In [ ]:
```
#Import Libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

In [ ]:
```
raw_data=pd.read_csv('temperatures.csv')
raw_data
```

Out[ ]:

	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	AI
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	
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	
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	
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	
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	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
112	2013	24.56	26.59	30.62	32.66	34.46	32.44	31.07	30.76	31.04	30.27	27.83	25.37	
113	2014	23.83	25.97	28.95	32.74	33.77	34.15	31.85	31.32	30.68	30.29	28.05	25.08	
114	2015	24.58	26.89	29.07	31.87	34.09	32.48	31.88	31.52	31.55	31.04	28.10	25.67	
115	2016	26.94	29.72	32.62	35.38	35.72	34.03	31.64	31.79	31.66	31.98	30.11	28.01	
116	2017	26.45	29.46	31.60	34.95	35.84	33.82	31.88	31.72	32.22	32.29	29.60	27.18	

117 rows × 18 columns

In [ ]:
```
raw_data.describe()
```

Out[ ]:

	YEAR	JAN	FEB	MAR	APR	MAY	JUN	
count	117.000000	117.000000	117.000000	117.000000	117.000000	117.000000	117.000000	117
mean	1959.000000	23.687436	25.597863	29.085983	31.975812	33.565299	32.774274	31
std	33.919021	0.834588	1.150757	1.068451	0.889478	0.724905	0.633132	0
min	1901.000000	22.000000	22.830000	26.680000	30.010000	31.930000	31.100000	29
25%	1930.000000	23.100000	24.780000	28.370000	31.460000	33.110000	32.340000	30
50%	1959.000000	23.680000	25.480000	29.040000	31.950000	33.510000	32.730000	31
75%	1988.000000	24.180000	26.310000	29.610000	32.420000	34.030000	33.180000	31
max	2017.000000	26.940000	29.720000	32.620000	35.380000	35.840000	34.480000	32

In [ ]:
```
raw_data.shape
```

Out[ ]:
```
(117, 18)
```

In [ ]:
```
raw_data.isnull().sum()
```

Out[ ]:
```
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
```

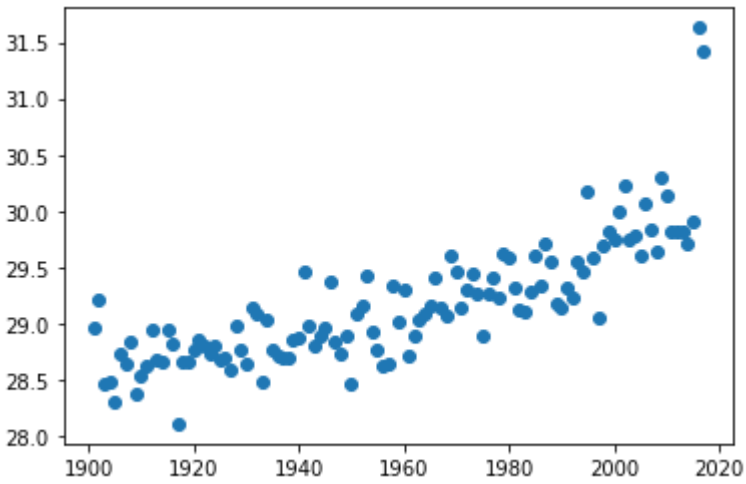
In [ ]:
```
column_names=list(raw_data)
print(column_names)
```

```
['YEAR', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP', 'OC
T', 'NOV', 'DEC', 'ANNUAL', 'JAN-FEB', 'MAR-MAY', 'JUN-SEP', 'OCT-DEC']
```

In [ ]:
```
x=raw_data['YEAR']
y=raw_data['ANNUAL']
```

In [ ]:
```
x = raw_data.iloc[:,0:1]
y = raw_data.iloc[:,13:14]

plt.scatter(x,y)
plt.plot()
```

Out[ ]:
```
[]
```



In [ ]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.30 , rand
y_test = np.array(y_test)
```

In [ ]:
```
x_train,x_test, y_train, y_test = train_test_split(x, y,test_size=0.40, rand
y_test = np.array(y_test)
print(x_train.shape)
print(x_test.shape)
```

```
print(y_train.shape)
print(y_test.shape)
```

```
(70, 1)
(47, 1)
(70, 1)
(47, 1)
```

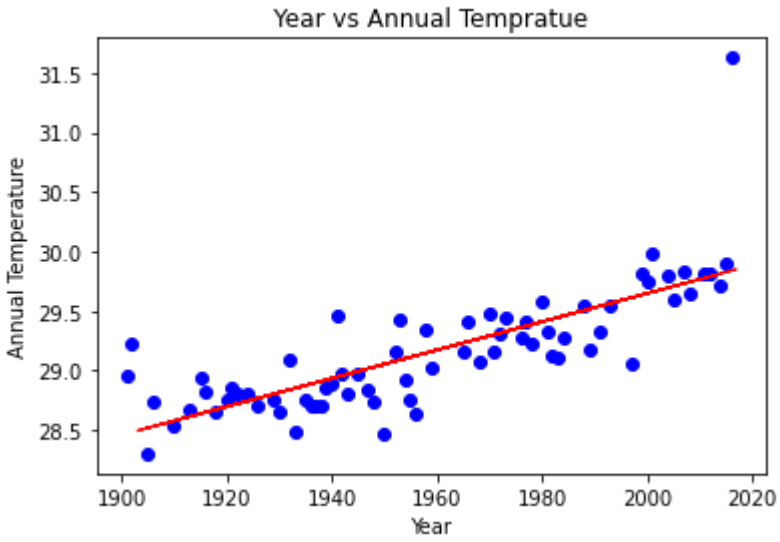
In [ ]:
```
model = LinearRegression()
model.fit(x_train,y_train)
```

Out[ ]:
```
▾ LinearRegression

LinearRegression()
```

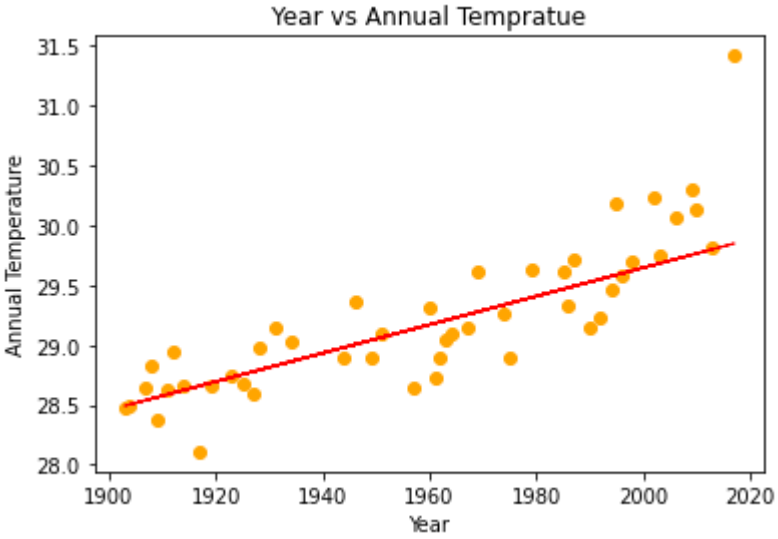
In [ ]:
```
y_pred = model.predict(x_test)
y_pred = np.array(y_pred)
print(y_pred.shape)
```

```
(47, 1)
```

In [ ]:
```
plt.scatter(x_train, y_train, color='blue')
plt.plot(x_test, y_pred, color='red', linewidth=1)
plt.title("Year vs Annual Tempratue")
plt.xlabel("Year")
plt.ylabel("Annual Temperature")
plt.show()
```



In [ ]:
```
plt.scatter(x_test, y_test, color='orange')
plt.plot(x_test, y_pred, color='red', linewidth=1)
plt.title("Year vs Annual Tempratue")
plt.xlabel("Year")
plt.ylabel("Annual Temperature")
plt.show()
```



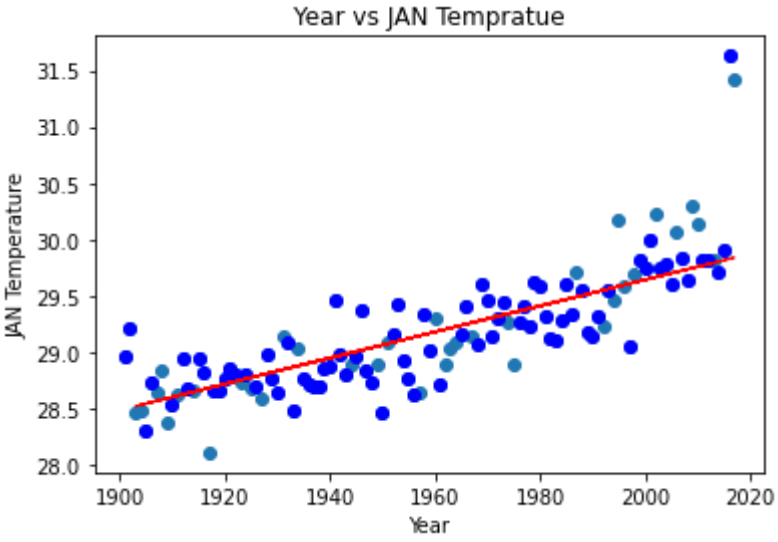
In [ ]:
```
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score
print(f"MSE:  {mean_squared_error(y_test,y_pred)}")
print(f"MAE:  {mean_absolute_error(y_test,y_pred)}")
print(f"R-Sqaure :  {r2_score(y_test,y_pred)}")
```

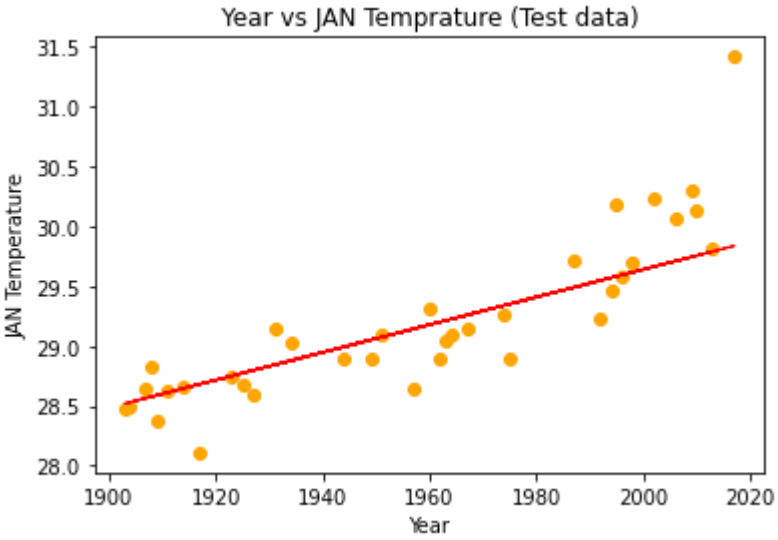
```
MSE:  0.12739407224025404
MAE:  0.2467987767427756
R-Sqaure :  0.6672373588443201
```

In [ ]:
```
def predict_monthwise(x,y,month):
  x = raw_data.iloc[:,0:1]
  y = raw_data.iloc[:,13:14]

  plt.scatter(x,y)
  plt.plot()
  x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.30 , ra
  y_test = np.array(y_test)

  model = LinearRegression()
  model.fit(x_train,y_train)
  y_pred = model.predict(x_test)
  y_pred = np.array(y_pred)

  plt.scatter(x_train, y_train, color='blue')
  plt.plot(x_test, y_pred, color='red', linewidth=1)
  plt.title("Year vs "+month+" Tempratue")
  plt.xlabel("Year")
  plt.ylabel(month+" Temperature")
  plt.show()
  plt.scatter(x_test, y_test, color='orange')
  plt.plot(x_test, y_pred, color='red', linewidth=1)
  plt.title("Year vs "+month+" Temprature (Test data)")
  plt.xlabel("Year")
  plt.ylabel(month+" Temperature")
  plt.show()
  print(f"MSE:  {mean_squared_error(y_test,y_pred)}")
  print(f"MAE:  {mean_absolute_error(y_test,y_pred)}")
  print(f"R-Sqaure :  {r2_score(y_test,y_pred)}")
```

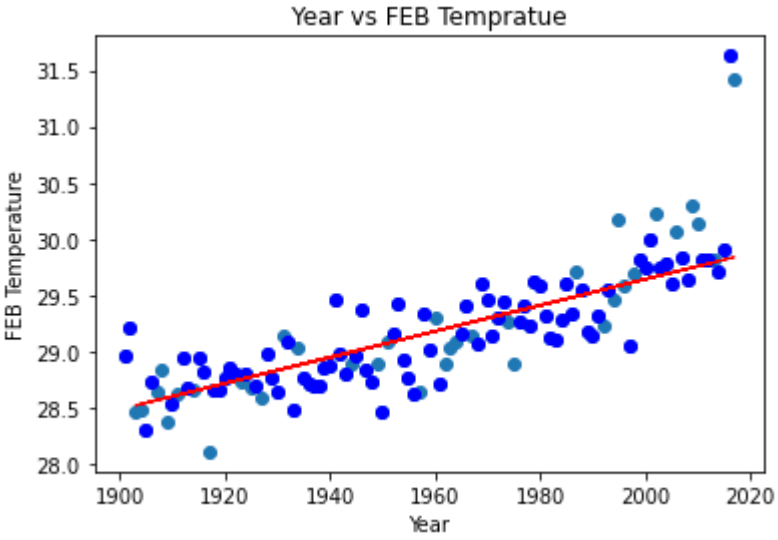
In [ ]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
```

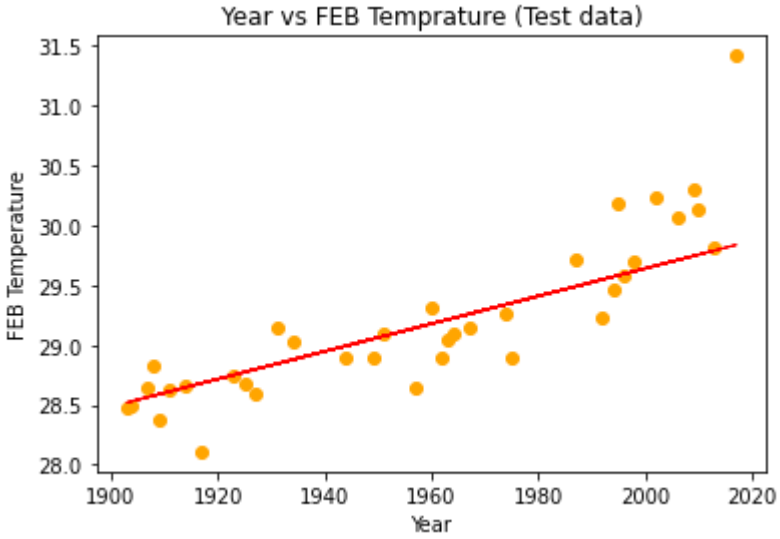
```
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score

for i in range(1,len(column_names)):
  x=raw_data['YEAR']
  y=raw_data[column_names[i]]
  predict_monthwise(x,y,column_names[i])
```



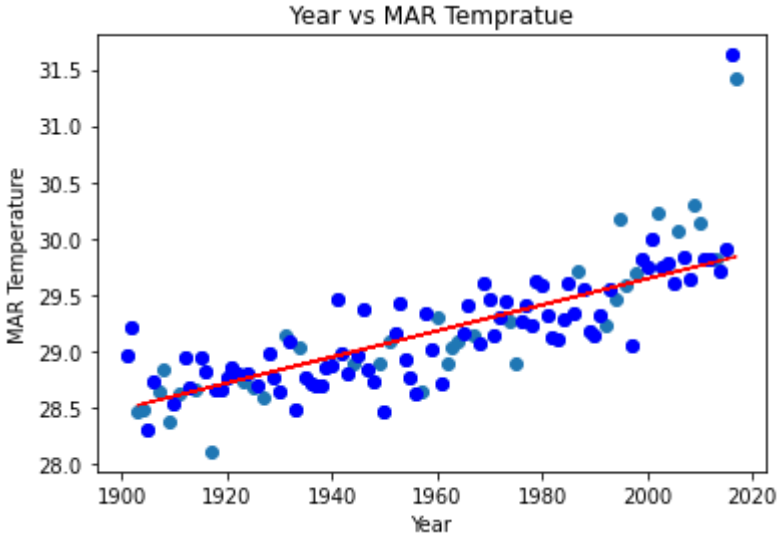


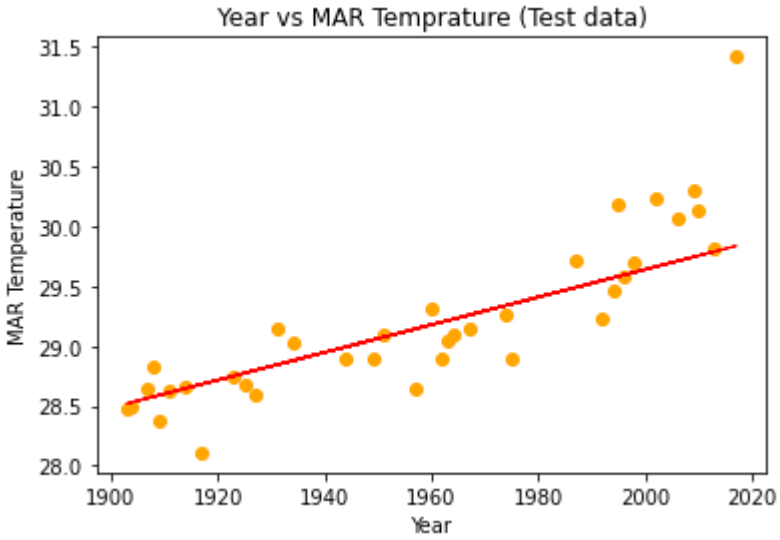

```
MSE:  0.14516479343842795
MAE:  0.2501753025103757
R-Sqaure :  0.6832202846026354
```



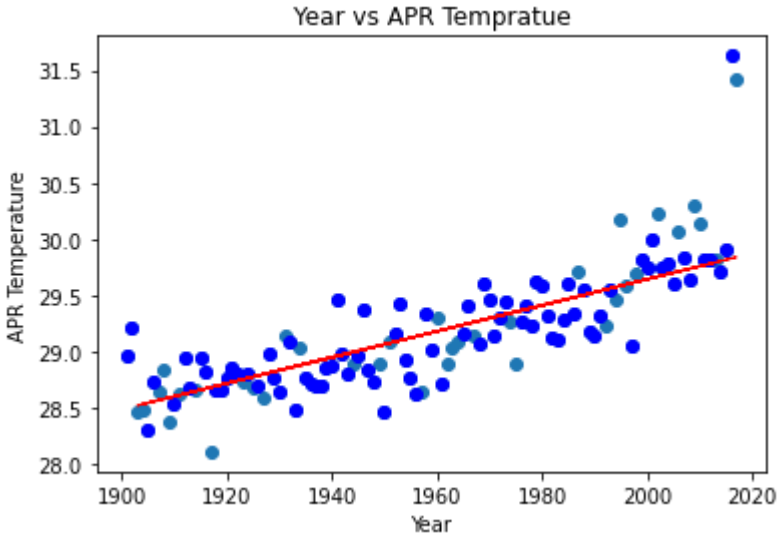


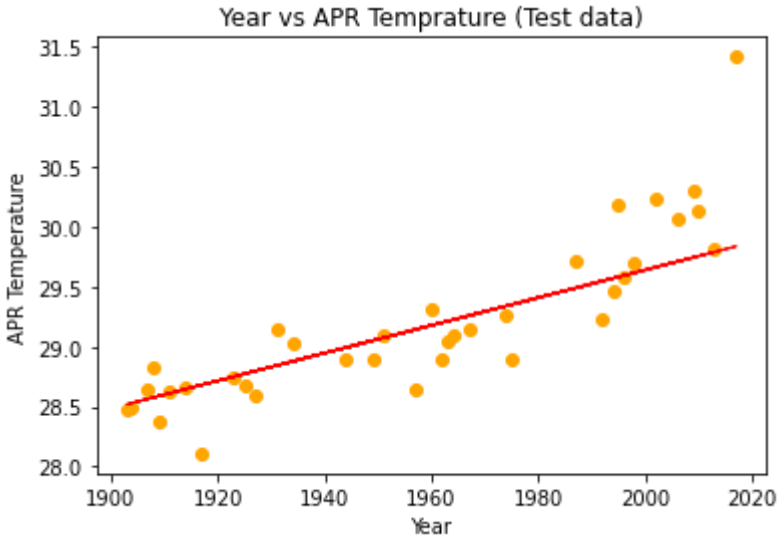
MSE: 0.14516479343842795
MAE: 0.2501753025103757
R-Sqaure : 0.6832202846026354



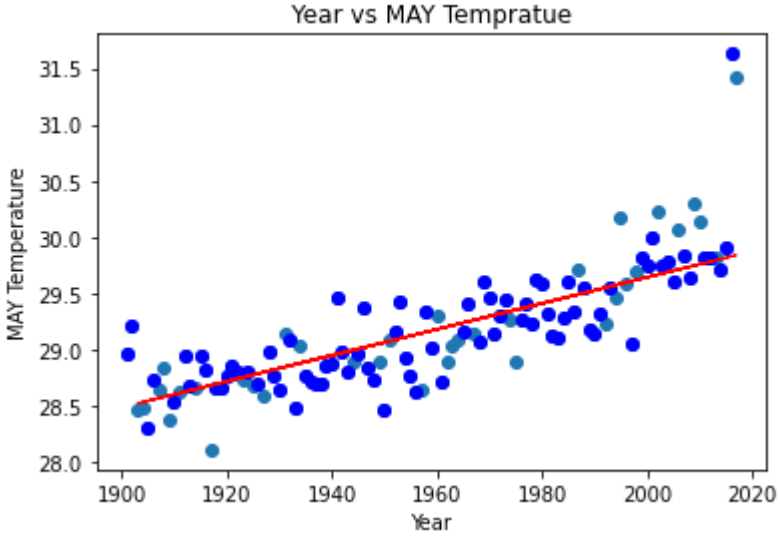


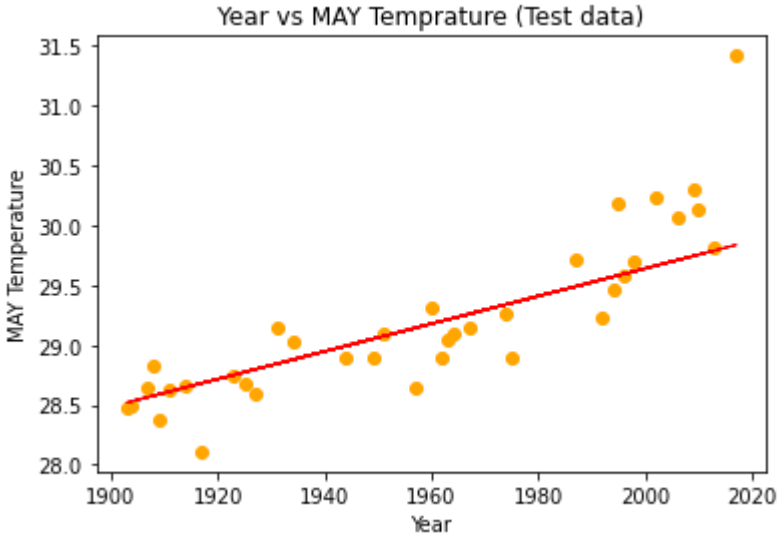
MSE: 0.14516479343842795
MAE: 0.2501753025103757
R-Sqaure : 0.6832202846026354



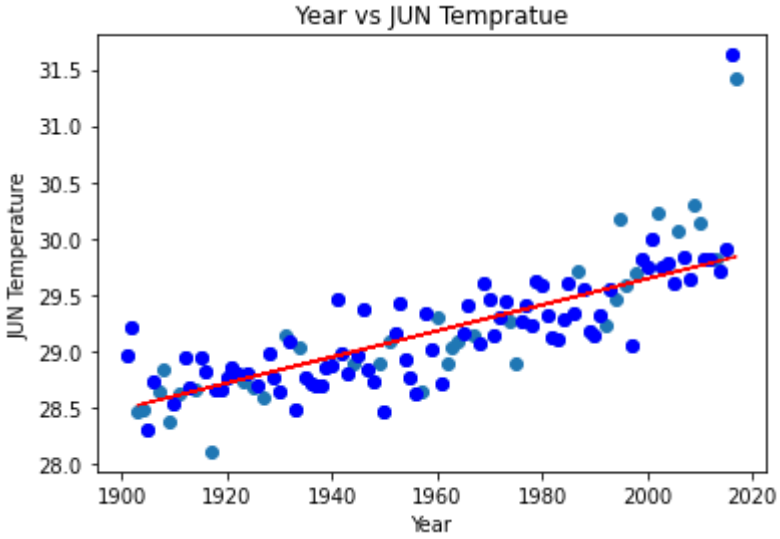


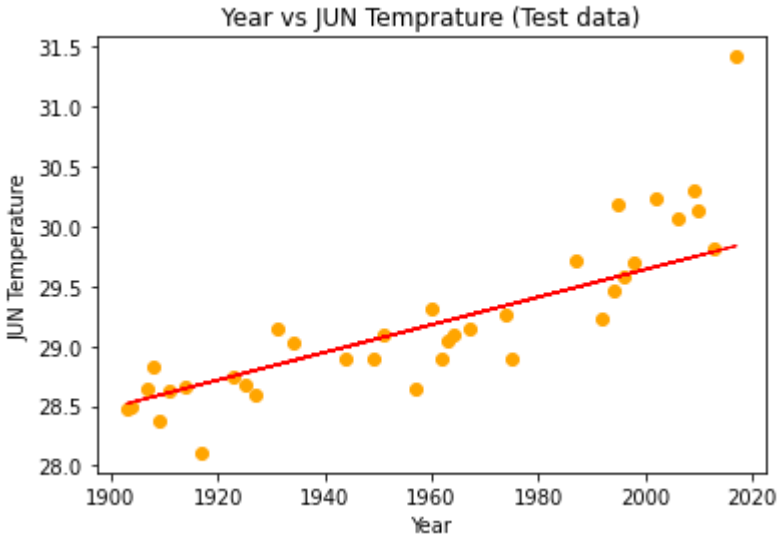

```
MSE:  0.14516479343842795
MAE:  0.2501753025103757
R-Sqaure :  0.6832202846026354
```



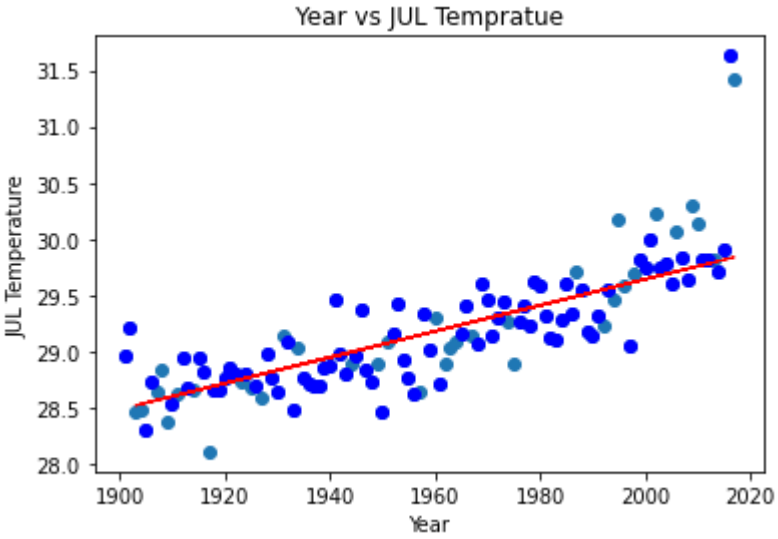


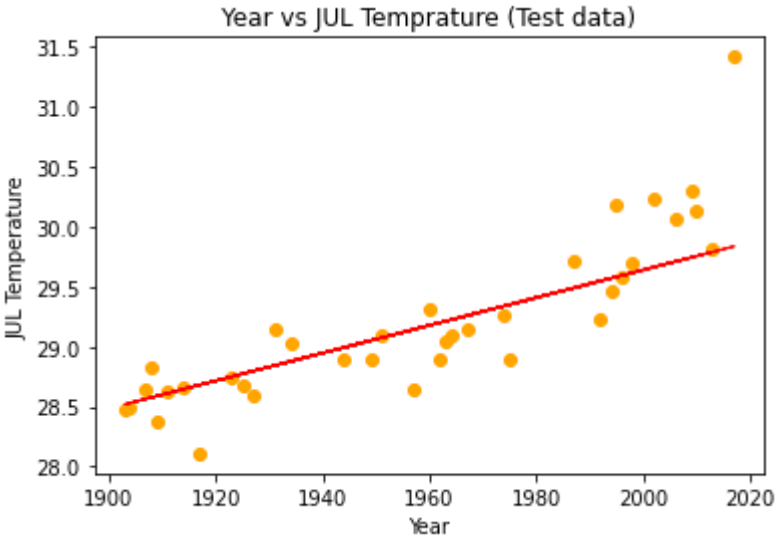
MSE: 0.14516479343842795
MAE: 0.2501753025103757
R-Sqaure : 0.6832202846026354



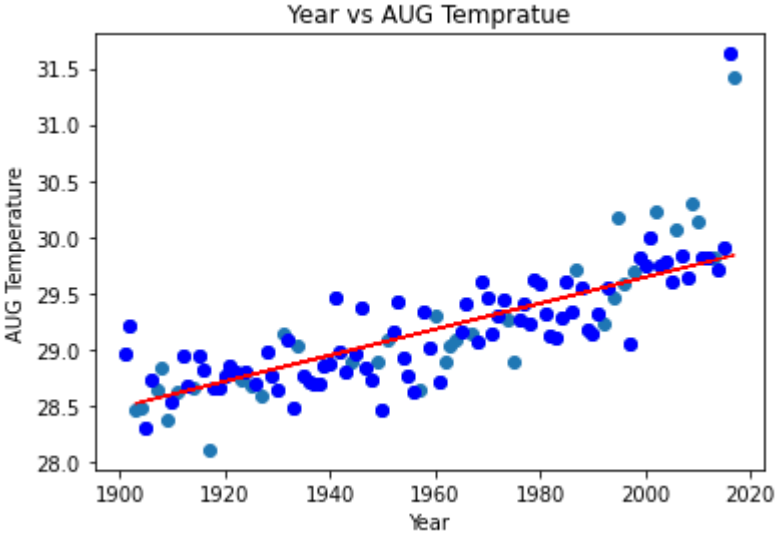


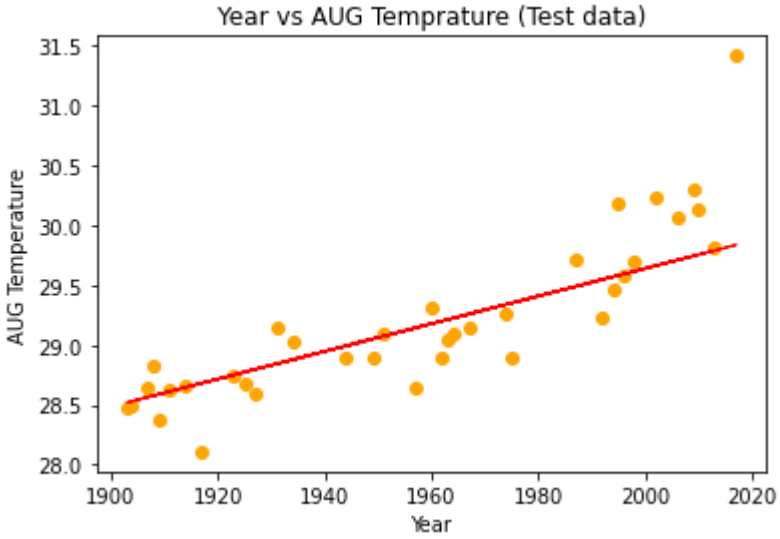
MSE: 0.14516479343842795
MAE: 0.2501753025103757
R-Sqaure : 0.6832202846026354



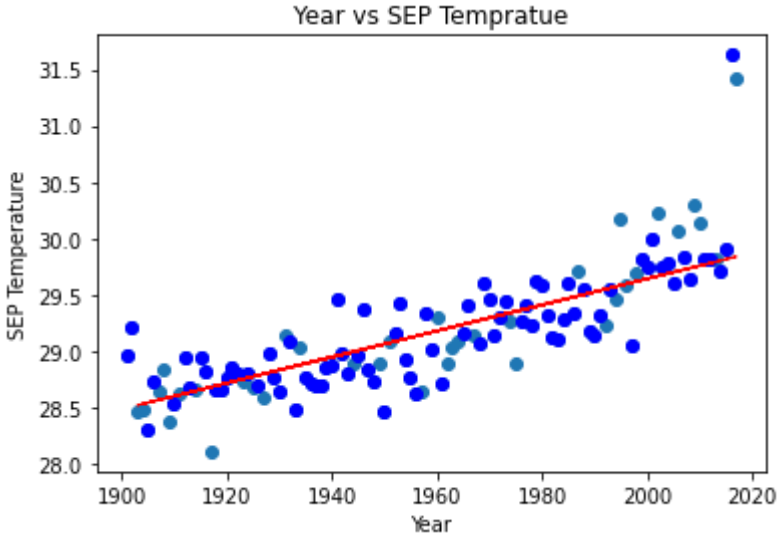


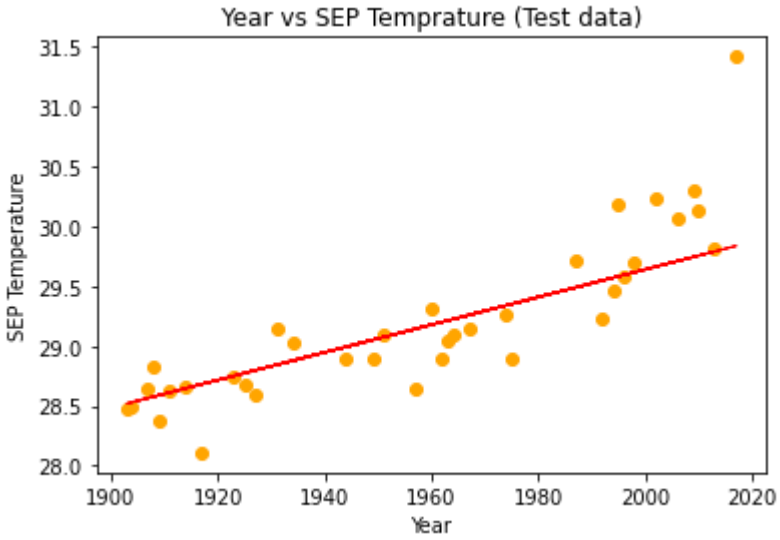

```
MSE:  0.14516479343842795
MAE:  0.2501753025103757
R-Sqaure :  0.6832202846026354
```



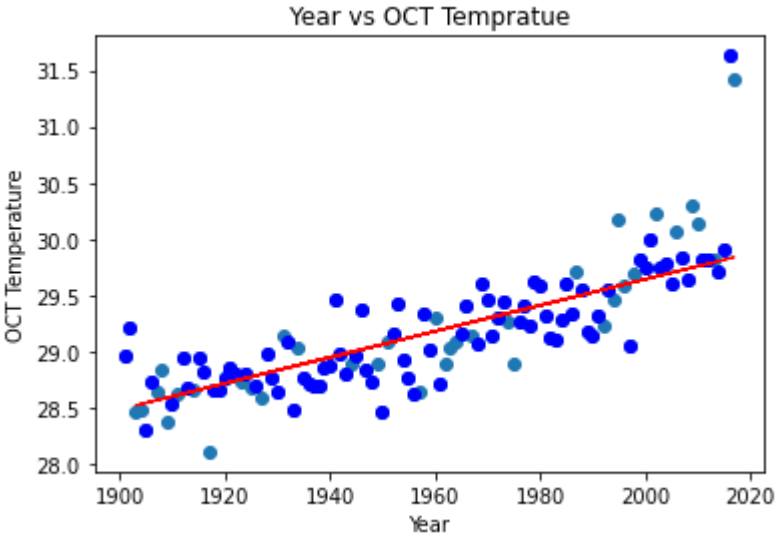


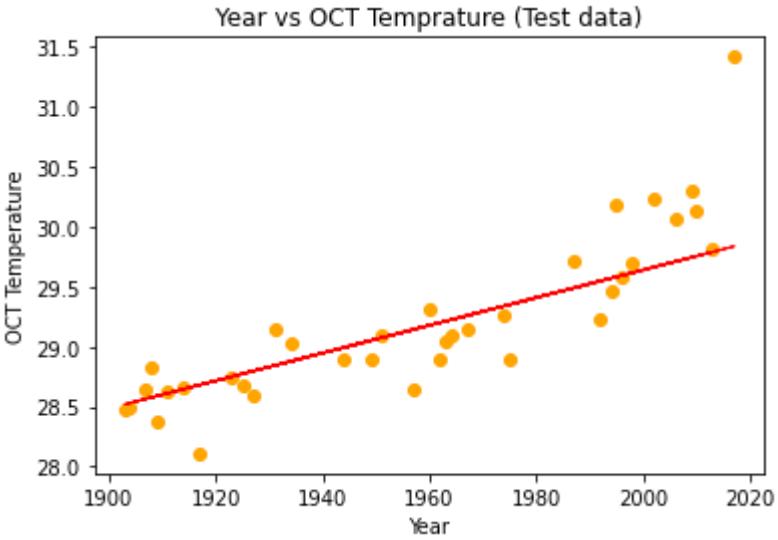
MSE: 0.14516479343842795
MAE: 0.2501753025103757
R-Sqaure : 0.6832202846026354



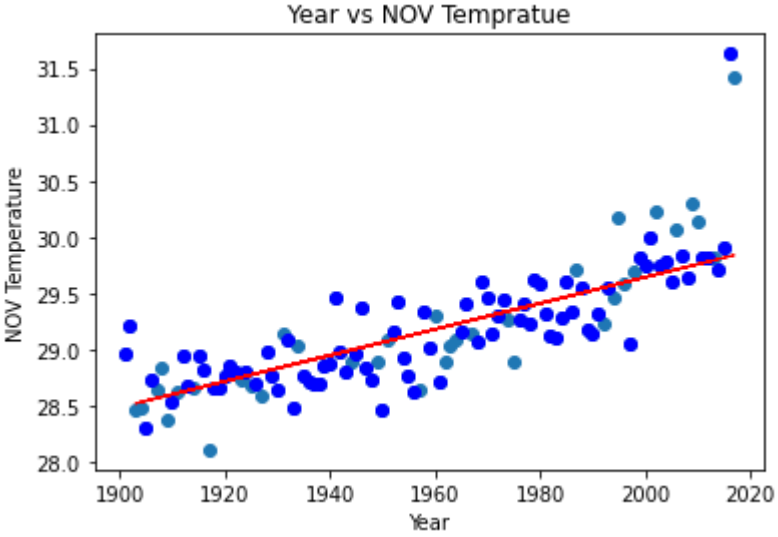


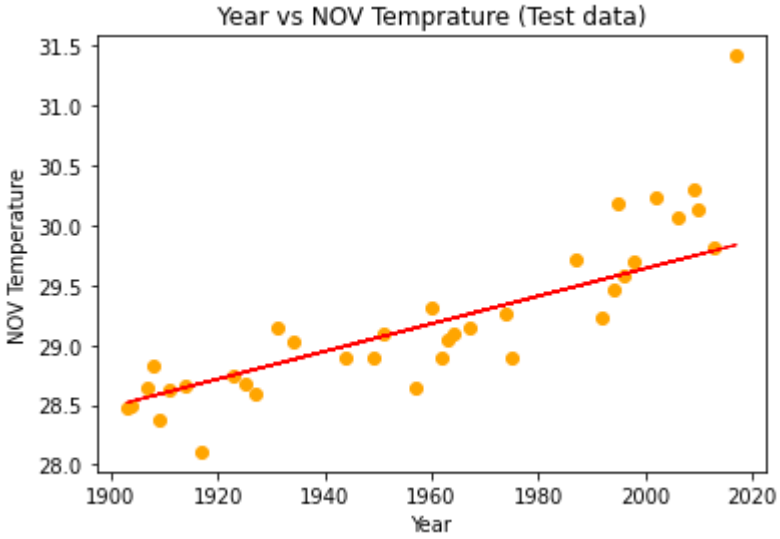
MSE: 0.14516479343842795
MAE: 0.2501753025103757
R-Sqaure : 0.6832202846026354



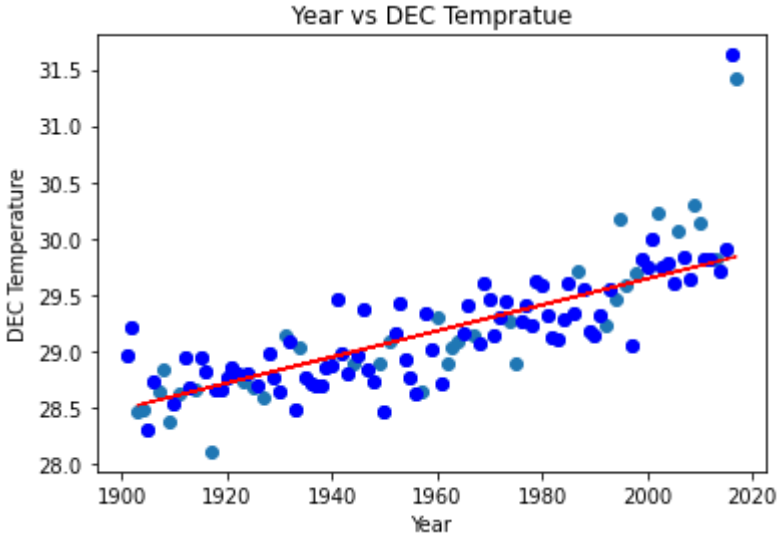


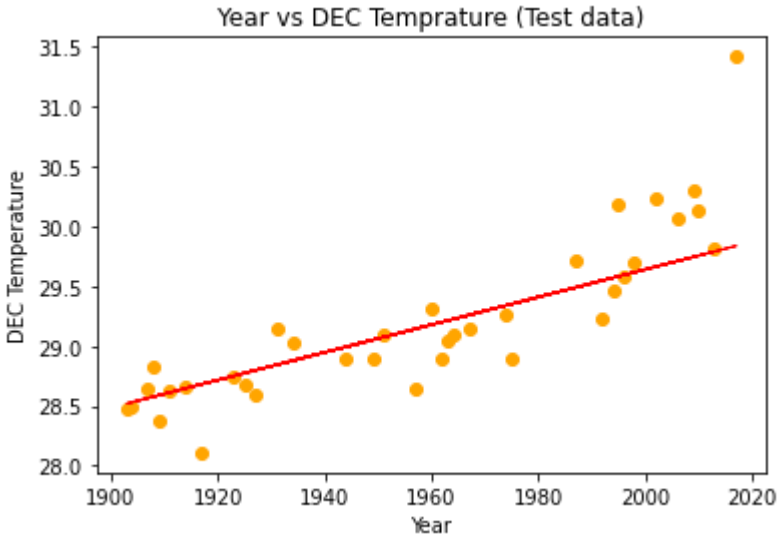

```
MSE:  0.14516479343842795
MAE:  0.2501753025103757
R-Sqaure :  0.6832202846026354
```



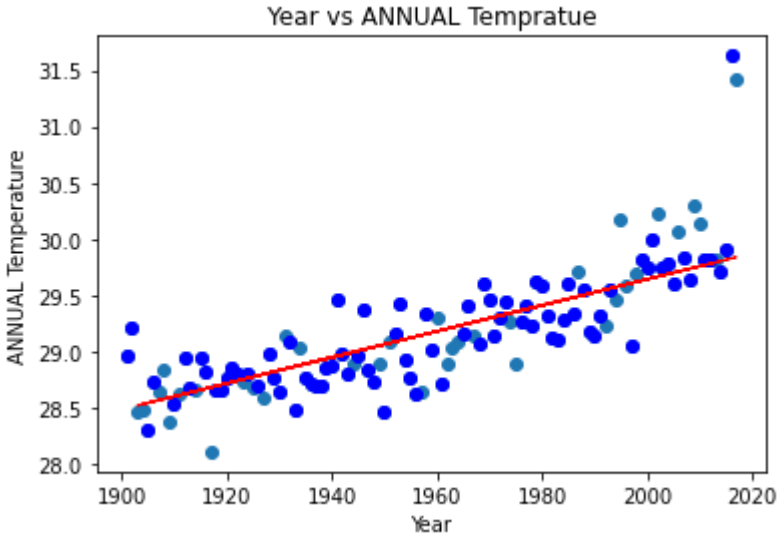


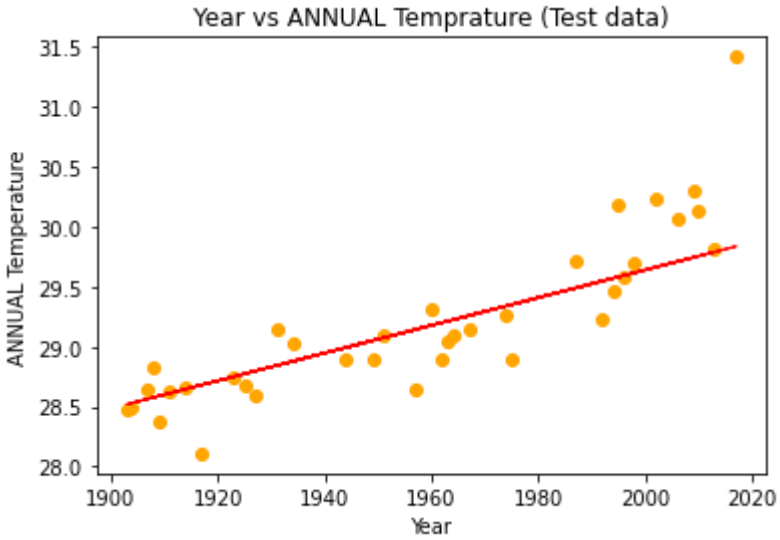

```
MSE:  0.14516479343842795
MAE:  0.2501753025103757
R-Sqaure :  0.6832202846026354
```



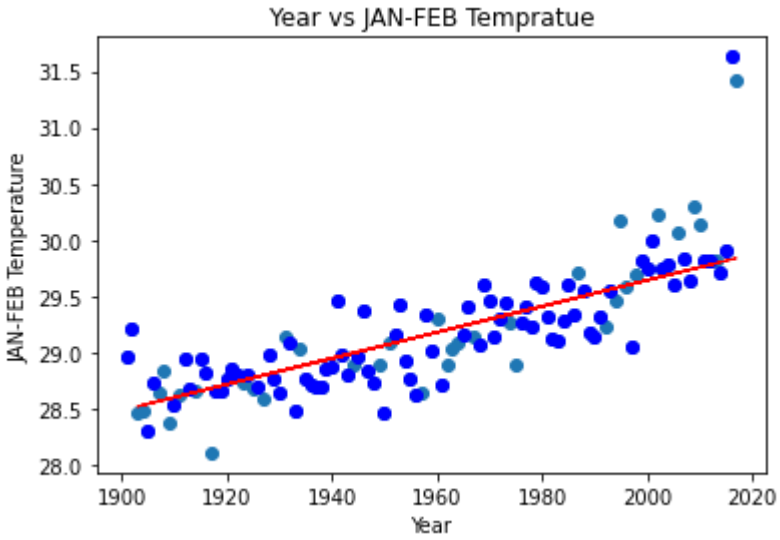


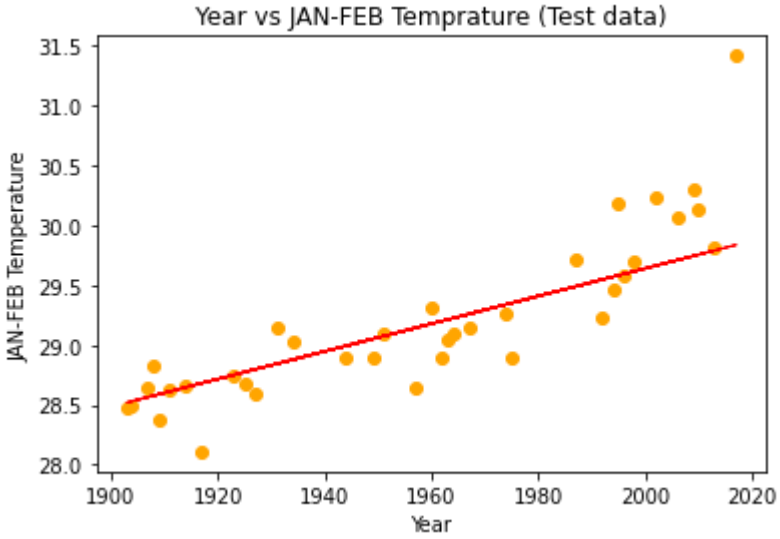

```
MSE:  0.14516479343842795
MAE:  0.2501753025103757
R-Sqaure :  0.6832202846026354
```



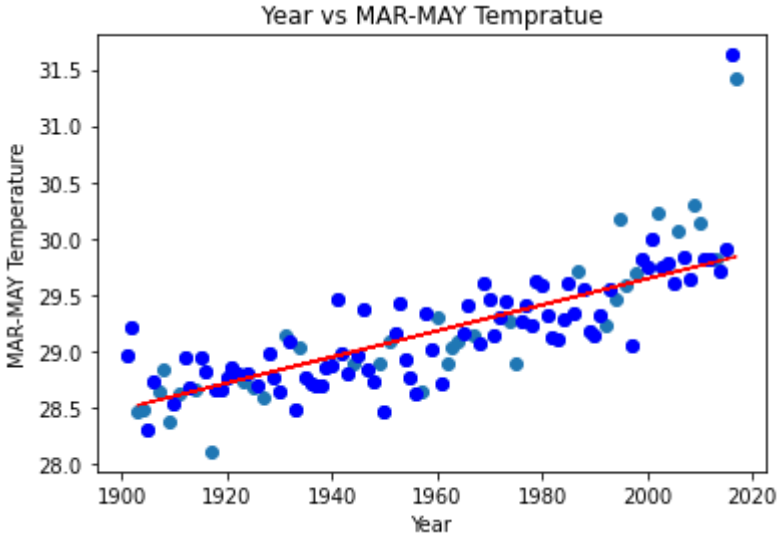


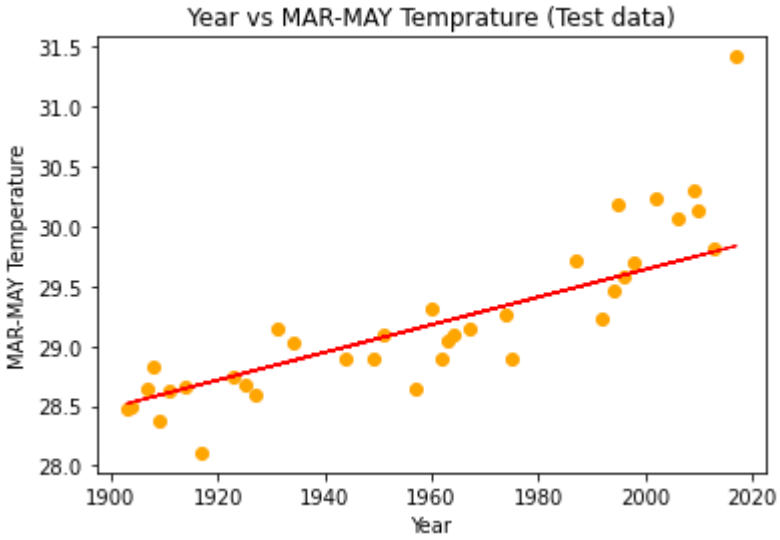

```
MSE:  0.14516479343842795
MAE:  0.2501753025103757
R-Sqaure :  0.6832202846026354
```



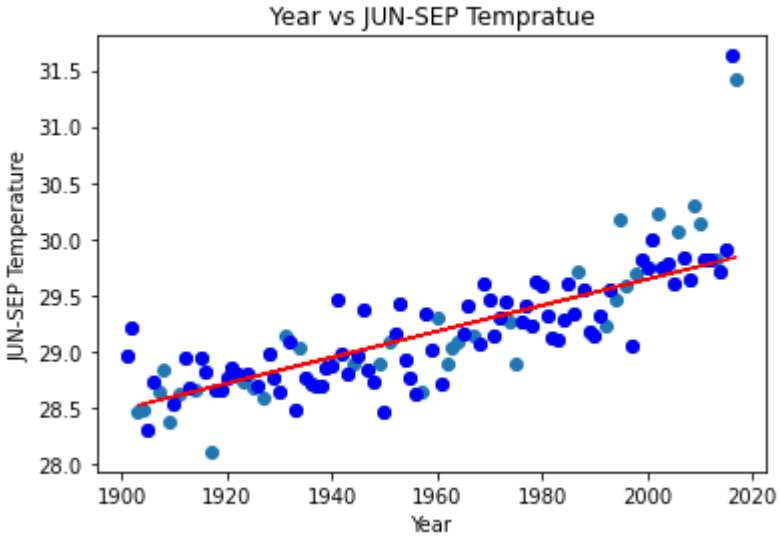


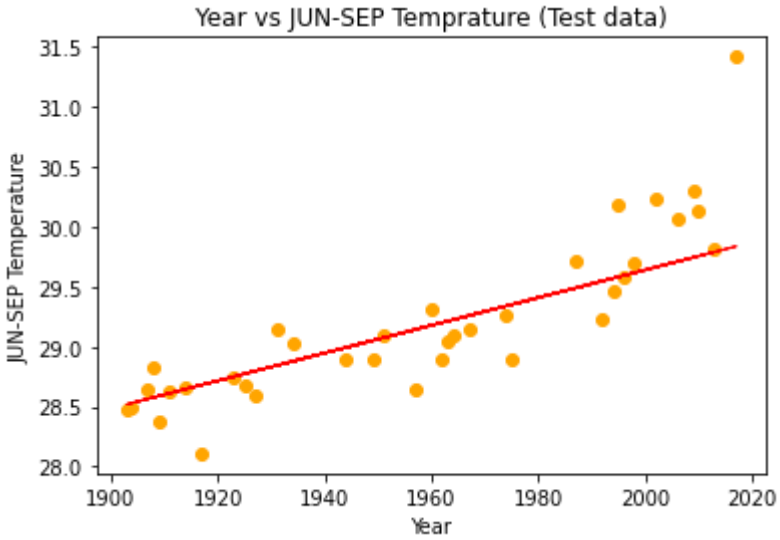
MSE: 0.14516479343842795
MAE: 0.2501753025103757
R-Sqaure : 0.6832202846026354



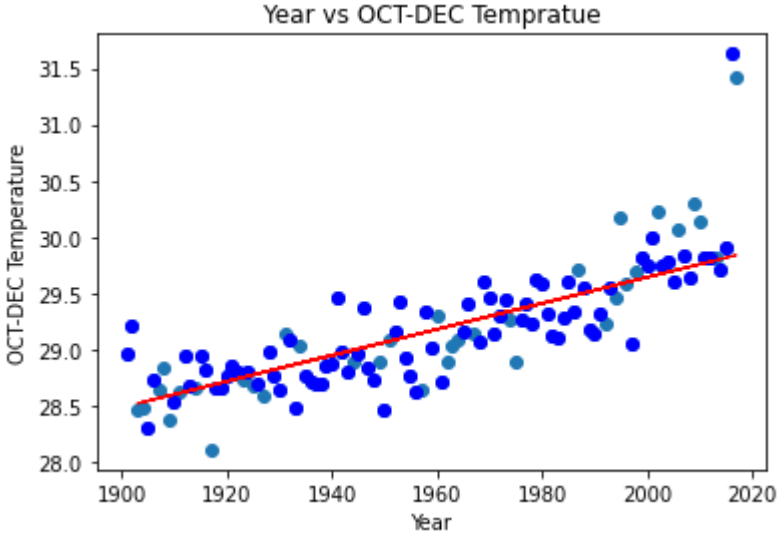


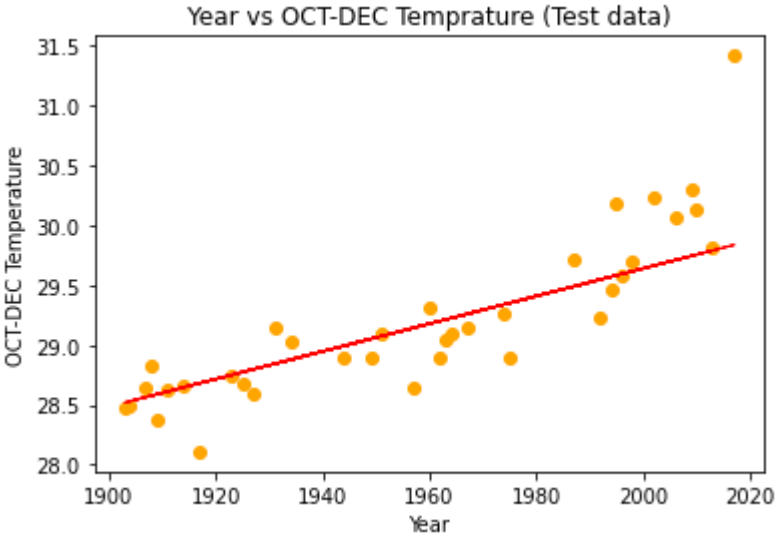
MSE: 0.14516479343842795
MAE: 0.2501753025103757
R-Sqaure : 0.6832202846026354






```
MSE:  0.14516479343842795
MAE:  0.2501753025103757
R-Sqaure :  0.6832202846026354
```






```
MSE:  0.14516479343842795
MAE:  0.2501753025103757
R-Sqaure :  0.6832202846026354
```