

# Rainfall Analysis Prediction

October 9, 2018

## India Rainfall Analysis

```
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
from subprocess import check_output
# print(check_output(["ls", "../input"]).decode("utf8"))
```

```
In [2]: data = pd.read_csv("../data/rainfall_in_india_1901-2015.csv", sep=",")
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            4112 non-null float64
FEB            4113 non-null float64
MAR            4110 non-null float64
APR            4112 non-null float64
MAY            4113 non-null float64
JUN            4111 non-null float64
JUL            4109 non-null float64
AUG            4112 non-null float64
SEP            4110 non-null float64
OCT            4109 non-null float64
NOV            4105 non-null float64
DEC            4106 non-null float64
ANNUAL         4090 non-null float64
Jan-Feb        4110 non-null float64
Mar-May        4107 non-null float64
Jun-Sep        4106 non-null float64
Oct-Dec        4103 non-null float64
dtypes: float64(17), int64(1), object(1)
memory usage: 611.0+ KB
```

```
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  \
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

```

```

In [4]: data.describe()

```

```

Out [4]:
          YEAR  JAN  FEB  MAR  APR  \
count  4116.000000  4112.000000  4113.000000  4110.000000  4112.000000
mean    1958.218659   18.957320   21.805325   27.359197   43.127432
std       33.140898   33.585371   35.909488   46.959424   67.831168
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    7.800000   15.700000
75%     1987.000000   22.200000   26.800000   31.300000   49.950000
max     2015.000000  583.700000  403.500000  605.600000  595.100000

          MAY  JUN  JUL  AUG  SEP  \
count  4113.000000  4111.000000  4109.000000  4112.000000  4110.000000
mean     85.745417   230.234444   347.214334   290.263497   197.361922
std    123.234904   234.710758   269.539667   188.770477   135.408345
min       0.000000    0.400000    0.000000    0.000000    0.100000
25%       8.600000   70.350000   175.600000   155.975000   100.525000
50%      36.600000  138.700000   284.800000   259.400000   173.900000
75%      97.200000  305.150000   418.400000   377.800000   265.800000
max    1168.600000  1609.900000  2362.800000  1664.600000  1222.000000

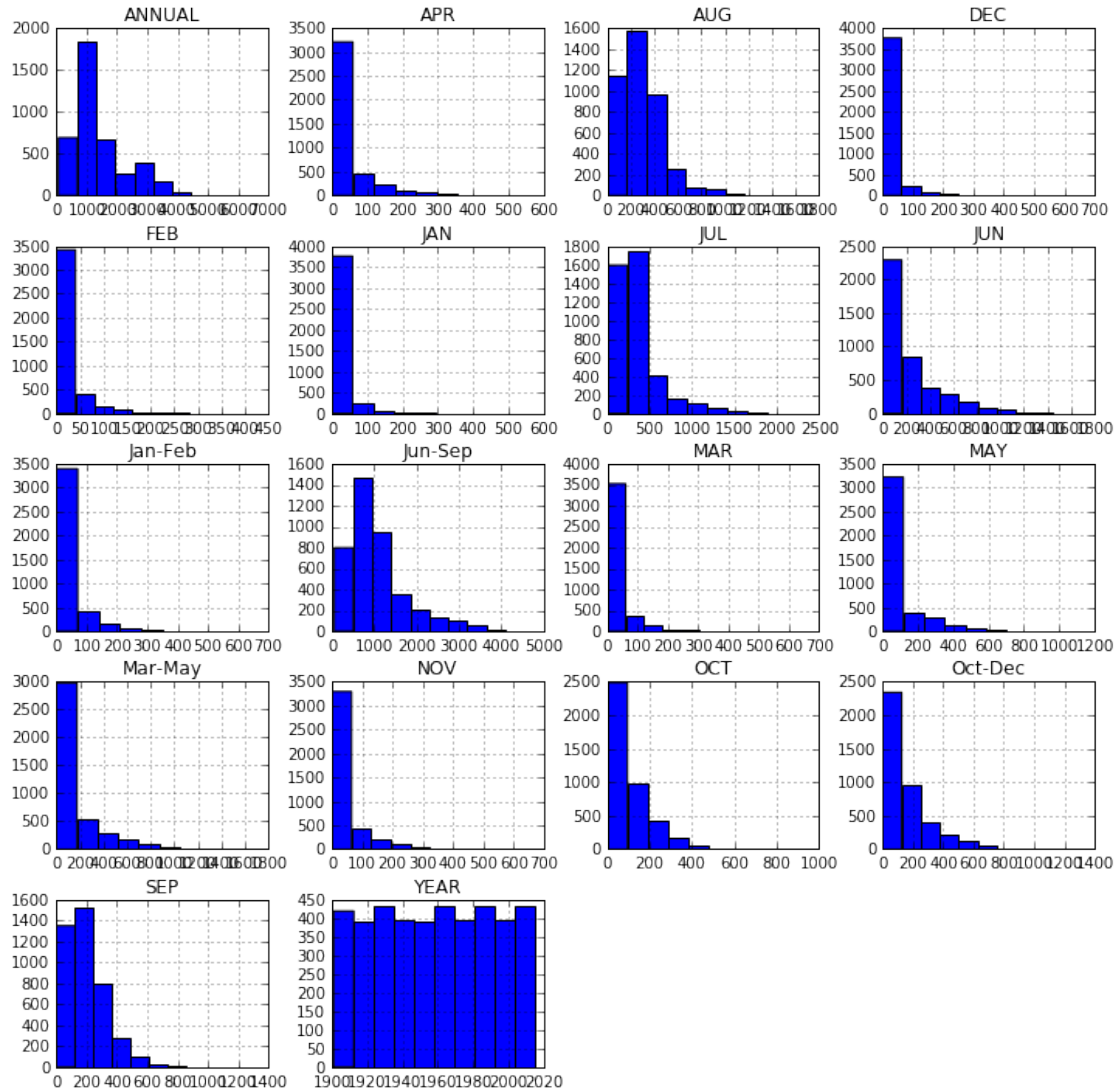
          OCT  NOV  DEC  ANNUAL  Jan-Feb  \
count  4109.000000  4105.000000  4106.000000  4090.000000  4110.000000
mean     95.507009   39.866163   18.870580  1411.008900   40.747786
std    99.519134   68.685410   42.369611   903.846565   59.308277
min       0.000000    0.000000    0.000000   62.300000    0.000000

```

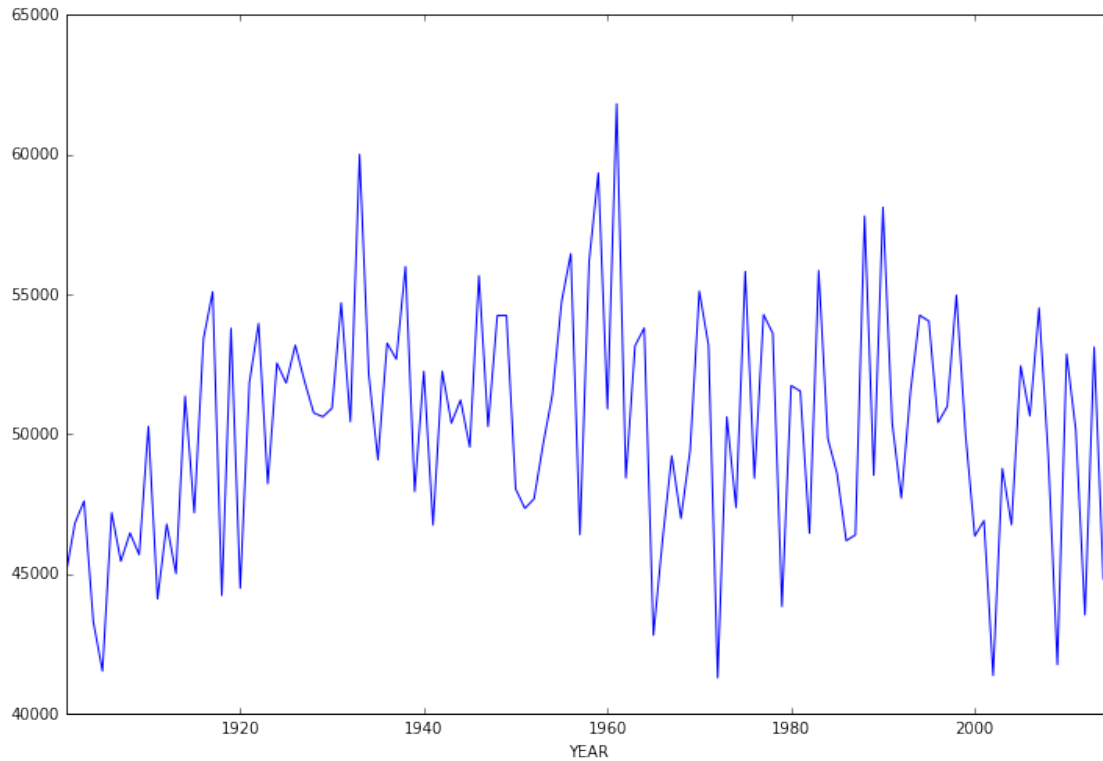
25%	14.600000	0.700000	0.100000	804.500000	4.100000
50%	65.200000	9.500000	3.000000	1121.300000	19.200000
75%	148.400000	46.100000	17.500000	1644.775000	50.375000
max	948.300000	648.900000	617.500000	6331.100000	699.500000

	Mar-May	Jun-Sep	Oct-Dec
count	4107.000000	4106.000000	4103.000000
mean	155.901753	1064.724769	154.100487
std	201.316965	707.741531	166.942660
min	0.000000	57.400000	0.000000
25%	24.050000	573.850000	34.200000
50%	74.800000	881.100000	98.200000
75%	196.950000	1288.175000	213.500000
max	1745.800000	4536.900000	1252.500000

```
In [5]: data.hist(figsize=(12,12));
```



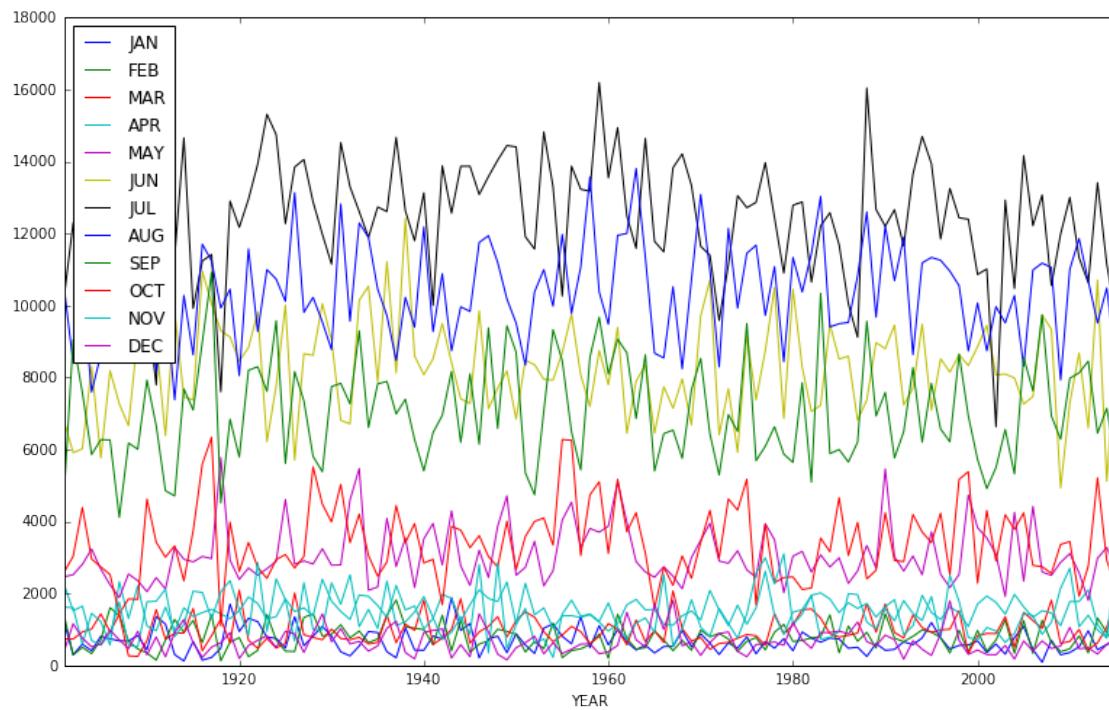
```
In [6]: data.groupby("YEAR").sum()['ANNUAL'].plot(figsize=(12,8));
```



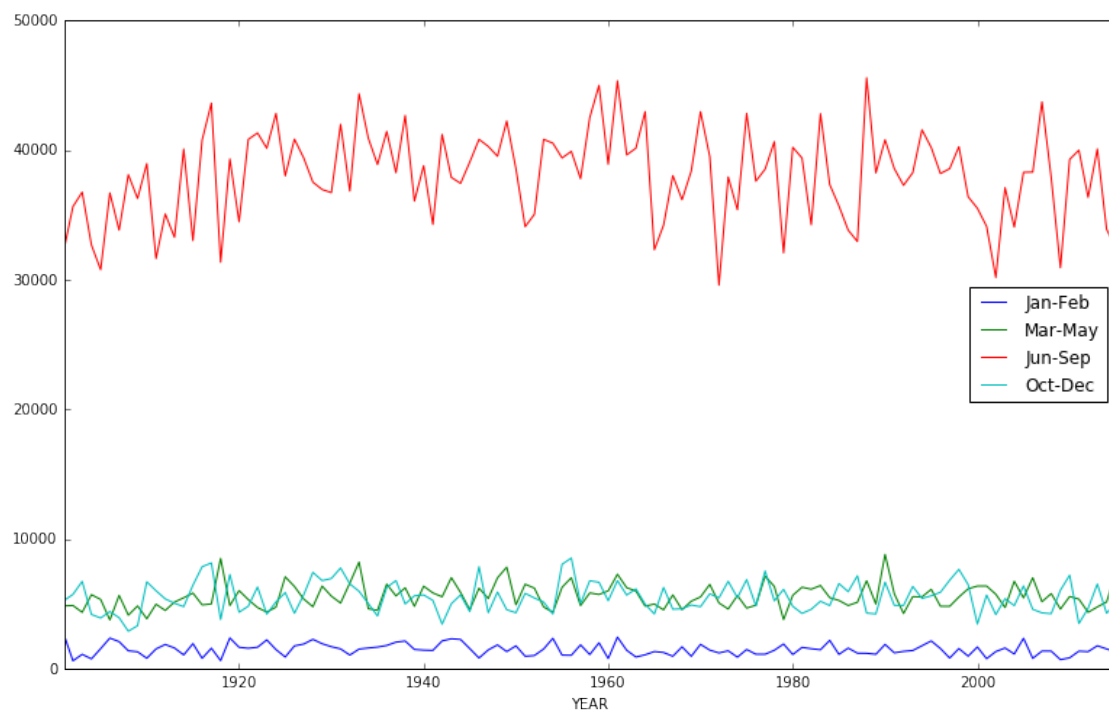
```
In [7]: data.columns
```

```
Out[7]: Index([u'SUBDIVISION', u'YEAR', u'JAN', u'FEB', u'MAR', u'APR', u'MAY', u'JUN',
              u'JUL', u'AUG', u'SEP', u'OCT', u'NOV', u'DEC', u'ANNUAL', u'Jan-Feb',
              u'Mar-May', u'Jun-Sep', u'Oct-Dec'],
              dtype='object')
```

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

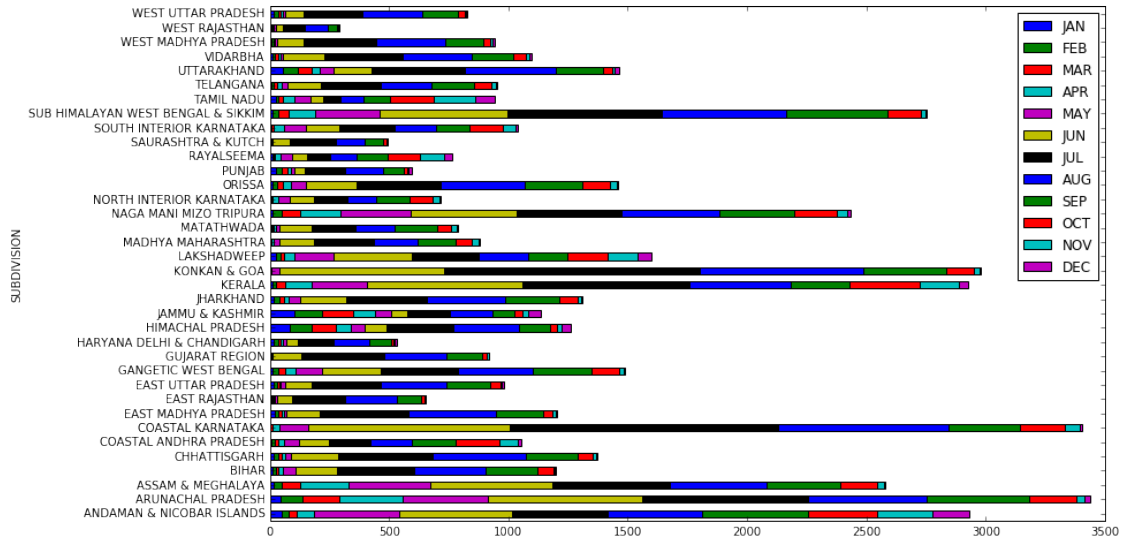


```
In [9]: data[['YEAR', 'Jan-Feb', 'Mar-May',
              'Jun-Sep', 'Oct-Dec']].groupby("YEAR").sum().plot(figsize=(13,8));
```



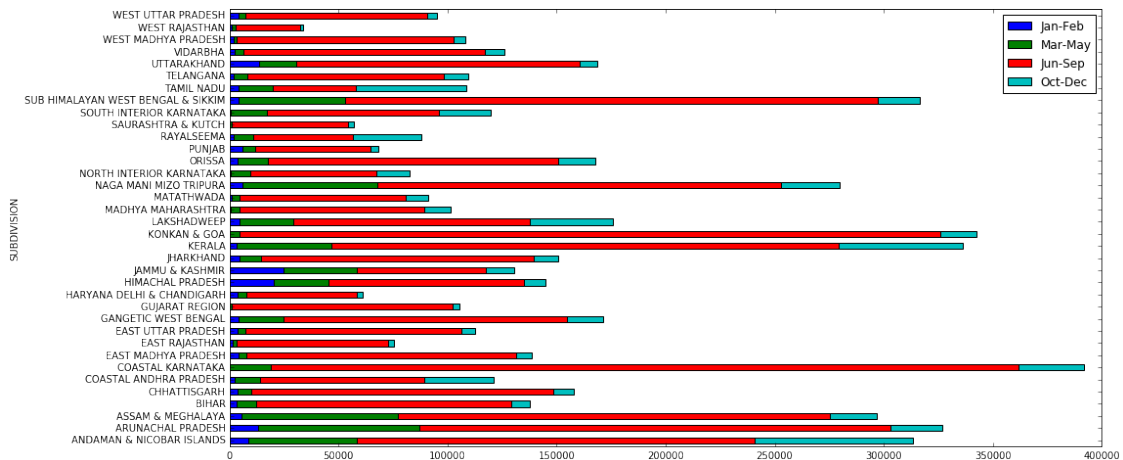
Span of Jun-Sep has maximum rainfall records.

```
In [10]: data[['SUBDIVISION', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
              'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].groupby("SUBDIVISION").mean().plot.barh(stacked=True)
```

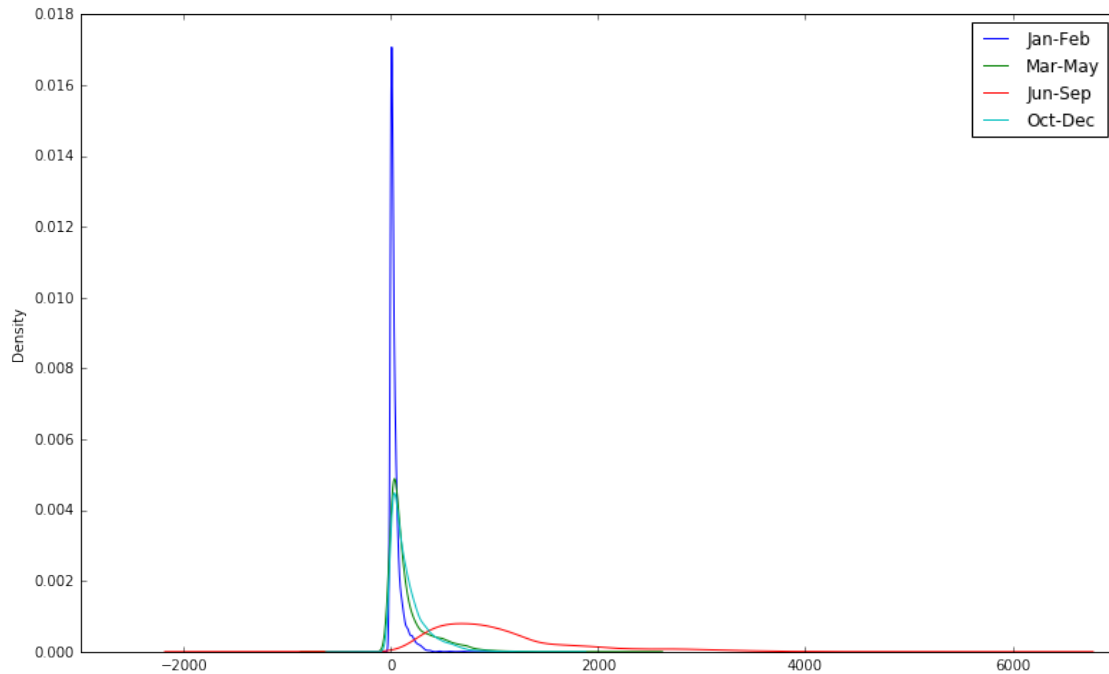


Graph shows top 3 subdivisions having high average rainfall: 1. ARUNACHAL PRADESH 2. COASTAL KARNATAKA 3. KONKAN & GOA

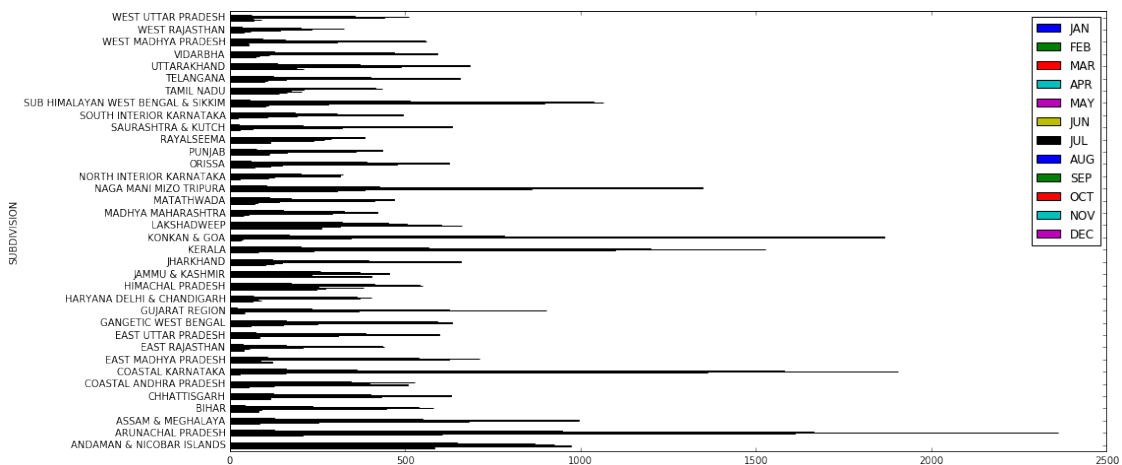
```
In [11]: data[['SUBDIVISION', 'Jan-Feb', 'Mar-May',
              'Jun-Sep', 'Oct-Dec']].groupby("SUBDIVISION").sum().plot.barh(stacked=True,figsize=(13,8))
```



```
In [12]: data[['Jan-Feb', 'Mar-May',
              'Jun-Sep', 'Oct-Dec']].plot(kind="kde",figsize=(13,8));
```



```
In [13]: data[['SUBDIVISION', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
               'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].groupby("SUBDIVISION").max().plot.barh(figsize=(15, 10))
```



Graph shows Max rainfall for specific month registered in ARUNACHAL PRADESH.  
District wise details

```
In [14]: district = pd.read_csv("../data/district_wise_rainfall_normal.csv", sep=",")
         district.info()
```

```

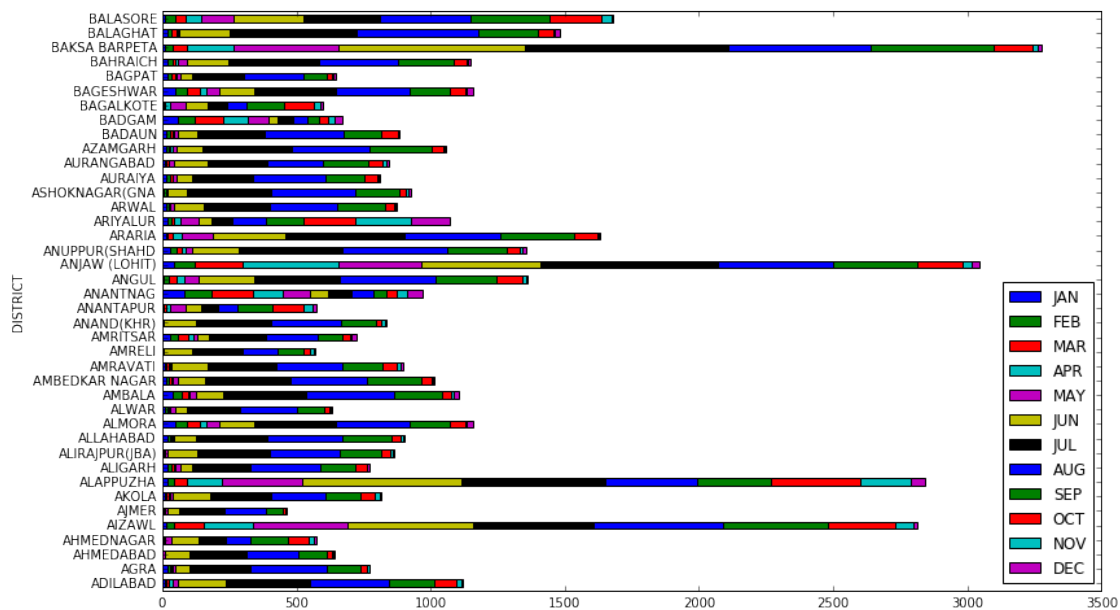
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 641 entries, 0 to 640
Data columns (total 19 columns):
STATE_UT_NAME    641 non-null object
DISTRICT         641 non-null object
JAN              641 non-null float64
FEB              641 non-null float64
MAR              641 non-null float64
APR              641 non-null float64
MAY              641 non-null float64
JUN              641 non-null float64
JUL              641 non-null float64
AUG              641 non-null float64
SEP              641 non-null float64
OCT              641 non-null float64
NOV              641 non-null float64
DEC              641 non-null float64
ANNUAL           641 non-null float64
Jan-Feb          641 non-null float64
Mar-May          641 non-null float64
Jun-Sep          641 non-null float64
Oct-Dec          641 non-null float64
dtypes: float64(17), object(2)
memory usage: 95.2+ KB

```

```

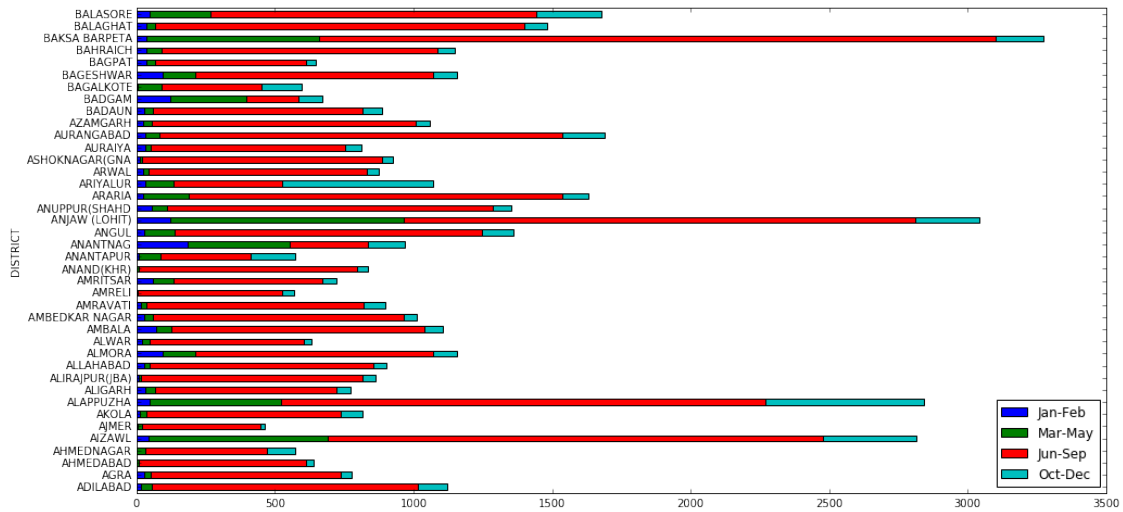
In [15]: district[['DISTRICT', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].groupby("DISTRICT").mean()[ :40].plot.barh(st

```





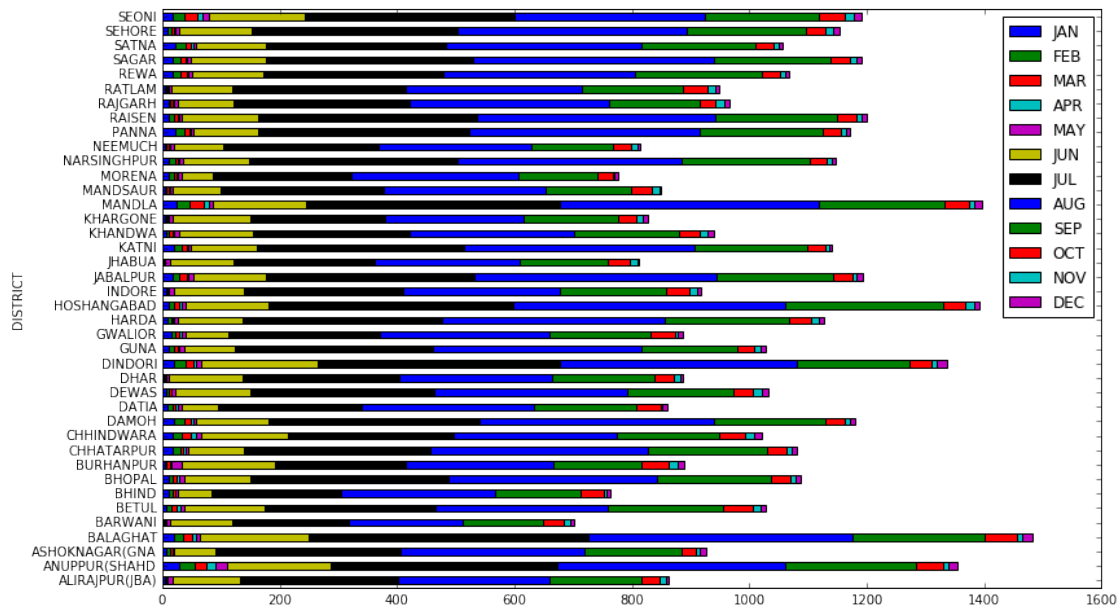
```
In [16]: district[['DISTRICT', 'Jan-Feb', 'Mar-May',
                  'Jun-Sep', 'Oct-Dec']].groupby("DISTRICT").sum()[:40].plot.barh(stacked=True,fig
```



## Madhya Pradesh Data

```
In [17]: mp_data = district[district['STATE_UT_NAME'] == 'MADHYA PRADESH']
```

```
In [18]: mp_data[['DISTRICT', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
                  'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].groupby("DISTRICT").mean()[:40].plot.barh(st
```



```
In [19]: mp_data[['DISTRICT', 'Jan-Feb', 'Mar-May',
                'Jun-Sep', 'Oct-Dec']].groupby("DISTRICT").sum()[ :40].plot.barh(stacked=True,fig
```

