

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

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

```
In [3]: df = pd.read_csv('../pr2/temperatures.csv')
```

```
In [4]: df.head(n=10)
```

```
Out[4]:
```

| | 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 |
| 5 | 1906 | 22.28 | 23.69 | 27.31 | 31.93 | 34.11 | 32.19 | 31.01 | 30.30 | 29.92 | 29.55 | 27.60 | 24.72 | 28.73 |
| 6 | 1907 | 24.46 | 24.01 | 27.04 | 31.79 | 32.68 | 31.92 | 31.05 | 29.58 | 30.67 | 29.87 | 27.78 | 24.44 | 28.65 |
| 7 | 1908 | 23.57 | 25.26 | 28.86 | 32.42 | 33.02 | 33.12 | 30.61 | 29.55 | 29.59 | 29.35 | 26.88 | 23.73 | 28.83 |
| 8 | 1909 | 22.67 | 24.36 | 29.22 | 30.79 | 33.06 | 31.70 | 29.81 | 29.81 | 30.06 | 29.25 | 27.69 | 23.69 | 28.38 |
| 9 | 1910 | 23.24 | 25.16 | 28.48 | 31.42 | 33.51 | 31.84 | 30.42 | 29.86 | 29.82 | 28.91 | 26.32 | 23.37 | 28.53 |

finding datatypes

df.dtypes

```
In [5]: df.dtypes
```

```
Out[5]: YEAR          int64
      JAN          float64
      FEB          float64
      MAR          float64
      APR          float64
      MAY          float64
      JUN          float64
      JUL          float64
      AUG          float64
      SEP          float64
      OCT          float64
      NOV          float64
      DEC          float64
      ANNUAL       float64
      JAN-FEB      float64
      MAR-MAY      float64
      JUN-SEP      float64
      OCT-DEC      float64
dtype: object
```

columns

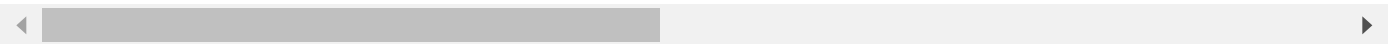
df.columns

```
In [6]: df.columns
```

```
Out[6]: 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')
```

```
In [7]: df.describe()
```

| | YEAR | JAN | FEB | MAR | APR | MAY | JUN | JUL |
|-------|-------------|------------|------------|------------|------------|------------|------------|------------|
| count | 117.000000 | 117.000000 | 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 | 32.774274 | 31.035812 |
| std | 33.919021 | 0.834588 | 1.150757 | 1.068451 | 0.889478 | 0.724905 | 0.633132 | 0.468812 |
| min | 1901.000000 | 22.000000 | 22.830000 | 26.680000 | 30.010000 | 31.930000 | 31.100000 | 29.760000 |
| 25% | 1930.000000 | 23.100000 | 24.780000 | 28.370000 | 31.460000 | 33.110000 | 32.340000 | 30.740000 |
| 50% | 1959.000000 | 23.680000 | 25.480000 | 29.040000 | 31.950000 | 33.510000 | 32.730000 | 31.000000 |
| 75% | 1988.000000 | 24.180000 | 26.310000 | 29.610000 | 32.420000 | 34.030000 | 33.180000 | 31.330000 |
| max | 2017.000000 | 26.940000 | 29.720000 | 32.620000 | 35.380000 | 35.840000 | 34.480000 | 32.760000 |

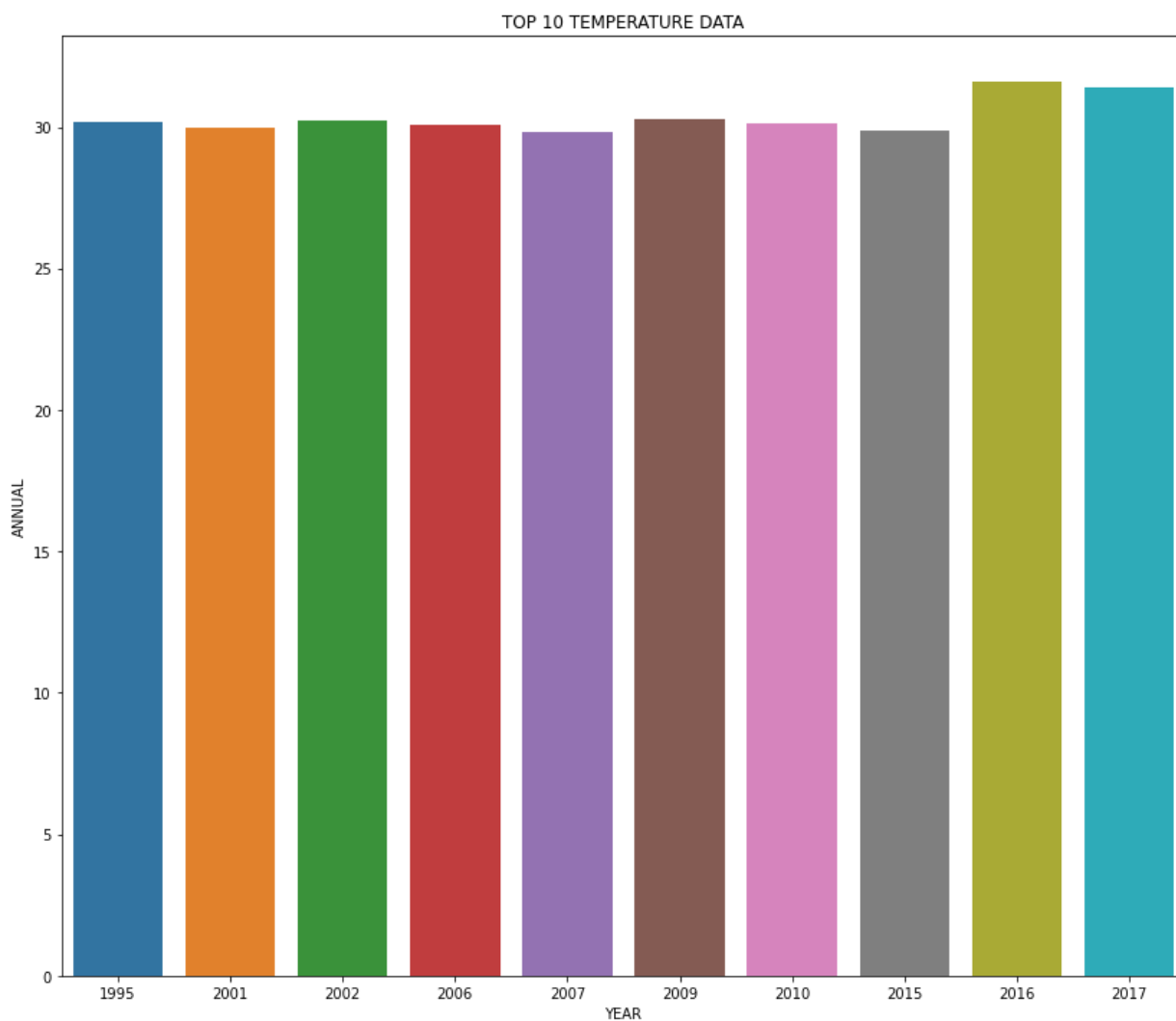


```
In [8]: #FINDING NULL VALUE
df.isnull().sum()
```

```
Out[8]: 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 [9]: top_10_data = df.nlargest(10,"ANNUAL")  
plt.figure(figsize=(14,12))  
plt.title("TOP 10 TEMPERATURE DATA")  
sns.barplot(x=top_10_data.YEAR,y=top_10_data.ANNUAL)
```

```
Out[9]: <AxesSubplot:title={'center':'TOP 10 TEMPERATURE DATA'}, xlabel='YEAR', ylabel='ANNUA  
L'>
```



```
In [12]: from sklearn import linear_model, metrics
```

```
In [13]: X = df[["YEAR"]]  
Y = df[["JAN"]]
```

```
In [15]: 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)
```

```
In [16]: len(X_train)
```

```
Out[16]: 93
```

```
In [17]: len(X_test)
```

```
Out[17]: 24
```

```
In [18]: df.shape
```

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

```
In [19]: reg = linear_model.LinearRegression()
```

```
In [20]: print(X_train)
```

```
      YEAR  
56    1957  
94    1995  
35    1936  
38    1939  
93    1994  
..     ...  
9     1910  
72    1973  
12    1913  
107   2008  
37    1938
```

```
[93 rows x 1 columns]
```

```
In [21]: print(Y_train)
```

```
      JAN  
56    22.98  
94    24.44  
35    23.10  
38    23.61  
93    24.67  
..     ...  
9     23.24  
72    24.02  
12    23.71  
107   23.97  
37    22.95
```

```
[93 rows x 1 columns]
```

```
In [22]: model = reg.fit(X_train, Y_train)
```

```
In [23]: r_sq = reg.score(X_train,Y_train)
```

```
In [24]: print("intercept:",model.intercept_)
```

```
intercept: [-5.35338281]
```

```
In [25]: print("coefficient:",r_sq)
```

```
coefficient: 0.3548045849122119
```

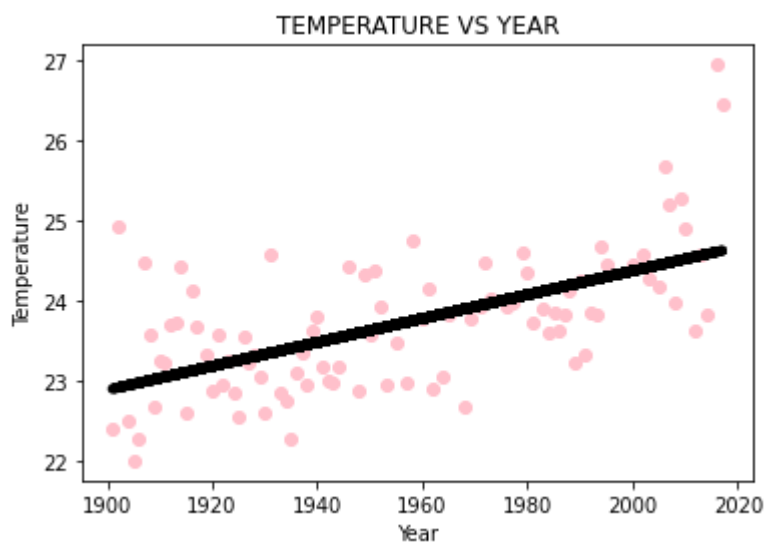
```
In [26]: print("slope:",model.coef_)
```

```
slope: [[0.01486008]]
```

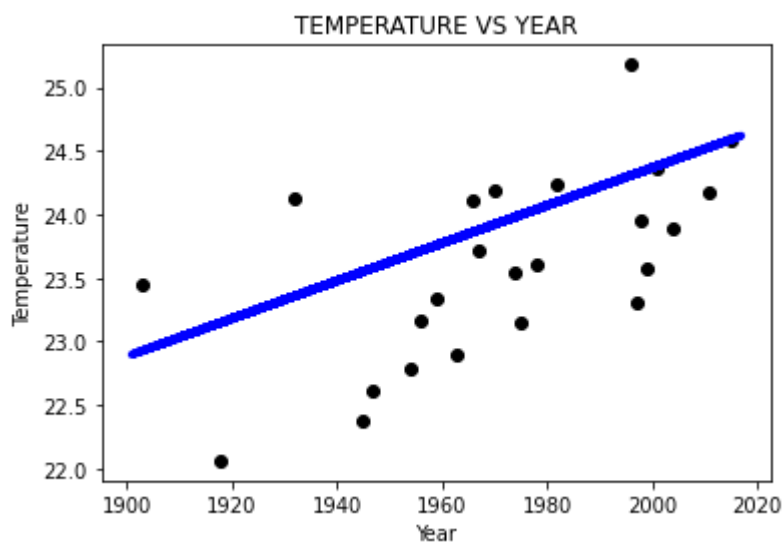
```
In [28]: Y_pred = model.predict(X_test)
print(Y_pred)
```

```
[[23.92097555]
 [23.5791937 ]
 [23.75751466]
 [24.58967916]
 [23.98041587]
 [24.35191788]
 [23.35629249]
 [23.68321426]
 [23.86153523]
 [24.32219772]
 [24.30733764]
 [24.3370578 ]
 [22.92535016]
 [23.81695498]
 [24.53023884]
 [23.71293442]
 [24.42621828]
 [24.38163804]
 [23.87639531]
 [23.54947354]
 [24.03985619]
 [23.14825137]
 [24.09929651]
 [23.99527595]]
```

```
In [29]: plt.scatter(X_train,Y_train,color='pink')
plt.plot(X_train,reg.predict(X_train), color='black',linewidth=5)
plt.title("TEMPERATURE VS YEAR")
plt.xlabel("Year")
plt.ylabel("Temperature")
plt.show()
```



```
In [30]: plt.scatter(X_test,Y_test,color='black')
plt.plot(X_train,reg.predict(X_train), color='blue',linewidth=4)
plt.title("TEMPERATURE VS YEAR")
plt.xlabel("Year")
plt.ylabel("Temperature")
plt.show()
```



```
In [ ]:
```