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

```
df = pd.read_csv('/content/temperatures.csv')
df
```

	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	D
0	1901	22.40	24.14	29.07	31.91	33.41	33.18	31.21	30.39	30.47	29.97	27.31	24.
1	1902	24.93	26.58	29.77	31.78	33.73	32.91	30.92	30.73	29.80	29.12	26.31	24.
2	1903	23.44	25.03	27.83	31.39	32.91	33.00	31.34	29.98	29.85	29.04	26.08	23.
3	1904	22.50	24.73	28.21	32.02	32.64	32.07	30.36	30.09	30.04	29.20	26.36	23.
4	1905	22.00	22.83	26.68	30.01	33.32	33.25	31.44	30.68	30.12	30.67	27.52	23.
...	...	...	...	...	...	...	...	...	...	...	...	...	
112	2013	24.56	26.59	30.62	32.66	34.46	32.44	31.07	30.76	31.04	30.27	27.83	25.
113	2014	23.83	25.97	28.95	32.74	33.77	34.15	31.85	31.32	30.68	30.29	28.05	25.
114	2015	24.58	26.89	29.07	31.87	34.09	32.48	31.88	31.52	31.55	31.04	28.10	25.
115	2016	26.94	29.72	32.62	35.38	35.72	34.03	31.64	31.79	31.66	31.98	30.11	28.
116	2017	26.45	29.46	31.60	34.95	35.84	33.82	31.88	31.72	32.22	32.29	29.60	27.

117 rows × 18 columns

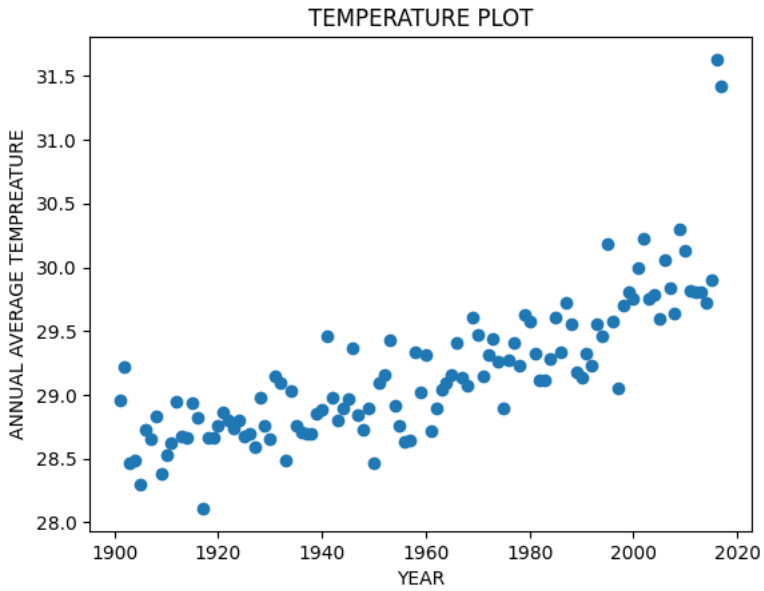
```
df.columns
```

```
Index(['YEAR', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP',
       'OCT', 'NOV', 'DEC', 'ANNUAL', 'JAN-FEB', 'MAR-MAY', 'JUN-SEP',
       'OCT-DEC'],
      dtype='object')
```

```
#input variable
x = df['YEAR']
#output variable
y = df['ANNUAL']
```

```
#Plotting the graph between area vs price
plt.title('TEMPERATURE PLOT')
plt.xlabel('YEAR')
plt.ylabel('ANNUAL AVERAGE TEMPREATURE')
plt.scatter(x,y)
```

```
<matplotlib.collections.PathCollection at 0x7caf0a5fb430>
```




```
type(x)
```

```
pandas.core.series.Series
```

```
x.shape
```

```
(117,)
```

```
x=x.reshape(117,1)
```

```
x.shape
```

```
(117, 1)
```

```
from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y,test_size=0.25)
print(f"x Training dataset: {x_train.shape}")
print(f"y Training dataset: {y_train.shape}")
print(f"x test dataset: {x_test.shape}")
print(f"y test dataset: {y_test.shape}")
```

```
x Training dataset: (87, 1)
y Training dataset: (87,)
x test dataset: (30, 1)
y test dataset: (30,)
```

```
from sklearn.linear_model import LinearRegression
```

```
model = LinearRegression()
```

```
model.fit(x_train,y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
model.coef_ #slope(m)
```

```
array([0.0136786])
```

```
model.intercept_ #intercept value(c)
```

```
2.414164132210292
```

```
model.predict([[2048]])#predicting the tempreature of 2048
```

```
array([30.42793254])
```

```
y_predicted = model.predict(x_train)#storing the predicted value in 'predicted' variable
```

Mean Absolute Error

```
np.mean(abs(y_train - y_predicted))
```

```
0.24814814996665818
```

```
#Another method to claculate MAE
from sklearn.metrics import mean_absolute_error
mean_absolute_error(y_train,y_predicted)#(actual_vlaue,predicted_value)
```

```
0.24814814996665818
```

Mean Squared Error

```
np.mean((y_train - y_predicted)**2)
```

```
0.1290921935808046
```

```
from sklearn.metrics import mean_squared_error
mean_squared_error(y_train ,y_predicted)
```

```
0.1290921935808046
```

r2-Score

```
from sklearn.metrics import r2_score
r2_score(y_train ,y_predicted)
```

```
0.6325395712553645
```

```
model.score(x_train,y_train)
```

```
0.6325395712553645
```

Visualising The Regression Model

```
plt.title('TEMPERATURE PLOT')
plt.xlabel('YEAR')
plt.ylabel('ANNUAL AVERAGE TEMPREATURE')
plt.scatter(x_train,y_train, label = 'actual',color='green')
plt.plot(x_train ,y_predicted ,label = 'predicted',color='red')
```

```
[<matplotlib.lines.Line2D at 0x7caf0a08c1f0>]
```