In [58]:

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

In [59]:

```
df = pd.read_csv('temperatures 2.csv')
```

In [60]:

```
df.head()
```

Out[60]:

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

In [61]:

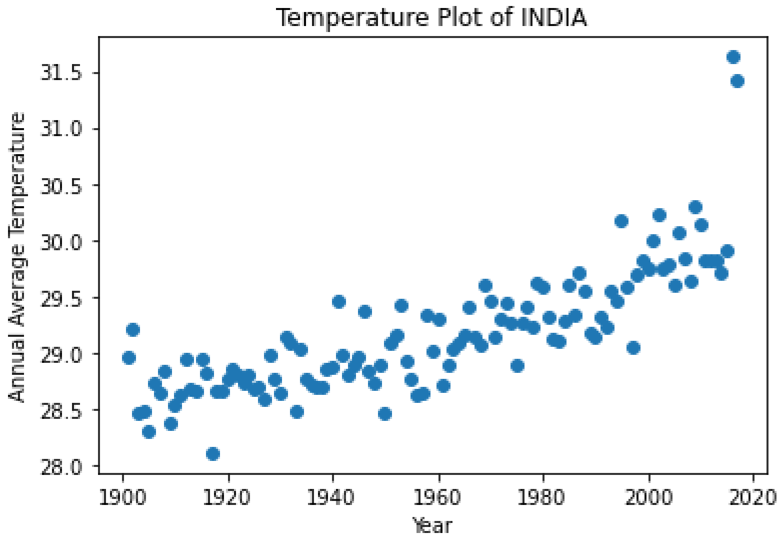
```
#input data
x = df['YEAR']

#output data
y = df['ANNUAL']
```

In [62]:

```
#plt.figure(figsize=(16,9))
plt.title('Temperature Plot of INDIA')
plt.xlabel('Year')
plt.ylabel('Annual Average Temperature')
plt.scatter(x,y)
```

Out[62]:

```
<matplotlib.collections.PathCollection at 0x7ff10d6443a0>
```



In [63]:

```
x.shape
```

Out[63]:

```
(117,)
```

In [64]:

```
x = x.values
```

In [65]:

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

In [66]:

```
x
```

```
       [1999],
       [2000],
       [2001],
       [2002],
       [2003],
       [2004],
       [2005],
       [2006],
       [2007],
       [2008],
       [2009],
       [2010],
       [2011],
       [2012],
       [2013],
       [2014],
       [2015],


       [2016],
```

In [67]:

```
x.shape
```

Out[67]:

```
(117, 1)
```

In [68]:

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

In [69]:

```
regressor = LinearRegression()
```

In [70]:

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

Out[70]:

```
LinearRegression()
```

In [72]:

```
regressor.coef_
```

Out[72]:

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

In [73]:

```
regressor.intercept_
```

Out[73]:

```
3.4761897126187016
```

In [75]:

```
regressor.predict([[2035]])
```

Out[75]:

```
array([30.1786077])
```

In [76]:

```
regressor.predict([[2055]])
```

Out[76]:

```
array([30.44103933])
```

In [77]:

```
regressor.predict([[2075]])
```

Out[77]:

```
array([30.70347095])
```

In [78]:

```
regressor.predict([[2099]])
```

Out[78]:

```
array([31.01838891])
```

In [79]:

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

In [80]:

```
predicted
```

Out[80]:

```
array([28.4203158 , 28.43343739, 28.44655897, 28.45968055, 28.4728021
3,
       28.48592371, 28.49904529, 28.51216687, 28.52528846, 28.5384100
4,
       28.55153162, 28.5646532 , 28.57777478, 28.59089636, 28.6040179
4,
       28.61713952, 28.63026111, 28.64338269, 28.65650427, 28.6696258
5,
       28.68274743, 28.69586901, 28.70899059, 28.72211218, 28.7352337
6,
       28.74835534, 28.76147692, 28.7745985 , 28.78772008, 28.8008416
6,
       28.81396324, 28.82708483, 28.84020641, 28.85332799, 28.8664495
7,
       28.87957115, 28.89269273, 28.90581431, 28.91893589, 28.9320574
8,
       28.94517906, 28.95830064, 28.97142222, 28.9845438 , 28.9976653
8,
       29.01078696, 29.02390855, 29.03703013, 29.05015171, 29.0632732
9,
       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, 29.2600970
1,
       29.27321859, 29.28634017, 29.29946175, 29.31258333, 29.3257049
2,
       29.3388265 , 29.35194808, 29.36506966, 29.37819124, 29.3913128
2,
       29.4044344 , 29.41755599, 29.43067757, 29.44379915, 29.4569207
3,
       29.47004231, 29.48316389, 29.49628547, 29.50940705, 29.5225286
4,
       29.53565022, 29.5487718 , 29.56189338, 29.57501496, 29.5881365
4,
       29.60125812, 29.6143797 , 29.62750129, 29.64062287, 29.6537444
5,
       29.66686603, 29.67998761, 29.69310919, 29.70623077, 29.7193523
6,
       29.73247394, 29.74559552, 29.7587171 , 29.77183868, 29.7849602
6,
       29.79808184, 29.81120342, 29.82432501, 29.83744659, 29.8505681
7,
       29.86368975, 29.87681133, 29.88993291, 29.90305449, 29.9161760
8,
       29.92929766, 29.94241924])
```

In [84]:

```
y
```

Out[84]:

```
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 [86]:

```
import numpy as np
```

In [87]:

```
# MEAN ABSOLUTE ERROR
np.mean(abs(y - predicted))
```

Out[87]:

```
0.22535284978630413
```

In [88]:

```
# MEAN ABSOLUTE ERROR
from sklearn.metrics import mean_absolute_error
mean_absolute_error(y,predicted)
```

Out[88]:

```
0.22535284978630413
```

In [89]:

```
#mean squared Error
from sklearn.metrics import mean_squared_error
mean_squared_error(y,predicted)
```

Out[89]:

```
0.10960795229110352
```

In [90]:

```
#mean squared Error
np.mean(abs(y - predicted)**2)
```

Out[90]:

```
0.10960795229110352
```

In [91]:

```
# R Square Error
from sklearn.metrics import r2_score
r2_score(y,predicted)
```

Out[91]:

```
0.6418078912783682
```

In [92]:

```
#R Square Error
regressor.score(x,y)
```

Out[92]:

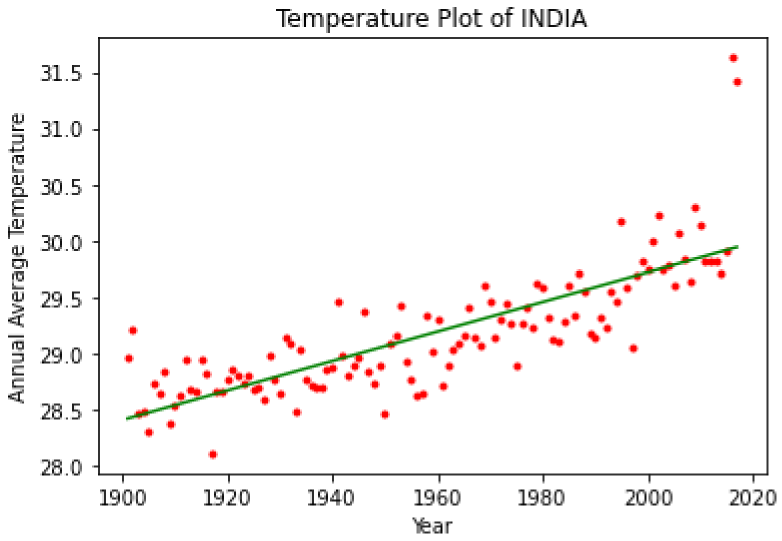
```
0.6418078912783682
```

In [103]:

```
plt.title('Temperature Plot of INDIA')
plt.xlabel('Year')
plt.ylabel('Annual Average Temperature')
plt.scatter(x,y,label ='actual',color='red',marker='.')
plt.plot(x,predicted,label ='predicted',color='g')
```

Out[103]:

```
[<matplotlib.lines.Line2D at 0x7ff10fbc9f70>]
```

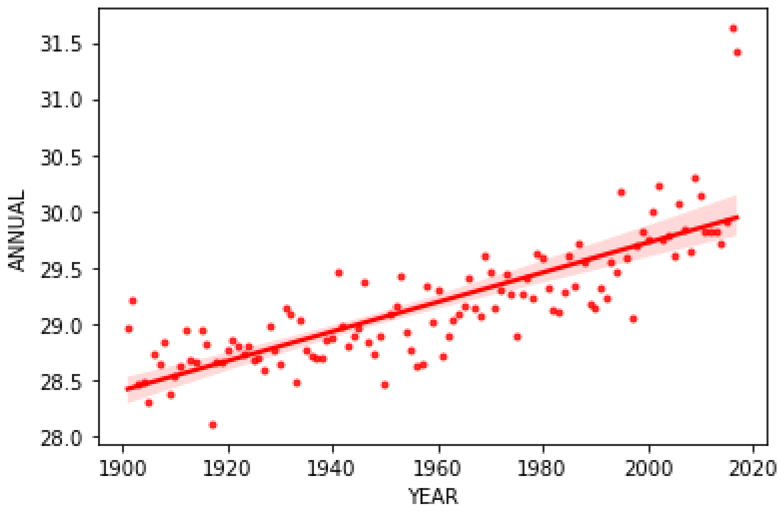


In [105]:

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

Out[105]:

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



In [ ]:

```

```