In [2]:
```
#Importing Necessary Libraries
import pandas as pd
```

In [3]:
```
#Reading CSV file
A= pd.read_csv(r"/Users/harshavardhan/Desktop/archive/temperatures.csv")
A
```

Out[3]:

	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

In [4]:
```
#input data
x1 = A[['YEAR']]
x1
```

Out[4]:

	YEAR
0	1901
1	1902
2	1903
3	1904
4	1905
...	...
112	2013
113	2014
114	2015
115	2016
116	2017

117 rows × 1 columns

In [5]:
```
# output data
y1 = A[['JAN']]
y1
```

Out[5]:

	JAN
0	22.40
1	24.93
2	23.44
3	22.50
4	22.00
...	...
112	24.56
113	23.83
114	24.58
115	26.94
116	26.45

117 rows × 1 columns

In [6]:
```
#splitting data
from sklearn.model_selection import train_test_split
x1_train,x1_test,y1_train,y1_test= train_test_split(x1,y1,test_size =0.2)
```

In [8]:
```
#importing linear regression and creating model
from sklearn.linear_model import LinearRegression
model1= LinearRegression()
model1.fit(x1_train,y1_train)
```

Out[8]:

▾ LinearRegression

LinearRegression()

In [9]:
```
print(model1.coef_)
print(model1.intercept_)
```

```
[[0.01319436]]
[-2.17865729]
```

In [10]:
```
predict = model1.predict(x1_test)
predict
```

Out[10]:
```
array([[24.35519689],
       [23.66909027],
       [23.08853851],
       [24.0385323 ],
       [23.95936615],
       [23.37881439],
       [23.4711749 ],
       [23.53714669],
       [24.42116868],
       [23.28645389],
       [23.99894922],
       [23.69547899],
       [23.49756362],
       [23.57672976],
       [24.14408716],
       [23.89339436],
       [24.40797433],
       [24.10450409],
       [23.61631284],
       [24.18367024],
       [24.15728152],
       [23.60311848],
       [24.1968646 ],
       [23.0621498 ]])
```

In [11]:
```
# calculating errors
from sklearn import metrics
metrics.mean_absolute_error(y1_test,predict)
```

Out[11]:
```
0.5053171608159269
```

In [12]:
```
MSE = metrics.mean_squared_error(y1_test,predict)
MSE
```

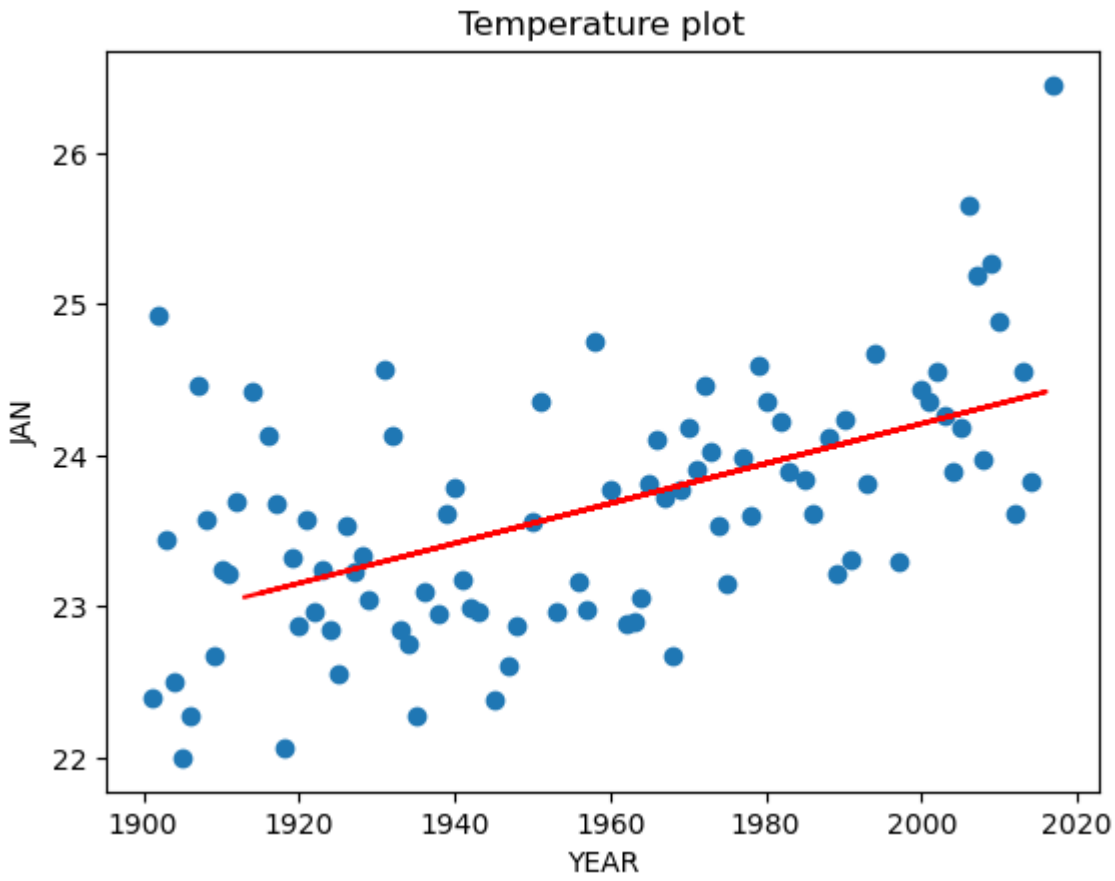
Out[12]:
```
0.5037120617548343
```

In [13]:
```
import math
RMSE = math.sqrt(MSE)
RMSE
```

Out[13]:
```
0.7097267514718847
```

In [14]:
```
#visualization
import matplotlib.pyplot as plt
plt.title('Temperature plot')
plt.xlabel('YEAR')
plt.ylabel('JAN')
plt.scatter(x1_train,y1_train)
plt.plot(x1_test,predict,color ='red')
```

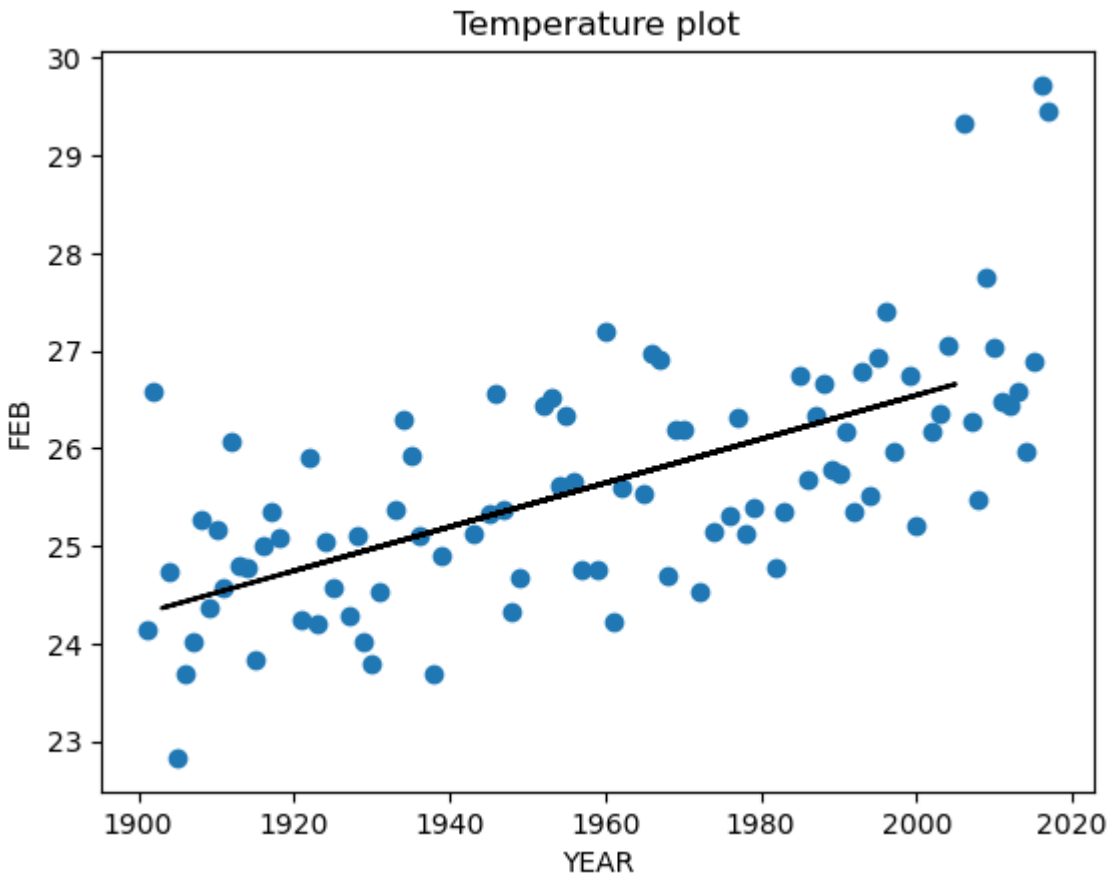
Out[14]:
```
[<matplotlib.lines.Line2D at 0x16ae91e90>]
```



In [15]:
```
#model 2 FEB
x2 = A[['YEAR']]
y2 = A[['FEB']]
x2_train,x2_test,y2_train,y2_test = train_test_split(x2,y2,test_size =0.2)
model2 = LinearRegression()
model2.fit(x2_train,y2_train)
print(model2.coef_)
print(model2.intercept_)
predict2= model2.predict(x2_test)
MAE2 = metrics.mean_absolute_error(y2_test,predict2)
print(MAE2)
MSE2 = metrics.mean_squared_error(y2_test,predict2)
print(MSE2)
RMSE2 = math.sqrt(MSE2)
print(RMSE2)
plt.title('Temperature plot')
plt.xlabel('YEAR')
plt.ylabel('FEB')
plt.scatter(x2_train,y2_train)
plt.plot(x2_test,predict2,color ='black')
```

```
[[0.0224837]]
[-18.42411434]
0.565686085996531
0.48173453504649105
0.6940709870369824
```

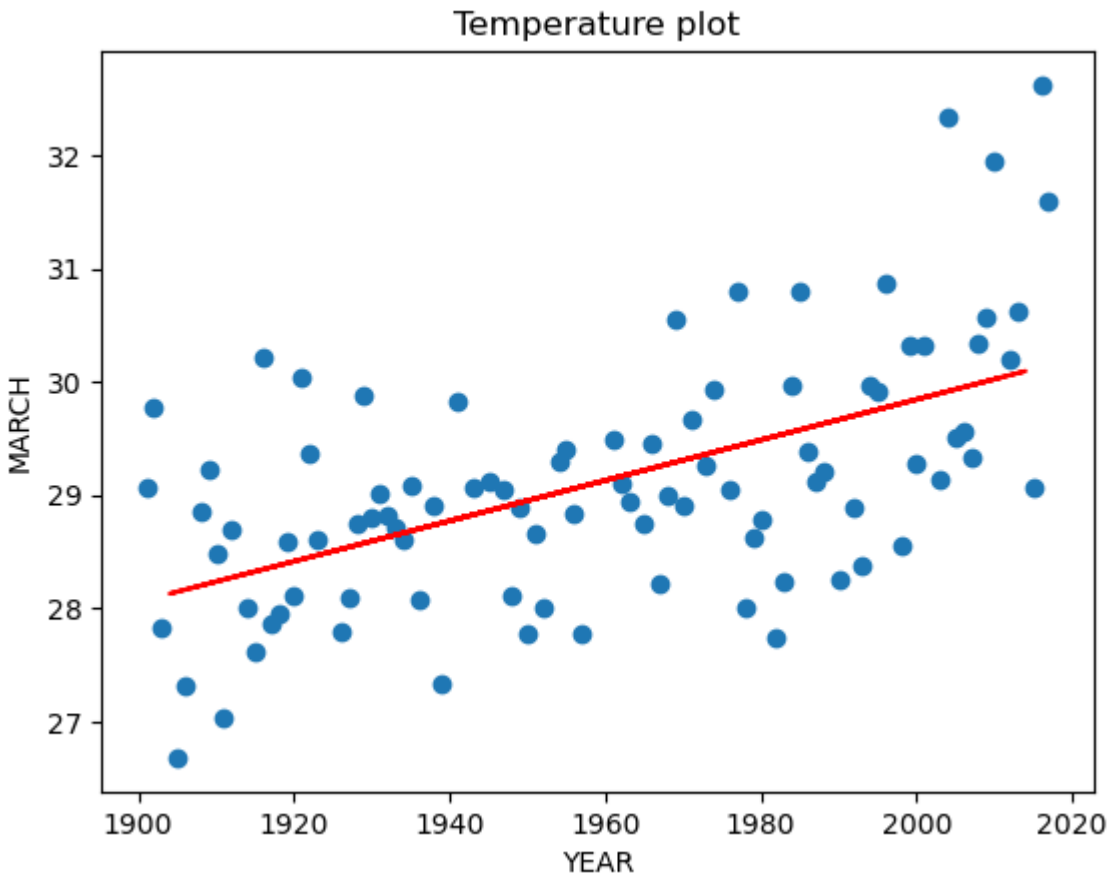
Out[15]:
```
[<matplotlib.lines.Line2D at 0x16aea5550>]
```



In [16]:
```
#model 3 MARCH
x3 = A[['YEAR']]
y3 = A[['MAR']]
x3_train,x3_test,y3_train,y3_test = train_test_split(x3,y3,test_size =0.2)
model3 = LinearRegression()
model3.fit(x3_train,y3_train)
print(model3.coef_)
print(model3.intercept_)
predict3= model3.predict(x3_test)
MAE3 = metrics.mean_absolute_error(y3_test,predict3)
print(MAE3)
MSE3 = metrics.mean_squared_error(y3_test,predict3)
print(MSE3)
RMSE3 = math.sqrt(MSE3)
print(RMSE3)
plt.title('Temperature plot')
plt.xlabel('YEAR')
plt.ylabel('MARCH')
plt.scatter(x3_train,y3_train)
plt.plot(x3_test,predict3,color ='red')
```

```
[[0.01786825]]
[-5.89319033]
0.7412197426734007
0.7156382848646081
0.8459540678220113
```

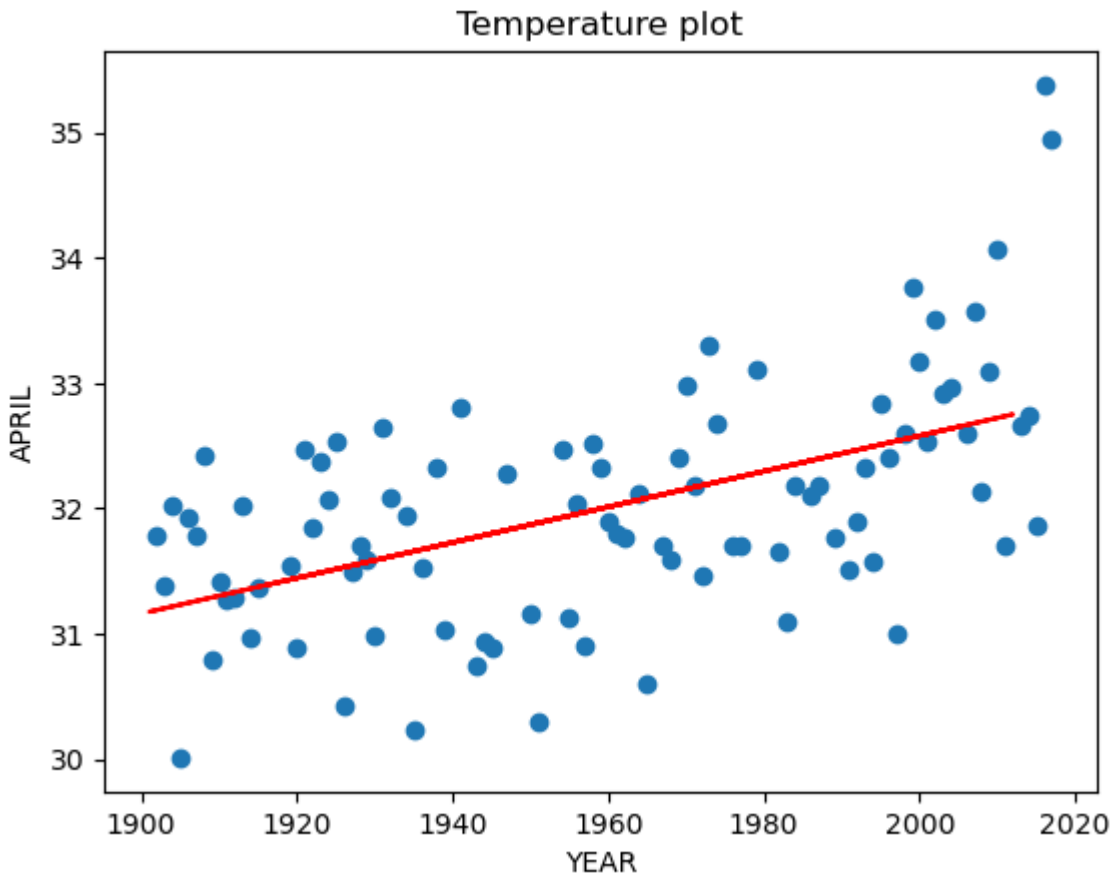
Out[16]:
```
[<matplotlib.lines.Line2D at 0x16adcf590>]
```



In [17]:
```
#model 4 april
x4 = A[['YEAR']]
y4 = A[['APR']]
x4_train,x4_test,y4_train,y4_test = train_test_split(x4,y4,test_size =0.2)
model4 = LinearRegression()
model4.fit(x4_train,y4_train)
print(model4.coef_)
print(model4.intercept_)
predict4= model4.predict(x4_test)
MAE4 = metrics.mean_absolute_error(y4_test,predict4)
print(MAE4)
MSE4 = metrics.mean_squared_error(y4_test,predict4)
print(MSE4)
RMSE4 = math.sqrt(MSE4)
print(RMSE4)
plt.title('Temperature plot')
plt.xlabel('YEAR')
plt.ylabel('APRIL')
plt.scatter(x4_train,y4_train)
plt.plot(x4_test,predict4,color ='red')
```

```
[[0.01419717]]
[4.18577757]
0.4488485147752958
0.31873867275916723
0.564569457869594
```

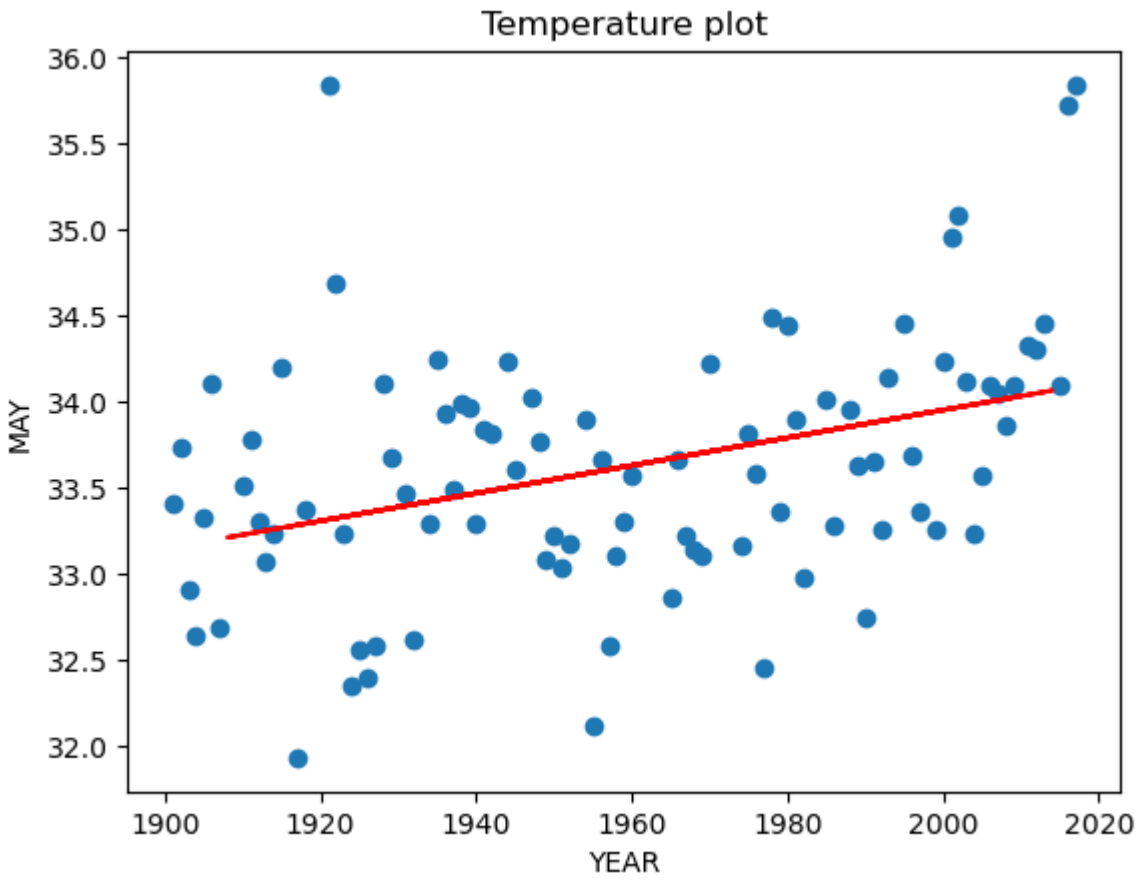
Out[17]:
```
[<matplotlib.lines.Line2D at 0x16b885890>]
```



In [18]:
```
#model 5 May
x5 = A[['YEAR']]
y5 = A[['MAY']]
x5_train,x5_test,y5_train,y5_test = train_test_split(x5,y5,test_size =0.2)
model5 = LinearRegression()
model5.fit(x5_train,y5_train)
print(model5.coef_)
print(model5.intercept_)
predict5= model5.predict(x5_test)
MAE5 = metrics.mean_absolute_error(y5_test,predict5)
print(MAE5)
MSE5 = metrics.mean_squared_error(y5_test,predict5)
print(MSE5)
RMSE5 = math.sqrt(MSE5)
print(RMSE5)
plt.title('Temperature plot')
plt.xlabel('YEAR')
plt.ylabel('MAY')
plt.scatter(x5_train,y5_train)
plt.plot(x5_test,predict5,color ='red')
```

```
[[0.00805879]]
[17.83342088]
0.48417227334560575
0.3482167972355404
0.5900989724067822
```

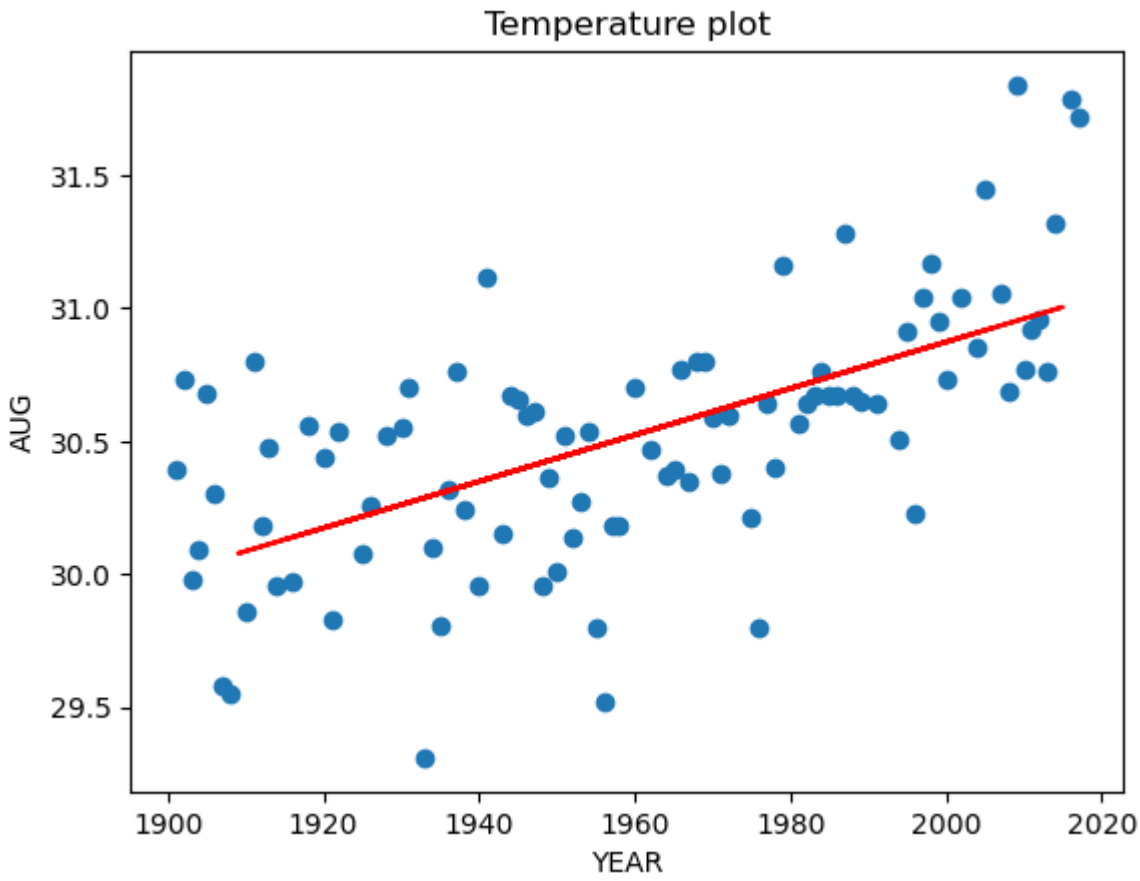
Out[18]:
```
[<matplotlib.lines.Line2D at 0x16ae16610>]
```



In [19]:
```
#model 6 AUG
x6 = A[['YEAR']]
y6 = A[['AUG']]
x6_train,x6_test,y6_train,y6_test = train_test_split(x6,y6,test_size =0.2)
model6 = LinearRegression()
model6.fit(x6_train,y6_train)
print(model6.coef_)
print(model6.intercept_)
predict6= model6.predict(x6_test)
MAE6 = metrics.mean_absolute_error(y6_test,predict6)
print(MAE6)
MSE6 = metrics.mean_squared_error(y6_test,predict6)
print(MSE6)
RMSE6 = math.sqrt(MSE6)
print(RMSE6)
plt.title('Temperature plot')
plt.xlabel('YEAR')
plt.ylabel('AUG')
plt.scatter(x6_train,y6_train)
plt.plot(x6_test,predict6,color ='red')
```

```
[[0.00874363]]
[13.38647985]
0.2531151429536993
0.09271664774391447
0.30449408490792473
```

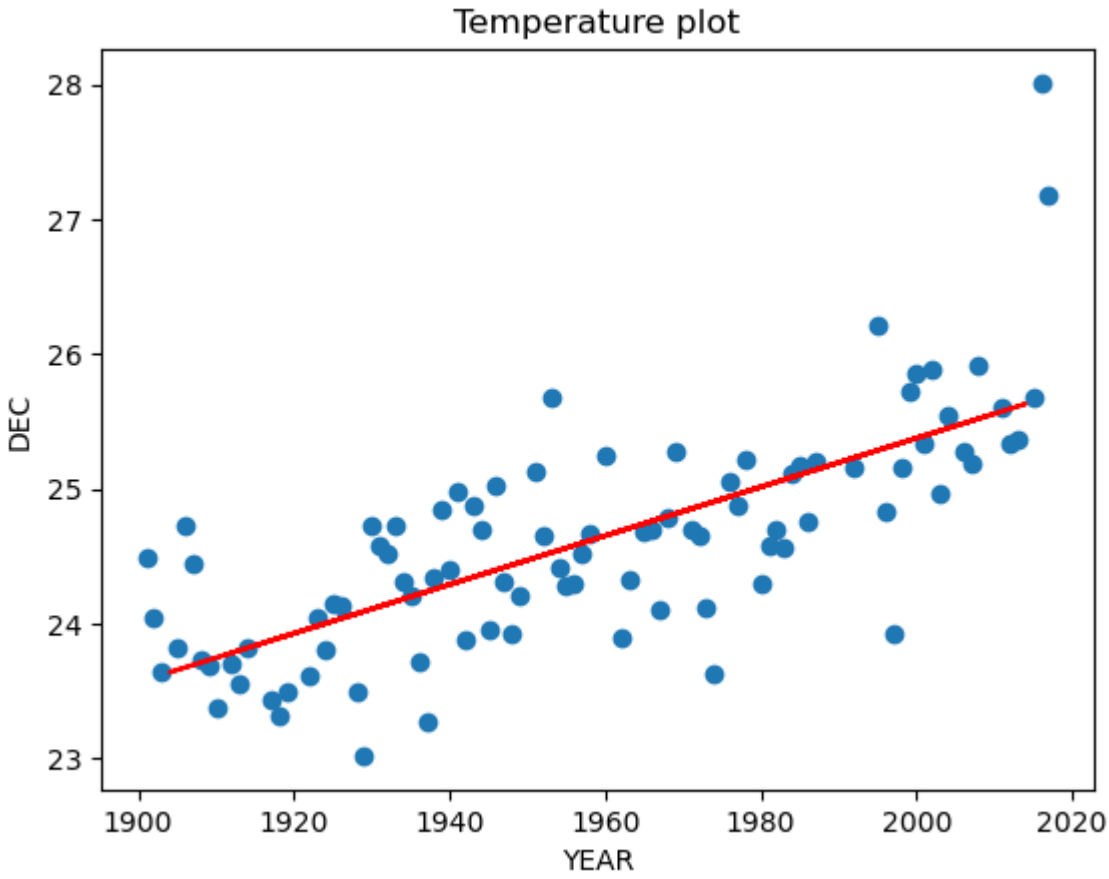
Out[19]:
```
[<matplotlib.lines.Line2D at 0x16b98bfd0>]
```



In [20]:
```
#model 7 DEC
x7 = A[['YEAR']]
y7 = A[['DEC']]
x7_train,x7_test,y7_train,y7_test = train_test_split(x7,y7,test_size =0.2)
model7 = LinearRegression()
model7.fit(x7_train,y7_train)
print(model7.coef_)
print(model7.intercept_)
predict7= model7.predict(x7_test)
MAE7 = metrics.mean_absolute_error(y7_test,predict7)
print(MAE7)
MSE7 = metrics.mean_squared_error(y7_test,predict7)
print(MSE7)
RMSE7 = math.sqrt(MSE7)
print(RMSE7)
plt.title('Temperature plot')
plt.xlabel('YEAR')
plt.ylabel('DEC')
plt.scatter(x7_train,y7_train)
plt.plot(x7_test,predict7,color ='red')
```

```
[[0.0180995]]
[-10.82308471]
0.3100439891258217
0.1696684574243046
0.41190831191456256
```

Out[20]:
```
[<matplotlib.lines.Line2D at 0x16ba050d0>]
```


Temperature plot
DEC
YEAR
28
27
26
25
24
23
1900
1920
1940
1960
1980
2000
2020