**4.Pandas:**

**Introduction:**

* Used for data manipulation(data cleaning,organizing data)
* Creates dataframes from excel,csv,txt,DBs
* Dataframe(rows and columns readable by python)
* Data cleaning by dropping or replacing with mean
* Visualize the data

**Ex: (series)**

import pandas as pd

names=["vasavi","harsha","hasini","pavan"]

index=[40,42,43,45]

ser1=pd.Series(names,index)

print(ser1)

**Output:**

40 vasavi

42 harsha

43 hasini

45 pavan

dtype: object

**Importing files:**

- for csv and txt:read\_csv("file path")

- for excel:read\_excel("file path")

**Ex-1:**

import pandas as pd

df=pd.read\_csv("/content/demoDT.csv")

df.head(10)

**Output:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | State | Literacy | Cleanliness | Crime\_Rate |
| 0 | A | 92 | 90 | 54 |
| 1 | B | 56 | 67 | 50 |
| 2 | C | 78 | 80 | 62 |
| 3 | D | 85 | 75 | 48 |
| 4 | E | 60 | 82 | 55 |
| 5 | F | 72 | 78 | 60 |
| 6 | G | 88 | 85 | 45 |
| 7 | H | 64 | 70 | 58 |
| 8 | I | 90 | 88 | 42 |
| 9 | J | 68 | 72 | 65 |

df.tail(10)

Output:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | State | Literacy | Cleanliness | Crime\_Rate |
| 16 | Q | 79 | 83 | 47 |
| 17 | R | 71 | 68 | 59 |
| 18 | S | 86 | 89 | 43 |
| 19 | T | 66 | 71 | 64 |
| 20 | U | 81 | 81 | 51 |
| 21 | V | 70 | 77 | 56 |
| 22 | W | 87 | 84 | 46 |
| 23 | X | 62 | 69 | 66 |
| 24 | Y | 76 | 73 | 53 |
| 25 | Z | 83 | 87 | 44 |

**Ex-2:**

df=pd.read\_csv("/content/grades.txt",sep=" ") #**sep is used to get align**

print(df.tail(8)) #returns

print(df.describe) **#total data of the list**

print(df.shape) **#get the no. of rows and columns**

print(df.shape[0]) **#get the no. of rows only**

print(df.shape[1]) **#get the no. of columnss only**

print(df.columns) **#get the column names**

**Output:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Names | | Initials | SEM1 | | SEM2 | | SEM3 | Grade |
| 8 | | Jess | | K | 9.8 | | 9.1 | | 9.9 | A+ |
| 9 | | Rajini | | M | 7.0 | | 9.1 | | 9.3 | A |
| 10 | | Kiran | | V | 9.9 | | 9.3 | | 9.2 | A |
| 11 | | Maya | | N | 7.7 | | 8.0 | | 7.1 | B |
| 12 | | Jolin | | K | 9.8 | | 9.1 | | 9.9 | A+ |
| 13 | | Riya | | M | 8.0 | | 9.1 | | 9.3 | A |
| 14 | | Sana | | V | 9.9 | | 9.3 | | 9.2 | A |
| 15 | | Mark | | N | 7.7 | | 8.0 | | 7.0 | B |
|  | | **Names** | | | **Initials** | | **SEM1** | | **SEM2** | **SEM3** | **Grade** |
| 0 | | Joe | | | K | | 9.8 | | 10.0 | 9.9 | A+ |
| 1 | | Rajesh | | | M | | 8.9 | | 9.1 | 9.3 | A |
| 2 | | Kissan | | | V | | 9.9 | | 9.3 | 9.2 | A |
| 3 | | Mary | | | N | | 7.7 | | 8.0 | 7.1 | B |
| 4 | | Jeen | | | K | | 9.8 | | 9.1 | 9.9 | A+ |
| 5 | | Raj | | | M | | 8.9 | | 9.1 | 9.3 | A |
| 6 | | Hassan | | | V | | 9.9 | | 9.0 | 9.2 | A |
| 7 | | Mari | | | N | | 7.7 | | 8.0 | 7.1 | B |
| 8 | | Jess | | | K | | 9.8 | | 9.1 | 9.9 | A+ |
| 9 | | Rajini | | | M | | 7.0 | | 9.1 | 9.3 | A |
| 10 | | Kiran | | | V | | 9.9 | | 9.3 | 9.2 | A |
| 11 | | Maya | | | N | | 7.7 | | 8.0 | 7.1 | B |
| 12 | | Jolin | | | K | | 9.8 | | 9.1 | 9.9 | A+ |
| 13 | | Riya | | | M | | 8.0 | | 9.1 | 9.3 | A |
| 14 | | Sana | | | V | | 9.9 | | 9.3 | 9.2 | A |
| 15 | | Mark | | | N | | 7.7 | | 8.0 | 7.0 | B |

(16, 6)

16

6

Index(['Names', 'Initials', 'SEM1', 'SEM2', 'SEM3', 'Grade'], dtype='object')

**Ex:**

print(df[2:5]) **#to access rows**

**Output:**

Names Initials SEM1 SEM2 SEM3 Grade

2 Kissan V 9.9 9.3 9.2 A

3 Mary N 7.7 8.0 7.1 B

4 Jeen K 9.8 9.1 9.9 A+

**Accessing data:**

* loc-accepts column names and index
* iloc-accepts only index

**Ex:**

print(df.loc[2:5],"Names") **#rows of specified columns**

print(df.iloc[2:5]) **#rows of specified columns**

print(df.iloc[2:5,:3])    **#iloc[row range,column range]=>index**

Names Initials SEM1 SEM2 SEM3 Grade

2 Kissan V 9.9 9.3 9.2 A

3 Mary N 7.7 8.0 7.1 B

4 Jeen K 9.8 9.1 9.9 A+

5 Raj M 8.9 9.1 9.3 A Names

Names Initials SEM1 SEM2 SEM3 Grade

2 Kissan V 9.9 9.3 9.2 A

3 Mary N 7.7 8.0 7.1 B

4 Jeen K 9.8 9.1 9.9 A+

Names Initials SEM1

2 Kissan V 9.9

3 Mary N 7.7

4 Jeen K 9.8

**Ex-3:**

dfn=pd.read\_csv("/content/grades\_withnulls.csv")

dfn

dfn.isnull().head(7) **#to get 7 rows**

dfn.isnull() **#in place of a null we get true**

dfn.isnull().sum() **#to view how many nulls we have per column**

dfn.isnull().sum().sum() **#to view total nulls**

**Output:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Names | Initials | SEM1 | SEM2 | SEM3 | Grade | Placed |
| 0 | Joe | K | 9.8 | 10.0 | 9.9 | A+ | 1 |
| 1 | Rajesh | M | 8.9 | 9.1 | 9.3 | A | 1 |
| 2 | Kissan | V | 9.9 | 9.8 | 10.0 | A | 0 |
| 3 | Mary | N | 7.7 | 8.0 | NaN | B | 0 |
| 4 | Jeen | K | 9.8 | 9.1 | 9.9 | A+ | 1 |
| 5 | Raj | M | 8.9 | 9.1 | 9.3 | A | 1 |
| 6 | Hassan | V | 9.9 | 9.0 | 9.2 | A | 1 |
| 7 | Mari | N | 7.7 | 8.0 | 7.1 | B | 1 |
| 8 | Jess | K | NaN | 9.1 | 9.9 | A+ | 1 |
| 9 | Rajini | M | NaN | 9.1 | 9.3 | A | 0 |
| 10 | Kiran | V | NaN | 9.3 | 9.2 | A | 0 |
| 11 | Maya | N | 7.7 | 8.0 | 7.1 | B | 0 |
| 12 | Jolin | K | 9.8 | 9.1 | 9.9 | A+ | 1 |
| 13 | Rajesh | M | 8.9 | 9.1 | 9.3 | A | 1 |
| 14 | Riya | M | 9.3 | 9.9 | 10.0 | A | 1 |
| 15 | Sana | V | 9.9 | 9.3 | 9.2 | A | 0 |
| 16 | Mark | N | 7.7 | 8.0 | 7.0 | B | 0 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Names | Initials | SEM1 | SEM2 | SEM3 | Grade | Placed |
| 0 | False | False | False | False | False | False | False |
| 1 | False | False | False | False | False | False | False |
| 2 | False | False | False | False | False | False | False |
| 3 | False | False | False | False | True | False | False |
| 4 | False | False | False | False | False | False | False |
| 5 | False | False | False | False | False | False | False |
| 6 | False | False | False | False | False | False | False |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Names | Initials | SEM1 | SEM2 | SEM3 | Grade | Placed |
| 0 | False | False | False | False | False | False | False |
| 1 | False | False | False | False | False | False | False |
| 2 | False | False | False | False | False | False | False |
| 3 | False | False | False | False | True | False | False |
| 4 | False | False | False | False | False | False | False |
| 5 | False | False | False | False | False | False | False |
| 6 | False | False | False | False | False | False | False |
| 7 | False | False | False | False | False | False | False |
| 8 | False | False | True | False | False | False | False |
| 9 | False | False | True | False | False | False | False |
| 10 | False | False | True | False | False | False | False |
| 11 | False | False | False | False | False | False | False |
| 12 | False | False | False | False | False | False | False |
| 13 | False | False | False | False | False | False | False |
| 14 | False | False | False | False | False | False | False |
| 15 | False | False | False | False | False | False | False |
| 16 | False | False | False | False | False | False | False |

Names 0

Initials 0

SEM1 3

SEM2 0

SEM3 1

Grade 0

Placed 0

dtype: int64

4

**Cleaning:**

**Dropping nulls:**

**Ex:**

dfc=dfn.dropna() #dropping all the rows with nulls

dfc

**Output:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Names | Initials | SEM1 | SEM2 | SEM3 | Grade | Placed |
| 0 | Joe | K | 9.8 | 10.0 | 9.9 | A+ | 1 |
| 1 | Rajesh | M | 8.9 | 9.1 | 9.3 | A | 1 |
| 2 | Kissan | V | 9.9 | 9.8 | 10.0 | A | 0 |
| 4 | Jeen | K | 9.8 | 9.1 | 9.9 | A+ | 1 |
| 5 | Raj | M | 8.9 | 9.1 | 9.3 | A | 1 |
| 6 | Hassan | V | 9.9 | 9.0 | 9.2 | A | 1 |
| 7 | Mari | N | 7.7 | 8.0 | 7.1 | B | 1 |
| 11 | Maya | N | 7.7 | 8.0 | 7.1 | B | 0 |
| 12 | Jolin | K | 9.8 | 9.1 | 9.9 | A+ | 1 |
| 13 | Rajesh | M | 8.9 | 9.1 | 9.3 | A | 1 |
| 14 | Riya | M | 9.3 | 9.9 | 10.0 | A | 1 |
| 15 | Sana | V | 9.9 | 9.3 | 9.2 | A | 0 |
| 16 | Mark | N | 7.7 | 8.0 | 7.0 | B | 0 |

**Filling nulls:**

**Ex:**

dfc1=dfn.fillna(5) **#null values are replaced with specified value**

dfc1

**Output:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Names | Initials | SEM1 | SEM2 | SEM3 | Grade | Placed |
| 0 | Joe | K | 9.8 | 10.0 | 9.9 | A+ | 1 |
| 1 | Rajesh | M | 8.9 | 9.1 | 9.3 | A | 1 |
| 2 | Kissan | V | 9.9 | 9.8 | 10.0 | A | 0 |
| 3 | Mary | N | 7.7 | 8.0 | 5.0 | B | 0 |
| 4 | Jeen | K | 9.8 | 9.1 | 9.9 | A+ | 1 |
| 5 | Raj | M | 8.9 | 9.1 | 9.3 | A | 1 |
| 6 | Hassan | V | 9.9 | 9.0 | 9.2 | A | 1 |
| 7 | Mari | N | 7.7 | 8.0 | 7.1 | B | 1 |
| 8 | Jess | K | 5.0 | 9.1 | 9.9 | A+ | 1 |
| 9 | Rajini | M | 5.0 | 9.1 | 9.3 | A | 0 |
| 10 | Kiran | V | 5.0 | 9.3 | 9.2 | A | 0 |
| 11 | Maya | N | 7.7 | 8.0 | 7.1 | B | 0 |
| 12 | Jolin | K | 9.8 | 9.1 | 9.9 | A+ | 1 |
| 13 | Rajesh | M | 8.9 | 9.1 | 9.3 | A | 1 |
| 14 | Riya | M | 9.3 | 9.9 | 10.0 | A | 1 |
| 15 | Sana | V | 9.9 | 9.3 | 9.2 | A | 0 |
| 16 | Mark | N | 7.7 | 8.0 | 7.0 | B | 0 |

**Cleaning with mean:**

**Ex:**

m=dfn['SEM2'].mean() #mean of sem2

print(m)

dfc2=dfn.fillna(m)

print(dfc2)

**Output:**

9.0

Names Initials SEM1 SEM2 SEM3 Grade Placed

0 Joe K 9.8 10.0 9.9 A+ 1

1 Rajesh M 8.9 9.1 9.3 A 1

2 Kissan V 9.9 9.8 10.0 A 0

3 Mary N 7.7 8.0 9.0 B 0

4 Jeen K 9.8 9.1 9.9 A+ 1

5 Raj M 8.9 9.1 9.3 A 1

6 Hassan V 9.9 9.0 9.2 A 1

7 Mari N 7.7 8.0 7.1 B 1

8 Jess K 9.0 9.1 9.9 A+ 1

9 Rajini M 9.0 9.1 9.3 A 0

10 Kiran V 9.0 9.3 9.2 A 0

11 Maya N 7.7 8.0 7.1 B 0

12 Jolin K 9.8 9.1 9.9 A+ 1

13 Rajesh M 8.9 9.1 9.3 A 1

14 Riya M 9.3 9.9 10.0 A 1

15 Sana V 9.9 9.3 9.2 A 0

16 Mark N 7.7 8.0 7.0 B 0

**Ex:**

dropped=dfc2.drop\_duplicates() #duplicates will be remmoved

print(dropped)

**Output:**

Names Initials SEM1 SEM2 SEM3 Grade Placed

0 Joe K 9.8 10.0 9.9 A+ 1

1 Rajesh M 8.9 9.1 9.3 A 1

2 Kissan V 9.9 9.8 10.0 A 0

3 Mary N 7.7 8.0 9.0 B 0

4 Jeen K 9.8 9.1 9.9 A+ 1

5 Raj M 8.9 9.1 9.3 A 1

6 Hassan V 9.9 9.0 9.2 A 1

7 Mari N 7.7 8.0 7.1 B 1

8 Jess K 9.0 9.1 9.9 A+ 1

9 Rajini M 9.0 9.1 9.3 A 0

10 Kiran V 9.0 9.3 9.2 A 0

11 Maya N 7.7 8.0 7.1 B 0

12 Jolin K 9.8 9.1 9.9 A+ 1

14 Riya M 9.3 9.9 10.0 A 1

15 Sana V 9.9 9.3 9.2 A 0

16 Mark N 7.7 8.0 7.0 B 0

**Ex: (rename)**

dfn.rename(columns={"Grade":"GPA"},inplace=True) #dataframe.rename(columns={"original\_name":"new\_name"},inplace=TRUE) dfn

**Output:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Names | Initials | SEM1 | SEM2 | SEM3 | GPA | Placed |
| 0 | Joe | K | 9.8 | 10.0 | 9.9 | A+ | 1 |
| 1 | Rajesh | M | 8.9 | 9.1 | 9.3 | A | 1 |
| 2 | Kissan | V | 9.9 | 9.8 | 10.0 | A | 0 |
| 3 | Mary | N | 7.7 | 8.0 | NaN | B | 0 |
| 4 | Jeen | K | 9.8 | 9.1 | 9.9 | A+ | 1 |
| 5 | Raj | M | 8.9 | 9.1 | 9.3 | A | 1 |
| 6 | Hassan | V | 9.9 | 9.0 | 9.2 | A | 1 |
| 7 | Mari | N | 7.7 | 8.0 | 7.1 | B | 1 |
| 8 | Jess | K | NaN | 9.1 | 9.9 | A+ | 1 |
| 9 | Rajini | M | NaN | 9.1 | 9.3 | A | 0 |
| 10 | Kiran | V | NaN | 9.3 | 9.2 | A | 0 |
| 11 | Maya | N | 7.7 | 8.0 | 7.1 | B | 0 |
| 12 | Jolin | K | 9.8 | 9.1 | 9.9 | A+ | 1 |
| 13 | Rajesh | M | 8.9 | 9.1 | 9.3 | A | 1 |
| 14 | Riya | M | 9.3 | 9.9 | 10.0 | A | 1 |
| 15 | Sana | V | 9.9 | 9.3 | 9.2 | A | 0 |
| 16 | Mark | N | 7.7 | 8.0 | 7.0 | B | 0 |

**Ex-2:**

dfn['Avg\_score']=(dfn['SEM1']+dfn['SEM2']+dfn['SEM3'])/3 **#df['newcol']=values**

dfn.head()

**Output:**

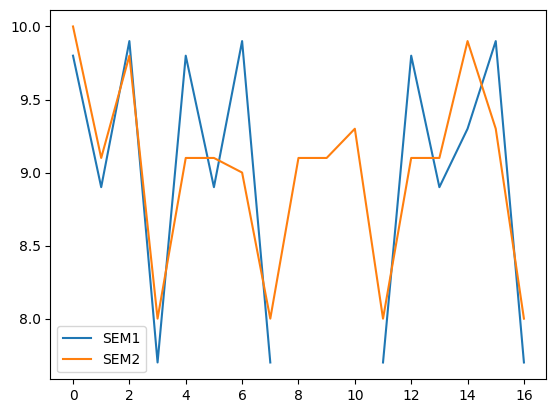
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Names | Initials | SEM1 | SEM2 | SEM3 | GPA | Placed | Avg\_score |
| 0 | Joe | K | 9.8 | 10.0 | 9.9 | A+ | 1 | 9.9 |
| 1 | Rajesh | M | 8.9 | 9.1 | 9.3 | A | 1 | 9.1 |
| 2 | Kissan | V | 9.9 | 9.8 | 10.0 | A | 0 | 9.9 |
| 3 | Mary | N | 7.7 | 8.0 | NaN | B | 0 | NaN |
| 4 | Jeen | K | 9.8 | 9.1 | 9.9 | A+ | 1 | 9.6 |

**Plotting with pandas:**

**Ex-1: (Two plots in one graph)**

dfn[['SEM1','SEM2']].plot.line()

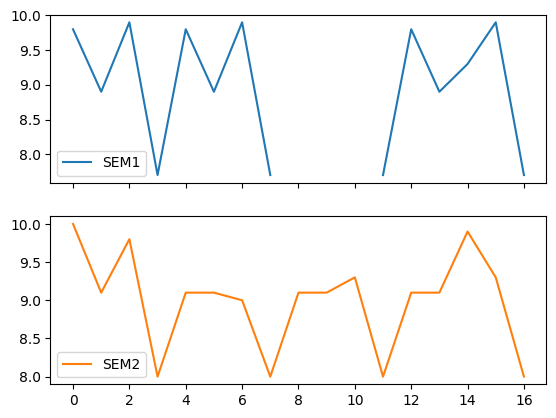
**Output:**



**Ex-2: (sub plots)**

dfn[['SEM1','SEM2']].plot.line(subplots=True)

**Output:**



**Ex-3:**

dfn.plot.line(subplots=True)

**Output:**

