# Manipulating Pandas DataFrames

* **Set index for easier filtering of dates**

df.set\_index(‘Col name’,inplace=True)

df.sort\_index()

df.reset\_index(inplace=True)

* **Convert column to datetime if not already**

df[‘Col name’]=pd.to\_datetime(df[‘Col name’])

* **Subset dataframe or series**

series=df[‘Col name’]

df\_new=df[[‘Col name 1’, ‘Col name 2’]]

df\_new=df.loc[‘Index name 1’:’Index name 2’,:]

(grp[‘Col name’].nunique()).loc[‘Index name’]

df\_new=df.iloc[:,0:3]

* **Filter data by multiple conditions**

Use >, <, >=, <=, ==, !=

filter = df[‘Col name’]>10

string\_filter=df[‘Col name’].str.contains(‘text’)

df\_filtered=df[filter]

Combine multiple conditions with (filter1) & (filter2), (filter1) | (filter2)

df\_filtered=df[(df[‘Col name 1’]>=number)|(df[‘Col name 2’]==’value’)]

* **Count values in a series**

df[‘Col name’].value\_counts()

* **Group data and do aggregate calculations**

grp= df.groupby(‘Col name’)

grp[‘Col name 1’, ‘Col name 2’].count()

.nunique(),.sum(),.mean(),.max(),.min()

grp[‘Col name 1’, ‘Col name 2’].agg([‘min’,’max’,’mean’])

grp.apply and grp.transform

* **Join data**

df3=pd.merge(df1,df2,how='outer',on=‘Col name’) #outer/inner/left/right

any columns with same name get appended with \_x (left) and \_y (right)

* **Self-join data**

when- you want to get a value from another row of same table

df.join(df.drop('m\_ids',1).set\_index('e\_ids'),on='m\_ids',rsuffix='e\_names')

* **Sort data**

df.sort\_values(by=[‘Col name 1’, ‘Col name 2’],ascending=True, inplace=True)

sorted(“series”) #sort series

* **Handle missing/incomplete data**

df[df.isna().any(axis=1)] #find missing values

df.dropna(axis=0,subset=[‘Col name’],inplace=True) #drop rows w/ cols na

df[‘Col name’].fillna(df[‘Col name’].mean(),inplace=True)

* **Handle duplicated data**

df[df[‘Col name’].duplicated(keep=False)==True] #identify duplicates

df.drop([row#,row#],axis=0,inplace=True) #drop specific rows

df.drop\_duplicates(subset=‘Col name’,keep=’first’,inplace=True) #keep first/last

* **Create a unique key**

df[‘key name’]=df[‘Col name 1’].astype(str)+’-’+df[‘Col name 2’].astype(str)