PANDAS

Python, pandas is a powerful and versatile library specifically designed for data manipulation and analysis

Pip install pandas
pip install --upgrade pandas
pip install jupyter
python -m notebook -> ek notebook server launch ho jata h

jupyter me Shift + enter code execute ya run hota hai

df = pd.DataFrame(dict) -> dictionary ke datas ko table me convert karta h

df.to_csv('friends.csv') -> Dictionary k data ko csv file me convert karta h

df.to_csv('friends_index.csv', index=False) -> index ko 1 se start karta h nhi to 0 se start

df.head(2) -> shuru k 2 row dikhata h

df.tail(2) -> last k 2 row dikhata h

df.describe() -> numeric coloumns ko calculate ya statistical analysis karta h

ankit = pd.read_csv('ankit.csv') -> python k andar csv file ko read karta h

ankit['username'] -> ek coloumn ka data show karta h

ankit['trainno.'][0] = 10 -> kisi column k data ko change kar sakte h index anusaar

ankit.to_csv('new_updated_csv') -> values ko change karke ek new csv bana sakte h

ankit.index = ['first', 'second', 'third', 'fourth'] -> string wale index add karta h

row ko access karne k liye row ka naam hota h aur
column ko access karne k liye column ka naam hotah

pandas
data analysis ki open source library h jo python me likhi gayi
ye numpy ki power aur speed istemaal karta h
numpy fast kaam karta h
numpy ek better storage deta h
har kaam ko karne functions provide karta h

pandas k types
series
indexes k saath one dimensional array hota h
ye kisi bhi tarhe ka data ko hold karta h

ye single row aur single column ko dataframe store karta h

dataframe
ye 2 dimensional data structure hota h
different types of data ko hold karta h
ek tabular spreadsheet jesa structure hota h
isme ek aur multiple column ho sakte h
har column ka datatype same hota h

jupyter notebook
ek open sourse web application h
jo documents ko cread aur share karta h
aur ye live code, equations, visulizations and narrative text ko contain karta h
jupyter 40 programming language support karta h
notebook ko email, dropbox, github par share kiya ja sakta h
jupyter ko data analysis me sabse jyaada use kiya jata h

ser = pd.Series(np.random.rand(10)) -> 0 se lekar 10 tak random series dega

newdf = pd.DataFrame(np.random.rand(334,5), index=np.arange(334)) -> saare row column dikhayega data frame k

newdf[0][0] = "ankitkumar" -> k first row aur first column k data ko change kar sakte h

newdf.index -> 0 se n tak index dega

newdf.columns -> o se lekar n tak column dega with steps

newdf.to_numpy() -> data ko numpy array me convert karta h

newdf.T -> row ko column aur column ko row me transpose karta h

newdf.sort_index(axis=0, ascending=False) -> table k rows ko oolta sort kardega

newdf.sort_index(axis=1, ascending=False) -> table k columns ko oolta sort kardega

axis = 0 for row
axis = 1 for coloumn

newdf[0] -> colomns k saara data show karega series me

series k combination se data frame banta h

newdf2 newdf ka view h
newdf2 = newdf -> to pointer ki tarhe h , agar newdf 2 ko modify kiya to newdf bhi change hoga

newdf2 = newdf.copy()
newdf2 =[0][0] = 1234

newdf
copy hota h

newdf.loc[0,0] = 563
newdf.head[2]
ek table ki cell me value ko change karta h

newdf.columns = list("ABCDE")
columns k index ko alphabet me convert karta h

newdf.loc[0,'A'] = 343
ise ese update karte h

newdf = newdf.drop(0, axis=1)
ye column k faltu no. ko remove karta h

newdf.loc[[1, 2], ['a', 'd']]
table k row aur columns ko view k perspective k liye change karta h sir show karne k liye permanent nhi karta h

newdf.loc[:,['C','D']] -> table k saare row deta aur c d column deta

newdf.loc[[1,2],:] -> table k saare column deta aur 12 row deta

newdf.loc[newdf['A']<0.3)] -> table 0.3 se badi values k data show karega

newdf.loc[(newdf['A']<0.3) & (newdf['C']>0.1)] -> boolean expression

newdf.iloc[0,4] -> 0 row par 4th column par jo value h wo dega ek single cell ki value dega

newdf.iloc[[0,5],[1,2]] -> 0 row 1 columne ki value dega aur 5 row 2 cloumn ki value dega

newdf.drop([0]) -> 0th row ko udaa dega by default 0 axis set hota h row k liye

newdf.drop([0], axis=1) -> pheli coloumn ko uda deta

newdf.drop(['0','1'],axis=1) -> 0 aur 1 wale column ko udaa dega

newdf.drop(['A','D'],axis=1,implace=True) -> selected data ko drop kar deta h

newdf.reset_index()
newdf.reset_index(drop=True)
newdf.reset(drop=True, inplace=True)
indexes ko reset kar deta h

loc और iloc डेटाफ़्रेम को लेबल या पोज़िशन के आधार पर चुनने के तरीके हैं (In Jupyter, loc and iloc are ways to select dataframes based on labels or positions
loc

labels (index names ya column names) ka use karke data ko select karta hai, jabki
iloc
integer positions ka use karke data ko select karta hai.

newdf['B'].isnull() -> jitne bhi row me zero un sab me yeh false dega

newdf.loc[:,['B']] = 40 -> b column k data me 40 replace kar dega

sabse jyaada loc ya iloc use karna chaiye

newdf['B'] = None -> b me saari values ko none kar dega

df = pd.DataFrame({
"name":['Alfred', 'Batman', 'catwoman'],
"toy": [np.nan, 'batmobile', 'bullwhip'],
"born": [pd.Nat, pd.Timestamp("2001-03-23"), pd.NaT]})

df.dropna() -> jha jha na h wha na hata dega

df.dropna(how='all', axis=1) -> same data k column ko remove karta h

df.drop_duplicates(subset=['name'],keep='first')
df.drop_duplicates(subset=['name'],keep='last')
df.drop_duplicates(subset=['name'],keep=False) -> duplicate data ko remove karta h

df.shape -> table ki shape dega

df.info() -> table ki info dega

df.['toy'].value_counts(dropna=False) -> na ko hatao ya mat hatao ne ka command

df.notnull() -> jha null nhi h wha true hota h aur jha null h wha false hota h

create a dataframe which contains only intergers with 3 rows and 2 columns run following dataframe methods on time:
df.descirbe()
df.mean()
df.corr()
df.count()
df.max()
df.min()
df.median()
df.std()
ans. import pandas as pd

data = [[1, 3], [5, 2], [6, 4]]
df = pd.DataFrame(data, columns=["Col1", "Col2"])

```
print(df, df.describe(), df.mean(), df.corr(), df.count(),
      df.max(), df.min(), df.median(), df.std(), sep="\n\n")
```

data = pd.read_excel('ankit.csv') -> excel ki sheet ko read karta h , ya multiple sheet ko read kiya ja sakta h

data.to_excel('data.xlsx', sheet_name='sheet2') -> sheet 2 k andar sheet1 ka modify data show karta h aur sheet1 gayab kar dega