• Numpy

1) a = np.array([1,2,5]) → list to array

2) a = np.array([[1,2], → 2D
[3,4]])

3) a = np.zeros((3,4)) → all 0

4) a = np.ones((3,4)) → all 1

5) a = np.ones((3,4))*8 → all 8

a = np.zeros((3,4))+8.0 → float

a = np.full((3,4),7) → all 7 → int

5) a = np.arange(0,10,3) → 0, 3, 6, 9

6) a = np.linspace(2,10,6) → equal space

→ $\frac{10-1}{6-1} = \frac{9}{5} = 1.8$

7) b.shape → shape

b.ndim → dimention

b.dtype → datatype

8) .reshape(3,5) → row 3, col 5

9) np.arange(1,13).reshape(3,4)

1 2 3 4
5 6 7 8
9 10 11 12

10) X[0] → 1st, X[-1] → last

11) X[:3] → 0 to 2

12) ~~X[r,c]~~ X[r,c]

13) X[rows, cols]

X(□:□, □:□) → sub 2D

15) v = a.copy() → copy

16) s = np.sqrt(v)

17) X = np.random.randn(5) → 5 sample.

18) X = np.random.randn(5,3) → row,col

19) mu = X.mean() → overall mean

= X.mean(axis=0) → col wise, row value → feature

= X.mean(axis=1) → row as col wise value. → sample

19)
- v = np.array([1,-1,2,3])
- mask = v>1 → T, F, F, T
- v[mask] → true value
  v[mask == False] → False "
- v[v>1] → True value (greater than)
- v[v<=1] → equal or less than value
- v[v>=4] → select upper 4
- v[v<4] = 0 → replace by 0
- np.where[v%2 == 0] → even value index

20) now → sample.
col → feature.

x.sum() → all
(axis=1) → now wise
(axis=0) → col wise

21) x.std() → standard deviation
x.min()
x.max()
x.mean()
x.var() → variance

22) standardization → score

$$= (X - \text{mean}) / (\text{std} + 1e^{-8})$$

23) covariance matrix
↳
cov = np.cov(X, rowvar = False) → feature-feature

cov = np.cov(X) → sample - sample relation

29) Transpose → X.T

25) np.concatenate([A,B], axis=1) → now array const

26) " " axis=0) → col "

27) np.vstack([A,B]) → vertical array const → new col
np.hstack([A,B]) → horizontal " " → row now

Common mistake :→

- matrix-vector multiplication = X @ w
- element wise mult = X * w

const
→ || → |
→ = → —

- 1,2,3
  ↳ reshape(1,-1)
- 1
  2 → reshape(-1,1)
  3

• Pandas :→

1) df=pd.read_csv('Toun.csv')
   -excel('Toun.xlsx')

2) ~~head()~~ df.head()

3) df.tail()

4) df.columns → all column

4) df.Temp
   df['Temp'] } see col

5) df[['A', 'B']] → mul col

⑥ df.Tem.max()
   .min()
   .mean()

| df['Temp'].max()
   .min()
   .mean()
   .std()

→ df.iloc[2]

⑦ df.loc[2] → 2 row

df.loc[1:5] → 1 to 5 row.

df.iloc[2:4] → 2,3 row.

2D :-

df.iloc[1:3, 2:4]

⑧ df.Events.unique() → col unique value see

⑨ df['EST'][df['Eve']=='Rain']
   ↳ query.

⑩ df[['EST', 'Temp']][df['Event']=='Rain'] ~~.~~

⑪ df[df['Event']=='Rain'] → Rain Related sob row,

⑫ df.isnull().sum() → null value ,

⑬ df.fillna(1, inplace=True) → নাল value 1 দিয়ে পূরণ করে

⑭ df.interpolate() → আশেপাশের value দিয়ে পূরণ করা

⑮ df.replace(1, np.nan, inplace=True) → 1 → 0 replace.

⑯ df.shape   ⑯' type(df['day']) → col type.

⑰ df['EST'][df['Events']==df['Event'].max()]

⑱ df.describe()   ⑲ df.describe(include='O')

⑲ df.index

→ df.loc['1/1/127':'2/2/0'] → string analysis

⑳ df.set_index('day', inplace=True) → day হবে index

㉑ df.reset_index(inplace=True)   ② df['Event'].dtype

Data frame create :→

(21) data = { — dic

'Nam' : ['A', 'B'],

'Roll' : [1, 2, 3]

}

df = pd.DataFrame(data)

tuple

data = [

('A', 1), (B, 2)

]

df = pd.DataFrame(data = 'data',

colamns = ['Nam', 'Roll'])

list of dic :-

data = [

{ 'Nam' : 'A', 'Roll' : 1 },

{ 'Nam' : 'B', 'Roll' : 2 }

]

(31) df.colamns = ['A', 'B', 'C'] → rename

↳ same no of col of previous

(32) loc → row থেকে কাজ

↳ index, → set index

↳ search

• pd.read_excel('Data', 'Sheet1')

• df = pd.read_csv('Data'.csv', skiprows = 1)

↳ প্রথম row টা বাদ

• df = pd..

, header = 2)

↳ row in 2 no.

• df-

, header = None, names = ['A', 'b', 'c'])

↳ header এমনি col নাম a, b, c

↳ আগের টা value.

•

, nrows = 3) → 1st 3 টা row

শুধু

• NaN করে → df = pd.read_c(,

na_values = [-1]

• na_values = {

'esp' : [-1],

'Name' : ['Tusan']

}

3 → প্রথম 3 টা বাদ.

```
def con_peo_name(cell):
    if cell == 'n.a.':
        return 'Sam'
    return cell
```

```
def con_peo_age(cell):
    if cell == 'n.a.':
        return 50
    return cell
```

```
df = pd.read_excel("data.xlsx", convetens = {
        'people' : con_peo_name,
        'age' : con_peo_age})
df
```

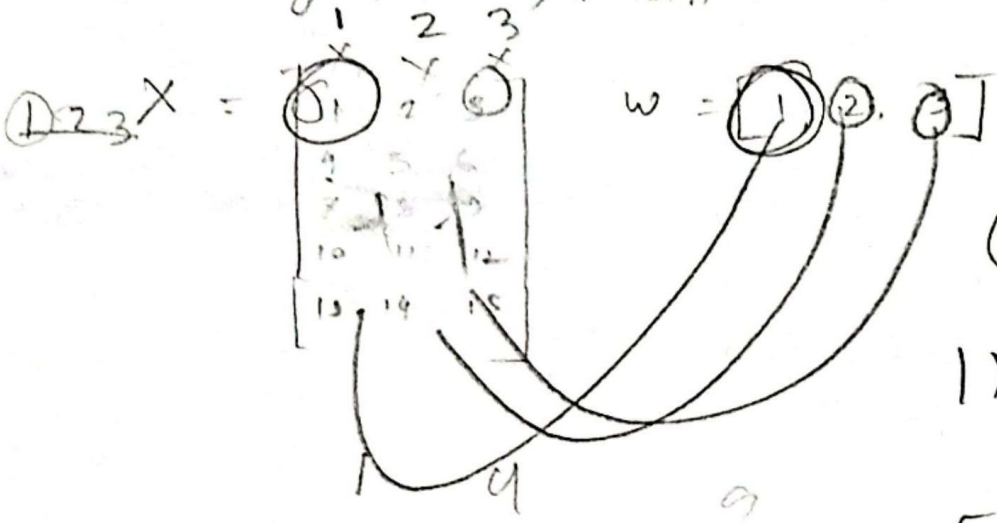
2 dataframe in two seperate excel -

```
data 1 = {}
data 2 = {}
```

```
with pd.ExcelWriter('OK.xlsx') as
        writer:
    data 1.to_excel(writer, sheet_name
        = 'she1')
    data 2.to_excel(writer, sheet_name
        = "she2")
```

np.zeros(,)

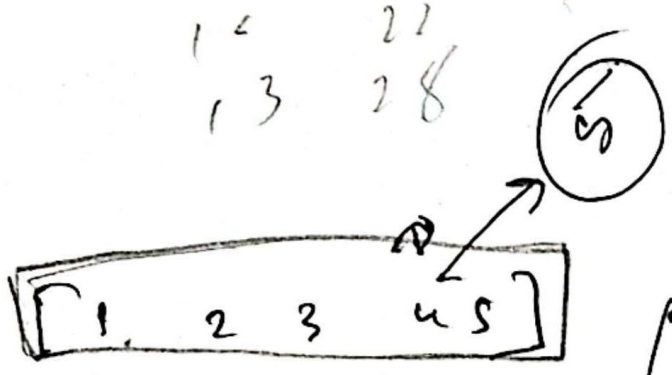
np.arange(0, 10, 2) # 0, 2, 4, 6, 8



(5) (1, -1)

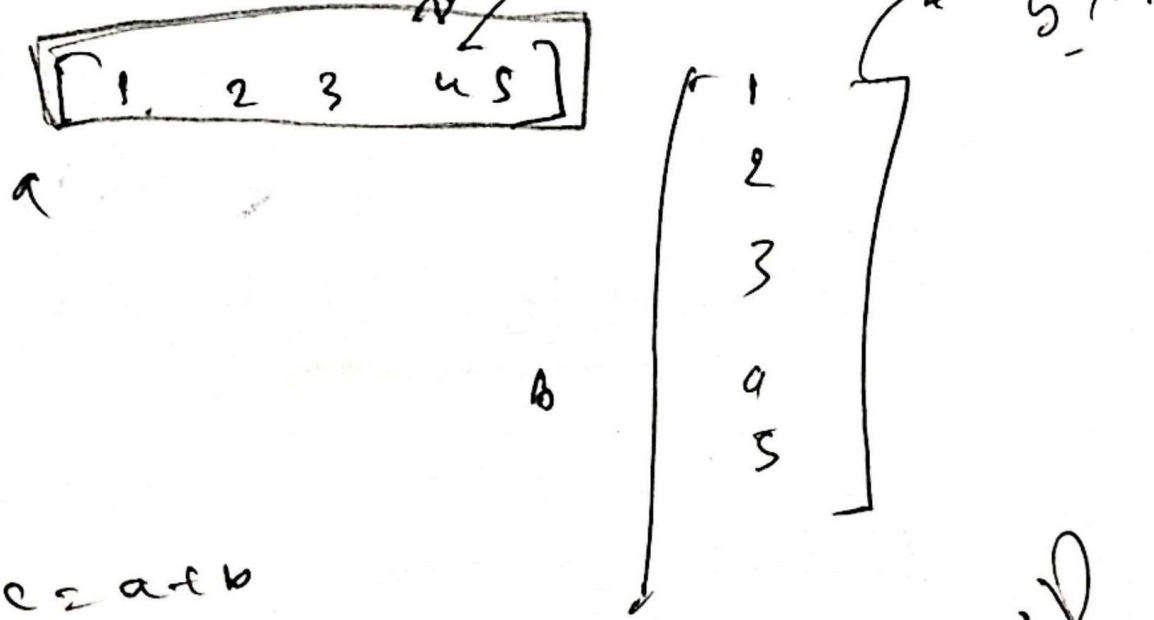
1 × 5 → 1 2 3 4 5

5 × 1 → 1 2 3 4 5



1 × 5

5 × 1



c = a + b

=> [1 4 6 8 10]

b

1, 2 3 .4 ,

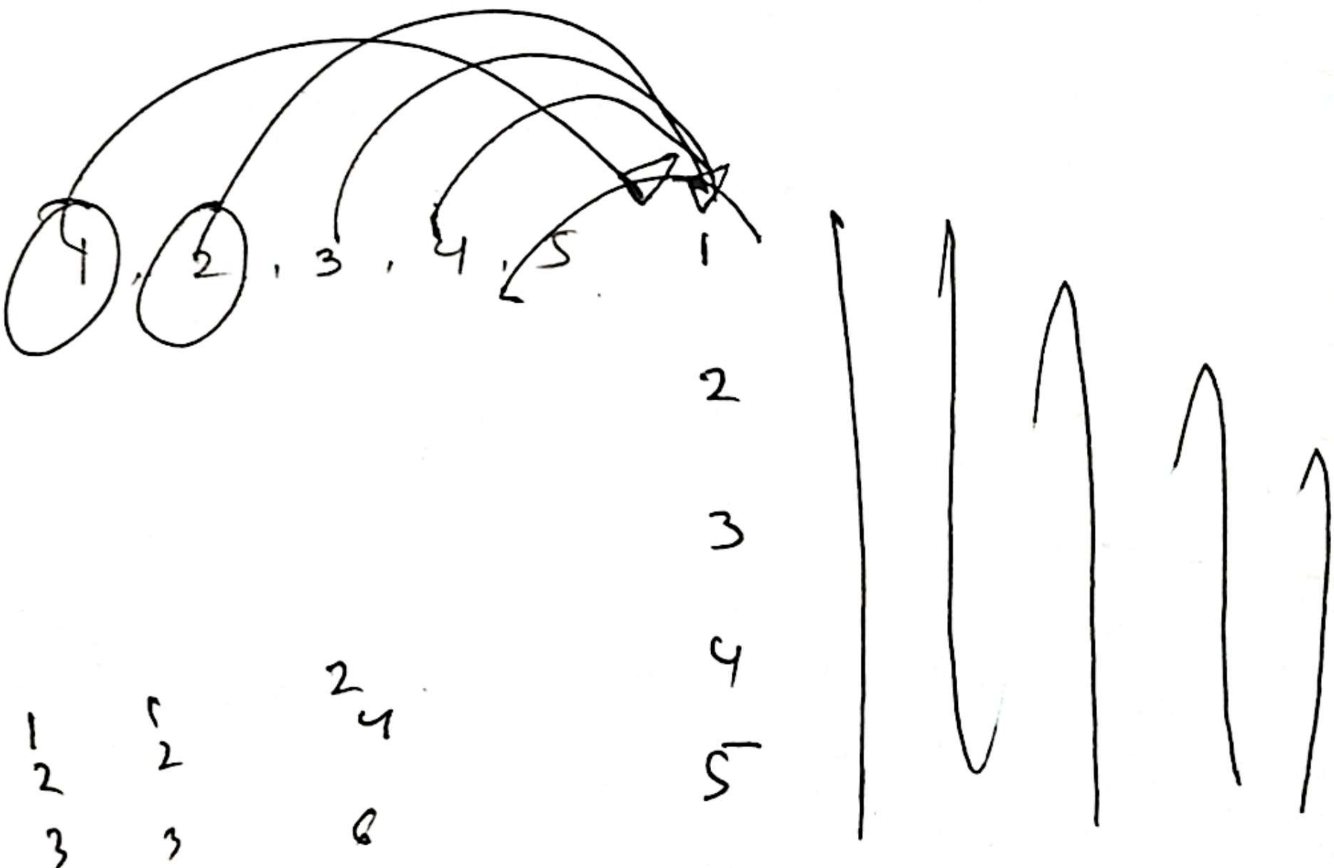
1

2 a

3 T

4

5 a = a.reshape(1, -1)



1	1	2
2	2	4
3	3	6
4	4	8
5	5	10