ASSIGNMENT 2

Download temperature data from below link. https://www.kaggle.com/venky73/temperatures-of-india?select=temperatures.csv This data consists of temperatures of INDIA averaging the temperatures of all places month wise. Temperatures values are recorded in CELSIUS a. Apply Linear Regression using suitable library function and predict the Month-wise temperature. b. Assess the performance of regression models using MSE, MAE and R-Square metrics c. Visualize simple regression model.

[1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import math
```

1. Read the temperature data

[3]:
```
df=pd.read_csv("/Users/tanmay2/Desktop/TANMAY/CODING/TE ML/temperatures.csv")
```

[4]:
```
df.head()
```

[4]:

	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	JAN-FEB	MAR-MAY	JUN-SEP	OCT-DEC
0	1901	22.40	24.14	29.07	31.91	33.41	33.18	31.21	30.39	30.47	29.97	27.31	24.49	28.96	23.27	31.46	31.27	27.25
1	1902	24.93	26.58	29.77	31.78	33.73	32.91	30.92	30.73	29.80	29.12	26.31	24.04	29.22	25.75	31.76	31.09	26.49
2	1903	23.44	25.03	27.83	31.39	32.91	33.00	31.34	29.98	29.85	29.04	26.08	23.65	28.47	24.24	30.71	30.92	26.26
3	1904	22.50	24.73	28.21	32.02	32.64	32.07	30.36	30.09	30.04	29.20	26.36	23.63	28.49	23.62	30.95	30.66	26.40
4	1905	22.00	22.83	26.68	30.01	33.32	33.25	31.44	30.68	30.12	30.67	27.52	23.82	28.30	22.25	30.00	31.33	26.57

2. Check if data has null or duplicated values

[5]:
```
df.isnull().sum()
```

[5]:
```
YEAR        0
JAN         0
FEB         0
MAR         0
APR         0
MAY         0
JUN         0
JUL         0
AUG         0
SEP         0
OCT         0
NOV         0
DEC         0
ANNUAL      0
JAN-FEB     0
MAR-MAY     0
JUN-SEP     0
OCT-DEC     0
dtype: int64
```

[6]:
```
df.duplicated().sum()
```

[6]:
```
0
```

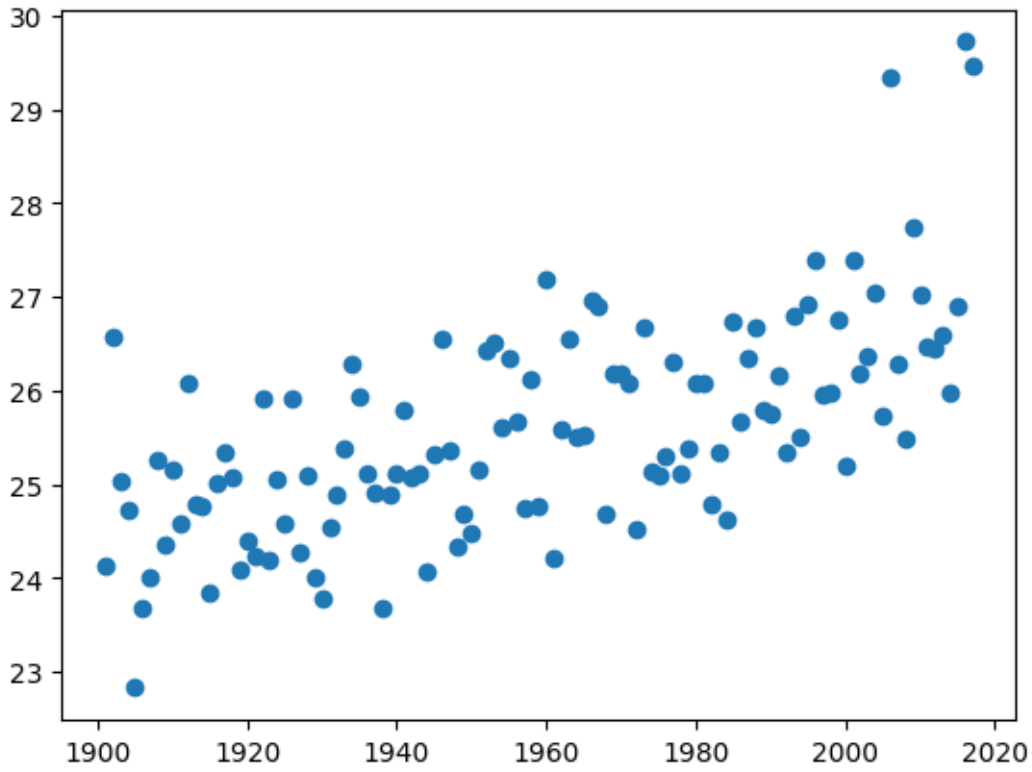
2 1) February Month Prediction

[7]:
```
X=df[['YEAR']]
Y=df[['FEB']]
```

3. Visualize the Whole Dataset

[8]:
```
plt.scatter(X,Y)
```

[8]:
```
<matplotlib.collections.PathCollection at 0x1663ade80>
```



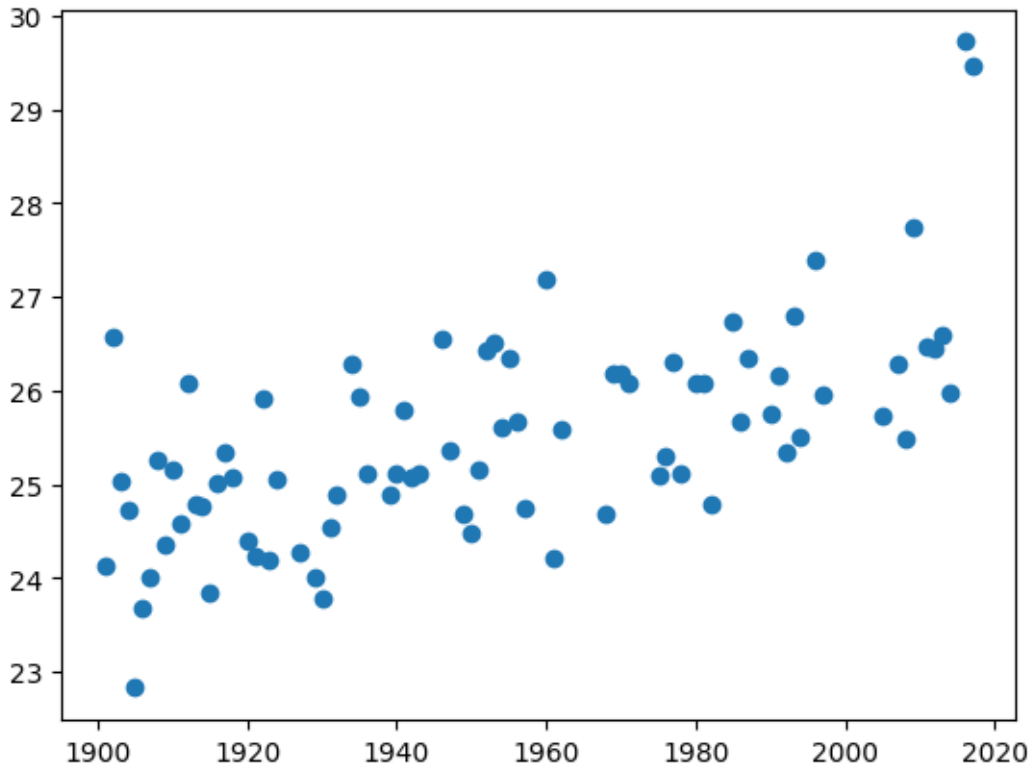
4. Split the dataset into training and testing data

[9]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

[10]:
```
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.3)
```

[11]:
```
plt.scatter(X_train,Y_train)
```

[11]:
```
<matplotlib.collections.PathCollection at 0x30264d430>
```



5. Train the model on Training data

[12]:
```
model=LinearRegression()
model.fit(X_train,Y_train)
```

[12]:
```
LinearRegression()
```

6. Make Predictions on test data

[13]:
```
Y_pred=model.predict(X_test)
```

7. Find Y intercept and Coefficient

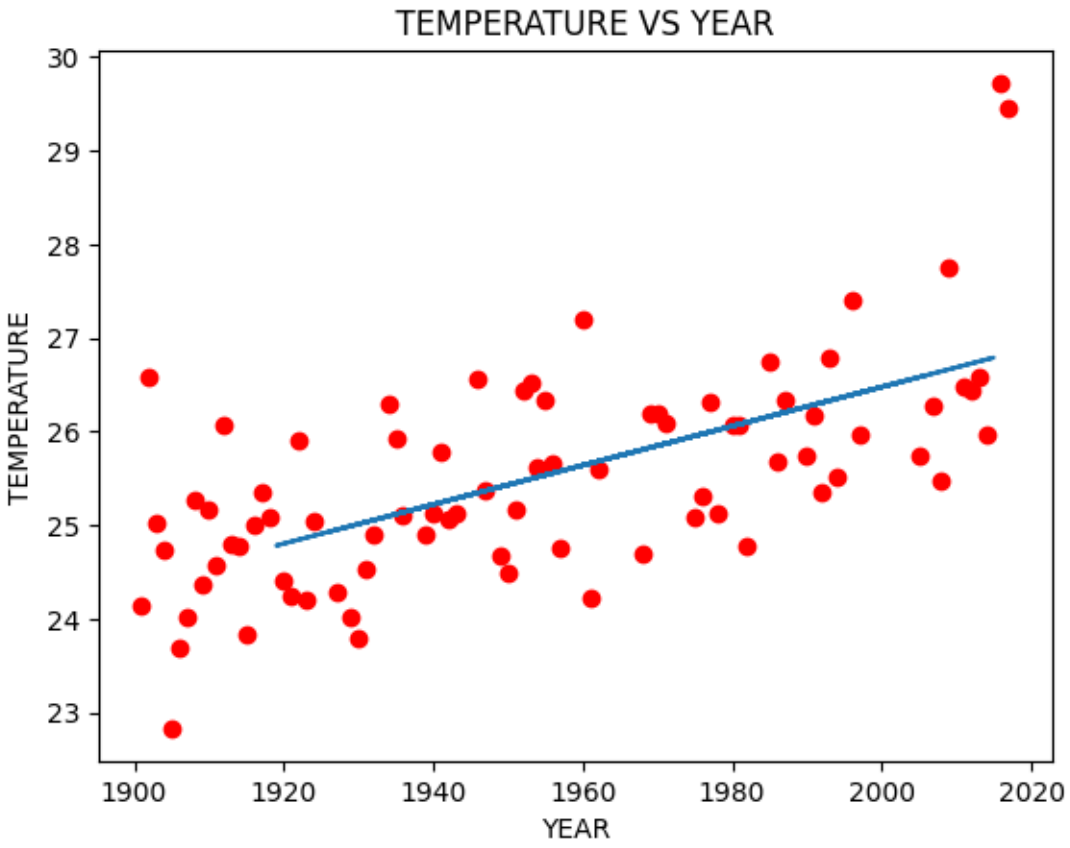
[14]:
```
print(f"Y Intercept={model.intercept_}")
print(f"Coefficient={model.coef_}")
```

```
Y Intercept=[-15.32430879]
Coefficient=[[0.02089956]]
```

8. Visualize the model with Training data

[15]:
```
plt.xlabel("YEAR")
plt.ylabel("TEMPERATURE")
plt.title("  TEMPERATURE VS YEAR")
```

```
plt.scatter(X_train,Y_train,color="red")
plt.plot(X_test,Y_pred)
plt.show()
```



9. Find the Performance using MAE, MSE and RMSE

[16]:
```
from sklearn.metrics import mean_absolute_error,mean_squared_error
```

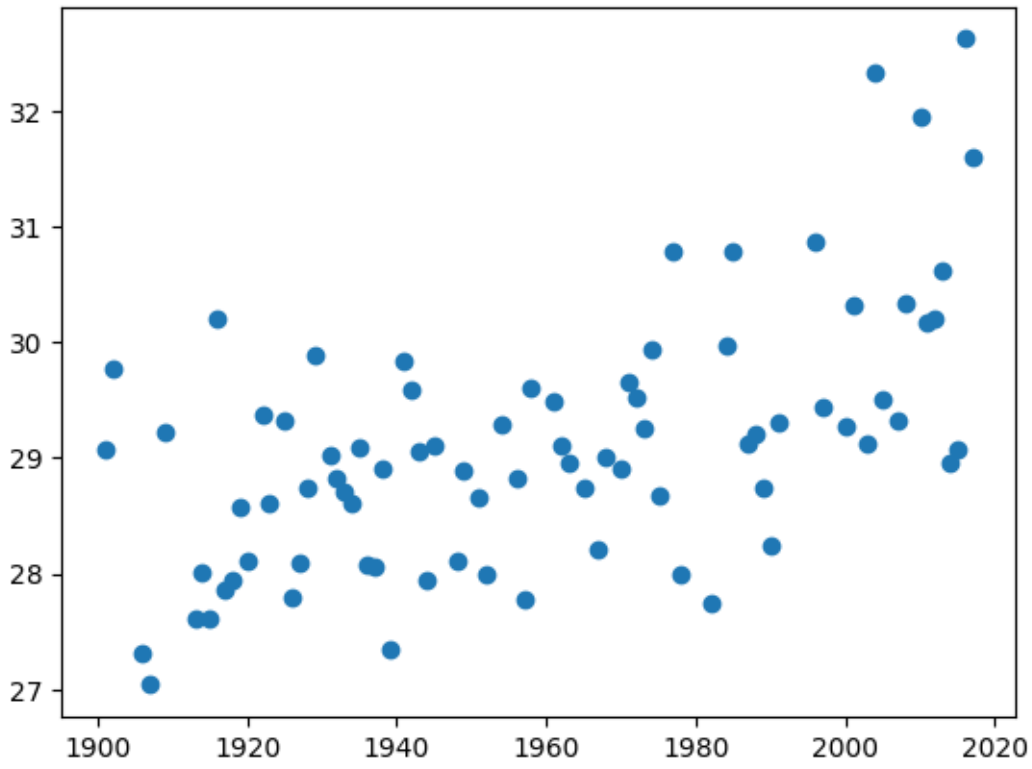
[17]:
```
print(f"MAE={mean_absolute_error(Y_test,Y_pred)}")
print(f"MSE={mean_squared_error(Y_test,Y_pred)}")
print(f"RMSE={math.sqrt(mean_squared_error(Y_test,Y_pred))}")
```

```
MAE=0.7190307308499649
MSE=0.8042309205310956
RMSE=0.8967892285989476
```

3 2) March Month Prediction

[18]:
```
X=df[['YEAR']]
Y=df[['MAR']]
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.3)
plt.scatter(X_train,Y_train)
```

[18]: <matplotlib.collections.PathCollection at 0x303e6f940>

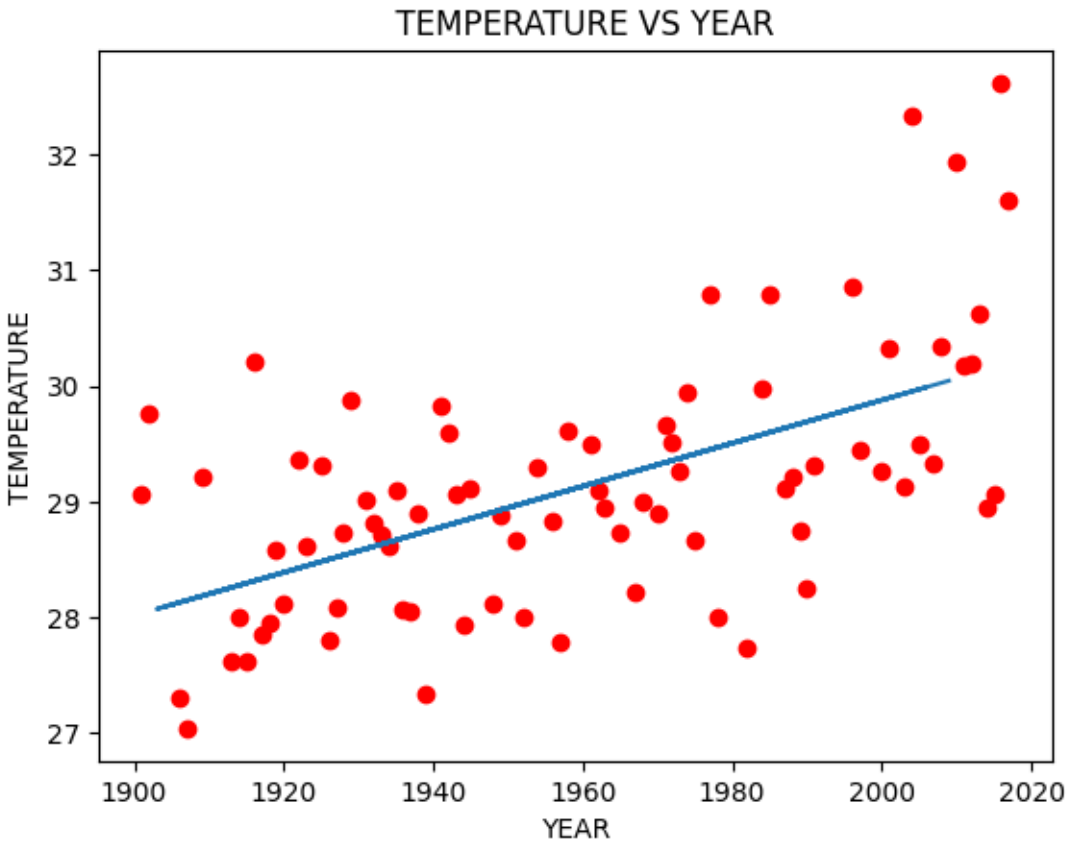


[19]:
```
model=LinearRegression()
model.fit(X_train,Y_train)
Y_pred=model.predict(X_test)
print(f"Y Intercept={model.intercept_}")
print(f"Coefficient={model.coef_}")
```

```
Y Intercept=[-7.36191071]
Coefficient=[[0.01862007]]
```

[20]:
```
plt.xlabel("YEAR")
plt.ylabel("TEMPERATURE")
plt.title("  TEMPERATURE VS YEAR")
plt.scatter(X_train,Y_train,color="red")
```

```
plt.plot(X_test,Y_pred)
plt.show()
```



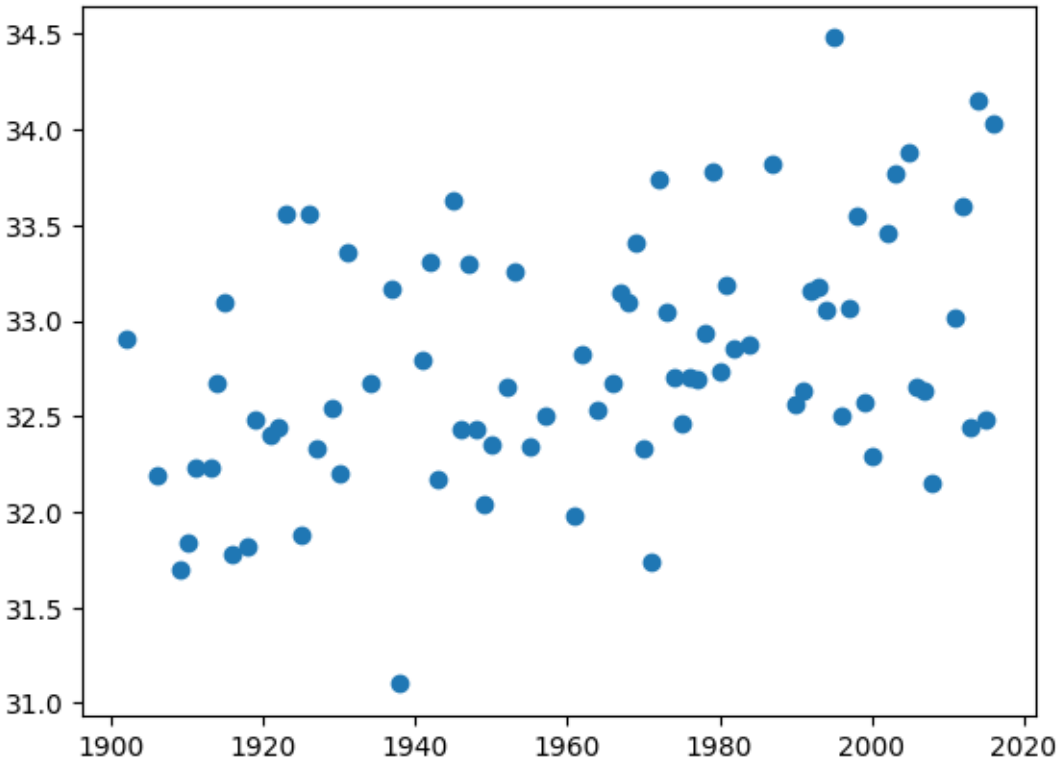
[21]:
```
print(f"MAE={mean_absolute_error(Y_test,Y_pred)}")
print(f"MSE={mean_squared_error(Y_test,Y_pred)}")
print(f"RMSE={math.sqrt(mean_squared_error(Y_test,Y_pred))}")
```

```
MAE=0.7381588945520998
MSE=0.7780352335415439
RMSE=0.8820630553092811
```

4 3) June Month Prediction

[22]:
```
X=df[['YEAR']]
Y=df[['JUN']]
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.3)
plt.scatter(X_train,Y_train)
```

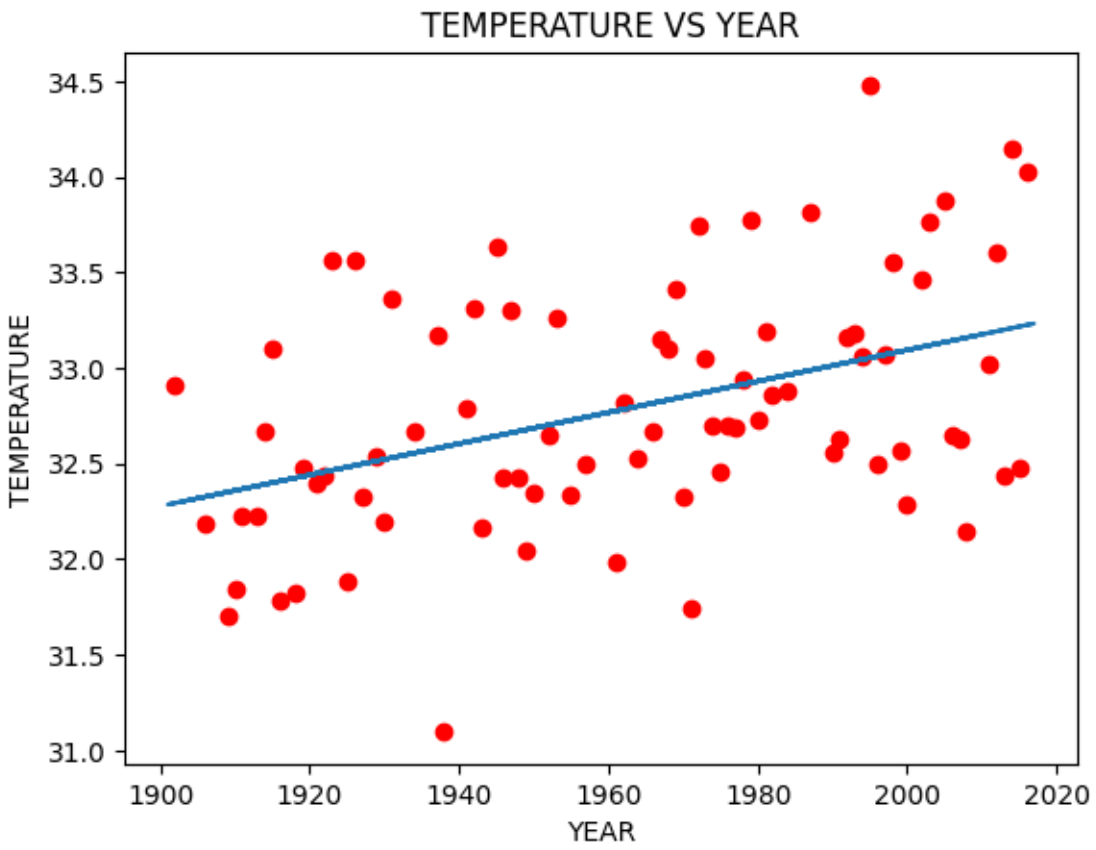
[22]: <matplotlib.collections.PathCollection at 0x303ec7400>



[23]:
```
model=LinearRegression()
model.fit(X_train,Y_train)
Y_pred=model.predict(X_test)
print(f"Y Intercept={model.intercept_}")
print(f"Coefficient={model.coef_}")
```

```
Y Intercept=[16.76423687]
Coefficient=[[0.00816493]]
```

[24]:
```
plt.xlabel("YEAR")
plt.ylabel("TEMPERATURE")
plt.title("  TEMPERATURE VS YEAR")
plt.scatter(X_train,Y_train,color="red")
plt.plot(X_test,Y_pred)
plt.show()
```



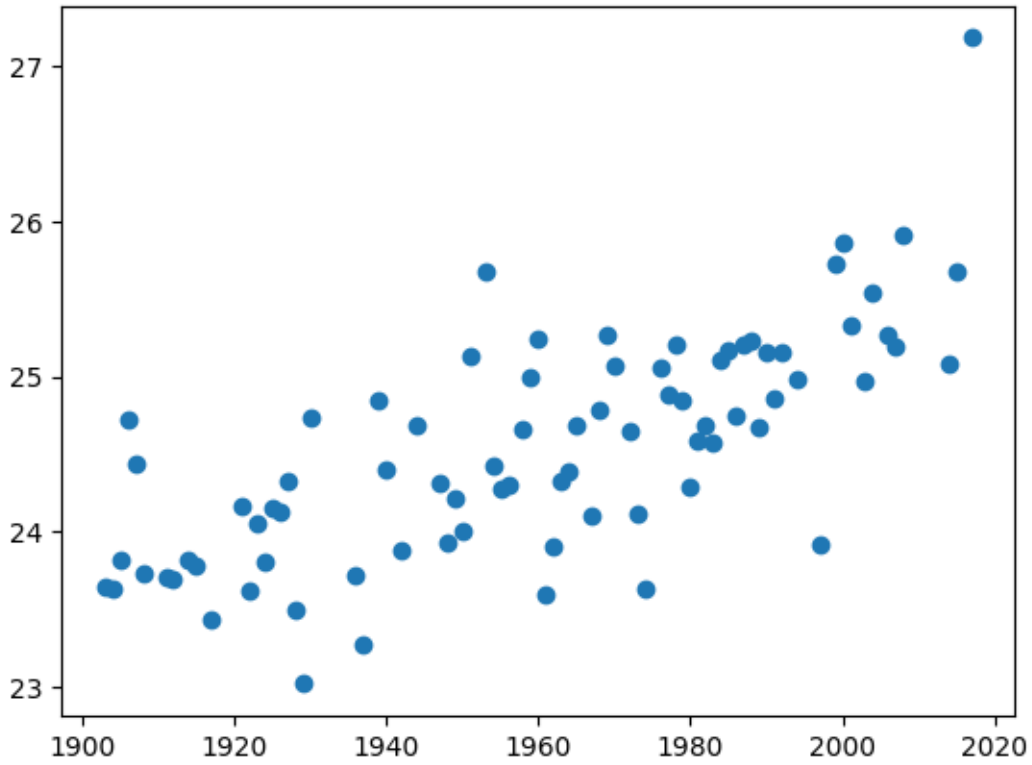
[25]:
```
print(f"MAE={mean_absolute_error(Y_test,Y_pred)}")
print(f"MSE={mean_squared_error(Y_test,Y_pred)}")
print(f"RMSE={math.sqrt(mean_squared_error(Y_test,Y_pred))}")
```

```
MAE=0.5181536440987545
MSE=0.38147720961682574
RMSE=0.6176384133267827
```

5 4) December Month Prediction

[26]:
```
X=df[['YEAR']]
Y=df[['DEC']]
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.3)
plt.scatter(X_train,Y_train)
```

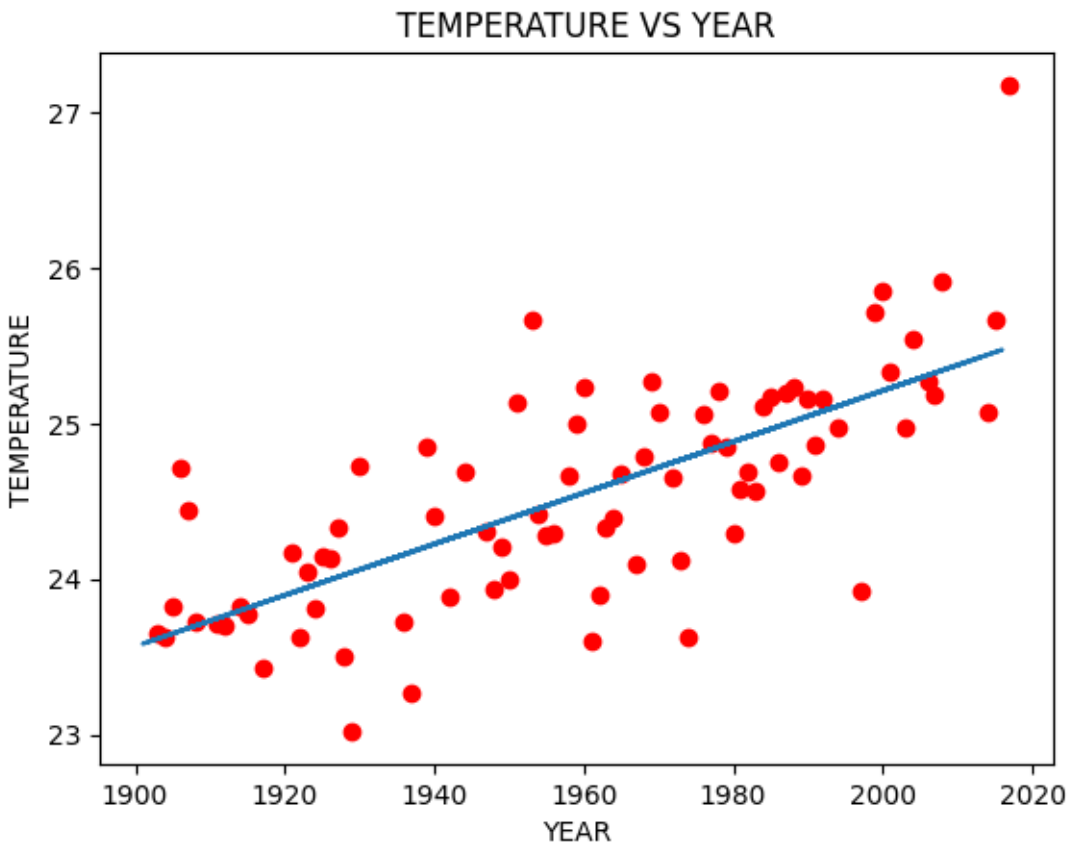
[26]: `<matplotlib.collections.PathCollection at 0x303fc0490>`



[27]:
```
model=LinearRegression()
model.fit(X_train,Y_train)
Y_pred=model.predict(X_test)
print(f"Y Intercept={model.intercept_}")
print(f"Coefficient={model.coef_}")
```

```
Y Intercept=[-7.70627673]
Coefficient=[[0.01645939]]
```

[28]:
```
plt.xlabel("YEAR")
plt.ylabel("TEMPERATURE")
plt.title("  TEMPERATURE VS YEAR")
plt.scatter(X_train,Y_train,color="red")
plt.plot(X_test,Y_pred)
plt.show()
```



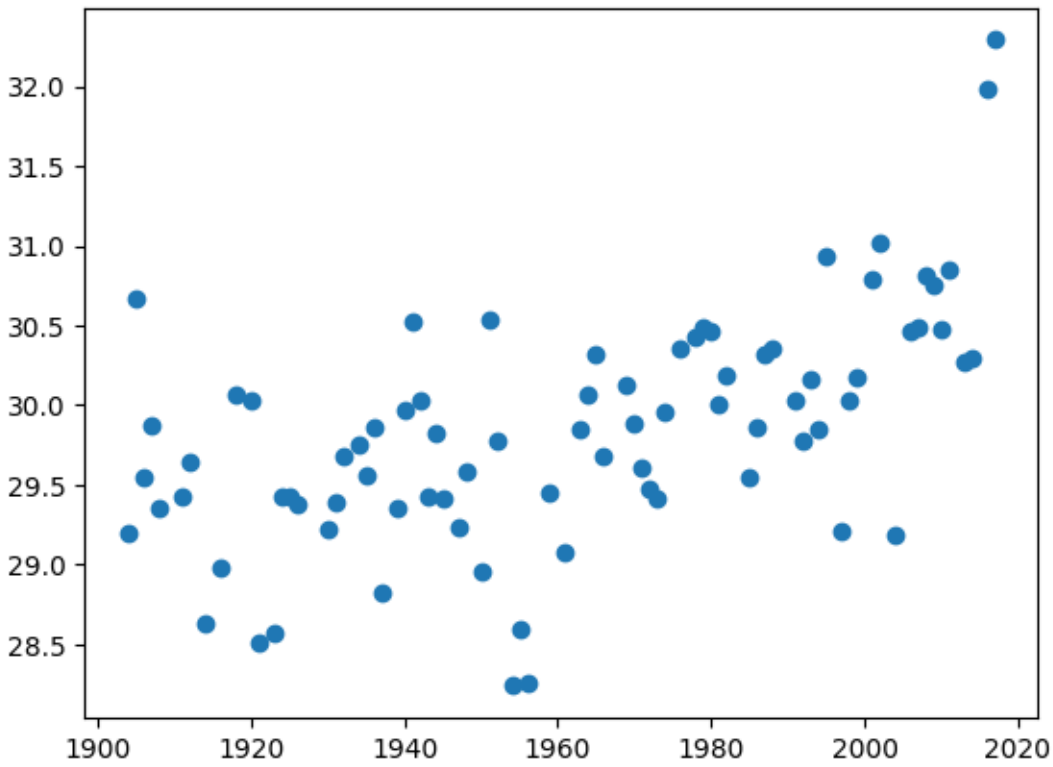
[29]:
```
print(f"MAE={mean_absolute_error(Y_test,Y_pred)}")
print(f"MSE={mean_squared_error(Y_test,Y_pred)}")
print(f"RMSE={math.sqrt(mean_squared_error(Y_test,Y_pred))}")
```

```
MAE=0.40819707933928373
MSE=0.3753159717298199
RMSE=0.6126303712107488
```

6 5) October Month Prediction

[30]:
```
X=df[['YEAR']]
Y=df[['OCT']]
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.3)
plt.scatter(X_train,Y_train)
```

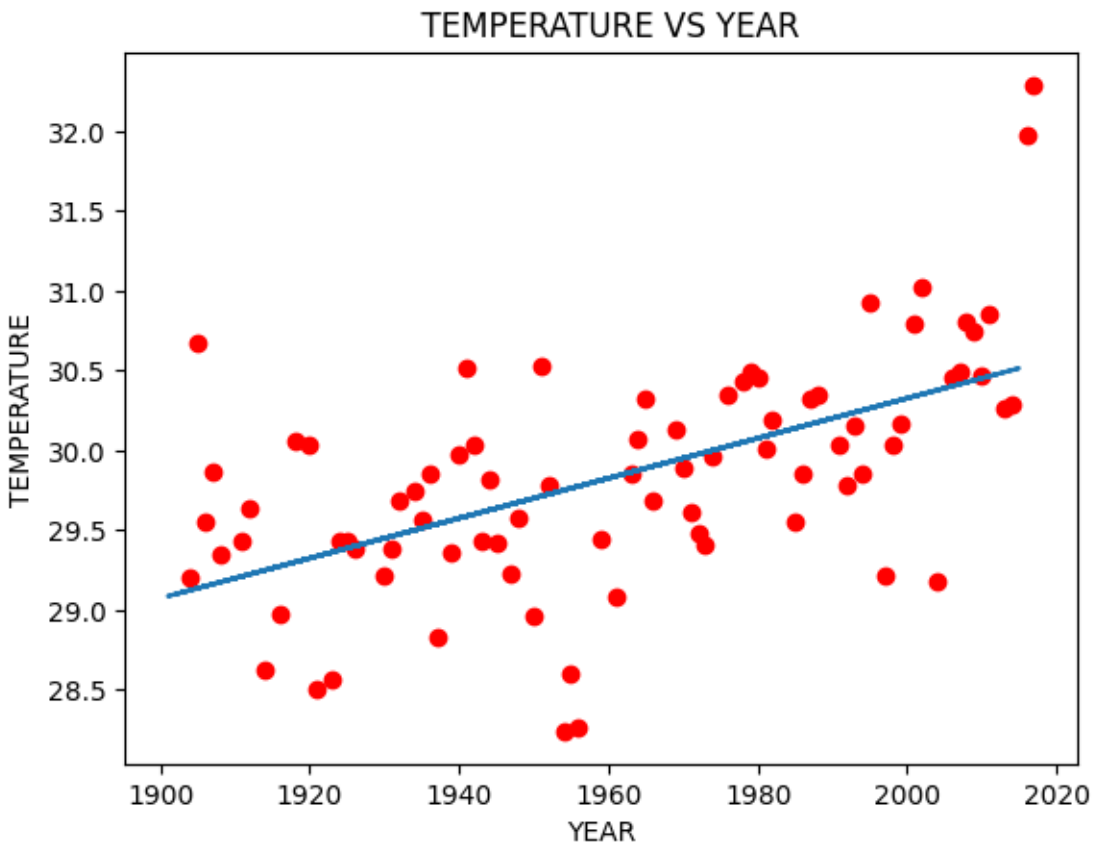
[30]: `<matplotlib.collections.PathCollection at 0x303f0faf0>`



[31]:
```
model=LinearRegression()
model.fit(X_train,Y_train)
Y_pred=model.predict(X_test)
print(f"Y Intercept={model.intercept_}")
print(f"Coefficient={model.coef_}")
```

```
Y Intercept=[5.26315549]
Coefficient=[[0.012532]]
```

[32]:
```
plt.xlabel("YEAR")
plt.ylabel("TEMPERATURE")
plt.title("  TEMPERATURE VS YEAR")
plt.scatter(X_train,Y_train,color="red")
plt.plot(X_test,Y_pred)
plt.show()
```



[33]:
```
print(f"MAE={mean_absolute_error(Y_test,Y_pred)}")
print(f"MSE={mean_squared_error(Y_test,Y_pred)}")
print(f"RMSE={math.sqrt(mean_squared_error(Y_test,Y_pred))}")
```

```
MAE=0.4089071147910718
MSE=0.26890176161508733
RMSE=0.5185573850742918
```