Week 5: Scipy and some miscellany

M P Gururajan, Hina Gokhale and N N Viswanathan

Department of Metallurgical Engineering & Materials Science
Indian Institute of Technology Bombay

September, 2024

Outline

Scipy

- Built on numpy
- Powerful: optimization, signal processing, interpolation, linear algebra, symbolic computation, statistics, FFT, ...
- How to import and work with scipy?

scipy.stats

- `scipy.stats`: statistics module
- More than 80 continuous and 10 discrete random variables
- Consider, for example, Rayleigh distribution
- $f_X(x;\sigma) = \frac{x}{\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right)$ for $x \geq 0$
- Generate random variates from Rayliegh distribution

```
from scipy import stats

x = stats.rayleigh.rvs(0,1,10)
print(x)
```

- `rvs(loc,scale,number of random variates)`

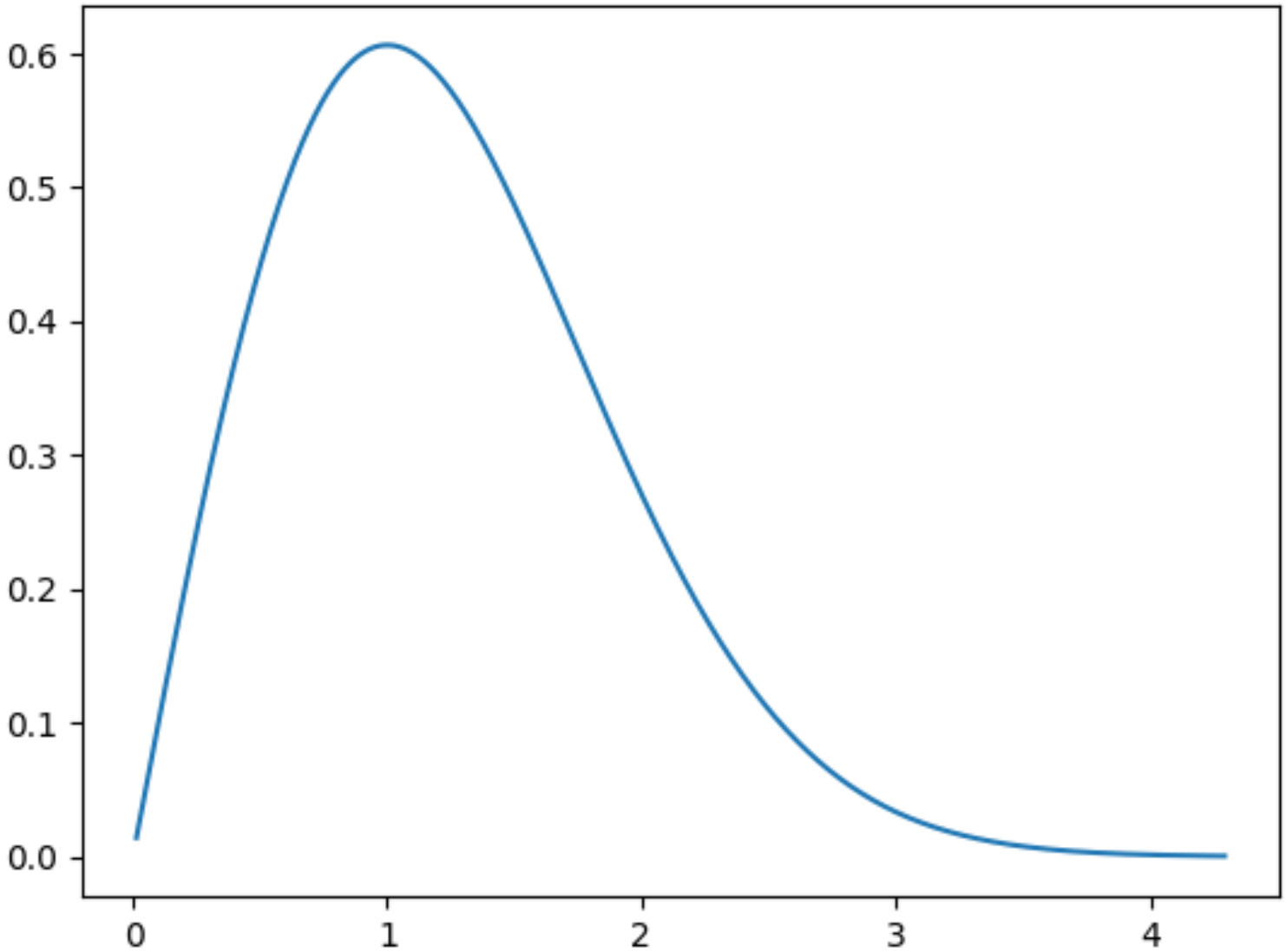
PDF of Raylrigh distribution

- Let us plot the PDF of Rayliegh distribution

```
from scipy import stats
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(stats.rayleigh.ppf(0.001),\
stats.rayleigh.ppf(0.999),1000)
print(x)
plt.plot(x,stats.rayleigh.pdf(x))
plt.show()
```

- ppf: to calculate percent point function
- ppf: inverse of cdf

PDF plot

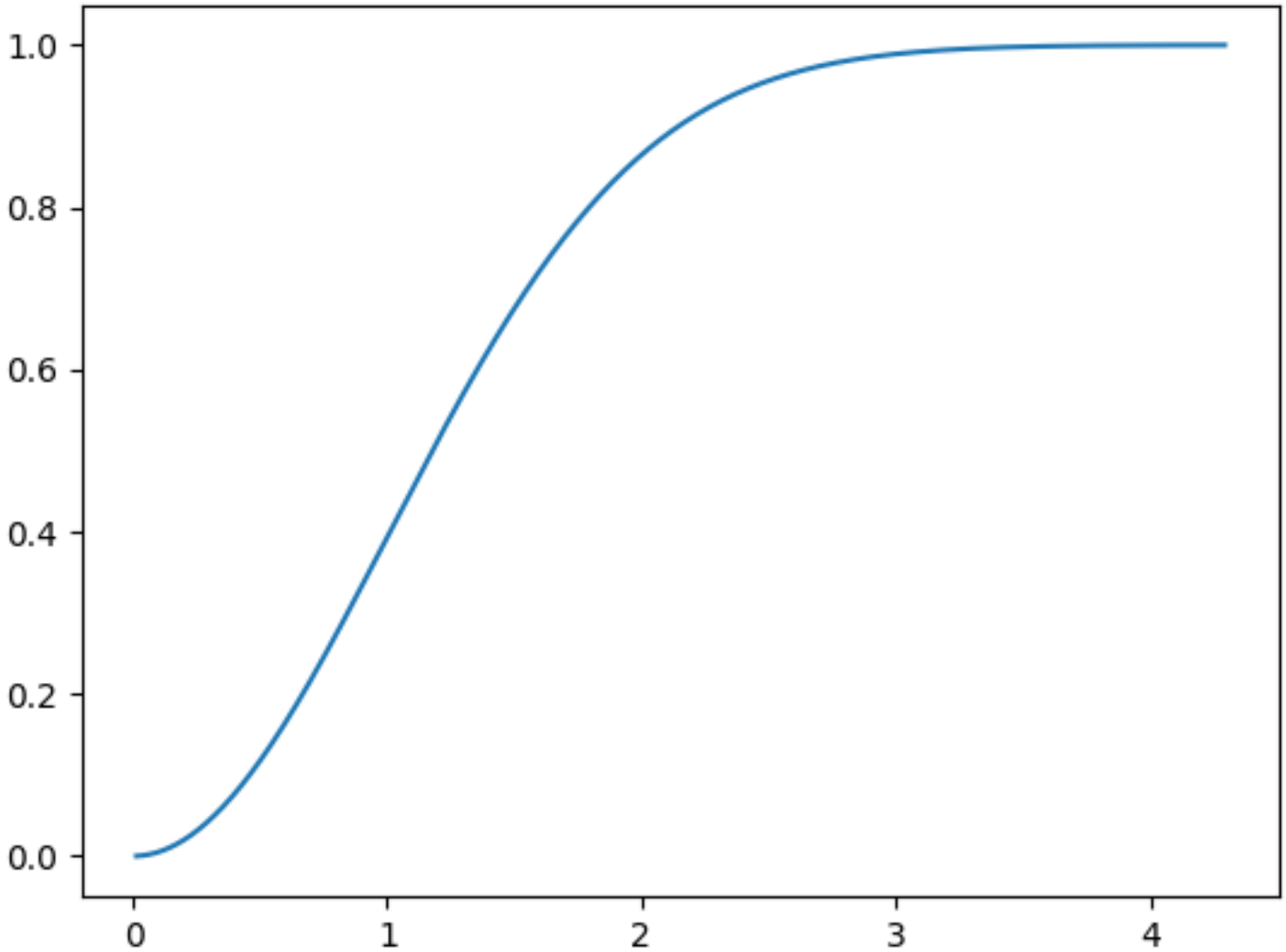


CDF of Rayleigh distribution

- Let us evaluate the CDF of Rayleigh distribution

```
from scipy import stats
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(stats.rayleigh.ppf(0.0001),\
        stats.rayleigh.ppf(0.9999),10000)
print(x)
plt.plot(x,stats.rayleigh.cdf(x))
plt.show()
```

CDF plot



CDF of Rayleigh distribution

- $f_X(x;\sigma) = \frac{x}{\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right)$ for $x \geq 0$
- CDF: integrate this function from 0 to x to obtain $F_X(x)$ – using `scipy`

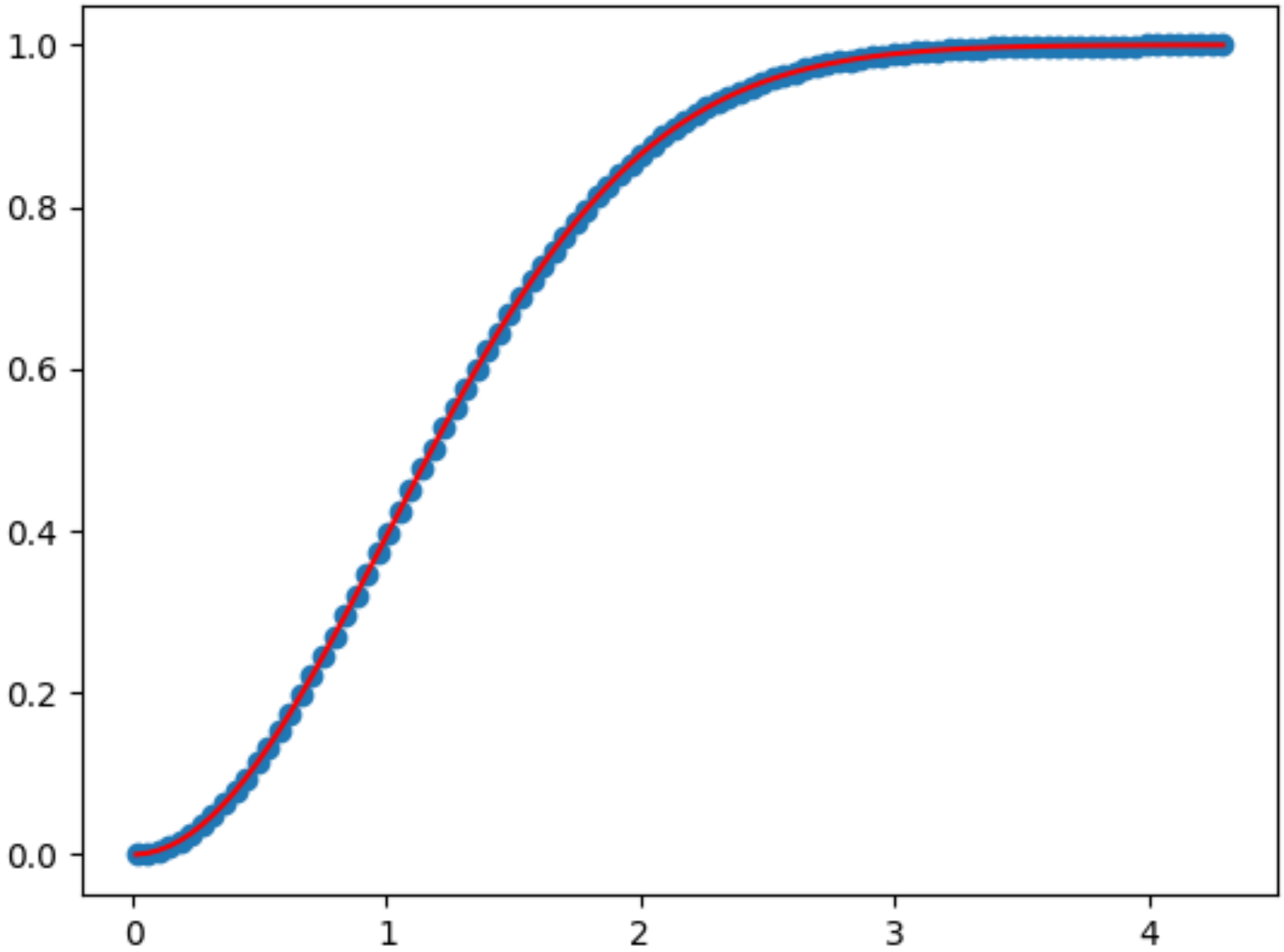
```
from sympy import *
x = Symbol('x')
sigma = Symbol('sigma')
integrate( (x/sigma**2) * exp(-0.5*x**2/sigma**2), \
(x, 0, x))
```

- $F_X(x) = 1 - \exp\left(-\frac{x^2}{2\sigma^2}\right)$

Analytical versus numerical

```
from scipy import stats
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(stats.rayleigh.ppf(0.0001),\
        stats.rayleigh.ppf(0.9999),100)
print(x)
plt.scatter(x,stats.rayleigh.cdf(x))
plt.plot(x,1-np.exp(-0.5*x**2),'r')
plt.show()
```

CDF plot and comparison



What does this code do?

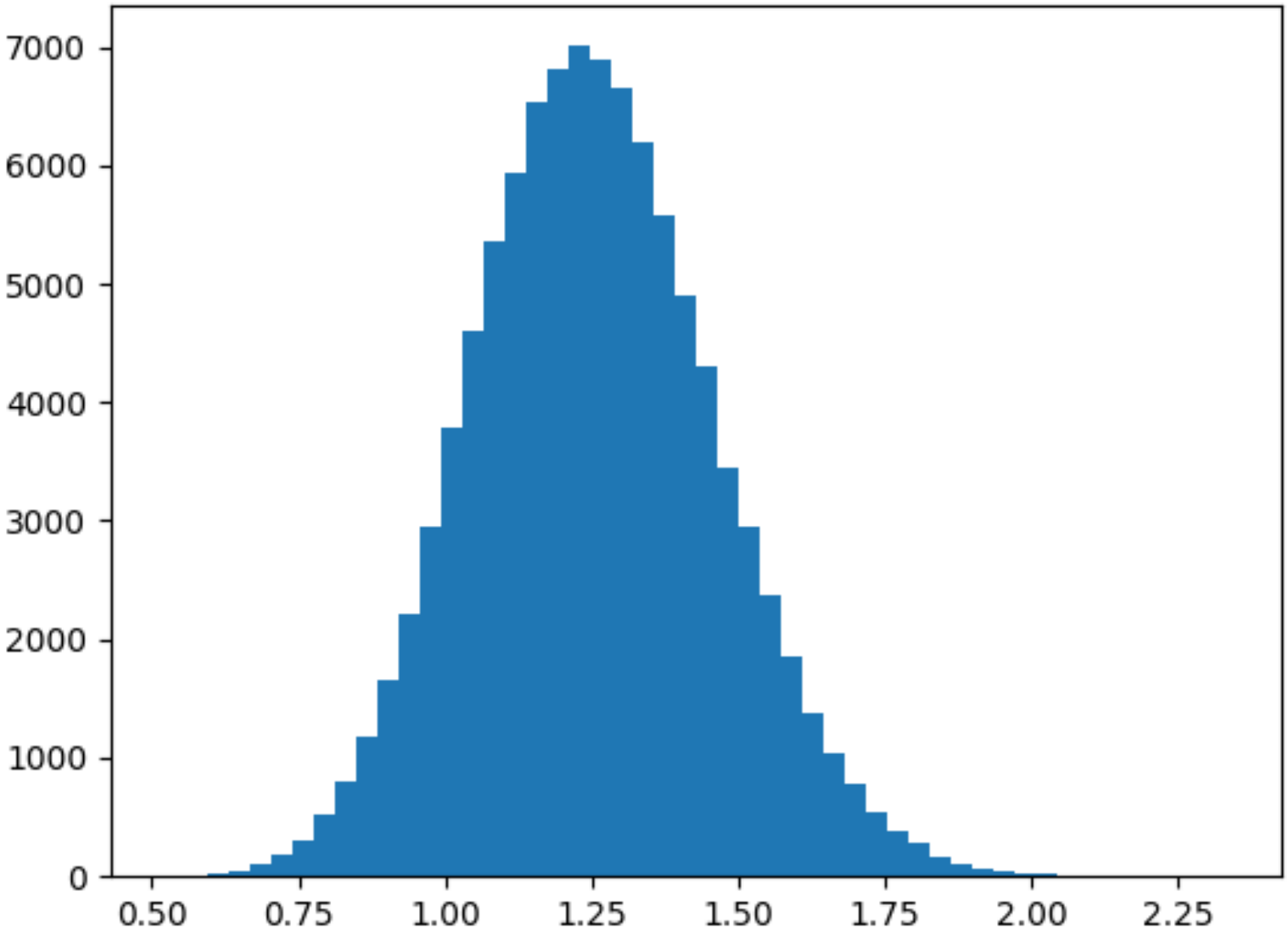
```
from scipy import stats
import matplotlib.pyplot as plt

N = 10

y = []
for i in range(N):
    x = stats.rayleigh.rvs(0,1,10)
    y.append(x.mean())

plt.hist(y,bins=5)
plt.show()
```

Effect of N

```
from scipy import stats
import matplotlib.pyplot as plt

N = 100

y = []
for i in range(N):
    x = stats.rayleigh.rvs(0,1,10)
    y.append(x.mean())

plt.hist(y,bins=5)
plt.show()
```

Keep increasing N and bins by a factor of 10. What happens?

N = 100000


7000
6000
5000
4000
3000
2000
1000
0
0.50
0.75
1.00
1.25
1.50
1.75
2.00
2.25

Home work

Use the above code for other distributions such as Weibull, cosine, hyperbolic secant, and Cauchy. Check what happens with large N!

Descriptive statistics

```
from scipy import stats
import matplotlib.pyplot as plt

N = 1000

y = []
for i in range(N):
    x = stats.rayleigh.rvs(0,1,10)
    y.append(x.mean())

print(stats.gmean(y))
print(stats.mode(y))
print(stats.gstd(y))
print(stats.skew(y))
print(stats.kurtosis(y))
print(stats.describe(y))
```

Miscellany

- Pandas dataframe: drop and apply functions

```
import pandas as pd
import matplotlib.pyplot as plt
Rainfall = pd.read_csv('SubDivisionWiseRainfall.csv')
RF = Rainfall.drop(Rainfall[Rainfall['YEAR'] \
== '1901-2015'].index)
RF['YEAR'] = RF['YEAR'].apply(int)
x = RF[(RF['SUBDIVISION']=='VIDARBHA')
        & (RF['YEAR']== 1982)
        & (RF['Parameter']=='Actual')]
print(x)
```

Miscellany

- Pandas dataframe: replace

```
import pandas as pd
import matplotlib.pyplot as plt
Rainfall = pd.read_csv('SubDivisionWiseRainfall.csv')
x = Rainfall[(Rainfall['SUBDIVISION']=='VIDARBHA')
        & (Rainfall['YEAR']== '1982')
        & (Rainfall['Parameter']=='Actual')]
Rainfall['SUBDIVISION']=\
Rainfall['SUBDIVISION'].replace('MATATHWADA','MARATHWADA')
y = Rainfall[(Rainfall['SUBDIVISION']=='MARATHWADA')
        & (Rainfall['YEAR']== '1982')
        & (Rainfall['Parameter']=='Actual')]
print(x)
print(y)
```

Miscellany

- Pandas dataframe: rename columns and dealing with NA

```
import pandas as pd
import matplotlib.pyplot as plt
Rainfall = pd.read_csv('SubDivisionWiseRainfall.csv')
Rainfall.rename(columns={'JF':'Winter',\
'MAM':'Summer','JJAS':'South West Monssoon',\
'OND':'North East Monsoon'}, inplace=True)
print(Rainfall.head())
```

- Dealing with NA: `df.dropna()` and `df.fillna()`

Thank You!

Questions, clarifications, comments?