Machine Learning Lab

Reg No: 23MCA1030

Name : Vinayak Kumar Singh

Exercise 2 : Linear Regression

A company manufactures an electronic device to be used in a very wide temperature range. The company knows that increased temperature shortens the life time of the device, and a study is therefore performed in which the life time is determined as a function of temperature.

Temperature in Celcius (t)	10	20	30	40	50	60	70	80	90
Life time in hours (y)	420	365	285	220	176	117	69	34	5

Calculate the 95% confidence interval for the slope in the usual linear re- gression model, which expresses the life time as a linear function of the temperature.

Collab url : https://colab.research.google.com/drive/1i6Mr35V1KS0ySU2CLhl_cOI8fH12OePV?usp=sharing

---

For Linear Regression Question

Importing library

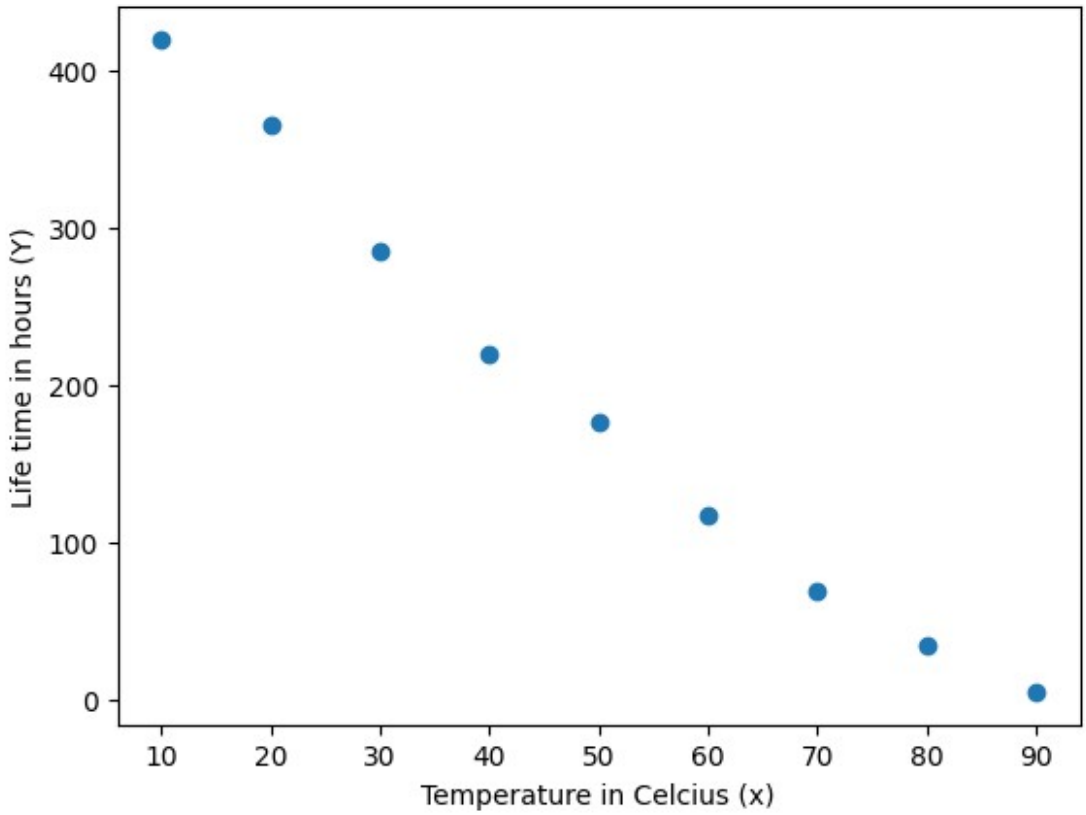
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Adding Data value of x and y values from the question

```
x= [10,20,30,40,50,60,70,80,90]
y= [420,365,285,220,176,117,69,34,5]
```

Performing the scatter plot

```
plt.xlabel('Temperature in Celcius (x)')
plt.ylabel('Life time in hours (Y)')
plt.scatter(x, y)
plt.show()
```



Performing linear regression on the data in the lists x and y.

```
from scipy import stats
B1, B0, r, p, std_err = stats.linregress(x, y)
print(B0,B1,r,std_err)
```

```
453.55555555555554 -5.3133333333333335 -0.9919863879175506
0.2557818184006401
```

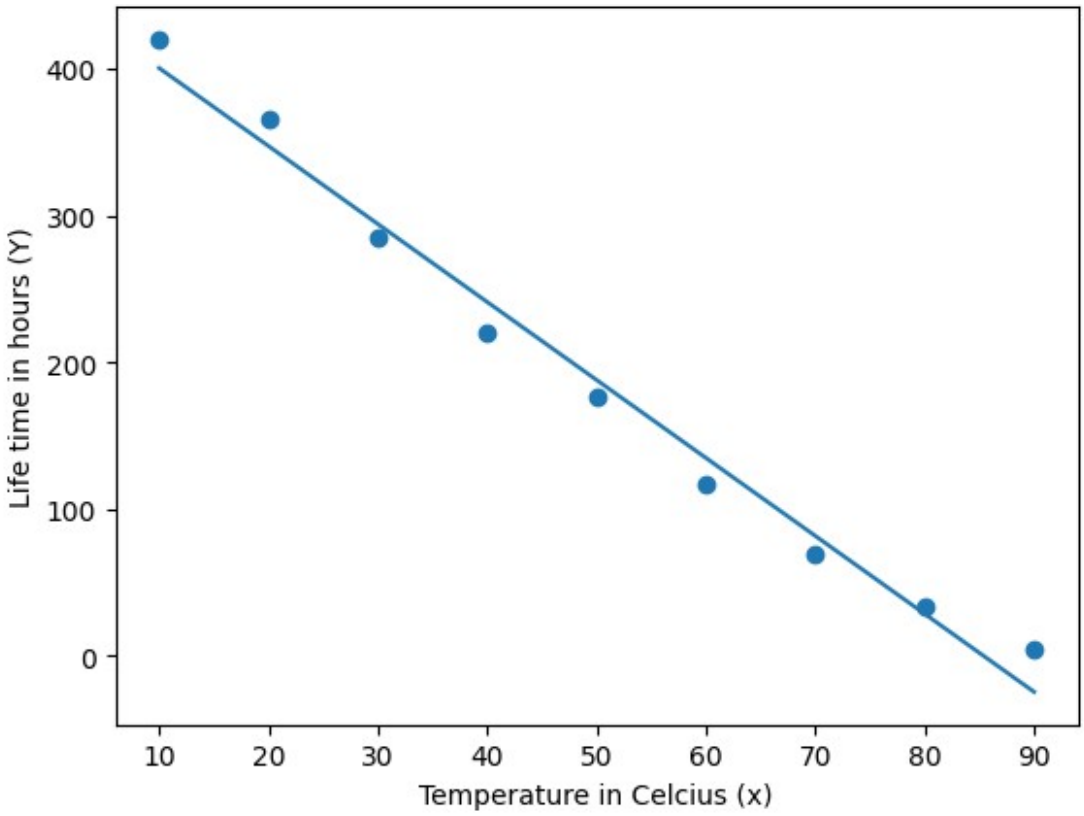
Returns the value of the linear equation

```
def myfunc(x):
  return B0 + B1 * x
```

Performing statsmodels module to fit a linear regression model.

```
import statsmodels.api as sm
mymodel = list(map(myfunc, x))
speed = myfunc(20)
plt.scatter(x, y)
plt.plot(x, mymodel)
plt.xlabel('Temperature in Celcius (x)')
plt.ylabel('Life time in hours (Y)')
plt.show()
```

```
print(speed)
x = sm.add_constant(x)
x = np.array(x)
model = sm.OLS(y, x).fit()
print(model.summary())
```




```
347.2888888888889
                            OLS Regression Results

==============================================================================
========
Dep. Variable:                          y   R-squared:
0.984
Model:                                OLS   Adj. R-squared:
0.982
Method:                     Least Squares   F-statistic:
431.5
Date:                   Thu, 18 Jan 2024   Prob (F-statistic):
1.51e-07
Time:                           15:56:09   Log-Likelihood:
-38.516
No. Observations:                       9   AIC:
81.03
Df Residuals:                           7   BIC:
```

```
81.43
Df Model:                           1

Covariance Type:            nonrobust

==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        453.5556     14.394     31.511      0.000     419.520     487.591
x1            -5.3133      0.256    -20.773      0.000      -5.918      -4.709
==============================================================================
Omnibus:                        2.018   Durbin-Watson:                   0.699
Prob(Omnibus):                  0.365   Jarque-Bera (JB):                0.965
Skew:                           0.415   Prob(JB):                        0.617
Kurtosis:                       1.628   Cond. No.                         123.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

/usr/local/lib/python3.10/dist-packages/scipy/stats/_stats_py.py:1806: UserWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=9
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```