**Pandas Cheat sheet**

1.creating a series/Dataframe using pandas

Seriesname=pandas.Series(data)

Dataframename=pandas.DataFrame(list)/nested list

Dataframename=pandas.DataFrame(tuple)

Dataframename=pandas.DataFrame(dictionary)

Dataframename=pandas.DataFrame(numpy\_array)

2.To find the order

df-dataframe

df.shape

3.To find the dimension

Df.ndim

4.to find the data type of element

Df.dtype

5. to find the type

Type(df)

6.to add an additional column to the dataframe

Df[‘New\_col\_name’]=[‘values’]

7. to display a perticulat column of dataframe

Df[‘colname’]

Df.col\_name

8.to drop a particular coloumn

Df.drop(‘colname’,axis=1)

Del df[‘colname’]

Df.pop(‘colname’)

9.head/tail

To dislay first 5 elements and last 5 elements

Df.head()

Df.tail()

10. df.values

Displays the values of df in an array form

11. df.value\_counts

Displays the count of the value in particular column

12.df.empty

To check wether the df is empty or not

13. df.size

To display the number of elements in array

14. to do a mathematical operation on particular column

Df.[‘colname’].sum()

Df.[‘colname’].mean()

Df.[‘colname’].median()

Df.[‘colname’].max()

Df.[‘colname’].min()

Df.[‘colname’].mode() etc..

15. to display the entire mathematical and statistical ops

Df.[‘colname’].describe()

16.to perform a mathematical operation of entire dataframe.

Note: the result will be shoen default column wise

Df.sum()

Df.count()

Df.mean()

Df.median()

Df.cumsum()

Df.std() etc..

16. pipe(): Table wise function applications in Pandas

Pandas function

D = df.pipe(lambda x,y:x+y,10)

Will add all elements with 10

17. apply(): Row or column wise function operation

Def sqr(x):

Return x\*\*2

Df.apply(sqr)

The dataframe will retuen the square of values

Axis = 1 column wisw

Axis=2 row wise

df.apply(lambda x: x.max()- x.min(),axis=1)

18. applymap(): Element-wise function applications In Pandas

df.applymap(lambda x: x\*100)

will multiply each elements in df with 100

df.[‘colname’].map(lambda x:x\*100)

column wise

19. loc and iloc

loc

Df.loc[‘index\_name’,’colname’]

To get particular value

Df.loc[[‘row\_name’,’r2’,’r3’],[‘c1,’c3’,’c5’]

To get particular column and rows

slicing

Df.loc[‘index\_start’:’end’,’col\_start’:’col\_end’]

Note: both the starting and ending values included in loc method slicing

Iloc

Interlock

Df.iloc[row\_ind:end,col\_ind:end]

Here it has same slicing standard of python i.e last not included

Df.iloc[[1,3,4],[0,1,2]]

20.concatanation of dataframe

B =pd.cncat([d1,d2])

it merges the two dataframe default columnwise one above another

pd.concat([one,three],keys=["x","y"]) #gives an unique identification

pd.concat([one,two,three],keys=["x","y","z"]) #if we only give 2 keys then the third df won't be displayed

pd.concat([two,three],ignore\_index=True) #to provide a normal indexing

pd.concat([one,two],axis="columns") #axis=1 # to join row wise

21.Append

Df1.append(df2)

Df1.append([df2,df3])

23. merge

To join two dataframe on particular series

Default inner join

pd.merge(one,two,on = "id") #the default merge is inner join

Left outer join

pd.merge(one,two,on = "subject\_id",how="left") #left outer join #displays first

Right outer join

d.merge(one,two,on = "subject\_id",how="right") #displays second

Outer join

d.merge(one,two,on = "subject\_id",how="outer") #complete

23. Group by

df.groupby(‘colname’).groups

24.reindex

Renaming rows

df = pd.DataFrame(data,index=['a','b','c','d','e'],columns=['A','B','C'])

df.reindex([‘set values’])

25. null value(isnull or isna)

Df.isnull()

Returns Boolean

Df.isnull().sum()

Returns sum of null values in a column

Df.isnull().any().sum()

Column wise addition of null values if presen

Ex: is 1st col has 2 null values it retuens 1 and adds with next colum if it has null val

24. csv file handeling

Df = pd.read\_csv(“location”)

25. filling up null values

Df.fillna(value)

We can replace a null value with the value of our choice, it can either be mean,median etc..