In [19]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
```

In [20]:
```
df=pd.read_csv("D:temperatures.csv")
```

In [21]:
```
df.describe()
```

Out[21]:

	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP
count	117.000000	117.000000	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	31.035897	30.507692	30.486752
std	33.919021	0.834588	1.150757	1.068451	0.889478	0.724905	0.633132	0.468818	0.476312	0.544295
min	1901.000000	22.000000	22.830000	26.680000	30.010000	31.930000	31.100000	29.760000	29.310000	29.070000
25%	1930.000000	23.100000	24.780000	28.370000	31.460000	33.110000	32.340000	30.740000	30.180000	30.120000
50%	1959.000000	23.680000	25.480000	29.040000	31.950000	33.510000	32.730000	31.000000	30.540000	30.520000
75%	1988.000000	24.180000	26.310000	29.610000	32.420000	34.030000	33.180000	31.330000	30.760000	30.810000
max	2017.000000	26.940000	29.720000	32.620000	35.380000	35.840000	34.480000	32.760000	31.840000	32.220000

In [22]:
```
df.head()
```

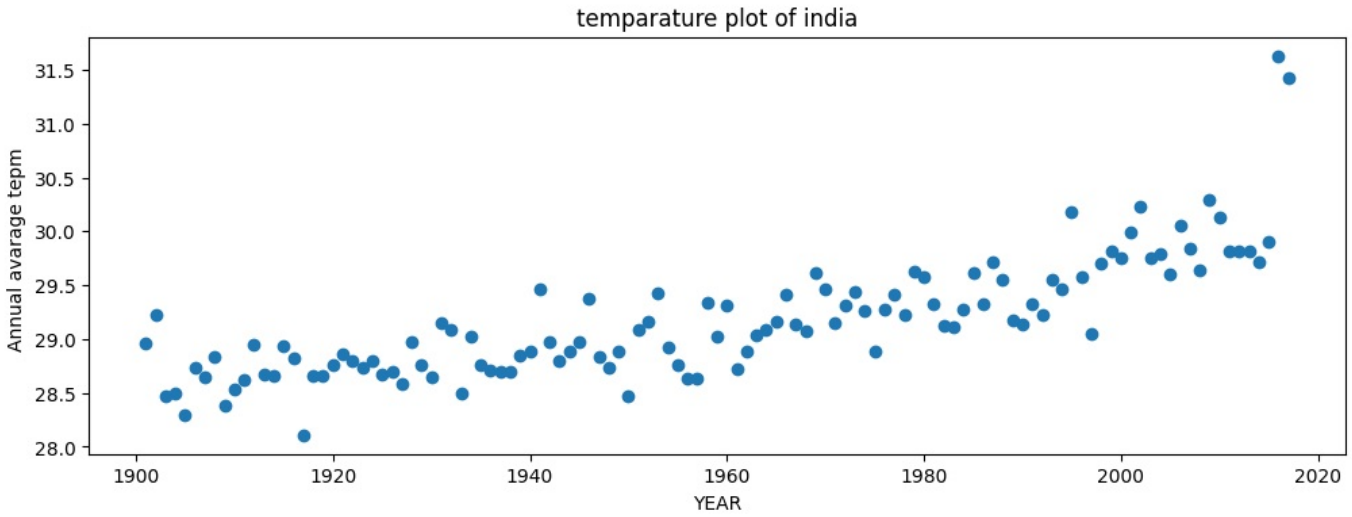
Out[22]:

	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 [23]:
```
x=df['YEAR']
y=df['ANNUAL']
```

In [24]:
```
plt.figure(figsize=(12,4))
plt.title('temparature plot of india')
plt.xlabel('YEAR')
plt.ylabel('Annual avarage tepm ')
plt.scatter(x,y)
```

Out[24]:
```
<matplotlib.collections.PathCollection at 0x1fc4f89bdc0>
```



In [25]:
```
x.shape
```

Out[25]:
```
(117,)
```

In [26]:
```
x=x.values
```

In [27]:
```
x=x.reshape(117,1)
```

In [28]:
```
y.shape
```

Out[28]:
```
(117,)
```

In [30]:
```
from sklearn.linear_model import LinearRegression
```

In [33]:
```
re=LinearRegression()
re.fit(x,y)
```

Out[33]:
```
LinearRegression()
```

In [34]:
```
re.coef_
```

Out[34]:
```
array([0.01312158])
```

In [35]:
```
re.intercept_
```

Out[35]:
```
np.float64(3.4761897126187016)
```

In [36]:
```
re.predict([[2024]])
```

Out[36]:
```
array([30.03427031])
```

In [37]:
```
predicted=re.predict(x)
```

In [38]:
```
predicted
```

Out[38]:
```
array([28.4203158 , 28.43343739, 28.44655897, 28.45968055, 28.47280213,
       28.48592371, 28.49904529, 28.51216687, 28.52528846, 28.53841004,
       28.55153162, 28.5646532 , 28.57777478, 28.59089636, 28.60401794,
       28.61713952, 28.63026111, 28.64338269, 28.65650427, 28.66962585,
       28.68274743, 28.69586901, 28.70899059, 28.72211218, 28.73523376,
       28.74835534, 28.76147692, 28.7745985 , 28.78772008, 28.80084166,
       28.81396324, 28.82708483, 28.84020641, 28.85332799, 28.86644957,
       28.87957115, 28.89269273, 28.90581431, 28.91893589, 28.93205748,
       28.94517906, 28.95830064, 28.97142222, 28.9845438 , 28.99766538,
       29.01078696, 29.02390855, 29.03703013, 29.05015171, 29.06327329,
       29.07639487, 29.08951645, 29.10263803, 29.11575961, 29.1288812 ,
       29.14200278, 29.15512436, 29.16824594, 29.18136752, 29.1944891 ,
       29.20761068, 29.22073227, 29.23385385, 29.24697543, 29.26009701,
       29.27321859, 29.28634017, 29.29946175, 29.31258333, 29.32570492,
       29.3388265 , 29.35194808, 29.36506966, 29.37819124, 29.39131282,
       29.4044344 , 29.41755599, 29.43067757, 29.44379915, 29.45692073,
       29.47004231, 29.48316389, 29.49628547, 29.50940705, 29.52252864,
       29.53565022, 29.5487718 , 29.56189338, 29.57501496, 29.58813654,
       29.60125812, 29.6143797 , 29.62750129, 29.64062287, 29.65374445,
       29.66686603, 29.67998761, 29.69310919, 29.70623077, 29.71935236,
       29.73247394, 29.74559552, 29.7587171 , 29.77183868, 29.78496026,
       29.79808184, 29.81120342, 29.82432501, 29.83744659, 29.85056817,
       29.86368975, 29.87681133, 29.88993291, 29.90305449, 29.91617608,
       29.92929766, 29.94241924])
```

In [39]:
```
y
```

Out[39]:
```
0      28.96
1      29.22
2      28.47
3      28.49
4      28.30
       ...
112    29.81
113    29.72
114    29.90
115    31.63
116    31.42
Name: ANNUAL, Length: 117, dtype: float64
```

In [40]:
```
y- predicted
```

Out[40]:
```
0      0.539684
1      0.786563
2      0.023441
3      0.030319
4     -0.172802
         ...
112   -0.079933
113   -0.183054
114   -0.016176
115    1.700702
116    1.477581
Name: ANNUAL, Length: 117, dtype: float64
```

In [41]:
```
import numpy as np
```

In [42]:
```
np.mean(abs(y-predicted)**2)
```

Out[42]:
```
np.float64(0.10960795229110352)
```

In [43]:
```
from sklearn.metrics import mean_squared_error
mean_squared_error(y,predicted)
```

Out[43]:
```
np.float64(0.10960795229110352)
```

In [44]:
```
from sklearn.metrics import r2_score
r2_score(y,predicted)
```

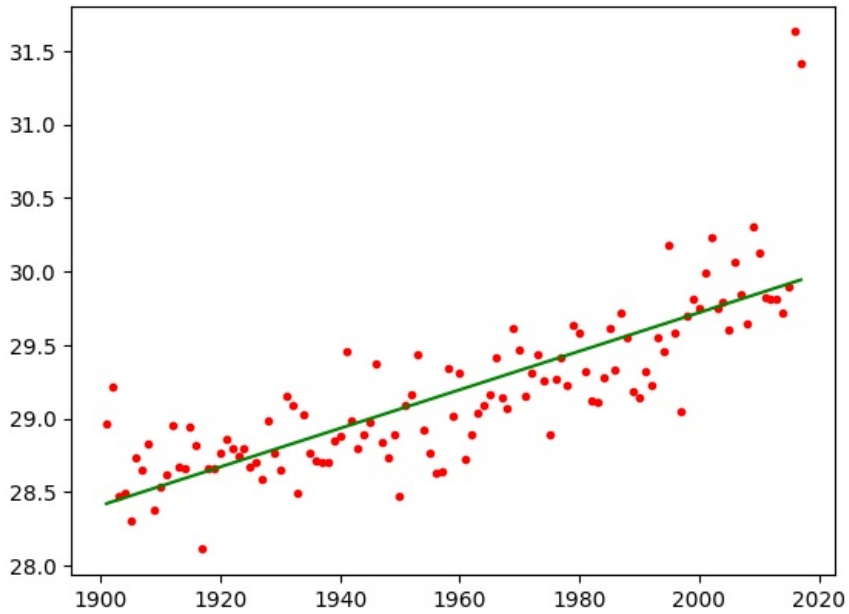
Out[44]:
```
0.6418078912783682
```

In [45]:
```
re.score(x,y)
```

Out[45]:
```
0.6418078912783682
```

In [47]:
```
plt.scatter (x,y,label='actual',color='r',marker='.')
plt.plot(x,predicted,label='predicted',color='g')
```

Out[47]:
```
[<matplotlib.lines.Line2D at 0x1fc791c0970>]
```



In [ ]:
```
```