Numpy — fast & convenient.

import numpy as np
l = range (1000)

l1 = range (SIZE)
l2 = range (size)
a1 = np.arrange (size)
a2 = np. arrange (size)
result = [(x+y) for x, y in zip (l1, l2)]

Advantage:-

1 - Convenient.
2 - Faster
3 - Less memory

a = np.array ([[1,2] , [3,4] , [5,6]])
a.ndim
↳ To print dimensions.

a.itemsize
↳ Byte size.

a = array([[1,2], [3, 4], [5,6]])
↳ array.size → 6.
a.shape = (3,2).

np.zeros ((3,4)) = 3x4 zero matrix.
np.ones ((3,4)) = 3x4 1 matrix.
np.arrange (1,5) → array ([1,2,3,4])
np.arrange (1,5,2) → array ([1,3]).
a.ravel () → Make 1-D.
a.min () , a.max (), a.sum (), a.sum (axis=0).
np. sqrt (a)
np.std (a) → standard deviation

Slicing, Indexing etc

n = [6,7,8]
n [0:2]
↳ [6,7]

a [0:2, 2]
0 -1 → 2nd element
([8,3]]

np.vstack ((a,b))
↳ one over other.

np.hsplit (a,3) → split into 3 array.

Iterate numpy array using nditer :-

```
for row in a :
    for cell in row.
        print cell.
```

→

0
1
2
3
1
10

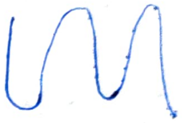
```
for cell in a.flatten() :
    print(cell)
```

→

0
1
2
1
10

```
Also, for x in np.nditer(a, order = 'F') :
          print(x) .
```

C order:

Fortran order :
(F)


Pandas

- Pandas is a python module that makes data science extremely easy & effective.
- Process of cleaning messy data is called data munging or data wrangling.

```
→   import pandas as pd
    df = pd.read_csv ( —)
    df.
```

Different ways of creating df :-

1 - df = pd.read_csv(—) or 2-pd.read_excel (—)

3- df = pd.DataFrame (ds, columns = [— —]).
    df   ↳ list

4 - ↳ But instead of list -Dict-

Readiy / Writiy in Excel or CSV

import pandas as pd
df = pd.read_csv(—)
df

pd.read_csv(—, skip_rows=1)
df.

pd.read_csv(—, header=None,)
↳ 0-header column.

, na_values = ("not available", "n.a."])

, header = False → Header deleted.

Read_Excel

import pandas as pd
df = pd.read_excel(—, "sheet1")
df.

Similarly we use excelwriter to import excel file etc.

Handliy missiy data

1) fillna() → df.fillna(0) → Fill null with 0.

df.fillna({
'temp': 0,
'wind': 0,
'event': no })

~~bfill~~
df.fillna(method = "bfill")
↳ Fill prev value either horizontaly or vertically.

df.interpolate()
↳ Give intepolarate value to blank/null.

df.dropna() → Drop null value...

df.dropna(how="all") → whose entire row is blank.

Replace function

df = df.replace (-999, np.NaN)

Regex

→ Regular expression.

df.replace ({'[A-Za-z]', '', regex= True)
↳ Remove alphabets.

Group By

g → df.groupby ('col')
g.max(), g.mean(), g.describe → Give all details.

Concat in Pandas

→ df = pd.concat([ind_wea, us_wea], ignore_index= True
↳ Continuous index.

, key=[ind, us], axis = 0 or 1

~~→ S=pd.series (name = "event")~~

→ s = pd. Series ([" _", "_", "_"], name="event")

df.concat ([temp_df], s], axis=1).

Merge in Python

→ df= pd.merge(df1, df2, on = "city", how="outer")

→ ✓

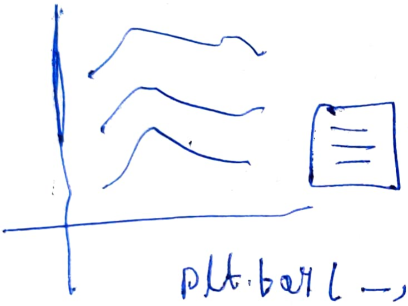
Pivot ~~&~~ Pivot_table

→ df.pivot(index='humidity', columns="city")
,aggfunc="sum"

Also margins use.

Matplotlib

```
import matplotlib.pyplot as plt
    x = [1,2, ___ ,7]
    y = [50,51,52, - - ,46]
 plt.plot[x,y, color = "green", linewidth=5 , linestyle="dotted")
  → plt.~~label~~ ('Day')
  plt.ylabel ('Temp')
```



→ ~~for~~ plt.plot (Loc = "best")
,shadow = "True"

```
plt.barl(—, ) ←for bar chart.
plt.bar(—, label = "Revenue")
plt.bar(—, label = "Profit")
```

Histograms

```
blood_sugar = [100, 110, 102, ___]
 plt.hist(blood sugar, bin=3, rwidth=0.95
        , label=[men, women], orientation='horizontal')
```

Pie Chart

```
exp_val = [1400, 600, 800, 700]
exp_labels = ["Home", "Wash", "Bike"—]

plt.axis ("equal")
plt.pie(exp_values, labels = exp_labels, radius = 1.5,
        autopct =   — )
```