In [1]:

```
import pandas as pd
import numpy as np
import math
import sklearn
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error,r2_score,mean_absolute_error
import matplotlib.pyplot as plt
import seaborn as sb
from sklearn import preprocessing
```

In [2]:

```
data=pd.read_csv("C:\\Users\\Nagarur\\Documents\\MachineLearning\\rainfall.csv")
print("Data heads:")
print(data.head())
print("Null values in the dataset before preprocessing:")
print(data.isnull().sum())
print("Filling null values with mean of that particular column")
data=data.fillna(np.mean(data))
print("Mean of data:")
print(np.mean(data))
print("Null values in the dataset after preprocessing:")
print(data.isnull().sum())
print("\n\nShape: ",data.shape)
```

```
Data heads:
                  SUBDIVISION  YEAR   JAN    FEB   MAR    APR    MAY    JUN  \
0  ANDAMAN & NICOBAR ISLANDS  1901  49.2   87.1  29.2    2.3  528.8  517.5
1  ANDAMAN & NICOBAR ISLANDS  1902   0.0  159.8  12.2    0.0  446.1  537.1
2  ANDAMAN & NICOBAR ISLANDS  1903  12.7  144.0   0.0    1.0  235.1  479.9
3  ANDAMAN & NICOBAR ISLANDS  1904   9.4   14.7   0.0  202.4  304.5  495.1
4  ANDAMAN & NICOBAR ISLANDS  1905   1.3    0.0   3.3   26.9  279.5  628.7

     JUL    AUG    SEP    OCT    NOV    DEC  ANNUAL  Jan-Feb  Mar-May  \
0  365.1  481.1  332.6  388.5  558.2   33.6  3373.2    136.3    560.3
1  228.9  753.7  666.2  197.2  359.0  160.5  3520.7    159.8    458.3
2  728.4  326.7  339.0  181.2  284.4  225.0  2957.4    156.7    236.1
3  502.0  160.1  820.4  222.2  308.7   40.1  3079.6     24.1    506.9
4  368.7  330.5  297.0  260.7   25.4  344.7  2566.7      1.3    309.7

   Jun-Sep  Oct-Dec
0   1696.3    980.3
1   2185.9    716.7
2   1874.0    690.6
3   1977.6    571.0
4   1624.9    630.8
Null values in the dataset before preprocessing:
SUBDIVISION     0
YEAR            0
JAN             4
FEB             3
MAR             6
APR             4
MAY             3
JUN             5
JUL             7
AUG             4
SEP             6
OCT             7
NOV            11
DEC            10
ANNUAL         26
Jan-Feb         6
Mar-May         9
Jun-Sep        10
Oct-Dec        13
dtype: int64
Filling null values with mean of that particular column
Mean of data:
YEAR       1958.218659
JAN          18.957320
```

```
JAN           18.957520
FEB           21.805325
MAR           27.359197
APR           43.127432
MAY           85.745417
JUN          230.234444
JUL          347.214334
AUG          290.263497
SEP          197.361922
OCT           95.507009
NOV           39.866163
DEC           18.870580
ANNUAL      1411.008900
Jan-Feb       40.747786
Mar-May      155.901753
Jun-Sep     1064.724769
Oct-Dec      154.100487
dtype: float64
Null values in the dataset after preprocessing:
SUBDIVISION    0
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


Shape:  (4116, 19)
```

In [3]:

```
print("Info:")
print(data.info())
```

```
Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4116 entries, 0 to 4115
Data columns (total 19 columns):
SUBDIVISION    4116 non-null object
YEAR           4116 non-null int64
JAN            4116 non-null float64
FEB            4116 non-null float64
MAR            4116 non-null float64
APR            4116 non-null float64
MAY            4116 non-null float64
JUN            4116 non-null float64
JUL            4116 non-null float64
AUG            4116 non-null float64
SEP            4116 non-null float64
OCT            4116 non-null float64
NOV            4116 non-null float64
DEC            4116 non-null float64
ANNUAL         4116 non-null float64
Jan-Feb        4116 non-null float64
Mar-May        4116 non-null float64
Jun-Sep        4116 non-null float64
Oct-Dec        4116 non-null float64
dtypes: float64(17), int64(1), object(1)
memory usage: 611.0+ KB
None
```

In [4]:

```
print("Group by:")
data.groupby('SUBDIVISION').size()
```

Group by:

Out[4]:

```
SUBDIVISION
ANDAMAN & NICOBAR ISLANDS             110
ARUNACHAL PRADESH                      97
ASSAM & MEGHALAYA                     115
BIHAR                                 115
CHHATTISGARH                          115
COASTAL ANDHRA PRADESH                115
COASTAL KARNATAKA                     115
EAST MADHYA PRADESH                   115
EAST RAJASTHAN                        115
EAST UTTAR PRADESH                    115
GANGETIC WEST BENGAL                  115
GUJARAT REGION                        115
HARYANA DELHI & CHANDIGARH            115
HIMACHAL PRADESH                      115
JAMMU & KASHMIR                       115
JHARKHAND                             115
KERALA                                115
KONKAN & GOA                          115
LAKSHADWEEP                           114
MADHYA MAHARASHTRA                    115
MATATHWADA                            115
NAGA MANI MIZO TRIPURA                115
NORTH INTERIOR KARNATAKA              115
ORISSA                                115
PUNJAB                                115
RAYALSEEMA                            115
SAURASHTRA & KUTCH                    115
SOUTH INTERIOR KARNATAKA              115
SUB HIMALAYAN WEST BENGAL & SIKKIM    115
TAMIL NADU                            115
TELANGANA                             115
UTTARAKHAND                           115
VIDARBHA                              115
WEST MADHYA PRADESH                   115
WEST RAJASTHAN                        115
WEST UTTAR PRADESH                    115
dtype: int64
```

In [5]:

```
print("Co-Variance =",data.cov())
print("Co-Relation =",data.corr())
```

```
Co-Variance =                 YEAR          JAN          FEB           MAR           APR  \
YEAR     1098.319127   -62.525455   -26.333846     31.608776     17.985597
JAN       -62.525455  1126.880700   549.299548    627.375837    475.344306
FEB       -26.333846   549.299548  1288.551221    974.300386    892.111938
MAR        31.608776   627.375837   974.300386   2201.972143   1766.214414
APR        17.985597   475.344306   892.111938   1766.214414   4596.594854
MAY        14.668989   535.377197   896.648721   2094.072720   5433.121872
JUN      -105.646427  -265.334667   283.409286   1819.002965   7239.347234
JUL      -144.815256  -465.128448   156.524886   1227.377060   4883.946095
AUG        40.268753    75.536029   488.385320   1195.021166   3275.971621
SEP       -29.893921   110.216877   388.537697   1133.597551   3507.500523
OCT         7.922814    41.252990   -16.324506    401.044464   2478.380760
NOV       -42.624271   154.517515   -57.342083     28.337749    767.024409
DEC       -26.804831   311.215537   200.895960    270.309451    379.022457
ANNUAL   -239.492392  3182.200682  5831.636967  13551.372253  34892.913572
Jan-Feb   -87.619668  1675.112786  1837.200844   1601.188055   1367.102182
Mar-May    70.841300  1632.972340  2754.690073   6057.993672  11751.856167
Jun-Sep  -220.389375  -537.708253  1291.239831   5354.542547  18800.990715
Oct-Dec   -56.011703   507.405040   130.324910    702.446620   3611.891111

                 MAY           JUN           JUL           AUG  \
YEAR       14.668989   -105.646427   -144.815256     40.268753
```

JAN	535.377197	-265.334667	-465.128448	75.536029	
FEB	896.648721	283.409286	156.524886	488.385320	
MAR	2094.072720	1819.002965	1227.377060	1195.021166	
APR	5433.121872	7239.347234	4883.946095	3275.971621	
MAY	15175.769642	16377.076471	10998.210205	7649.136573	
JUN	16377.076471	55022.202897	46742.156704	28995.034435	
JUL	10998.210205	46742.156704	72528.044881	34865.908024	
AUG	7649.136573	28995.034435	34865.908024	35599.654379	
SEP	8195.841594	17494.413820	18689.175537	12689.249607	
OCT	6466.235613	11407.373968	8011.623303	4698.672264	
NOV	2957.825540	3683.106889	787.996420	225.143088	
DEC	1297.885757	878.194764	-221.331169	13.095578	
ANNUAL	76832.142562	186947.838753	196502.700078	128005.124370	
Jan-Feb	1430.274263	14.101413	-304.676604	568.034975	
Mar-May	22630.231857	25297.228770	16923.240288	12061.613866	
Jun-Sep	43086.245286	147777.656772	172787.647754	111996.536530	
Oct-Dec	10689.487044	15929.294431	8543.095960	4889.354198	

	SEP	OCT	NOV	DEC	ANNUAL	\
YEAR	-29.893921	7.922814	-42.624271	-26.804831	-239.492392	
JAN	110.216877	41.252990	154.517515	311.215537	3182.200682	
FEB	388.537697	-16.324506	-57.342083	200.895960	5831.636967	
MAR	1133.597551	401.044464	28.337749	270.309451	13551.372253	
APR	3507.500523	2478.380760	767.024409	379.022457	34892.913572	
MAY	8195.841594	6466.235613	2957.825540	1297.885757	76832.142562	
JUN	17494.413820	11407.373968	3683.106889	878.194764	186947.838753	
JUL	18689.175537	8011.623303	787.996420	-221.331169	196502.700078	
AUG	12689.249607	4698.672264	225.143088	13.095578	128005.124370	
SEP	18308.685373	5160.065598	1423.645365	625.337757	86880.184870	
OCT	5160.065598	9887.210250	3255.412215	1181.840586	52370.830566	
NOV	1423.645365	3255.412215	4705.074413	1306.412604	18933.853987	
DEC	625.337757	1181.840586	1306.412604	1790.821416	7849.560149	
ANNUAL	86880.184870	52370.830566	18933.853987	7849.560149	811776.910930	
Jan-Feb	497.514696	22.024800	92.528943	512.180397	9013.688346	
Mar-May	12753.170806	9323.753288	3734.819277	1929.952379	125275.096761	
Jun-Sep	67119.849971	29204.838323	6116.134359	1265.311019	598334.762240	
Oct-Dec	7199.381976	14283.355731	9219.400872	4279.018309	79154.756536	

	Jan-Feb	Mar-May	Jun-Sep	Oct-Dec
YEAR	-87.619668	70.841300	-220.389375	-56.011703
JAN	1675.112786	1632.972340	-537.708253	507.405040
FEB	1837.200844	2754.690073	1291.239831	130.324910
MAR	1601.188055	6057.993672	5354.542547	702.446620
APR	1367.102182	11751.856167	18800.990715	3611.891111
MAY	1430.274263	22630.231857	43086.245286	10689.487044
JUN	14.101413	25297.228770	147777.656772	15929.294431
JUL	-304.676604	16923.240288	172787.647754	8543.095960
AUG	568.034975	12061.613866	111996.536530	4889.354198
SEP	497.514696	12753.170806	67119.849971	7199.381976
OCT	22.024800	9323.753288	29204.838323	14283.355731
NOV	92.528943	3734.819277	6116.134359	9219.400872
DEC	512.180397	1929.952379	1265.311019	4279.018309
ANNUAL	9013.688346	125275.096761	598334.762240	79154.756536
Jan-Feb	3512.342973	4385.069178	756.357901	630.248247
Mar-May	4385.069178	40439.879705	66774.885424	14946.235220
Jun-Sep	756.357901	66774.885424	499680.824907	36465.214924
Oct-Dec	630.248247	14946.235220	36465.214924	27781.806104

Co-Relation =

	YEAR	JAN	FEB	MAR	APR	MAY	JUN	\
YEAR	1.000000	-0.056202	-0.022136	0.020325	0.008005	0.003593	-0.013590	
JAN	-0.056202	1.000000	0.455847	0.398275	0.208858	0.129463	-0.033697	
FEB	-0.022136	0.455847	1.000000	0.578410	0.366564	0.202766	0.033658	
MAR	0.020325	0.398275	0.578410	1.000000	0.555162	0.362252	0.165256	
APR	0.008005	0.208858	0.366564	0.555162	1.000000	0.650513	0.455211	
MAY	0.003593	0.129463	0.202766	0.362252	0.650513	1.000000	0.566751	
JUN	-0.013590	-0.033697	0.033658	0.165256	0.455211	0.566751	1.000000	
JUL	-0.016225	-0.051449	0.016191	0.097122	0.267485	0.331508	0.739923	
AUG	0.006440	0.011926	0.072109	0.134973	0.256094	0.329090	0.655136	
SEP	-0.006666	0.024265	0.079993	0.178535	0.382341	0.491688	0.551191	
OCT	0.002404	0.012359	-0.004574	0.085951	0.367632	0.527885	0.489080	
NOV	-0.018750	0.067105	-0.023288	0.008804	0.164933	0.350037	0.228909	
DEC	-0.019113	0.219077	0.132250	0.136122	0.132105	0.248963	0.088470	
ANNUAL	-0.008021	0.105213	0.180311	0.320523	0.571217	0.692228	0.884572	
Jan-Feb	-0.044611	0.841989	0.863589	0.575755	0.340239	0.195905	0.001014	
Mar-May	0.010630	0.241899	0.381608	0.641975	0.861953	0.913500	0.536289	
Jun-Sep	-0.009408	-0.022660	0.050887	0.161425	0.392298	0.494785	0.891237	
Oct-Dec	-0.010140	0.090685	0.021782	0.089811	0.319622	0.520597	0.407425	

```
              JUL       AUG       SEP       OCT       NOV       DEC    ANNUAL  \
YEAR    -0.016225  0.006440 -0.006666  0.002404 -0.018750 -0.019113 -0.008021
JAN     -0.051449  0.011926  0.024265  0.012359  0.067105  0.219077  0.105213
FEB      0.016191  0.072109  0.079993 -0.004574 -0.023288  0.132250  0.180311
MAR      0.097122  0.134973  0.178535  0.085951  0.008804  0.136122  0.320523
APR      0.267485  0.256094  0.382341  0.367632  0.164933  0.132105  0.571217
MAY      0.331508  0.329090  0.491688  0.527885  0.350037  0.248963  0.692228
JUN      0.739923  0.655136  0.551191  0.489080  0.228909  0.088470  0.884572
JUL      1.000000  0.686160  0.512872  0.299179  0.042657 -0.019421  0.809836
AUG      0.686160  1.000000  0.497032  0.250447  0.017396  0.001640  0.752985
SEP      0.512872  0.497032  1.000000  0.383522  0.153388  0.109209  0.712646
OCT      0.299179  0.250447  0.383522  1.000000  0.477294  0.280864  0.584567
NOV      0.042657  0.017396  0.153388  0.477294  1.000000  0.450061  0.306364
DEC     -0.019421  0.001640  0.109209  0.280864  0.450061  1.000000  0.205874
ANNUAL   0.809836  0.752985  0.712646  0.584567  0.306364  0.205874  1.000000
Jan-Feb -0.019089  0.050799  0.062041  0.003737  0.022761  0.204220  0.168805
Mar-May  0.312482  0.317891  0.468689  0.466282  0.270758  0.226786  0.691419
Jun-Sep  0.907639  0.839722  0.701740  0.415501  0.126138  0.042298  0.939463
Oct-Dec  0.190319  0.155471  0.319217  0.861813  0.806379  0.606649  0.527082

            Jan-Feb   Mar-May   Jun-Sep   Oct-Dec
YEAR    -0.044611  0.010630 -0.009408 -0.010140
JAN      0.841989  0.241899 -0.022660  0.090685
FEB      0.863589  0.381608  0.050887  0.021782
MAR      0.575755  0.641975  0.161425  0.089811
APR      0.340239  0.861953  0.392298  0.319622
MAY      0.195905  0.913500  0.494785  0.520597
JUN      0.001014  0.536289  0.891237  0.407425
JUL     -0.019089  0.312482  0.907639  0.190319
AUG      0.050799  0.317891  0.839722  0.155471
SEP      0.062041  0.468689  0.701740  0.319217
OCT      0.003737  0.466282  0.415501  0.861813
NOV      0.022761  0.270758  0.126138  0.806379
DEC      0.204220  0.226786  0.042298  0.606649
ANNUAL   0.168805  0.691419  0.939463  0.527082
Jan-Feb  1.000000  0.367937  0.018054  0.063802
Mar-May  0.367937  1.000000  0.469745  0.445909
Jun-Sep  0.018054  0.469745  1.000000  0.309494
Oct-Dec  0.063802  0.445909  0.309494  1.000000
```

In [6]:

```
corr_cols=data.corr()['ANNUAL'].sort_values()[::-1]
print("Index of correlation columns:",corr_cols.index)
```

```
Index of correlation columns: Index(['ANNUAL', 'Jun-Sep', 'JUN', 'JUL', 'AUG', 'SEP', 'MAY', 'Mar-
May',
       'OCT', 'APR', 'Oct-Dec', 'MAR', 'NOV', 'DEC', 'FEB', 'Jan-Feb', 'JAN',
       'YEAR'],
      dtype='object')
```

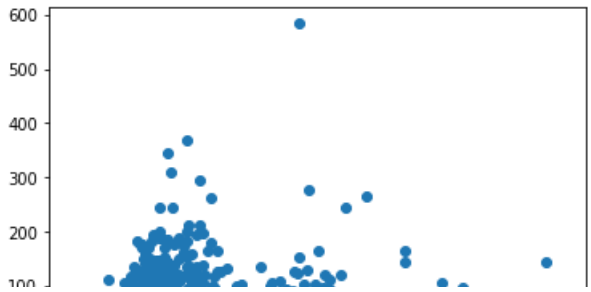
In [7]:

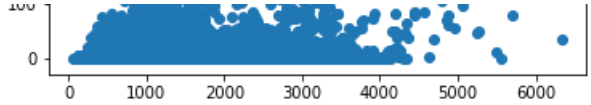
```
print("Scatter plot of annual and january attributes")
plt.scatter(data.ANNUAL,data.JAN)
```

```
Scatter plot of annual and january attributes
```

Out[7]:

```
<matplotlib.collections.PathCollection at 0x102dc7c940>
```





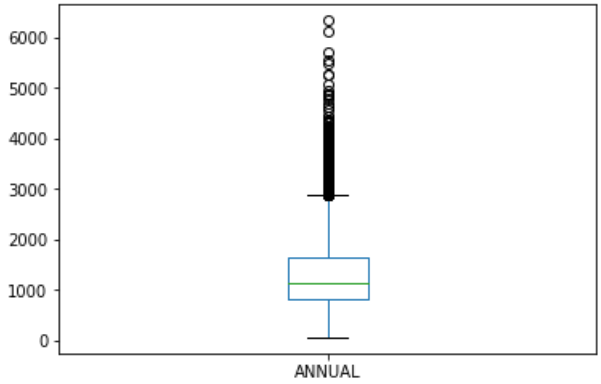
In [8]:

```
print("Box Plot of annual rainfall data in years 1901-2015")
data['ANNUAL'].plot(kind='box',sharex=False,sharey=False)
```

```
Box Plot of annual rainfall data in years 1901-2015
```

Out[8]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x102dc59048>
```



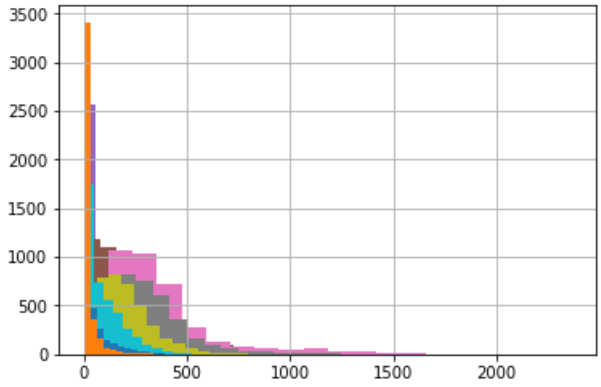
In [9]:

```
print("Histograms showing the data from attributes (JANUARY to DECEMBER) of the years 1901-2015:")
data['JAN'].hist(bins=20)
data['FEB'].hist(bins=20)
data['MAR'].hist(bins=20)
data['APR'].hist(bins=20)
data['MAY'].hist(bins=20)
data['JUN'].hist(bins=20)
data['JUL'].hist(bins=20)
data['AUG'].hist(bins=20)
data['SEP'].hist(bins=20)
data['OCT'].hist(bins=20)
data['NOV'].hist(bins=20)
data['DEC'].hist(bins=20)
```

```
Histograms showing the data from attributes (JANUARY to DECEMBER) of the years 1901-2015:
```

Out[9]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x102dc080b8>
```



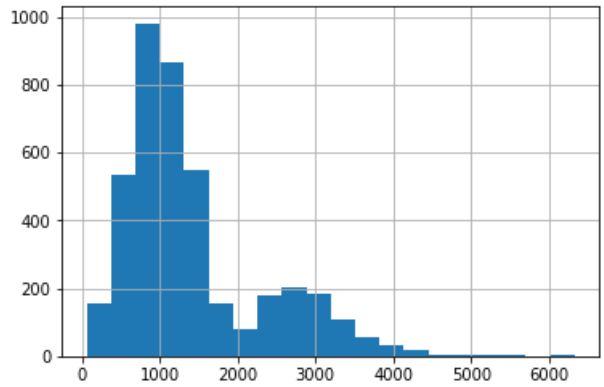
In [10]:

```
print("Histogram showing the annual rainfall of the all states:")
data['ANNUAL'].hist(bins=20)
```

Histogram showing the annual rainfall of the all states:

Out[10]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x102e38b550>
```



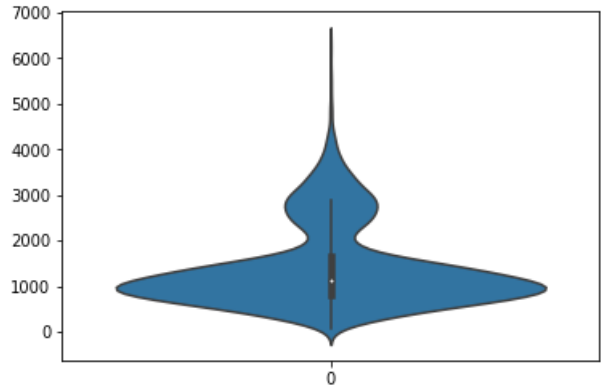
In [11]:

```
print("Violin plot of the ANNUAL attribute :-")
sb.violinplot(data=data['ANNUAL'])
```

Violin plot of the ANNUAL attribute :-

Out[11]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x102e5ffc18>
```



In [12]:

```
d2=data.drop(['SUBDIVISION','YEAR','ANNUAL','Jan-Feb','Mar-May','Jun-Sep','Oct-Dec'],axis=1)
k=((d2.head().sum()))
month=list(d2.head())
print("Months are: ",month)
print(k)
s=0
for i in d2.sum():
    s=s+i
print("Total recorded rainfall in these 12 months",s)
probability=list(k/s)
print(probability)
max_rainfall=max(probability)
for i in range(len(month)):
    if probability[i]==max_rainfall:
        print("Maximum Rainfall will be in the month of",month[i])
min_rainfall=min(probability)
for i in range(len(month)):
    if probability[i]==min_rainfall:
        print("Minimum Rainfall will be in the month of",month[i])
```

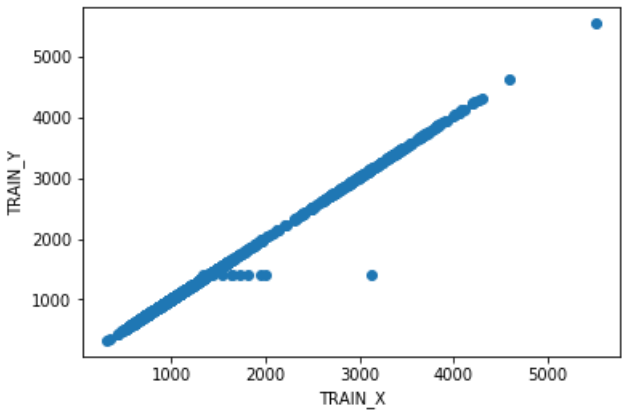
```
Months are:  ['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC']
JAN      72.6
FEB     405.6
```

```
MAR      44.7
APR     232.6
MAY    1794.0
JUN    2658.3
JUL    2193.1
AUG    2052.1
SEP    2455.2
OCT    1249.8
NOV    1535.7
DEC     803.9
dtype: float64
Total recorded rainfall in these 12 months 5829542.827152476
[1.2453806782557342e-05, 6.957663954552696e-05, 7.667839713227454e-06, 3.9900212914915124e-05,
0.00030774282875906155, 0.00045600488388529176, 0.0003762044580554615, 0.00035201731265132127,
0.00042116510210103014, 0.0002143907399013797, 0.0002634340368591365, 0.00013790103681126512]
Maximum Rainfall will be in the month of JUN
Minimum Rainfall will be in the month of MAR
```

In [13]:

```
from sklearn import linear_model
print("___Multiple Linear regression model between annual rainfall and the periodic rainfall___")
y=data['ANNUAL']
x=data[['Jan-Feb','Mar-May','Jun-Sep','Oct-Dec']]
train_x,test_x,train_y,test_y=train_test_split(x,y,test_size=0.3,shuffle=False)
'''train_x=train_x[:,np.newaxis]
test_x=test_x[:,np.newaxis]'''
print("Train x shape",train_x.shape,"; Test_x",test_x.shape)
print("Train y shape",train_y.shape,"; Test_y",test_y.shape)
lm=linear_model.LinearRegression()
lm.fit(train_x,train_y)
pred=lm.predict(test_x)
#print(test_y)
#print(pred)
print("Mean Squared Error =",mean_squared_error(test_y,pred))
print("Root Mean Squared Error =",np.sqrt(mean_squared_error(test_y,pred)))
print("Mean Absolute Error =",mean_absolute_error(test_y,pred))
print("r2_score =",r2_score(test_y,pred))
plt.scatter(pred,test_y)
plt.xlabel('TRAIN_X')
plt.ylabel('TRAIN_Y')
plt.show()
```

```
___Multiple Linear regression model between annual rainfall and the periodic rainfall___
Train x shape (2881, 4) ; Test_x (1235, 4)
Train y shape (2881,) ; Test_y (1235,)
Mean Squared Error = 3326.4157535418863
Root Mean Squared Error = 57.67508780697162
Mean Absolute Error = 10.953757241508946
r2_score = 0.9958637383726687
```



In [14]:

```
expected=[]
for i in test_y:
    if i>2000:
        expected.append("high")
    else:
```

```
        expected.append("low")
predicted=[]
for i in pred:
    if i>2000:
        predicted.append("high")
    else:
        predicted.append("low")
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
acc=accuracy_score(predicted,expected)
matrix=confusion_matrix(predicted,expected)
clas=classification_report(predicted,expected)
print("accuracy")
print(acc)
print("\n")
print("classification")
print(clas)
```

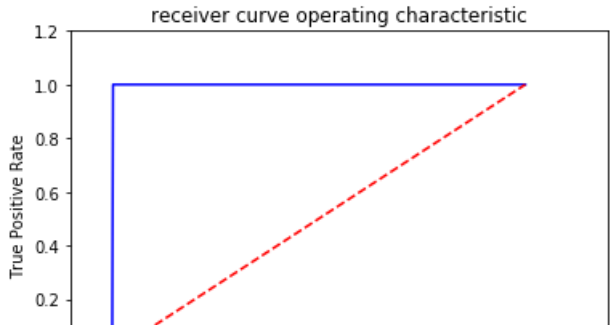
```
accuracy
0.9983805668016195


classification
             precision    recall  f1-score   support

       high       1.00      0.99      1.00       237
        low       1.00      1.00      1.00       998

avg / total       1.00      1.00      1.00      1235
```

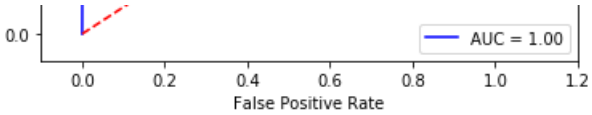
In [15]:

```
exp=[]
pre=[]
for i in expected:
    if i=='high':
        exp.append(1)
    else:
        exp.append(0)
for i in predicted:
    if i=='high':
        pre.append(1)
    else:
        pre.append(0)
```

In [16]:

```
from sklearn.metrics import roc_curve,auc
import random
fpr,tpr,threshold=roc_curve(exp,pre)
roc_auc=auc(fpr,tpr)
plt.title("receiver curve operating characteristic")
plt.plot(fpr,tpr,'b',label='AUC = %0.2f'%roc_auc)
plt.legend(loc='lower right')
plt.plot([0,1],[0,1],'r--')
plt.xlim([-0.1,1.2])
plt.ylim([-0.1,1.2])
plt.ylabel("True Positive Rate")
plt.xlabel("False Positive Rate")
plt.show()
```




0.0
AUC = 1.00
0.0
0.2
0.4
0.6
0.8
1.0
1.2
False Positive Rate