

Work in progress. This is draft...

### NUMPY: 2d Array

```
list1 = [  
    [111,141,171],  
    [211,241,271],  
    [311,341,371],  
    [411,441,471]  
]  
arr1 = np.array(list1)  
a34 = np.arange(1,13).reshape(3,4)
```

```
arr3 = arr1+arr2  
arr4 = arr1*arr2  
arr5 = arr1-arr2  
a33 = a34.dot(a43)  
a33t = a33.T
```

```
arr1[1]  
arr1[3][2]; arr1[3,2]  
arr1[1:4,1:3]  
arr1[[1,3,2],[0,1,0]]  
arr1[1:4,1]; arr1[1,1:3]  
arr1[1:4,[2,1]]; arr1[[1,3],0:2]
```

### NUMPY: 1d Array

```
import numpy as np  
seq1 = [111,222,333]  
arr1 = np.array(seq1)
```

```
arr1.shape  
arr1.ndim  
arr1.itemsize  
arr1.size
```

```
arr1 = np.arange(2,20,2)  
arr3 = arr1+arr2  
arr4 = arr1*arr2  
arr5 = arr1-arr2  
arr1[0]  
arr1[1]  
arr1[2]
```

### NUMPY: random

```
x = np.random.randint(25,size=(3,4))  
x = np.random.rand(4,30)  
x = np.random.random((4,30))  
x = np.random.randn(4,30)
```

### NUMPY: condition

```
z = np.random.randint(25,size=(4,5,3))  
mask = z > 15; x = z[mask]  
z[z%2 !=0]  
np.count_nonzero(z>20)  
np.sum(z > 20)  
np.any(z > 20)  
np.all(z >= 10)
```

### NUMPY: Statistical functions

```
a = np.array([[3,7,5],[8,4,6],[2,1,9]])  
np.amax(a); np.amin(a)  
np.amin(a,axis = 0); np.amin(a,axis= 1)  
np.amax(a,axis = 0); np.amax(a,axis= 1)  
np.sum(a); np.mean(a)  
np.var(a); np.std(a)  
np.median(a); np.median(a,axis=1)
```

### NUMPY: Predefined

```
np.empty((3,2); np.ones( (3,4)  
np.zeros( (2,3,4)); np.full( (2,2))  
np.identity((3)); np.eye(4, k= 0)  
np.argsort(np_arr)
```

### NUMPY: 2d Array Broadcast

```
a = np.arange(1,25, dtype=np.int32).reshape(6, 4)  
b1 = 3  
b2 = [2,3,4,5]  
b3 = [ [2], [3], [4], [5], [6], [7]]  
b4= np.arange(1,7,dtype=np.int32).reshape(6, 1)  
b5 = np.arange(1,5,dtype=np.int32).reshape(1, 4)  
c = a + b1
```

### Pandas DataFrame

```
dict1 = {"c1": [11,12,13,14,15], "c2": [21,22,23,24,25], "c3":  
[31,32,33,34,35]}
```

```
df = pd.DataFrame(dict1)
```

```
df = pd.DataFrame(dict1, index = ['r1', 'r2', 'r3', 'r4', 'r5'])
```

```
print(df)
```

```
listofd = [ {"Location":"Bangalore","Jan":26,"Feb":25},  
            {"Location":"Pune","Jan":28,"Feb":27},  
            {"Location":"Delhi","Jan":19,"Mar":12}  
          ]
```

```
df = pd.DataFrame(listofd)
```

```
mylist1 = [  
    ("Bangalore",150,80),  
    ("Chennai",200,70)  
  ]
```

```
Labels = ["Location","Jan","Feb"]
```

```
df = pd.DataFrame(mylist1,columns=Labels)
```

### Pandas Melt

```
df2 = pd.melt(df, id_vars=["location", "Temperature"], var_name="Date", value_name="Value")
```

### Pandas Pivot Table

```
df2 = pd.pivot_table(df, index=['Exam_Months', 'Subject'], aggfunc='mean')
df2 = pd.pivot_table(df, index=['Name'], aggfunc='sum')
df3 = df.pivot_table(index='Name', columns = 'Subject', values='Score', aggfunc='mean')
```

### Pandas Merge

```
df1 = pd.read_csv("cmpdata1.csv")
pd.set_option("display.max_rows", None)
pd.set_option("display.max_columns", None)
pd.set_option('display.width', 1000)
dfc = pd.merge(df1, df2, on='emp', how='inner')
dfc = pd.merge(df1, df2, on='emp', how='outer', indicator=True)
dfc = pd.merge(df1, df2, on='emp', how='left', indicator=True)
dfc = pd.merge(df1, df2, on='emp', how='right', indicator=True, suffixes=('_left', '_right'))
dfi = pd.merge(df1, df2, left_on='location', right_on='location1', how='inner', indicator=True)
dfi = pd.merge(df1, df2, left_on='location', right_on='location1', how='outer', indicator=True)
```

### Pandas Access

```
df1.loc['Tata1']
df1.loc['Tata1', 'Tech1']
df1.loc['Tata1': 'Birla1', 'Tech1': 'Group']
df1.loc['Tata1': 'Birla1': 2, 'Tech1': 'Group']
df1.loc[:, 'Tech1': 'Energy1']
df1.loc[['Tata1', 'Adani1', 'Marico1'], ['Infra1', 'Defense1', 'Tech1']]
df1.loc['Reliance1': 2, ['Infra1', 'Tech1', 'Energy1']]

mask = df1.loc[:, :] < 145
df1 = df1[mask]
```

### Pandas duplicates

```
Data = df.drop_duplicates()
df.drop_duplicates(['Name'], keep='last')
df.duplicated()
df.duplicated(['date'])
df.duplicated(['date', 'city'])
df.duplicated(['date', 'city'], keep='last')
df.drop_duplicates(['city'])
```

### Pandas Map, Rename, replace, groupby

```
Season = {'new york': 'winter', 'mumbai': 'summer'}
df['Season'] = df['city'].map(Season)
df['city'].replace('new york', 'Italy')
df.rename(columns= {'temperature': 'temp'})
Data = df.groupby(['city', 'Season'])
df.groupby('city').mean()
df.groupby('city').mean()['temperature']
```

### Pandas Stack, Unstack

```
stacked_df = df.stack()
stacked_df = df.stack(level=0)
stacked_df = df.stack(level=1)
unstacked_df = stacked_df.unstack()
```

### Pandas Concat

```
df1 = pd.read_csv("cmpdata1.csv")
pd.set_option("display.max_rows", None)
pd.set_option("display.max_columns", None)
pd.set_option('display.width', 1000)
dfc = pd.concat([df1, df2], axis=0, join="outer", sort=False)
dfc = pd.concat([df1, df2], axis = 0, join = 'inner', sort=False)
dfk = pd.concat([df1, df2], axis = 0, join_axes=[df1.columns], sort=False)
dfk = pd.concat([df1, df2], axis = 0, join = 'outer', keys=['set1', 'set2'], sort=False)
```

### Pandas Rank

```
x = df.rank(axis=0)
y = df.rank(axis=1)
df['srdes'] = df['Score'].rank(ascending=0)
df['srasc'] = df['Score'].rank(ascending=1)
df['srmn'] = df['Score'].rank(ascending=1, method='min')
df['srmx'] = df['Score'].rank(ascending=1, method='max')
df['srdns'] = df['Score'].rank(ascending=1, method='dense')
df['sragv'] = df['Score'].rank(ascending=1, method='average', pct=True)
df["grank"] = df.groupby("Subject")["Score"].rank(ascending=1, method='dense')
```

### Pandas Access

```
df1['Infra1']
df1.Infra1
df1[['Infra1', 'Group', 'Tech1']]

df1.iloc[0,0]
df1.iloc[[0,2,4],1]
df1.iloc[[0,2,4],[1,3,4]]
df1.iloc[0:3,1:5]
df1.iloc[1:4,2,0:4:2]
df1.iloc[:, [1,3,4]]
df1.iloc[2:5,[1,3,4]]
```