In [1]:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
```

In [2]:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
```

In [7]:

```
df = pd.read_csv("D:/temperatures.csv")
```

In [9]:

```
df.describe()
```

Out[9]:

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

In [10]:

```
df.head
```

```
Out[10]: <bound method NDFrame.head of       YEAR    JAN    FEB    MAR    APR    MAY
         JUN    JUL    AUG    SEP  \
         0    1901  22.40  24.14  29.07  31.91  33.41  33.18  31.21  30.39  30.47
         1    1902  24.93  26.58  29.77  31.78  33.73  32.91  30.92  30.73  29.80
         2    1903  23.44  25.03  27.83  31.39  32.91  33.00  31.34  29.98  29.85
         3    1904  22.50  24.73  28.21  32.02  32.64  32.07  30.36  30.09  30.04
         4    1905  22.00  22.83  26.68  30.01  33.32  33.25  31.44  30.68  30.12
         ..    ...    ...    ...    ...    ...    ...    ...    ...    ...    ...
         112  2013  24.56  26.59  30.62  32.66  34.46  32.44  31.07  30.76  31.04
         113  2014  23.83  25.97  28.95  32.74  33.77  34.15  31.85  31.32  30.68
         114  2015  24.58  26.89  29.07  31.87  34.09  32.48  31.88  31.52  31.55
         115  2016  26.94  29.72  32.62  35.38  35.72  34.03  31.64  31.79  31.66
         116  2017  26.45  29.46  31.60  34.95  35.84  33.82  31.88  31.72  32.22

                OCT    NOV    DEC  ANNUAL  JAN-FEB  MAR-MAY  JUN-SEP  OCT-DEC
         0    29.97  27.31  24.49   28.96    23.27    31.46    31.27    27.25
         1    29.12  26.31  24.04   29.22    25.75    31.76    31.09    26.49
         2    29.04  26.08  23.65   28.47    24.24    30.71    30.92    26.26
         3    29.20  26.36  23.63   28.49    23.62    30.95    30.66    26.40
         4    30.67  27.52  23.82   28.30    22.25    30.00    31.33    26.57
         ..     ...    ...    ...     ...      ...      ...      ...      ...
         112  30.27  27.83  25.37   29.81    25.58    32.58    31.33    27.83
         113  30.29  28.05  25.08   29.72    24.90    31.82    32.00    27.81
         114  31.04  28.10  25.67   29.90    25.74    31.68    31.87    28.27
         115  31.98  30.11  28.01   31.63    28.33    34.57    32.28    30.03
         116  32.29  29.60  27.18   31.42    27.95    34.13    32.41    29.69

         [117 rows x 18 columns]>
```

In [11]:

```
type(df)
```

```
Out[11]: pandas.core.frame.DataFrame
```

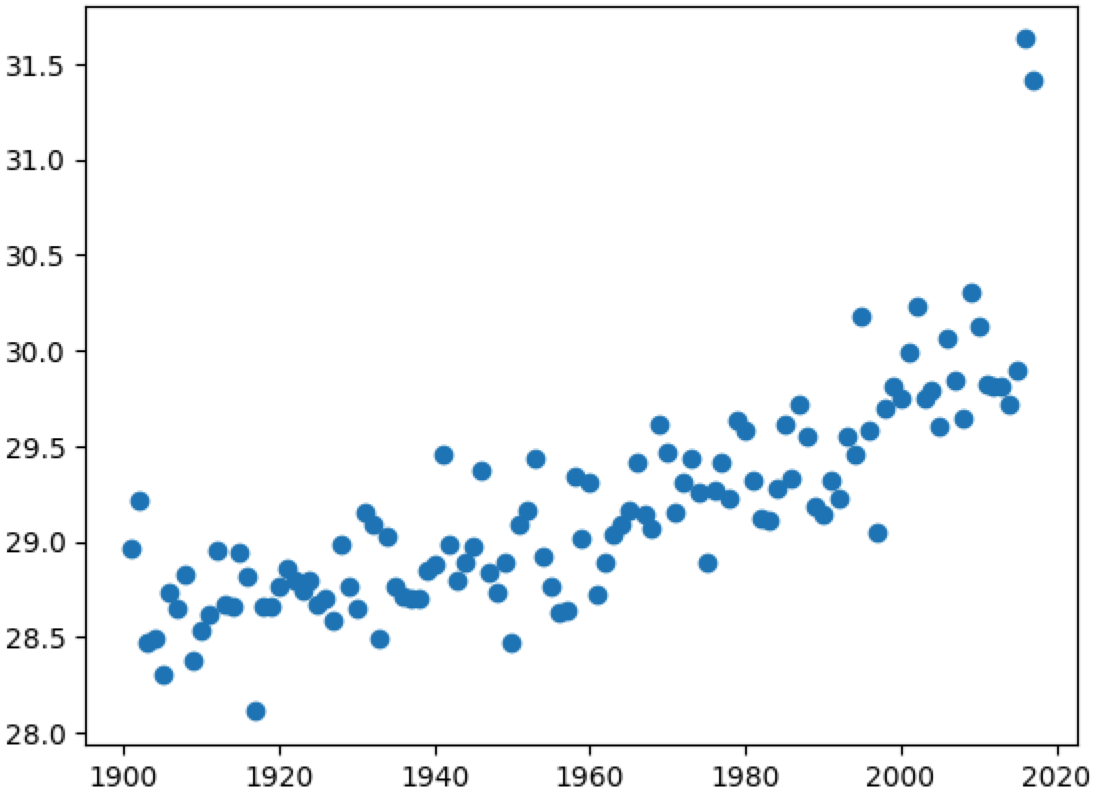
In [13]:

```
x = df["YEAR"]
y = df["ANNUAL"]
```

In [14]:

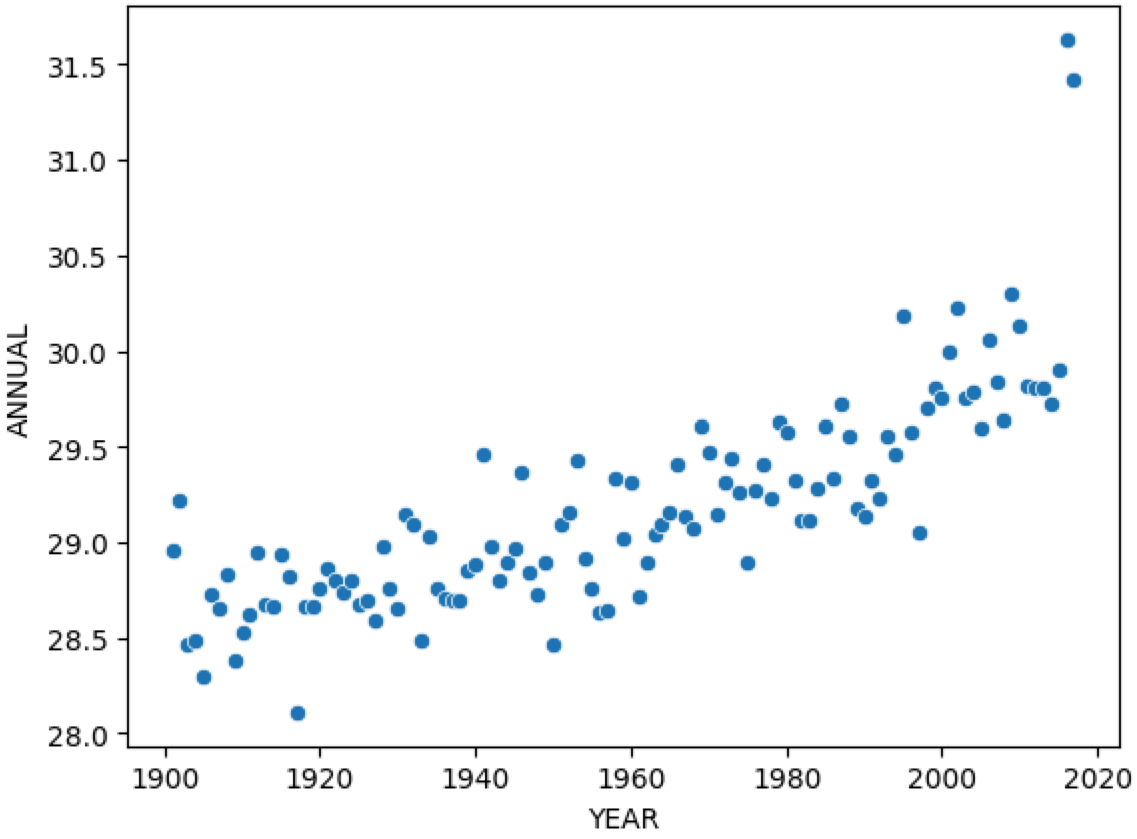
```
plt.plot(x,y,'o')
```

```
Out[14]: [<matplotlib.lines.Line2D at 0x1e2d422e9b0>]
```



```
In [15]: sns.scatterplot(x=x,y=y,data=df)
```

```
Out[15]: <Axes: xlabel='YEAR', ylabel='ANNUAL'>
```



In [16]:

```
type(x)
```

Out[16]:

```
pandas.core.series.Series
```

In [17]:

```
x = x.values
```

In [18]:

```
x = x.reshape(117,1)
```

In [19]:

```
x.shape
```

Out[19]:

```
(117, 1)
```

In [20]:

```
type(x)
```

Out[20]:

```
numpy.ndarray
```

In [21]:

```
x_train, x_test, y_train, y_test = train_test_split(x, y,test_size=0.25)

print(f"x Training dataset: {x_train.shape}")
print(f"y Training dataset: {y_train.shape}")
print(f"x test dataset: {x_test.shape}")
print(f"y test dataset: {y_test.shape}")
```

```
x Training dataset: (87, 1)
y Training dataset: (87,)
x test dataset: (30, 1)
y test dataset: (30,)
```

In [22]:

```
model = LinearRegression()
```

In [23]:

```
model.fit(x_train,y_train)
```

Out[23]:

▾ LinearRegression ⓘ ?

Parameters

⧉	fit_intercept	True
⧉	copy_X	True
⧉	tol	1e-06
⧉	n_jobs	None
⧉	positive	False

In [24]:

```
model.coef_ #w
```

Out[24]:

```
array([0.01384177])
```

In [25]:

```
model.intercept_ #b
```

Out[25]:

```
np.float64(2.0730307540883572)
```

In [26]:

```
y_pred = model.predict(x_test)
```

In [27]:

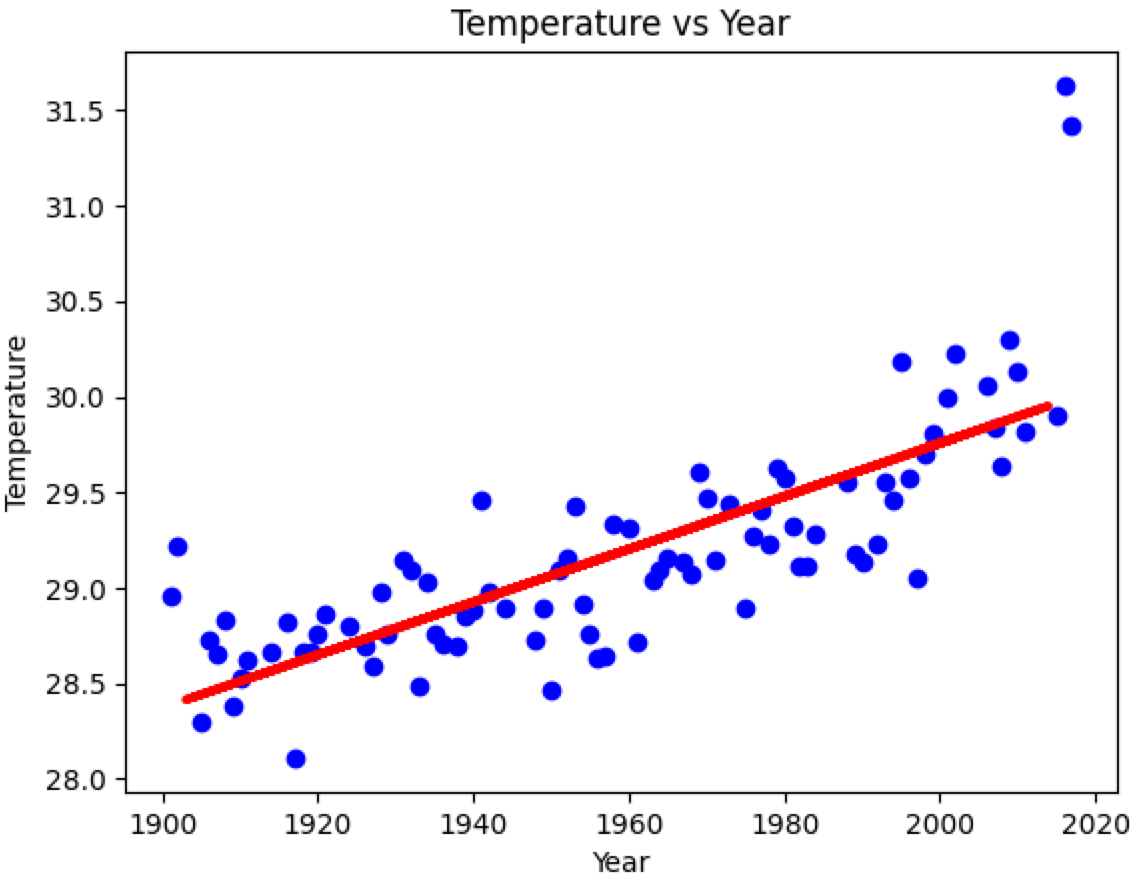
```
y_pred.shape
```

Out[27]:

```
(30,)
```

In [28]:

```
plt.scatter(x_train, y_train, color='blue')
plt.plot(x_test, y_pred, color='red', linewidth=3)
plt.title("Temperature vs Year")
plt.xlabel("Year")
plt.ylabel("Temperature")
plt.show()
```



In [ ]:

```
```