```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df = pd.read_csv('temperatures.csv')
```

```
df
```

	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	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
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

```
df.head()
```

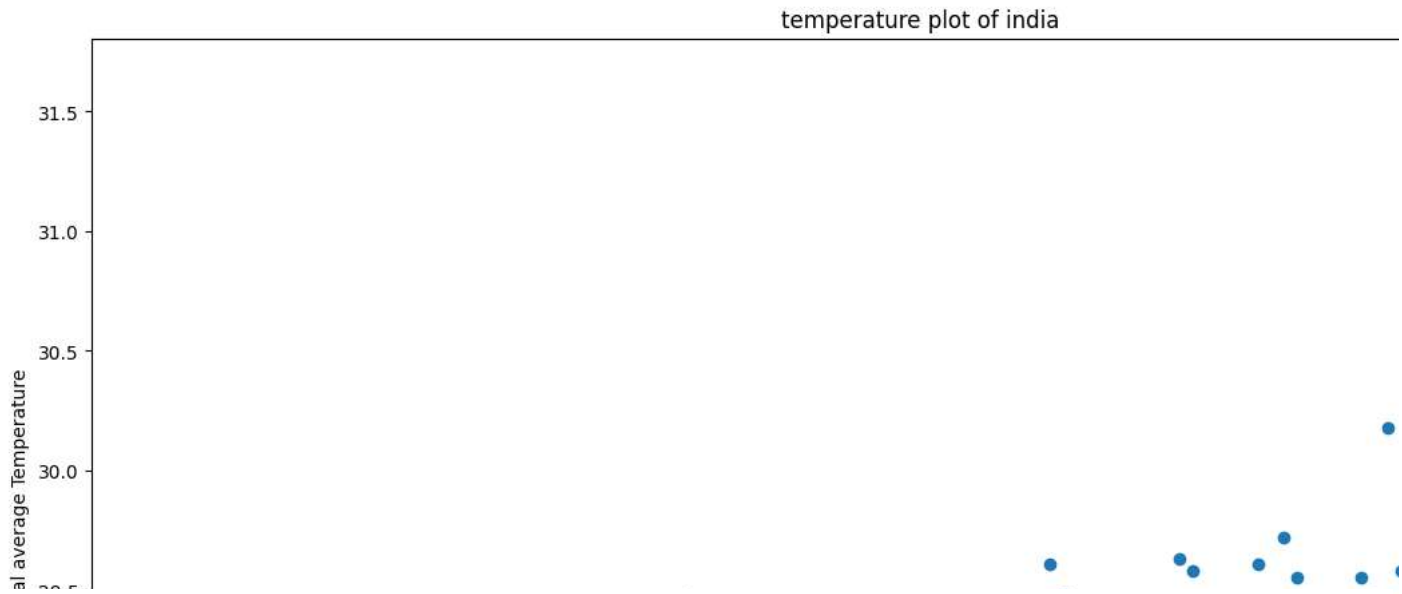
	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	JAN-FEB	MAR-MAY	JUN-SEP	OCT-DEC
0	1901	22.40	24.14	29.07	31.91	33.41	33.18	31.21	30.39	30.47	29.97	27.31	24.49	28.96	23.27	31.46	31.27	27.25
1	1902	24.93	26.58	29.77	31.78	33.73	32.91	30.92	30.73	29.80	29.12	26.31	24.04	29.22	25.75	31.76	31.09	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

```
x = df['YEAR']
```

```
y= df['ANNUAL']
```

```
plt.figure(figsize=(16,9))
plt.title('temperature plot of india')
plt.xlabel('Year')
plt.ylabel('annual average Temperature')
plt.scatter(x,y)
```

```
<matplotlib.collections.PathCollection at 0x7c2d8a2785e0>
```




```
x.shape
```

```
(117,)
```

```
x = x.values
```

```
x
```

```
array([1901, 1902, 1903, 1904, 1905, 1906, 1907, 1908, 1909, 1910, 1911,
       1912, 1913, 1914, 1915, 1916, 1917, 1918, 1919, 1920, 1921, 1922,
       1923, 1924, 1925, 1926, 1927, 1928, 1929, 1930, 1931, 1932, 1933,
       1934, 1935, 1936, 1937, 1938, 1939, 1940, 1941, 1942, 1943, 1944,
       1945, 1946, 1947, 1948, 1949, 1950, 1951, 1952, 1953, 1954, 1955,
       1956, 1957, 1958, 1959, 1960, 1961, 1962, 1963, 1964, 1965, 1966,
       1967, 1968, 1969, 1970, 1971, 1972, 1973, 1974, 1975, 1976, 1977,
       1978, 1979, 1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988,
       1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999,
       2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010,
       2011, 2012, 2013, 2014, 2015, 2016, 2017])
```

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

```
x.shape
```

```
(117, 1)
```

```
from sklearn.linear_model import LinearRegression
```

```
regressor = LinearRegression()
```

```
regressor.fit(x,y)
```

```
▾ LinearRegression
LinearRegression()
```

```
regressor.coef_
```

```
array([0.01312158])
```

```
regressor.intercept_
```

```
3.4761897126187016
```

```
regressor.predit(2024)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-20-f4ecb24de878> in <cell line: 1>()
----> 1 regressor.predit(2024)

AttributeError: 'LinearRegression' object has no attribute 'predit'
```

SEARCH STACK OVERFLOW

```
regressor.predict([[2024]])
```

```
array([30.03427031])
```

```
regressor.predict([[2074]])
```

```
array([30.69034937])
```

```
predicted = regressor.predict(x)
```

```
import numpy as np
```

```
#mean absolute error
np.mean(abs(y - predicted))
```

```
0.22535284978630413
```

```
from sklearn.metrics import mean_absolute_error
mean_absolute_error(y, predicted)
```

```
0.22535284978630413
```

```
from sklearn.metrics import mean_squared_error
mean_squared_error(y, predicted)
```

```
0.10960795229110352
```

```
from sklearn.metrics import r2_score
r2_score(y, predicted)
```

```
0.6418078912783682
```

```
sns.regplot(x='YEAR' , y='ANNUAL', data=df)
```

```
<Axes: xlabel='YEAR', ylabel='ANNUAL'>
```