In [1]:
```
import pandas as pd
```

In [2]:
```
import numpy as np
```

In [3]:
```
df=pd.read_csv(r'C:\Users\DELL\Desktop\temperatures.csv')
df
```

Out[3]:

	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
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	29.81	25.58	32.58	31.33	27.83
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	29.72	24.90	31.82	32.00	27.81
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	29.90	25.74	31.68	31.87	28.27
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	31.63	28.33	34.57	32.28	30.03
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	31.42	27.95	34.13	32.41	29.69

117 rows × 18 columns

In [4]:
```
from sklearn import linear_model
import matplotlib.pyplot as ply
```

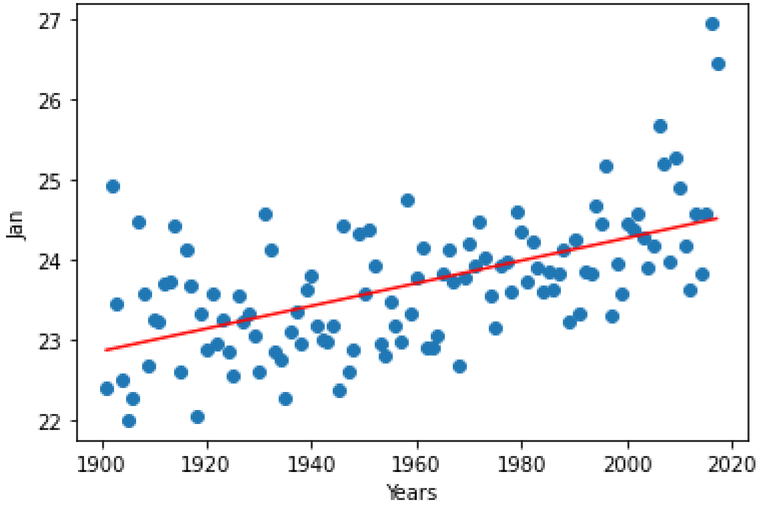
In [5]:
```
import matplotlib.pyplot as plt
reg=linear_model.LinearRegression()
reg.fit(df[['YEAR']],df.JAN)
print("Predicted result =",reg.predict([[2017]]))

%matplotlib inline
```

```
plt.xlabel("Years")
plt.ylabel("Jan")
plt.scatter(df.YEAR,df.JAN)
plt.plot(df.YEAR,reg.predict(df[['YEAR']]),color='Red')
```

```
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegre
ssion was fitted with feature names
  warnings.warn(
Predicted result = [24.50873533]
```

Out[5]:

```
[<matplotlib.lines.Line2D at 0x2319bedf670>]
```



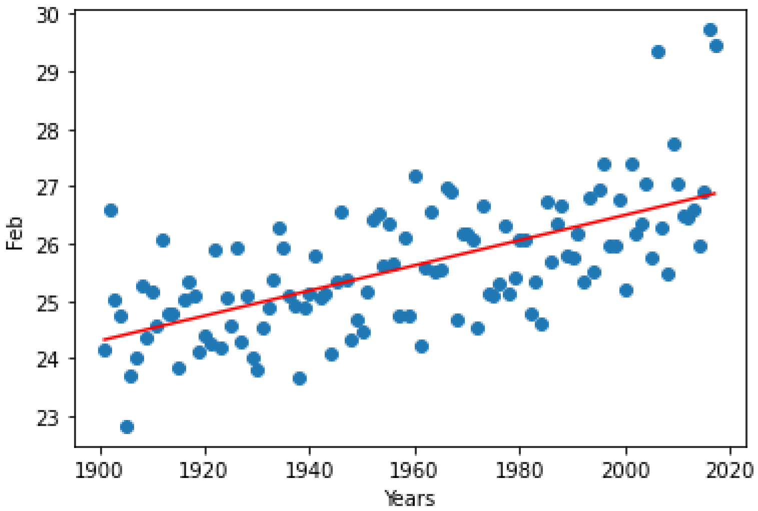
In [6]:

```
import matplotlib.pyplot as plt
reg=linear_model.LinearRegression()
reg.fit(df[['YEAR']],df.FEB)
print("Predicted result =",reg.predict([[2017]]))

%matplotlib inline
plt.xlabel("Years")
plt.ylabel("Feb")
plt.scatter(df.YEAR,df.FEB)
plt.plot(df.YEAR,reg.predict(df[['YEAR']]),color='Red')
```

```
Predicted result = [26.8711227]
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegre
ssion was fitted with feature names
  warnings.warn(
```

Out[6]:

```
[<matplotlib.lines.Line2D at 0x2319c39d610>]
```



In [7]:

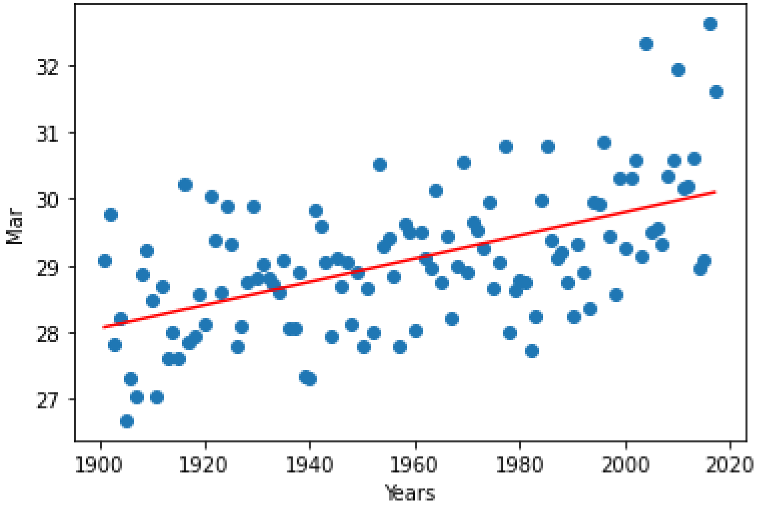
```
import matplotlib.pyplot as plt
reg=linear_model.LinearRegression()
reg.fit(df[['YEAR']],df.MAR)
print("Predicted result =",reg.predict([[2017]]))

%matplotlib inline
plt.xlabel("Years")
plt.ylabel("Mar")
plt.scatter(df.YEAR,df.MAR)
plt.plot(df.YEAR,reg.predict(df[['YEAR']]),color='Red')
```

```
Predicted result = [30.09793423]
```

```
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegre
ssion was fitted with feature names
  warnings.warn(
```

Out[7]:

```
[<matplotlib.lines.Line2D at 0x2319c41c3d0>]
```



In [8]:

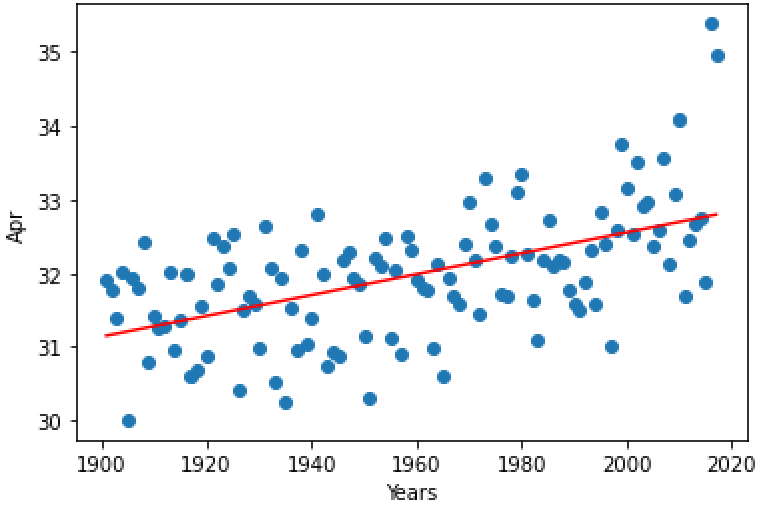
```
import matplotlib.pyplot as plt
reg=linear_model.LinearRegression()
reg.fit(df[['YEAR']],df.APR)
print("Predicted result =",reg.predict([[2017]]))

%matplotlib inline
plt.xlabel("Years")
plt.ylabel("Apr")
plt.scatter(df.YEAR,df.APR)
plt.plot(df.YEAR,reg.predict(df[['YEAR']]),color='Red')
```

```
Predicted result = [32.79814139]
```

```
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegre
ssion was fitted with feature names
  warnings.warn(
```

Out[8]:

```
[<matplotlib.lines.Line2D at 0x2319c489130>]
```



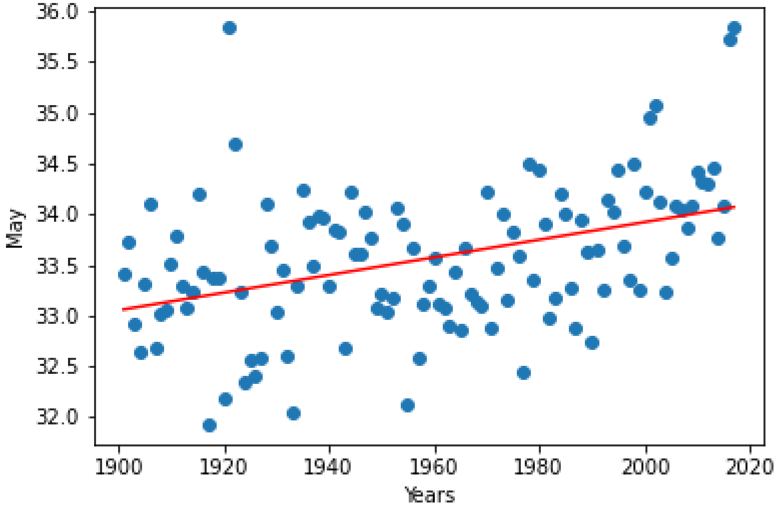
In [9]:

```
import matplotlib.pyplot as plt
reg=linear_model.LinearRegression()
reg.fit(df[['YEAR']],df.MAY)
print("Predicted result =",reg.predict([[2017]]))

%matplotlib inline
plt.xlabel("Years")
plt.ylabel("May")
plt.scatter(df.YEAR,df.MAY)
plt.plot(df.YEAR,reg.predict(df[['YEAR']]),color='Red')
```

```
Predicted result = [34.07060119]
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegre
ssion was fitted with feature names
  warnings.warn(
```

Out[9]:

```
[<matplotlib.lines.Line2D at 0x2319c4e5f10>]
```



In [10]:

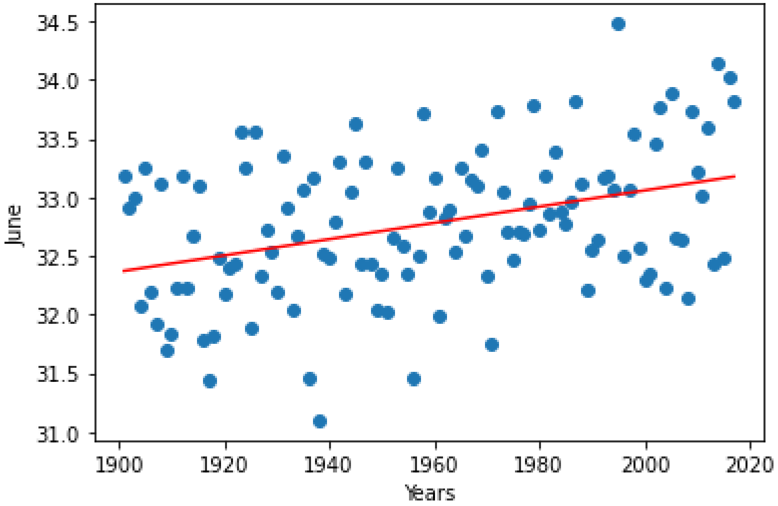
```
import matplotlib.pyplot as plt
reg=linear_model.LinearRegression()
reg.fit(df[['YEAR']],df.JUN)
print("Predicted result =",reg.predict([[2017]]))

%matplotlib inline
plt.xlabel("Years")
plt.ylabel("June")
plt.scatter(df.YEAR,df.JUN)
plt.plot(df.YEAR,reg.predict(df[['YEAR']]),color='Red')
```

```
Predicted result = [33.17683761]
```

```
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegre
ssion was fitted with feature names
  warnings.warn(
```

Out[10]:

```
[<matplotlib.lines.Line2D at 0x2319c55afd0>]
```



In [11]:

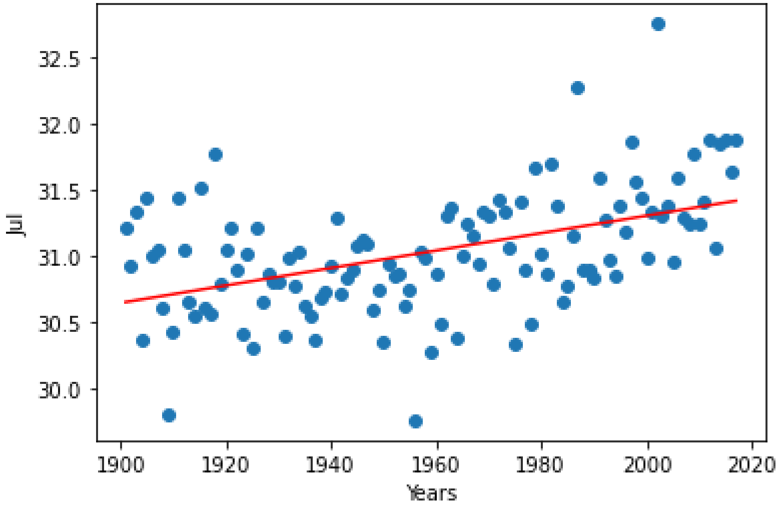
```
import matplotlib.pyplot as plt
reg=linear_model.LinearRegression()
reg.fit(df[['YEAR']],df.JUL)
print("Predicted result =",reg.predict([[2017]]))

%matplotlib inline
plt.xlabel("Years")
plt.ylabel("Jul")
plt.scatter(df.YEAR,df.JUL)
plt.plot(df.YEAR,reg.predict(df[['YEAR']]),color='Red')
```

```
Predicted result = [31.41950022]
```

```
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegre
ssion was fitted with feature names
  warnings.warn(
```

Out[11]:

```
[<matplotlib.lines.Line2D at 0x2319c5d4400>]
```



In [12]:

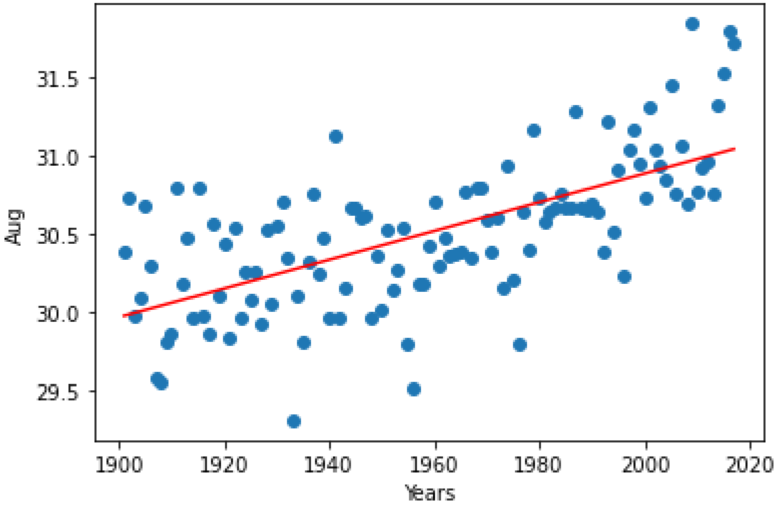
```
import matplotlib.pyplot as plt
reg=linear_model.LinearRegression()
reg.fit(df[['YEAR']],df.AUG)
print("Predicted result =",reg.predict([[2017]]))

%matplotlib inline
plt.xlabel("Years")
plt.ylabel("Aug")
plt.scatter(df.YEAR,df.AUG)
plt.plot(df.YEAR,reg.predict(df[['YEAR']]),color='Red')
```

```
Predicted result = [31.04046936]
```

```
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegre
ssion was fitted with feature names
  warnings.warn(
```

Out[12]:

```
[<matplotlib.lines.Line2D at 0x2319c633d90>]
```



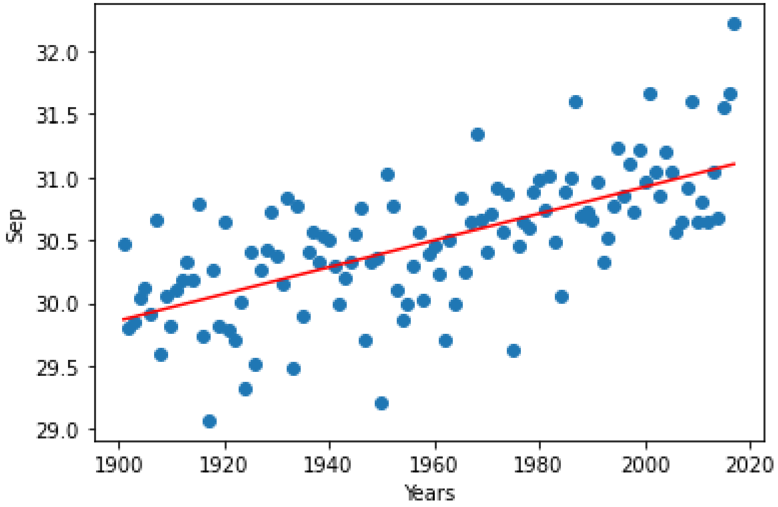
In [13]:

```
import matplotlib.pyplot as plt
reg=linear_model.LinearRegression()
reg.fit(df[['YEAR']],df.SEP)
print("Predicted result =",reg.predict([[2017]]))

%matplotlib inline
plt.xlabel("Years")
plt.ylabel("Sep")
plt.scatter(df.YEAR,df.SEP)
plt.plot(df.YEAR,reg.predict(df[['YEAR']]),color='Red')
```

```
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegre
ssion was fitted with feature names
  warnings.warn(
Predicted result = [31.10475735]
```

Out[13]:

```
[<matplotlib.lines.Line2D at 0x2319c6a2af0>]
```



In [14]:

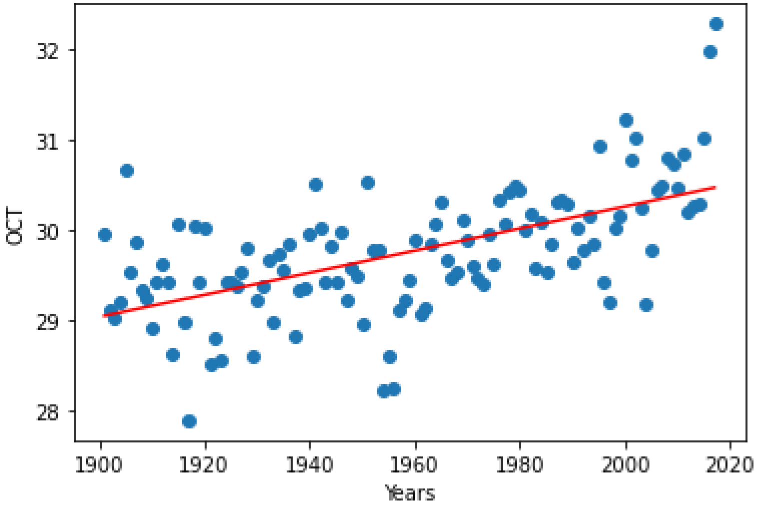
```
import matplotlib.pyplot as plt
reg=linear_model.LinearRegression()
reg.fit(df[['YEAR']],df.OCT)
print("Predicted result =",reg.predict([[2017]]))

%matplotlib inline
plt.xlabel("Years")
plt.ylabel("OCT")
plt.scatter(df.YEAR,df.OCT)
plt.plot(df.YEAR,reg.predict(df[['YEAR']]),color='Red')
```

```
Predicted result = [30.4772157]
```

```
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegre
ssion was fitted with feature names
  warnings.warn(
```

Out[14]:

```
[<matplotlib.lines.Line2D at 0x2319d6e4520>]
```



In [15]:

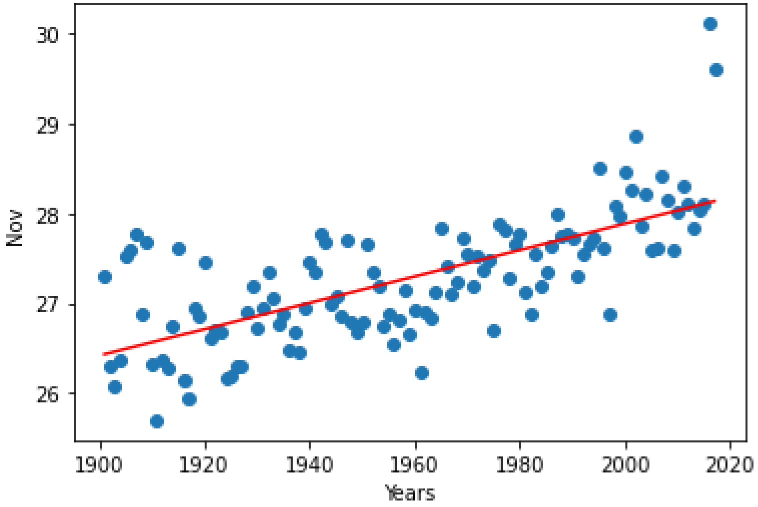
```
import matplotlib.pyplot as plt
reg=linear_model.LinearRegression()
reg.fit(df[['YEAR']],df.NOV)
print("Predicted result =",reg.predict([[2017]]))

%matplotlib inline
plt.xlabel("Years")
plt.ylabel("Nov")
plt.scatter(df.YEAR,df.NOV)
plt.plot(df.YEAR,reg.predict(df[['YEAR']]),color='Red')
```

```
Predicted result = [28.13814284]
```

```
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegre
ssion was fitted with feature names
  warnings.warn(
```

Out[15]:

```
[<matplotlib.lines.Line2D at 0x2319d73c730>]
```



In [16]:

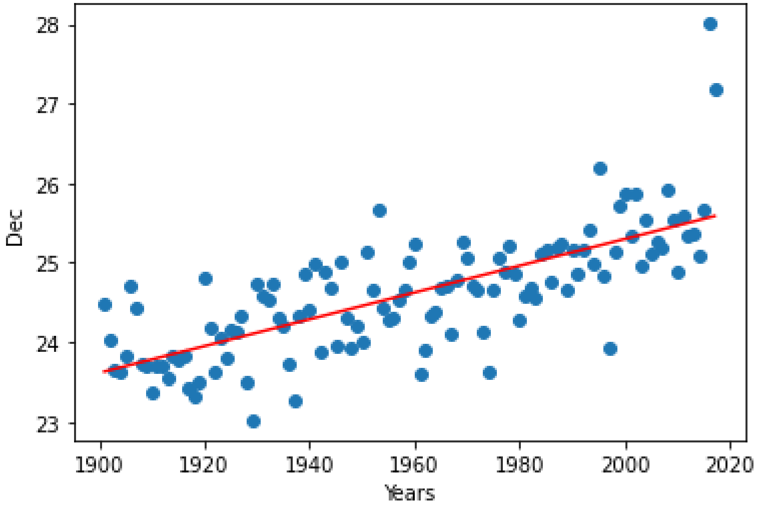
```
import matplotlib.pyplot as plt
reg=linear_model.LinearRegression()
reg.fit(df[['YEAR']],df.DEC)
print("Predicted result =",reg.predict([[2017]]))

%matplotlib inline
plt.xlabel("Years")
plt.ylabel("Dec")
plt.scatter(df.YEAR,df.DEC)
plt.plot(df.YEAR,reg.predict(df[['YEAR']]),color='Red')
```

```
Predicted result = [25.58821237]
```

```
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegre
ssion was fitted with feature names
  warnings.warn(
```

Out[16]:

```
[<matplotlib.lines.Line2D at 0x2319d798910>]
```



In [17]:

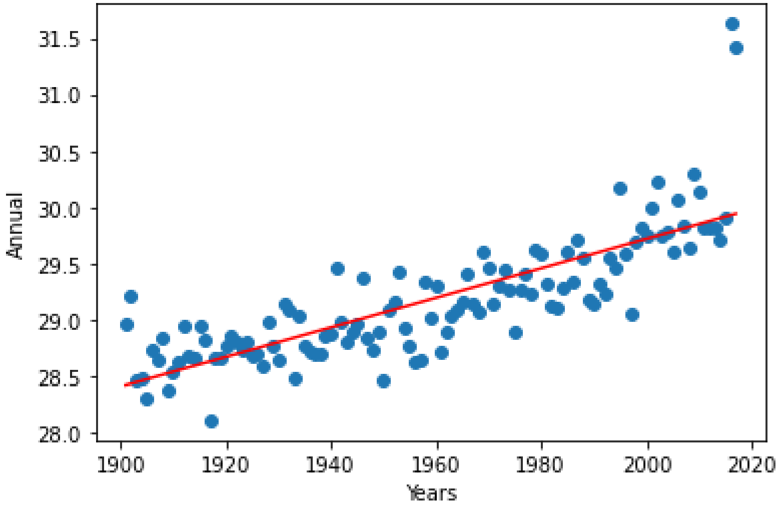
```
import matplotlib.pyplot as plt
reg=linear_model.LinearRegression()
reg.fit(df[['YEAR']],df.ANNUAL)
print("Predicted result =",reg.predict([[2017]]))

%matplotlib inline
plt.xlabel("Years")
plt.ylabel("Annual")
plt.scatter(df.YEAR,df.ANNUAL)
plt.plot(df.YEAR,reg.predict(df[['YEAR']]),color='Red')
```

```
Predicted result = [29.94241924]
```

```
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegre
ssion was fitted with feature names
  warnings.warn(
```

Out[17]:

```
[<matplotlib.lines.Line2D at 0x2319d805a30>]
```



In [18]:

```
import numpy as np
from sklearn.metrics import mean_absolute_error, mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
X=df[["YEAR"]]
Y=df["JAN"]
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.3, random_state=0)
regressor = LinearRegression()
regressor.fit(X_train,Y_train)
Y_pred = regressor.predict(X_test)
df_preds = pd.DataFrame({'Actual': Y_test.squeeze(), 'Predicted': Y_pred.squeeze()})
mae = mean_absolute_error(Y_test, Y_pred)
mse = mean_squared_error(Y_test, Y_pred)
rmse = np.sqrt(mse)
print("Jan Analysis")
print(f'Mean absolute error: {mae:.2f}')
print(f'Mean squared error: {mse:.2f}')
print(f'Root mean squared error: {rmse:.2f}')
```

```
Jan Analysis
Mean absolute error: 0.62
Mean squared error: 0.61
Root mean squared error: 0.78
```

In [19]:

```
X=df[["YEAR"]]
Y=df["FEB"]
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.3, random_state=0)
```

```
regressor = LinearRegression()
regressor.fit(X_train,Y_train)
Y_pred = regressor.predict(X_test)
df_preds = pd.DataFrame({'Actual': Y_test.squeeze(), 'Predicted': Y_pred.squeeze()})
mae = mean_absolute_error(Y_test, Y_pred)
mse = mean_squared_error(Y_test, Y_pred)
rmse = np.sqrt(mse)
print("Feb Analysis")
print(f'Mean absolute error: {mae:.2f}')
print(f'Mean squared error: {mse:.2f}')
print(f'Root mean squared error: {rmse:.2f}')
```

```
Feb Analysis
Mean absolute error: 0.73
Mean squared error: 0.95
Root mean squared error: 0.97
```

In [20]:

```
X=df[["YEAR"]]
Y=df["MAR"]
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.3, random_state=0)
regressor = LinearRegression()
regressor.fit(X_train,Y_train)
Y_pred = regressor.predict(X_test)
df_preds = pd.DataFrame({'Actual': Y_test.squeeze(), 'Predicted': Y_pred.squeeze()})
mae = mean_absolute_error(Y_test, Y_pred)
mse = mean_squared_error(Y_test, Y_pred)
rmse = np.sqrt(mse)
print("Mar Analysis")
print(f'Mean absolute error: {mae:.2f}')
print(f'Mean squared error: {mse:.2f}')
print(f'Root mean squared error: {rmse:.2f}')
```

```
Mar Analysis
Mean absolute error: 0.70
Mean squared error: 0.72
Root mean squared error: 0.85
```

In [21]:

```
X=df[["YEAR"]]
Y=df["APR"]
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.3, random_state=0)
regressor = LinearRegression()
regressor.fit(X_train,Y_train)
Y_pred = regressor.predict(X_test)
df_preds = pd.DataFrame({'Actual': Y_test.squeeze(), 'Predicted': Y_pred.squeeze()})
```

```
mae = mean_absolute_error(Y_test, Y_pred)
mse = mean_squared_error(Y_test, Y_pred)
rmse = np.sqrt(mse)
print("Apr Analysis")
print(f'Mean absolute error: {mae:.2f}')
print(f'Mean squared error: {mse:.2f}')
print(f'Root mean squared error: {rmse:.2f}')
```

```
Apr Analysis
Mean absolute error: 0.58
Mean squared error: 0.60
Root mean squared error: 0.77
```

In [22]:

```
X=df[["YEAR"]]
Y=df["MAY"]
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.3, random_state=0)
regressor = LinearRegression()
regressor.fit(X_train,Y_train)
Y_pred = regressor.predict(X_test)
df_preds = pd.DataFrame({'Actual': Y_test.squeeze(), 'Predicted': Y_pred.squeeze()})
mae = mean_absolute_error(Y_test, Y_pred)
mse = mean_squared_error(Y_test, Y_pred)
rmse = np.sqrt(mse)
print("May Analysis")
print(f'Mean absolute error: {mae:.2f}')
print(f'Mean squared error: {mse:.2f}')
print(f'Root mean squared error: {rmse:.2f}')
```

```
May Analysis
Mean absolute error: 0.55
Mean squared error: 0.47
Root mean squared error: 0.68
```

In [23]:

```
X=df[["YEAR"]]
Y=df["JUN"]
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.3, random_state=0)
regressor = LinearRegression()
regressor.fit(X_train,Y_train)
Y_pred = regressor.predict(X_test)
df_preds = pd.DataFrame({'Actual': Y_test.squeeze(), 'Predicted': Y_pred.squeeze()})
mae = mean_absolute_error(Y_test, Y_pred)
mse = mean_squared_error(Y_test, Y_pred)
rmse = np.sqrt(mse)
print("Jun Analysis")
print(f'Mean absolute error: {mae:.2f}')
print(f'Mean squared error: {mse:.2f}')
print(f'Root mean squared error: {rmse:.2f}')
```

```
Jun Analysis
Mean absolute error: 0.49
Mean squared error: 0.35
Root mean squared error: 0.59
```

In [24]:

```
X=df[["YEAR"]]
Y=df["JUL"]
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.3, random_state=0)
regressor = LinearRegression()
regressor.fit(X_train,Y_train)
Y_pred = regressor.predict(X_test)
df_preds = pd.DataFrame({'Actual': Y_test.squeeze(), 'Predicted': Y_pred.squeeze()})
mae = mean_absolute_error(Y_test, Y_pred)
mse = mean_squared_error(Y_test, Y_pred)
rmse = np.sqrt(mse)
print("July Analysis")
print(f'Mean absolute error: {mae:.2f}')
print(f'Mean squared error: {mse:.2f}')
print(f'Root mean squared error: {rmse:.2f}')
```

```
July Analysis
Mean absolute error: 0.37
Mean squared error: 0.24
Root mean squared error: 0.49
```

In [25]:

```
X=df[["YEAR"]]
Y=df["AUG"]
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.3, random_state=0)
regressor = LinearRegression()
regressor.fit(X_train,Y_train)
Y_pred = regressor.predict(X_test)
df_preds = pd.DataFrame({'Actual': Y_test.squeeze(), 'Predicted': Y_pred.squeeze()})
mae = mean_absolute_error(Y_test, Y_pred)
mse = mean_squared_error(Y_test, Y_pred)
rmse = np.sqrt(mse)
print("August Analysis")
print(f'Mean absolute error: {mae:.2f}')
print(f'Mean squared error: {mse:.2f}')
print(f'Root mean squared error: {rmse:.2f}')
```

```
August Analysis
Mean absolute error: 0.30
Mean squared error: 0.13
Root mean squared error: 0.36
```

In [26]:

```
X=df[["YEAR"]]
Y=df["SEP"]
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.3, random_state=0)
regressor = LinearRegression()
regressor.fit(X_train,Y_train)
Y_pred = regressor.predict(X_test)
df_preds = pd.DataFrame({'Actual': Y_test.squeeze(), 'Predicted': Y_pred.squeeze()})
mae = mean_absolute_error(Y_test, Y_pred)
mse = mean_squared_error(Y_test, Y_pred)
rmse = np.sqrt(mse)
print("Sept Analysis")
print(f'Mean absolute error: {mae:.2f}')
print(f'Mean squared error: {mse:.2f}')
print(f'Root mean squared error: {rmse:.2f}')
```

```
Sept Analysis
Mean absolute error: 0.33
Mean squared error: 0.21
Root mean squared error: 0.46
```

In [27]:

```
X=df[["YEAR"]]
Y=df["OCT"]
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.3, random_state=0)
regressor = LinearRegression()
regressor.fit(X_train,Y_train)
Y_pred = regressor.predict(X_test)
df_preds = pd.DataFrame({'Actual': Y_test.squeeze(), 'Predicted': Y_pred.squeeze()})
mae = mean_absolute_error(Y_test, Y_pred)
mse = mean_squared_error(Y_test, Y_pred)
rmse = np.sqrt(mse)
print("October Analysis")
print(f'Mean absolute error: {mae:.2f}')
print(f'Mean squared error: {mse:.2f}')
print(f'Root mean squared error: {rmse:.2f}')
```

```
October Analysis
Mean absolute error: 0.40
Mean squared error: 0.32
Root mean squared error: 0.57
```

In [28]:

```
X=df[["YEAR"]]
Y=df["NOV"]
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.3, random_state=0)
regressor = LinearRegression()
regressor.fit(X_train,Y_train)
Y_pred = regressor.predict(X_test)
df_preds = pd.DataFrame({'Actual': Y_test.squeeze(), 'Predicted': Y_pred.squeeze()})
mae = mean_absolute_error(Y_test, Y_pred)
mse = mean_squared_error(Y_test, Y_pred)
rmse = np.sqrt(mse)
print("November Analysis")
print(f'Mean absolute error: {mae:.2f}')
print(f'Mean squared error: {mse:.2f}')
print(f'Root mean squared error: {rmse:.2f}')
```

```
November Analysis
Mean absolute error: 0.44
Mean squared error: 0.32
Root mean squared error: 0.57
```

In [29]:

```
X=df[["YEAR"]]
Y=df["DEC"]
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.3, random_state=0)
regressor = LinearRegression()
regressor.fit(X_train,Y_train)
Y_pred = regressor.predict(X_test)
df_preds = pd.DataFrame({'Actual': Y_test.squeeze(), 'Predicted': Y_pred.squeeze()})
mae = mean_absolute_error(Y_test, Y_pred)
mse = mean_squared_error(Y_test, Y_pred)
rmse = np.sqrt(mse)
print("December Analysis")
print(f'Mean absolute error: {mae:.2f}')
print(f'Mean squared error: {mse:.2f}')
print(f'Root mean squared error: {rmse:.2f}')
```

```
December Analysis
Mean absolute error: 0.35
Mean squared error: 0.26
Root mean squared error: 0.51
```

In [30]:

```
X=df[["YEAR"]]
Y=df["ANNUAL"]
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.3, random_state=0)
regressor = LinearRegression()
```

```
regressor.fit(X_train,Y_train)
Y_pred = regressor.predict(X_test)
df_preds = pd.DataFrame({'Actual': Y_test.squeeze(), 'Predicted': Y_pred.squeeze()})
mae = mean_absolute_error(Y_test, Y_pred)
mse = mean_squared_error(Y_test, Y_pred)
rmse = np.sqrt(mse)
print("Annual Analysis")
print(f'Mean absolute error: {mae:.2f}')
print(f'Mean squared error: {mse:.2f}')
print(f'Root mean squared error: {rmse:.2f}')
```

```
Annual Analysis
Mean absolute error: 0.25
Mean squared error: 0.15
Root mean squared error: 0.38
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```