In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df=pd.read_csv('temperatures.csv')
```

In [3]:
```
df.head()
```

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

In [4]:
```
df
```

Out[4]:

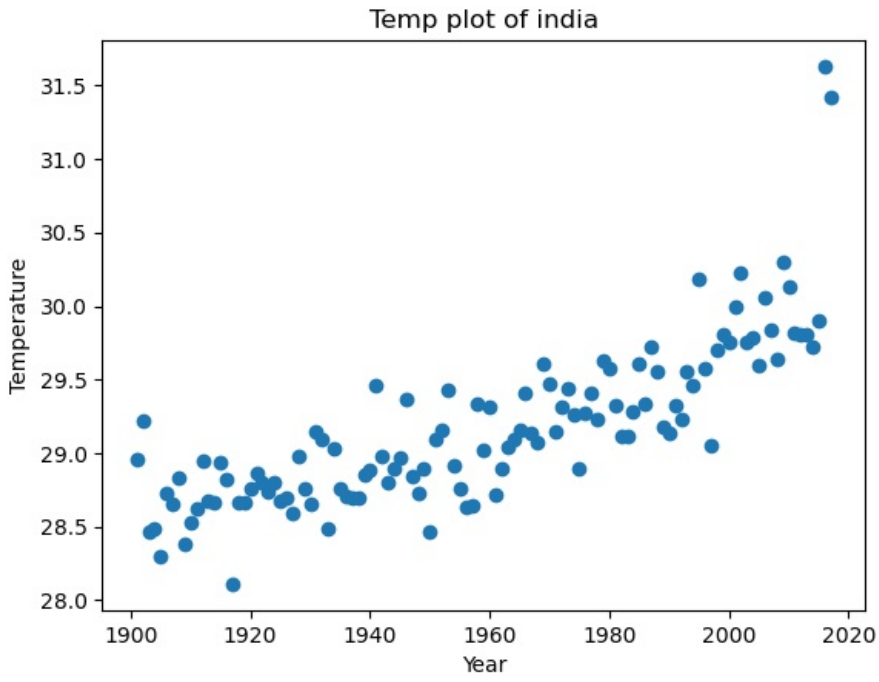
	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	JAN-FEB	MAR-MAY	JUN-SEP	OCT-DEC
0	1901	22.40	24.14	29.07	31.91	33.41	33.18	31.21	30.39	30.47	29.97	27.31	24.49	28.96	23.27	31.46	31.27	27.25
1	1902	24.93	26.58	29.77	31.78	33.73	32.91	30.92	30.73	29.80	29.12	26.31	24.04	29.22	25.75	31.76	31.09	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 [5]:
```
#imput data
x=df['YEAR']#Annual temp is dependent on year
#output data
y=df['ANNUAL']
```

In [6]:
```
#plt.figure(figsize=(16,9))
plt.title('Temp plot of india')
plt.xlabel('Year')
plt.ylabel('Temperature')
plt.scatter(x,y)
```

Out[6]:
```
<matplotlib.collections.PathCollection at 0x128f6536cf0>
```



In [7]:
```
x.shape
```

Out[7]: (117,)

In [8]:
```
x=x.values
```

In [9]:
```
x=x.reshape(117,1)
```

In [10]:
```
x.shape
```

Out[10]: (117, 1)

In [11]:
```
from sklearn.linear_model import  LinearRegression
```

In [12]:
```
regressor=LinearRegression()
```

In [13]:
```
regressor.fit(x,y)
```

Out[13]:
```
▾ LinearRegression
▶ Parameters
```

In [15]:
```
regressor.coef_
```

Out[15]: array([0.01312158])

In [16]:
```
regressor.intercept_
```

Out[16]: np.float64(3.4761897126187016)

In [17]:
```
#predication
```

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

Out[17]:
```
array([30.03427031])
```

In [18]:
```
predicted=regressor.predict(x)
```

In [19]:
```
#MEan absolute error
np.mean(abs(y-predicted))
```

Out[19]:
```
np.float64(0.22535284978630413)
```

In [20]:
```
#by using library
from sklearn.metrics import mean_absolute_error
mean_absolute_error(y,predicted)
```

Out[20]:
```
0.22535284978630413
```

In [21]:
```
#mean squared error
np.mean((y-predicted)**2)
```

Out[21]:
```
np.float64(0.10960795229110352)
```

In [22]:
```
from sklearn.metrics import mean_squared_error
mean_squared_error(y,predicted)
```

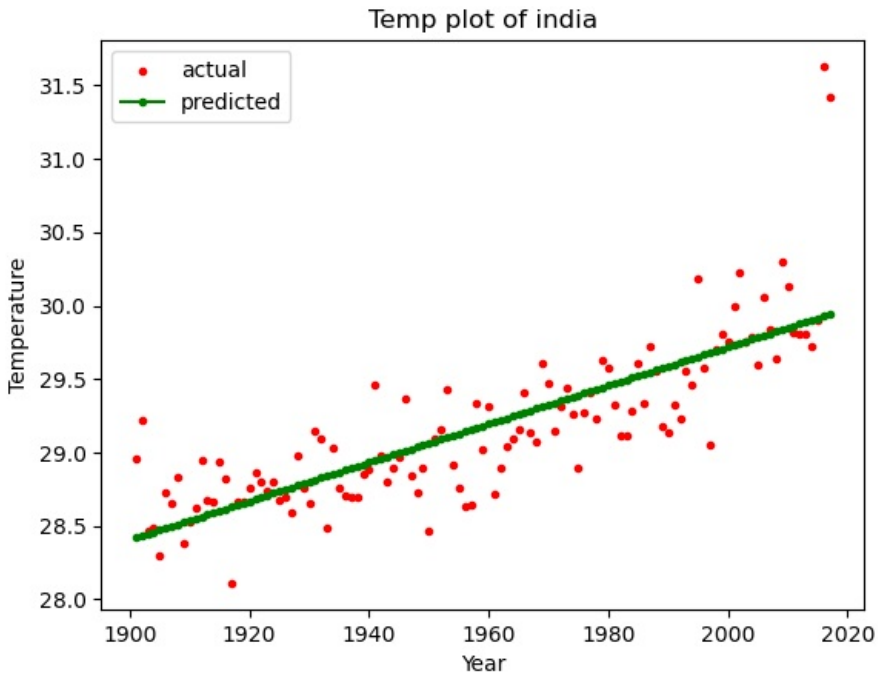
Out[22]:
```
0.10960795229110352
```

In [23]:
```
#r squared method is use to find linearity of model,performance measurement
from sklearn.metrics import r2_score
r2_score(y,predicted)
```

Out[23]:
```
0.6418078912783682
```

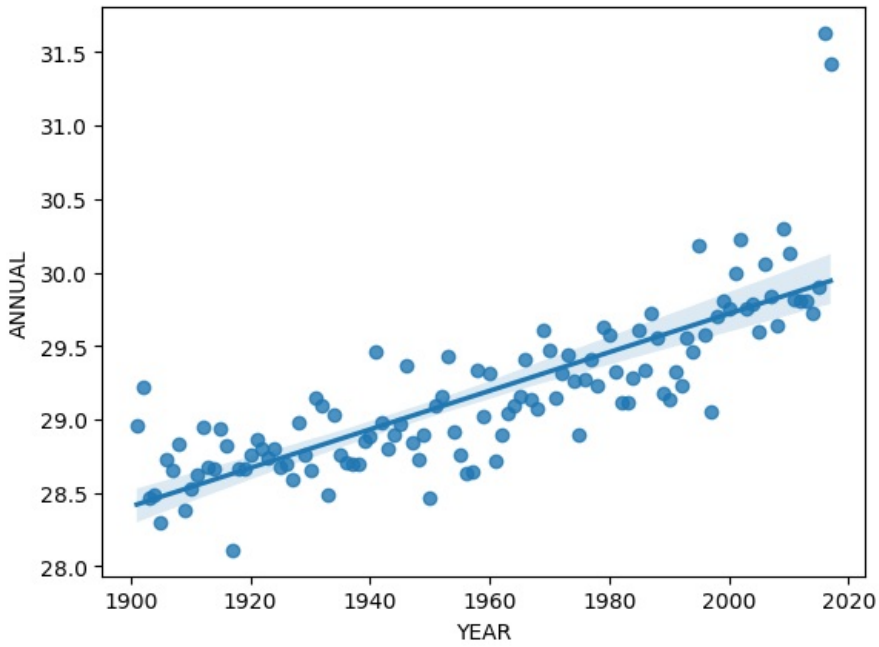
In [26]:
```
#plt.figure(figsize=(16,9))
plt.title('Temp plot of india')
plt.xlabel('Year')
plt.ylabel('Temperature')
plt.scatter(x,y,label='actual',color='r',marker='.')
plt.plot(x,predicted,label='predicted',color='g',marker='.')
plt.legend()
```

Out[26]:
```
<matplotlib.legend.Legend at 0x128f8dbcc20>
```



In [27]:
```
sns.regplot(x='YEAR',y='ANNUAL',data=df)
```

Out[27]:
```
<Axes: xlabel='YEAR', ylabel='ANNUAL'>
```


31.5
31.0
30.5
30.0
29.5
29.0
28.5
28.0
ANNUAL
1900
1920
1940
1960
1980
2000
2020
YEAR

In [ ]: