```
# !pip install matplotlib
# !pip install numpy
# !pip install pandas


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

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

df.head()
```

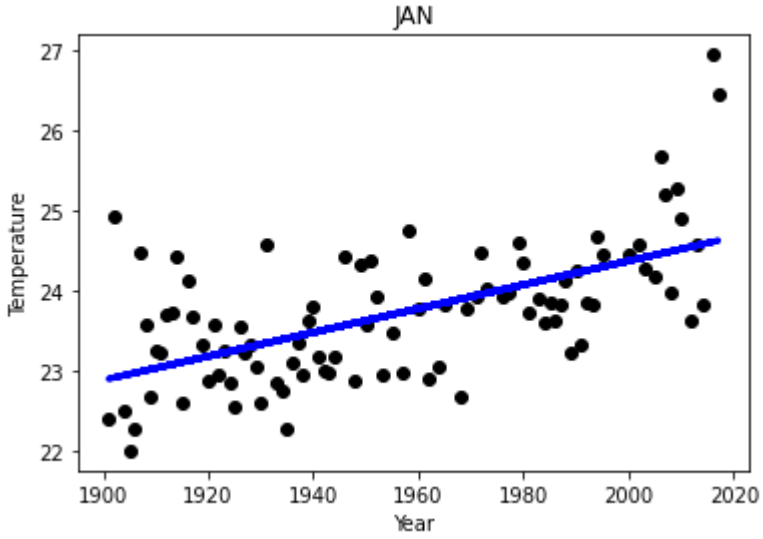
	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL
0	1901	22.40	24.14	29.07	31.91	33.41	33.18	31.21	30.39	30.47	29.97	27.31	24.49	28.96
1	1902	24.93	26.58	29.77	31.78	33.73	32.91	30.92	30.73	29.80	29.12	26.31	24.04	29.22
2	1903	23.44	25.03	27.83	31.39	32.91	33.00	31.34	29.98	29.85	29.04	26.08	23.65	28.47
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
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

```
df.dtypes
df.columns
```

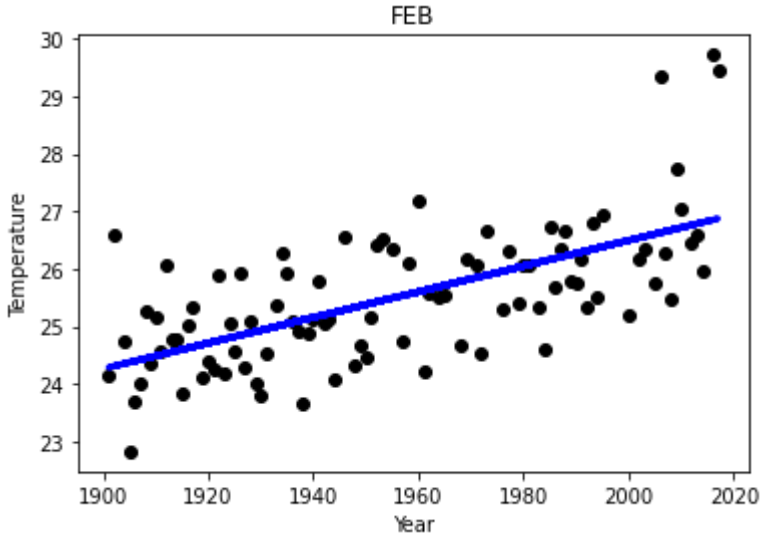
```
     Index(['YEAR', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP',
            'OCT', 'NOV', 'DEC', 'ANNUAL', 'JAN-FEB', 'MAR-MAY', 'JUN-SEP',
            'OCT-DEC'],
           dtype='object')
```

```
from sklearn import linear_model, metrics


def predictLinear(label, df):
  X=df[["YEAR"]]
  Y=df[[label]]

  from sklearn.model_selection import train_test_split
  X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=1)

  reg = linear_model.LinearRegression()

  model = reg.fit(X_train, Y_train)

  r_sq = reg.score(X_train, Y_train)

  Y_pred = model.predict(X_test)

  plt.scatter(X_train, Y_train, color='black')
  plt.plot(X_train, reg.predict(X_train), color='blue', linewidth=3)
  plt.title(label)
  plt.xlabel("Year")
  plt.ylabel("Temperature")
  plt.show()

  print(label + ' slope : ', model.coef_)
```

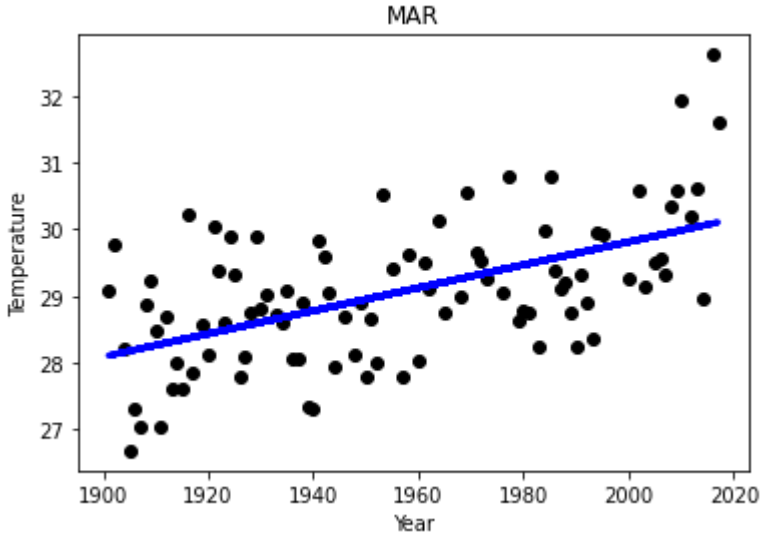
```
  print(label + '  slope :  ', model.coef_)
  #Model Evaluation using R-Square
  from sklearn import metrics
  r_square = metrics.r2_score(Y_test, Y_pred)
  print('R-Square Error:', r_square)

  print('Mean Squared Error:', metrics.mean_squared_error(Y_test, Y_pred))

  print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(Y_test, Y_pred)))

  print('Mean Absolute Error:', metrics.mean_absolute_error(Y_test, Y_pred))

  print('--------------------------------------------------------------------------------')


for label in df.columns:
  if(label == 'YEAR'):
    print('--------------------------------------------------------------------------------')
    continue

  predictLinear(label, df)
```



JAN slope : [[0.01486008]]
R-Square Error: 0.13772507149336943
Mean Squared Error: 0.4474127538283697
Root Mean Squared Error: 0.6688891939838538
Mean Absolute Error: 0.575735016242341

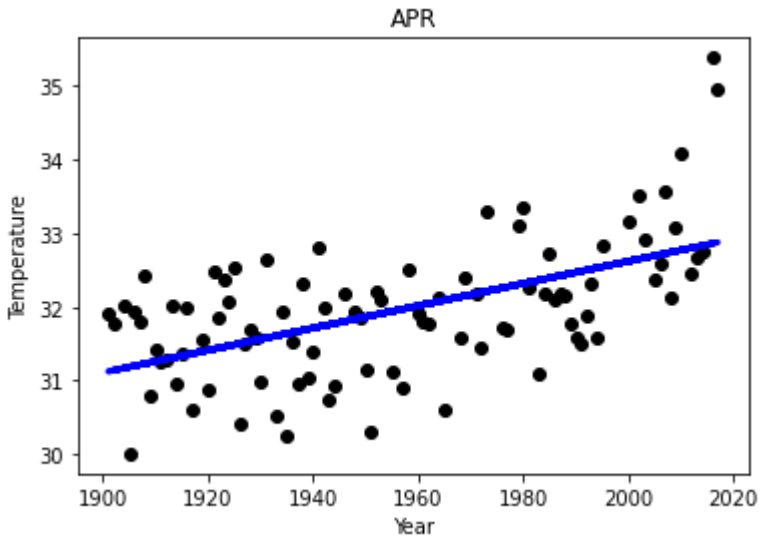


FEB slope : [[0.02228699]]
R-Square Error: 0.3619088707621523
Mean Squared Error: 0.4809982998658562
Root Mean Squared Error: 0.6935404096848692
Mean Absolute Error: 0.5735691679328223

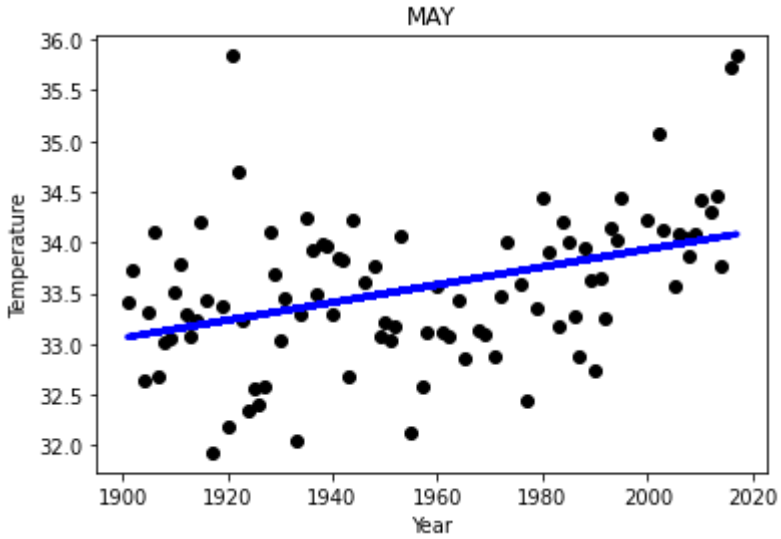


MAR slope : [[0.01723055]]
R-Square Error: 0.2906200173574891
Mean Squared Error: 0.766404157590963

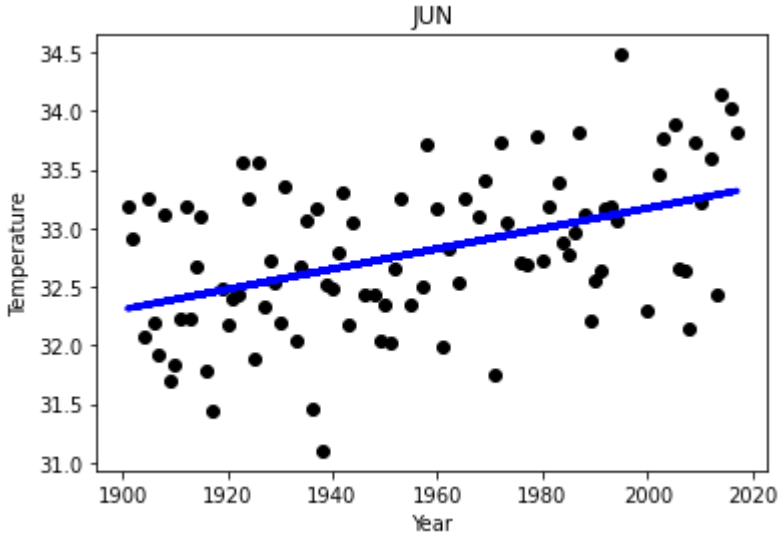
Root Mean Squared Error: 0.8754451196910992
Mean Absolute Error: 0.661365976919735



APR slope : [[0.01509518]]
R-Square Error: 0.11774988363579952
Mean Squared Error: 0.4593360178322951
Root Mean Squared Error: 0.6777433273978395
Mean Absolute Error: 0.5369162476615214

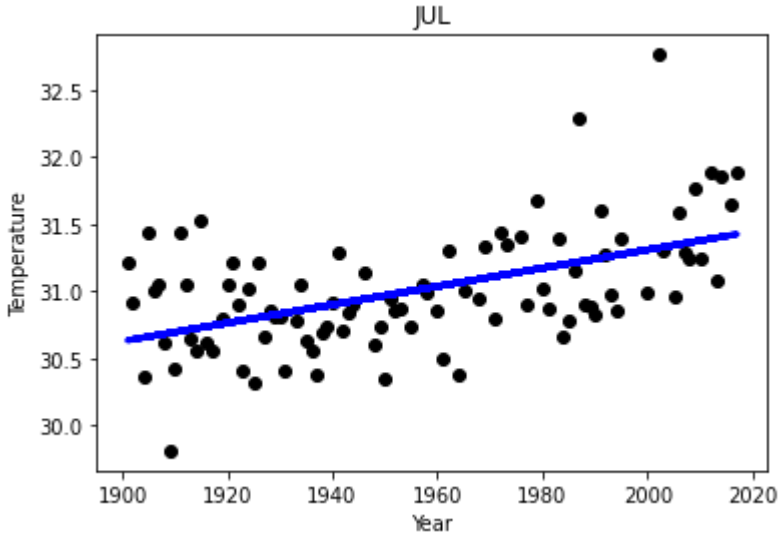


MAY slope : [[0.00874355]]
R-Square Error: 0.19965721218710397
Mean Squared Error: 0.2683015805678108
Root Mean Squared Error: 0.5179783591693873
Mean Absolute Error: 0.43834801587371847

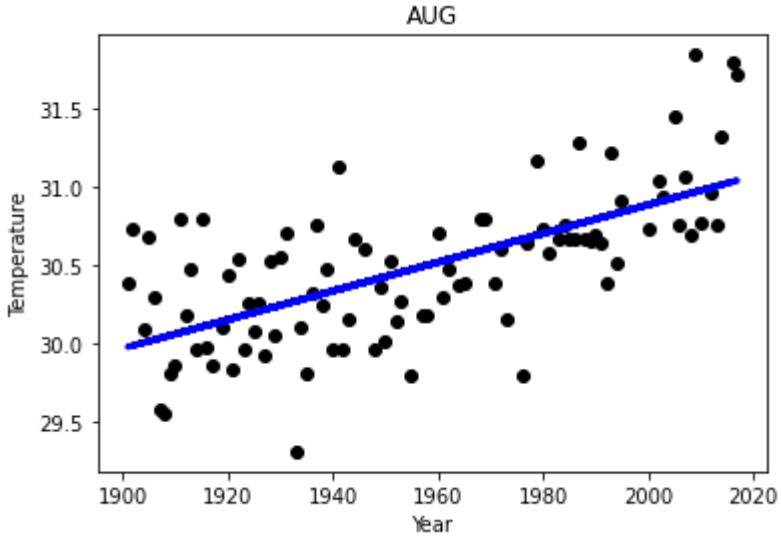


JUN slope : [[0.00865567]]
R-Square Error: -0.4312743365517233

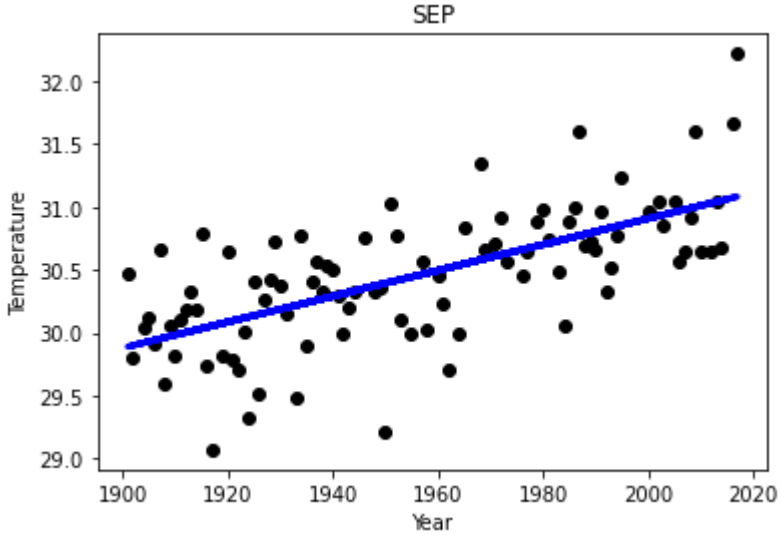
Mean Squared Error: 0.33658180268421645
Root Mean Squared Error: 0.5801567052824749
Mean Absolute Error: 0.4732476590314845



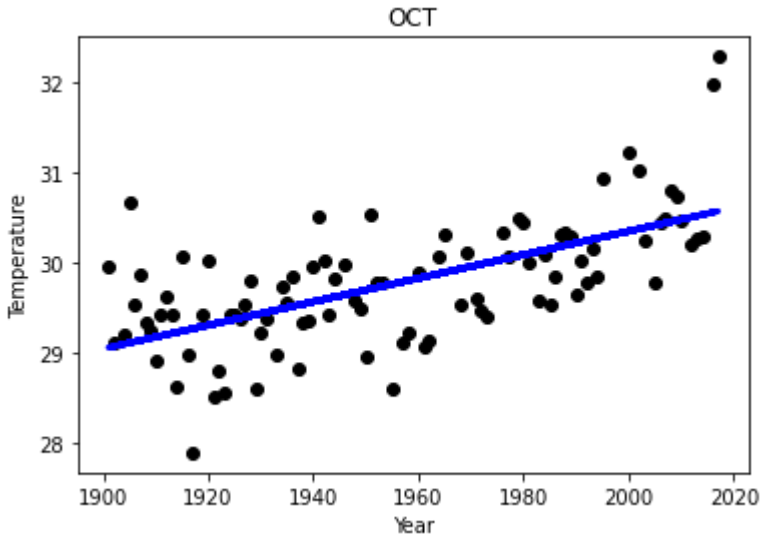
JUL slope : [[0.00681605]]
R-Square Error: 0.050539783681266326
Mean Squared Error: 0.25306015893332984
Root Mean Squared Error: 0.503050851240041
Mean Absolute Error: 0.3750678079512535



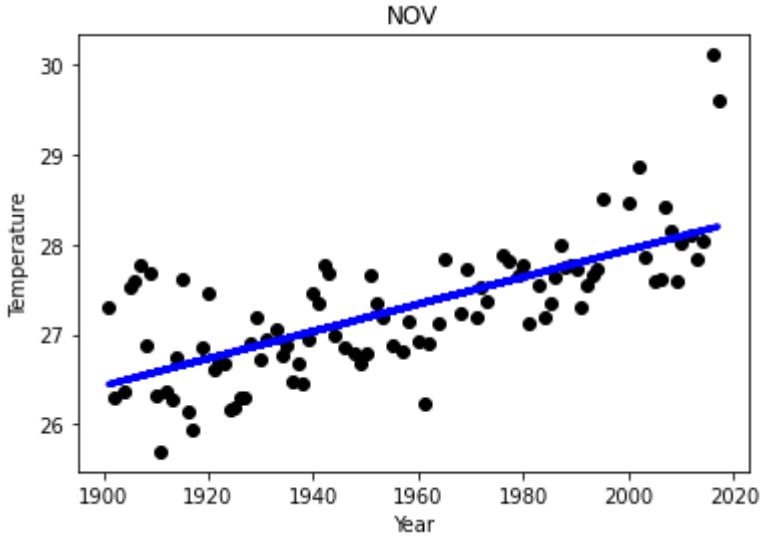
AUG slope : [[0.00915435]]
R-Square Error: 0.3884997566094218
Mean Squared Error: 0.11215424047319361
Root Mean Squared Error: 0.334894372113348
Mean Absolute Error: 0.25436576068595773



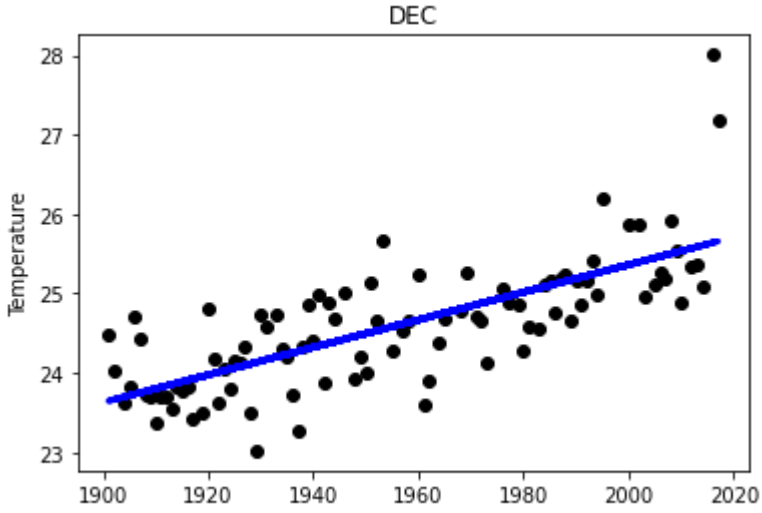
SEP slope : [[0.01028789]]
R-Square Error: 0.4519806690343964
Mean Squared Error: 0.1537224668876904
Root Mean Squared Error: 0.3920745680195164
Mean Absolute Error: 0.3004007404913646



OCT slope : [[0.01302672]]
R-Square Error: 0.08551960971161066
Mean Squared Error: 0.4303682841339444
Root Mean Squared Error: 0.6560246063479208
Mean Absolute Error: 0.4873641919879457

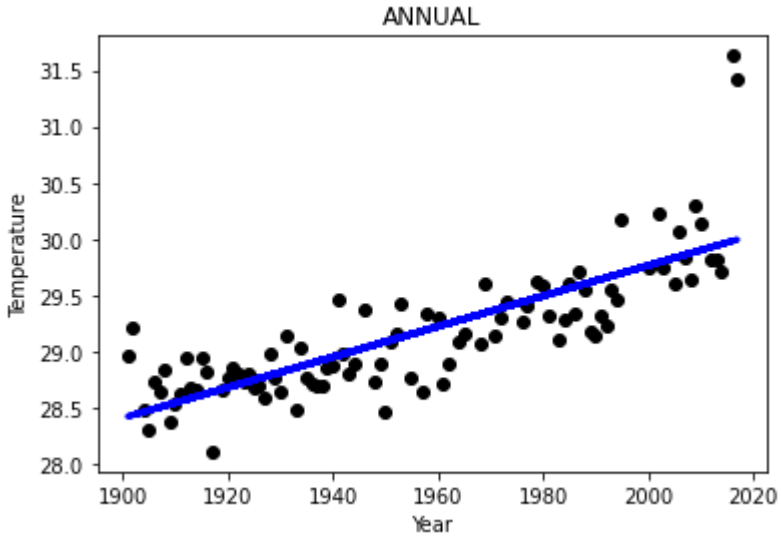


NOV slope : [[0.01511686]]
R-Square Error: 0.4089650689541704
Mean Squared Error: 0.21693403570835565
Root Mean Squared Error: 0.4657617799995569
Mean Absolute Error: 0.3706911585197501

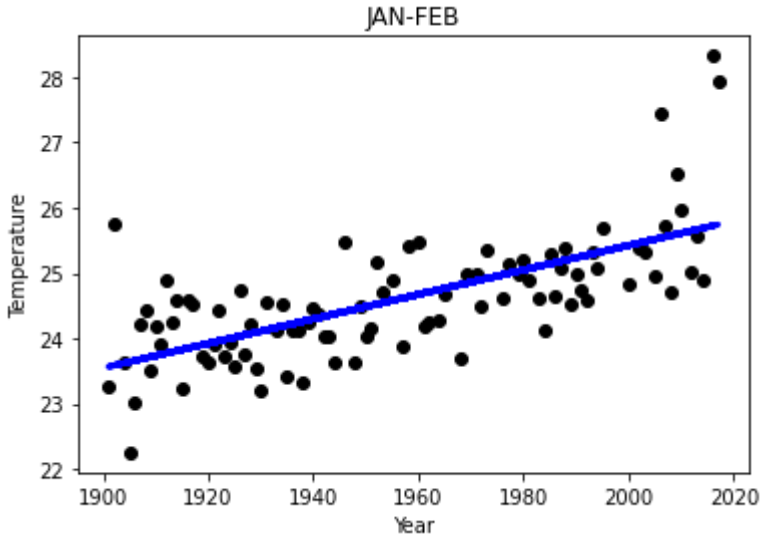


Year

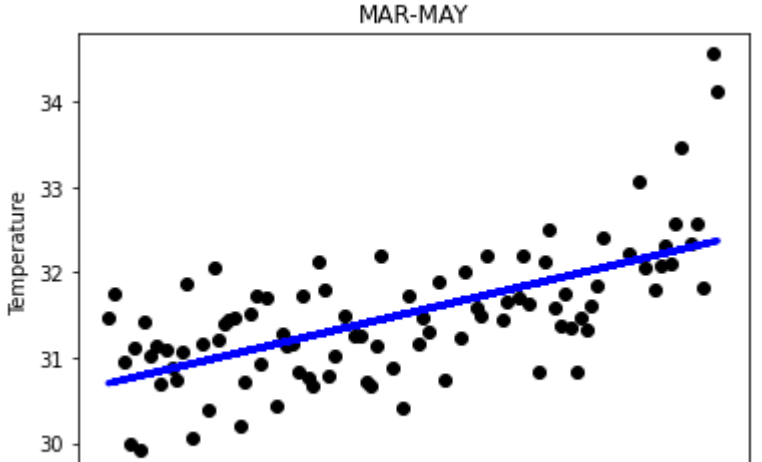
```
DEC slope :  [[0.01733832]]
R-Square Error: 0.45437806942738346
Mean Squared Error: 0.2449038244279631
Root Mean Squared Error: 0.4948775852955588
Mean Absolute Error: 0.3561237905861276
```

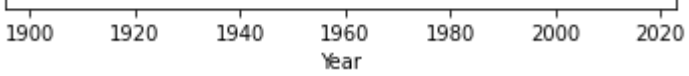



```
ANNUAL slope :  [[0.01355333]]
R-Square Error: 0.6173955861493867
Mean Squared Error: 0.06858077839268387
Root Mean Squared Error: 0.26187932028452315
Mean Absolute Error: 0.19674009476355758
```



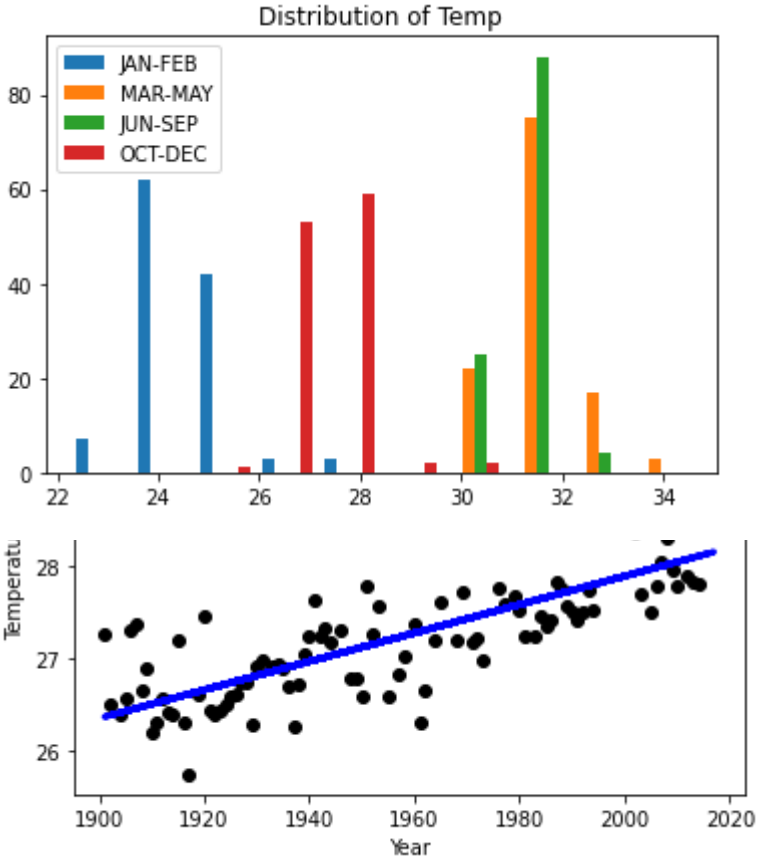

```
JAN-FEB slope :  [[0.01877599]]
R-Square Error: 0.4536678557176955
Mean Squared Error: 0.26487462820769275
Root Mean Squared Error: 0.5146597207939366
Mean Absolute Error: 0.4522009522737003
```






```
MAR-MAY slope :  [[0.01442118]]
R-Square Error: 0.39660003472754746
Mean Squared Error: 0.1858367136100738
Root Mean Squared Error: 0.4310878258662309
Mean Absolute Error: 0.33389470491091017
--------------------------------------------------------------------------------
```


```
plt.title('Distribution of Temp')

plt.hist([df["JAN-FEB"], df["MAR-MAY"], df["JUN-SEP"], df["OCT-DEC"]],
         stacked=False);

plt.legend(['JAN-FEB',  'MAR-MAY',  'JUN-SEP',  'OCT-DEC']);
```




```
OCT-DEC slope :  [[0.01537362]]
R-Square Error: 0.4251696087088045
Mean Squared Error: 0.18565065618624113
Root Mean Squared Error: 0.43087197192001375
Mean Absolute Error: 0.3360041612104608
--------------------------------------------------------------------------------
```