# MINI PROJECT

# RAINFALL PREDICTION SYSTEM

## SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

## FOR

## LAB PRACTICAL II

## BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

## SUBMITTED BY


| | |
|---|---|
| **Name: JAY JADHAV** | **Exam Seat No: B150614248** |
| **Name: SHARMIL ADROJA** | **Exam Seat No: B150614201** |
| **Name: KRUSHNA AVHAD** | **Exam Seat No: B150614208** |


## UNDER THE GUIDANCE OF
## PROF. VIVEK WAGHMARE



## DEPARTMENT OF COMPUTER ENGINEERING
**SANDIP INSTITUTE OF TECHNOLOGY & RESEARCH CENTRE,**
**NASHIK-422001**
**SAVITRIBAI PHULE PUNE UNIVERSITY**

**2020-21**

# 1 India Rainfall Analysis

## 1.1 Introduction

The climate of India consists of a wide range of weather conditions across a vast geographic scale and varied topography, making generalizations difficult. Climate in South India is generally hotter and extremely humid than that of North India. South India is more humid due to nearby coasts. The southern half of the nation doesn't experience temperatures below 10 °C (50 °F) in winter, and the temperature usually tends to exceed 40 °C (104 °F) during summer. Based on the Kappen system, India hosts six major climatic sub types, ranging from arid deserts in the west, alpine tundra and glaciers in the north, and humid tropical regions supporting rain forests in the southwest and the island territories. Many regions have starkly different microclimates, making it one of the most climatically diverse countries in the world. The country's meteorological department follows the international standard of four seasons with some local adjustments: winter (January and February), summer (March, April and May), monsoon (rainy) season (June to September), and a post-monsoon period (October to December).
India's geography and geology are climatically pivotal: the Thar Desert in the northwest and the Himalayas in the north work in tandem to create a culturally and economically important monsoonal regime.

## 1.2 Motivation and Description

Monsoon prediction is clearly of great importance for India.
Two types of rainfall predictions can be done, they are

- Long term predictions: Predict rainfall over few weeks/months in advance.
- Short term predictions: Predict rainfall a few days in advance in specific locations.

Indian meteorological department provides forecasting data required for project. In this project we are planning to work on long term predictions of rainfall. The main motive of the project is to predict the amount of rainfall in a particular division or state well in advance. We predict the amount of rainfall using past data.

## 1.3 Dataset

- Dataset1(dataset1) This dataset has average rainfall from 1951-2000 for each district, for every month.
- Dataset2(dataset2) This dataset has average rainfall for every year from 1901-2015 for each state.

## 1.4 Methodology

- Converting data in to the correct format to conduct experiments.
- Make a good analysis of data and observe variation in the patterns of rainfall.
- Finally, we try to predict the average rainfall by separating data into various visualizations. We apply various visualization technique to make analysis over various different data. By using various approaches, we try to acquire what we want.

- In [1]:

- import numpy as np # linear algebra

- import pandas as pd (data processing, CSV file I/O (e.g. pd. read_csv))

- import matplotlib. pyplot as plt import seaborn as sns

## 1.5 Types of graphs

- Bar graphs showing distribution of amount of rainfall.
- Distribution of amount of rainfall yearly, monthly, groups of months.
- Distribution of rainfall in subdivisions, districts form each month, groups of months.

In [2]:
```
data = pd.read_csv("../data/rainfall_in_india_1901-2015.csv",sep=",") data =
    data.fillna(data.mean()) data.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
```

## 1.6 Dataset-1 Description

- Data has 36 sub divisions and 19 attributes (individual months, annual, combinations of 3 consecutive months).
- For some of the subdivisions data is from 1950 to 2015.
- All the attributes has the sum of amount of rainfall in mm.

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

Out [3]:

| | 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 | Jun-Sep | Oct-Dec |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |
| 3 | 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 | | |

```
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
```

In [4]: data.describe()

Out [4]:

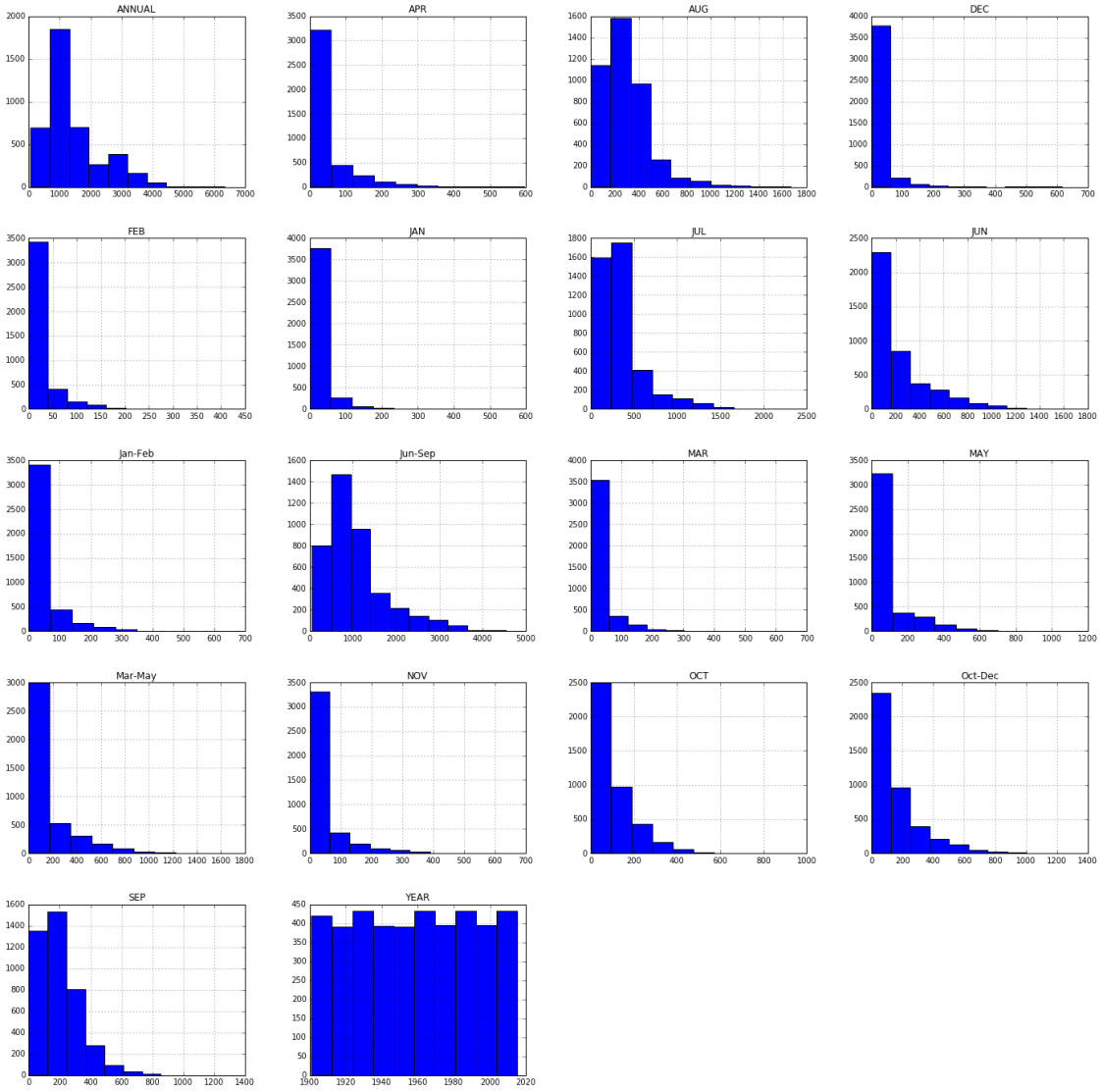
| | YEAR | JAN | FEB | MAR | APR |
|---|---|---|---|---|---|
| count | 4116.000000 | 4116.000000 | 4116.000000 | 4116.000000 | 4116.000000 |
| mean | 1958.218659 | 18.957320 | 21.805325 | 27.359197 | 43.127432 |
| std | 33.140898 | 33.569044 | 35.896396 | 46.925176 | 67.798192 |
| min | 1901.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 25% | 1930.000000 | 0.600000 | 0.600000 | 1.000000 | 3.000000 |
| 50% | 1958.000000 | 6.000000 | 6.700000 | 7.900000 | 15.700000 |
| 75% | 1987.000000 | 22.125000 | 26.800000 | 31.225000 | 49.825000 |
| max | 2015.000000 | 583.700000 | 403.500000 | 605.600000 | 595.100000 |

| | MAY | JUN | JUL | AUG | SEP |
|---|---|---|---|---|---|
| count | 4116.000000 | 4116.000000 | 4116.000000 | 4116.000000 | 4116.000000 |
| mean | 85.745417 | 230.234444 | 347.214334 | 290.263497 | 197.361922 |
| std | 123.189974 | 234.568120 | 269.310313 | 188.678707 | 135.309591 |
| min | 0.000000 | 0.400000 | 0.000000 | 0.000000 | 0.100000 |
| 25% | 8.600000 | 70.475000 | 175.900000 | 156.150000 | 100.600000 |
| 50% | 36.700000 | 138.900000 | 284.900000 | 259.500000 | 174.100000 |
| 75% | 96.825000 | 304.950000 | 418.225000 | 377.725000 | 265.725000 |
| max | 1168.600000 | 1609.900000 | 2362.800000 | 1664.600000 | 1222.000000 |

| | OCT | NOV | DEC | ANNUAL | Jan-Feb |
|---|---|---|---|---|---|
| count | 4116.000000 | 4116.000000 | 4116.000000 | 4116.000000 | 4116.000000 |
| mean | 95.507009 | 39.866163 | 18.870580 | 1411.008900 | 40.747786 |
| std | 99.434452 | 68.593545 | 42.318098 | 900.986632 | 59.265023 |
| min | 0.000000 | 0.000000 | 0.000000 | 62.300000 | 0.000000 |
| 25% | 14.600000 | 0.700000 | 0.100000 | 806.450000 | 4.100000 |
| 50% | 65.750000 | 9.700000 | 3.100000 | 1125.450000 | 19.300000 |
| 75% | 148.300000 | 45.825000 | 17.700000 | 1635.100000 | 50.300000 |
| max | 948.300000 | 648.900000 | 617.500000 | 6331.100000 | 699.500000 |

| | Mar-May | Jun-Sep | Oct-Dec |
|---|---|---|---|
| count | 4116.000000 | 4116.000000 | 4116.000000 |
| mean | 155.901753 | 1064.724769 | 154.100487 |
| std | 201.096692 | 706.881054 | 166.678751 |
| min | 0.000000 | 57.400000 | 0.000000 |
| 25% | 24.200000 | 574.375000 | 34.200000 |
| 50% | 75.200000 | 882.250000 | 98.800000 |
| 75% | 196.900000 | 1287.550000 | 212.600000 |
| max | 1745.800000 | 4536.900000 | 1252.500000 |

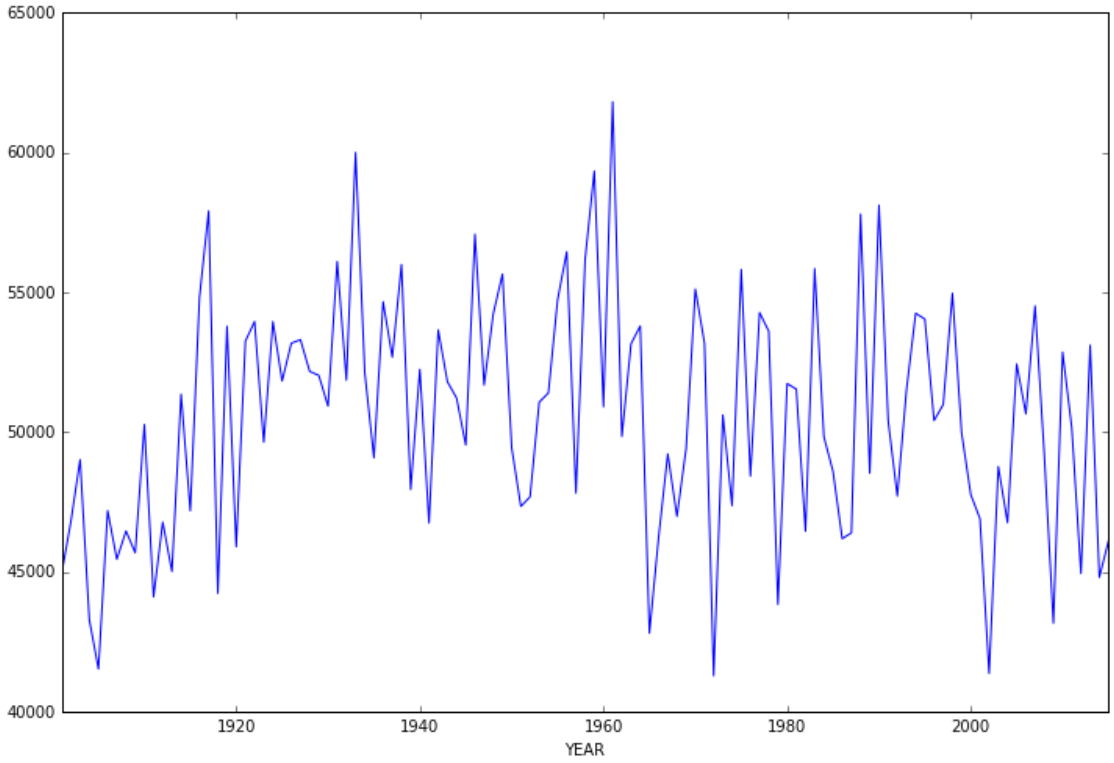
In [5]: data.hist(figsize=(24,24));




## 1.7 Observations

- Above histograms show the distribution of rainfall over months.
- Observed increase in amount of rainfall over months July, August, September.

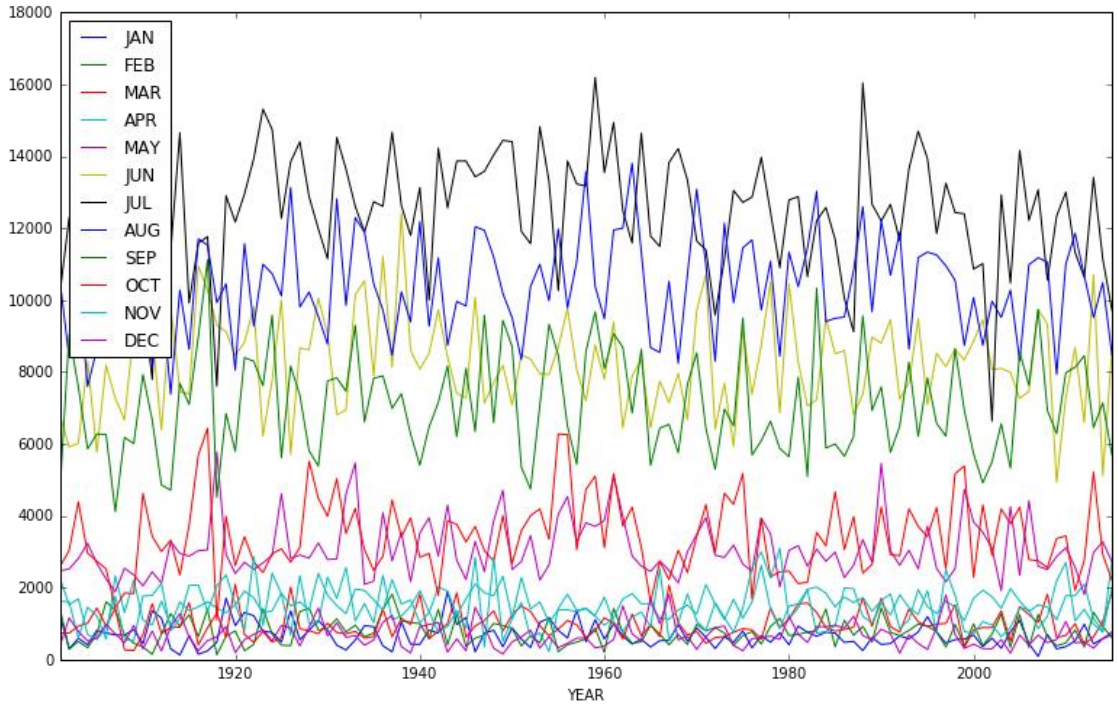
In [6]: data.groupby("YEAR").sum()['ANNUAL'].plot(figsize=(12,8));



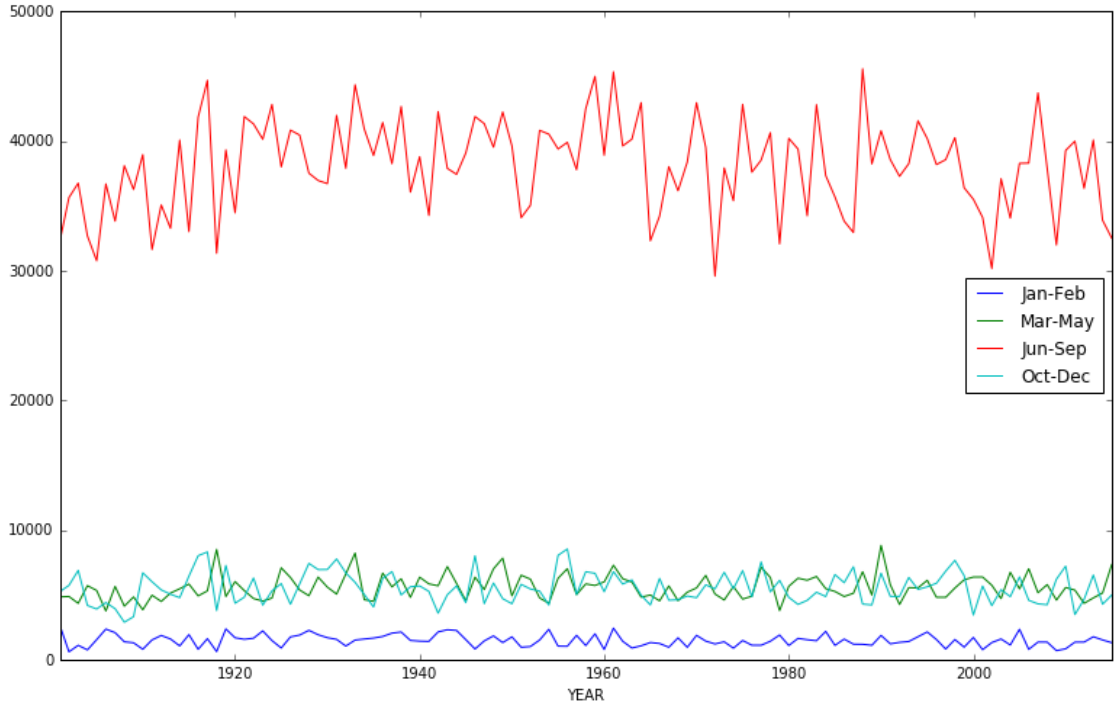

## 1.8 Observations

- Shows distribution of rainfall over years.
- Observed high amount of rainfall in 1950s.

```
In [7]: data[['YEAR', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
                                        'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].group by("YEAR").sum().plot(figsize=(13,8));
```




In [8]: data[['YEAR','Jan-Feb', 'Mar-May',
'Jun-Sep', 'Oct-Dec']].groupby("YEAR").sum().plot(figsize=(13,8));

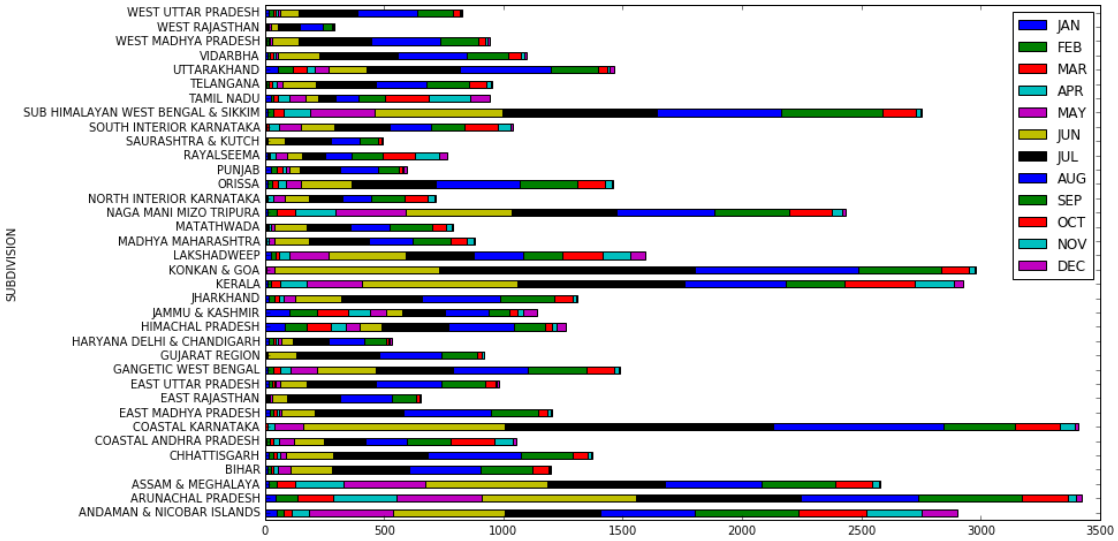



## 1.9 Observations

- The above two graphs show the distribution of rainfall over months.

- The graphs clearly shows that amount of rainfall in high in the months July, August, September which is monsoon season in India.

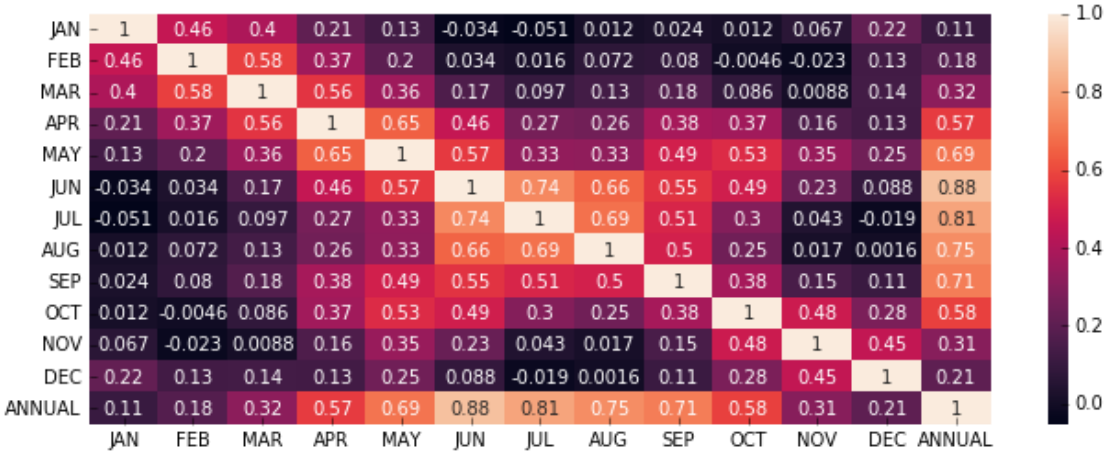
In [9]: data [['SUBDIVISION', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL','AUG', 'SEP', 'OCT', 'NOV', 'DEC']].groupby("SUBDIVISION").mean().plot.barh(stacked=True




## 1.10 Observations

- Above graph shows that the amount of rainfall is reasonably good in the months of march, April, May in eastern India.

In [12]: plt.figure(figsize=(11,4)) sns.heatmap(data[['JAN','FEB','MAR','APR','MAY','JUN','JUL','AUG','SEP','OCT','NOV','DEC','ANNU
plt.show()



## 1.11 Observations

- **Heat Map** shows the co-relation(dependency) between the amounts of rainfall over months.
- From above it is clear that if amount of rainfall is high in the months of July, August, September then the amount of rainfall will be high annually.
- It is also observed that if amount of rainfall in good in the months of October, November, December then the rainfall is going to be good in the overall year.

## 1.12 Predictions

- For prediction we formatted data in the way, given the rainfall in the last three months we try to predict the rainfall in the next consecutive month.
- For all the experiments we used 80:20 training and test ratio.
  - Linear regression
  - SVR
  - Artificial neural nets
- Testing metrics: We used Mean absolute error to train the models.
- We also shown the amount of rainfall actually and predicted with the histogram plots.
- We did two types of trainings once training on complete dataset and other with training with only telangana data
- All means are standard deviation observations are written, first one represents ground truth, second one represents predictions.

```
In [14]: # seperation of training and testing data from sklearn.model_selection
         import train_test_split from sklearn.metrics import
         mean_absolute_error
         division_data = np.asarray(data[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC']])

         X = None; y = None
         for i in range(division_data.shape[1]-3):
             if X is None:
                 X = division_data[:, i:i+3] y =
                 division_data[:, i+3]
             else:
                 X = np.concatenate((X, division_data[:, i:i+3]), axis=0) y =
                 np.concatenate((y, division_data[:, i+3]), axis=0)

         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)

In [15]: #test 2010 temp = data[['SUBDIVISION','JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP',
         'OCT', 'NOV', 'DEC']].loc[data['YEAR'] == 2010]

         data_2010 = np.asarray(temp[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
                 'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[temp['SUBDIVISION'] == 'TELANGANA'])

         X_year_2010 = None; y_year_2010 = None for i in
         range(data_2010.shape[1]-3):
             if X_year_2010 is None:
                 X_year_2010 = data_2010[:, i:i+3] y_year_2010 =
                 data_2010[:, i+3]
             else:
```

```
                X_year_2010 = np.concatenate((X_year_2010, data_2010[:, i:i+3]), axis=0) y_year_2010 =
                np.concatenate((y_year_2010, data_2010[:, i+3]), axis=0)

In [16]: #test 2005 temp = data[['SUBDIVISION','JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP',
        'OCT', 'NOV', 'DEC']].loc[data['YEAR'] == 2005]

        data_2005 = np.asarray(temp[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
                'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[temp['SUBDIVISION'] == 'TELANGANA'])

        X_year_2005 = None; y_year_2005 = None for i in
        range(data_2005.shape[1]-3):
            if X_year_2005 is None:
                X_year_2005 = data_2005[:, i:i+3] y_year_2005 =
                data_2005[:, i+3]
            else:
                X_year_2005 = np.concatenate((X_year_2005, data_2005[:, i:i+3]), axis=0) y_year_2005 =
                np.concatenate((y_year_2005, data_2005[:, i+3]), axis=0)

In [17]: #terst 2015 temp = data[['SUBDIVISION','JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP',
        'OCT', 'NOV', 'DEC']].loc[data['YEAR'] == 2015]

        data_2015 = np.asarray(temp[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
                'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[temp['SUBDIVISION'] == 'TELANGANA'])

        X_year_2015 = None; y_year_2015 = None for i in
        range(data_2015.shape[1]-3): if X_year_2015 is
        None:
                X_year_2015 = data_2015[:, i:i+3] y_year_2015 =
                data_2015[:, i+3]
            else:
                X_year_2015 = np.concatenate((X_year_2015, data_2015[:, i:i+3]), axis=0) y_year_2015 =
                np.concatenate((y_year_2015, data_2015[:, i+3]), axis=0)

MEAN 2005
121.2111111111111 134.68699821349824
Standard deviation 2005
123.77066107608005 90.86310230416397
MEAN 2010
139.93333333333334 144.8050132651592
Standard deviation 2010
135.71320250194282 95.94931363601675
MEAN 2015
88.52222222222223 119.64752006738864
Standard deviation 2015
86.62446123324875 62.36355370163346
```

```
In [20]: from sklearn.svm import SVR # SVM model
         clf = SVR(gamma='auto', C=0.1, epsilon=0.2)
         clf.fit(X_train, y_train) y_pred = clf.predict(X_test)
         print mean_absolute_error(y_test, y_pred)
```

127.1600615632603

## 1.13 Prediction Observations

### 1.13.1 Training on complete dataset

| Algorithm | MAE |
|---|---|
| Linear Regression | 94.94821727619338 |
| SVR | 127.74073860203839 |
| Artificial neural nets | 85.2648713528865 |

### 1.13.2 Training on Kerala dataset

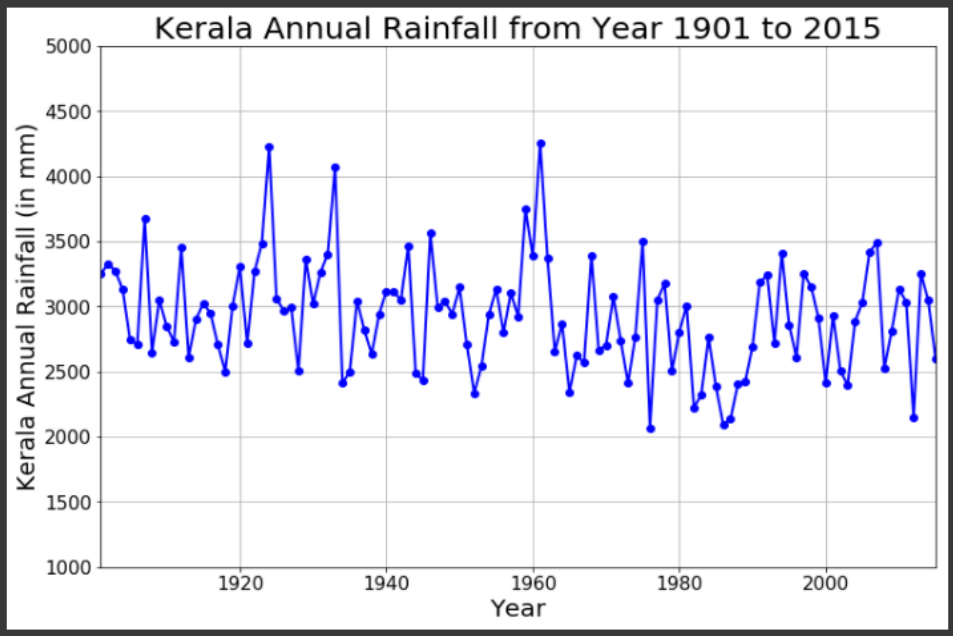
| Algorithm | MAE |
|---|---|
| Linear Regression | 70.61463829282977 |
| SVR | 90.30526775954294 |
| Artificial neural nets | 59.95190786532157 |

- Observed MAE is very high which indicates machine learning models won't work well for prediction of rainfall.
- Telangana data has a single pattern that can be learned by models, rather than learning different patterns of all states. so has high accuracy.
- Analysed individual year rainfall patterns for 2005, 2010, 2015.
- Approximately close means, noticed fewer standard deviations.

## 1.14 District wise details

- Similar to above the number of attributes is same, we don't have year in this.
- The amount of rainfall in mm for each district is added from 1950-2000.
- We analyse the data individually for the state **Kerala.**

```
ax=Kerala.groupby("YEAR").mean()['ANNUAL'].plot(ylim=(1000,5000),color='b',marker='o',linestyle='-',linewidth=2,figsize=(12,8));
#Kerala['MA10'] = Kerala.groupby('YEAR').mean()['ANNUAL'].rolling(10).mean()
#Kerala.MA10.plot(color='r',linewidth=4)
plt.xlabel('Year',fontsize=20)
plt.ylabel('Kerala Annual Rainfall (in mm)',fontsize=20)
plt.title('Kerala Annual Rainfall from Year 1901 to 2015',fontsize=25)
ax.tick_params(labelsize=15)
plt.grid()
plt.ioff()
```



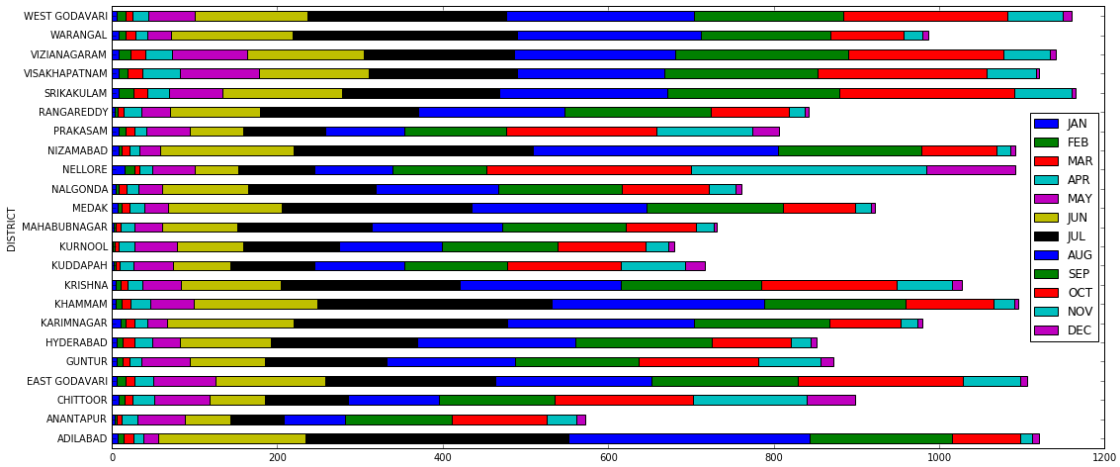

## 1.15 Observations

- The above graph shows the distribution of rain in Kerala.
- As there are large number of districts only 40 were shown in the graphs.

**Andhra Pradesh Data**

In [36]: ap_data = district[district['STATE_UT_NAME'] == 'ANDHRA PRADESH']

In [37]: ap_data[['DISTRICT', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL','AUG', 'SEP', 'OCT', 'NOV', 'DEC']].groupby("DISTRICT").mean()[:40].plot.barh(stacked=Tr

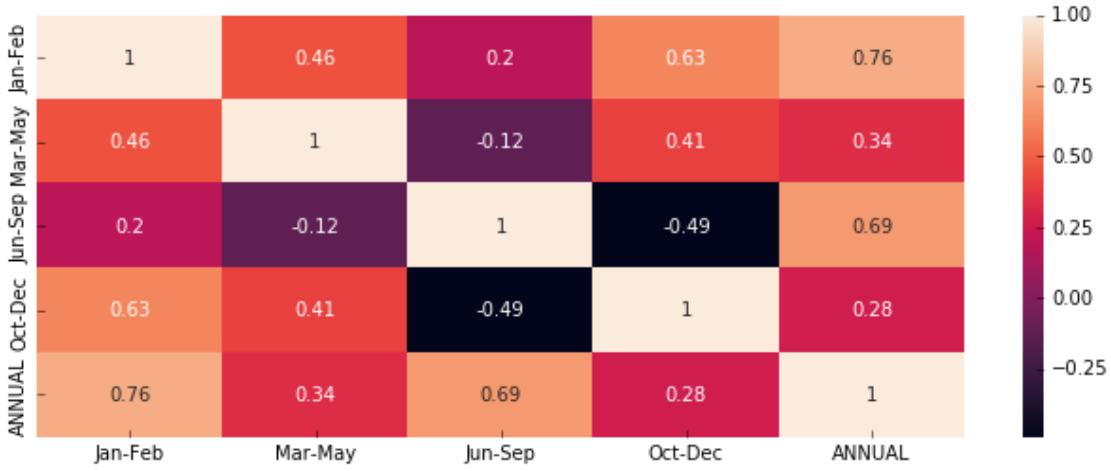


In [38]: ap_data[['DISTRICT', 'Jan-Feb', 'Mar-May','Jun-Sep', 'Oct
Dec']].groupby("DISTRICT").sum()[:40].plot.barh(stacked=True,figsize=(16,8)

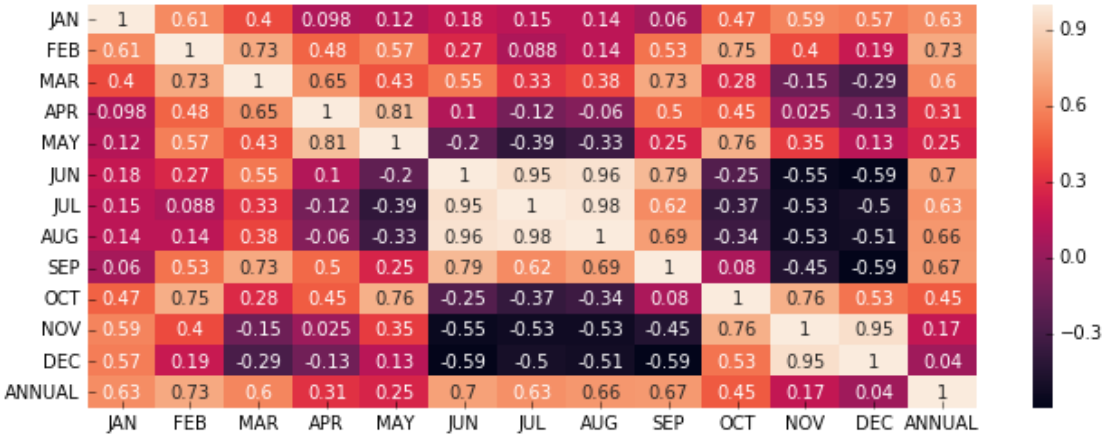

## 1.16 Observations

- The above two graphs show the distribution of over each district in **Andhra Pradesh**.
- The above graphs show that more amount of rainfall is found in Srikakulam district, least amount of rainfall is found in Anantapur district.
- It also shows that almost all states have more amount of rainfall have more amount of rainfall in the months June, July, September.

In [39]: plt.figure(figsize=(11,4)) sns.heatmap(ap_data[['Jan-Feb','Mar-May','Jun-Sep','Oct-Dec','ANNUAL']].corr(),annot=True)
plt.show()




In [40]: plt.figure(figsize=(11,4)) sns.heatmap(ap_data[['JAN','FEB','MAR','APR','MAY','JUN','JUL','AUG','SEP','OCT','NOV','DEC','
plt.show()




## 1.17 Conclusions

- Various visualizations of data are observed which helps in implementing the approaches for prediction.
- Prediction of amount of rainfall for both the types of dataset.
- Observations indicates machine learning models won't work well for prediction of rainfall due to fluctutaions in rainfall.

## 1.18 Technologies

- Programming language : *Python*
- Libraries : *numpy, pandas, matplotlib, seaborn, scipy, sklearn*