2 linear_regression

July 27, 2023

[2]:
```
import matplotlib.pyplot as plt
from scipy import stats

#Create the arrays that represent the values of the x and y axis
x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y = [99,86,87,88,111,86,103,87,94,78,77,85,86]

#Execute a method that returns some important key values of Linear Regression
slope, intercept, r, p, std_err = stats.linregress(x, y)

# measure the correlation
corr, _ = stats.pearsonr(x, y)
print('Pearsons correlation: %.3f' % corr)

#Create a function that uses the slope and intercept values to return a new␣
 ↪value.
#This new value represents where on the y-axis the corresponding x value will␣
 ↪be placed
def myfunc(x):
  return slope * x + intercept

#Run each value of the x array through the function. This will result in a new␣
 ↪array with new values for the y-axis
mymodel = list(map(myfunc, x))

#Draw the original scatter plot & the line of linear regression
plt.scatter(x, y)
plt.plot(x, mymodel)
plt.show()
```

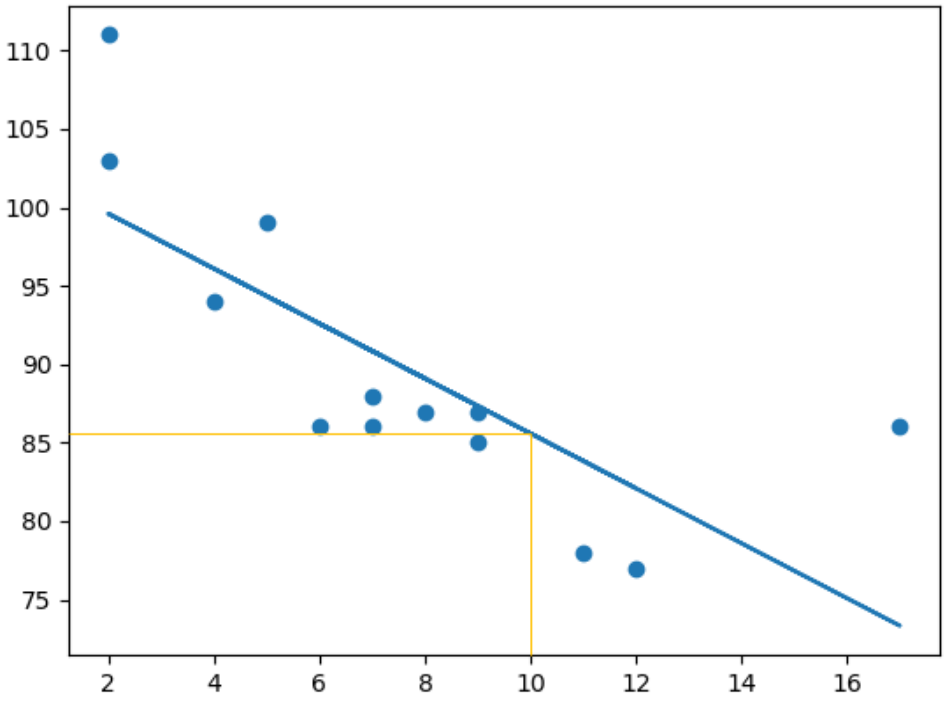
```
Pearsons correlation: -0.759
```



0.1 Predict Future Values

[4]:
```
from scipy import stats

x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y = [99,86,87,88,111,86,103,87,94,78,77,85,86]

slope, intercept, r, p, std_err = stats.linregress(x, y)

def myfunc(x):
  return intercept + slope * x

speed = myfunc(10)

print(speed)
```

```
85.59308314937454
```



If x=10 then predicted y is 85.59

[ ]: