Pandas

pandas is a data manipulation package in Python for tabular data. That is, data in the form of rows and columns, also known as DataFrames.

pandas is used throughout the data analysis workflow. With pandas, you can: -Import datasets from databases, spreadsheets, comma-separated values (CSV) files, and more. -Clean datasets, for example, by dealing with missing values. -Tidy datasets by reshaping their structure into a suitable format for analysis. -Aggregate data by calculating summary statistics such as the mean of columns, correlation between them, and more. -Visualize datasets and uncover insights.

- We can access the elements with loc(starts from 0) or iloc(starts from 1)

```
#importing pandas
import pandas as pd

import pandas as pd
a = ["Jwalitha", 'Ramya',
'Durga','Jahnavi','Lahari','Sunny','Dhanush']
r = pd.Series(a, index = [67,43,44,89,34,45,23])
print(r)
```

```
67    Jwalitha
43       Ramya
44       Durga
89     Jahnavi
34      Lahari
45       Sunny
23     Dhanush
dtype: object
```

```
df = pd.read_csv("D:\Jwalitha\Datasets\diabetcsv.csv")
print(df)
```

```
     preg  plas  pres  skin  insu  mass   pedi  age            class
0       6   148    72    35     0  33.6  0.627   50  tested_positive
1       1    85    66    29     0  26.6  0.351   31  tested_negative
2       8   183    64     0     0  23.3  0.672   32  tested_positive
3       1    89    66    23    94  28.1  0.167   21  tested_negative
4       0   137    40    35   168  43.1  2.288   33  tested_positive
..    ...   ...   ...   ...   ...   ...    ...  ...              ...
763    10   101    76    48   180  32.9  0.171   63  tested_negative
764     2   122    70    27     0  36.8  0.340   27  tested_negative
765     5   121    72    23   112  26.2  0.245   30  tested_negative
766     1   126    60     0     0  30.1  0.349   47  tested_positive
767     1    93    70    31     0  30.4  0.315   23  tested_negative

[768 rows x 9 columns]
```

```
df = pd.read_csv("D:\Jwalitha\Datasets\grades.txt", sep=" ")
print(df)
```

```
     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
```

```
df = pd.read_excel("D:\Jwalitha\Datasets\diabetes.xlsx")
print(df)
```

```
     preg  plas  pres  skin  insu  mass   pedi  age            class
0       6   148    72    35     0  33.6  0.627   50  tested_positive
1       1    85    66    29     0  26.6  0.351   31  tested_negative
2       8   183    64     0     0  23.3  0.672   32  tested_positive
3       1    89    66    23    94  28.1  0.167   21  tested_negative
4       0   137    40    35   168  43.1  2.288   33  tested_positive
..    ...   ...   ...   ...   ...   ...    ...  ...              ...
763    10   101    76    48   180  32.9  0.171   63  tested_negative
764     2   122    70    27     0  36.8  0.340   27  tested_negative
765     5   121    72    23   112  26.2  0.245   30  tested_negative
766     1   126    60     0     0  30.1  0.349   47  tested_positive
767     1    93    70    31     0  30.4  0.315   23  tested_negative

[768 rows x 9 columns]
```

```
df = pd.read_excel("D:\Jwalitha\Datasets\diabetes.xlsx",sheet_name=1)
print(df)

#also you can use :  df = pd.read_json("diabetes.json") for loading
json files
```

```
  Dead Alive
0  yes    no
1  yes    no
2  yes    no
3  yes    no
4  yes    no
```

```
df = pd.read_csv("D:\Jwalitha\Datasets\grades.txt", sep=" ")
print(df.head())
print(df.tail(n=10))
```

```
    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+
     Names Initials  SEM1  SEM2  SEM3 Grade
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
```

```
df = pd.read_csv("D:\Jwalitha\Datasets\diabetcsv.csv")
print(df.describe)
```

```
<bound method NDFrame.describe of      preg  plas  pres  skin  insu
mass   pedi  age            class
0       6   148    72    35     0  33.6  0.627   50  tested_positive
1       1    85    66    29     0  26.6  0.351   31  tested_negative
2       8   183    64     0     0  23.3  0.672   32  tested_positive
3       1    89    66    23    94  28.1  0.167   21  tested_negative
4       0   137    40    35   168  43.1  2.288   33  tested_positive
..    ...   ...   ...   ...   ...   ...    ...  ...              ...
763    10   101    76    48   180  32.9  0.171   63  tested_negative
764     2   122    70    27     0  36.8  0.340   27  tested_negative
765     5   121    72    23   112  26.2  0.245   30  tested_negative
766     1   126    60     0     0  30.1  0.349   47  tested_positive
767     1    93    70    31     0  30.4  0.315   23  tested_negative

[768 rows x 9 columns]>
```

```
df = pd.read_csv("D:\Jwalitha\Datasets\grades.txt", sep=" ")
print(df.describe().T)
```

```
      count     mean       std  min    25%   50%    75%   max
SEM1   16.0  8.90000  1.077652  7.0  7.700  9.35  9.825   9.9
SEM2   16.0  8.91250  0.589774  8.0  8.750  9.10  9.150  10.0
SEM3   16.0  8.86875  1.104970  7.0  8.675  9.25  9.450   9.9
```

df.info(show_counts=True, memory_usage=True, verbose=True) => also used to view and understand data

```
df = pd.read_csv("D:\Jwalitha\Datasets\grades.txt", sep=" ")
print(df.shape) # Get the number of rows and columns
print(df.shape[0]) # Get the number of rows only
print(df.shape[1]) # Get the number of columns only
```

```
(16, 6)
16
6
```

```
#creating a list with all column names
df = pd.read_csv("D:\Jwalitha\Datasets\grades.txt", sep=" ")
print(df.columns)

l1 = list(df.columns)
print(l1)
```

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

```
#making null values
df = pd.read_csv("D:\Jwalitha\Datasets\grades.txt", sep=" ")
df2 = df.copy()
df2.loc[2:5,'SEM1'] = None
df2.head(7)
```

	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	NaN	9.3	9.2	A
3	Mary	N	NaN	8.0	7.1	B
4	Jeen	K	NaN	9.1	9.9	A+
5	Raj	M	NaN	9.1	9.3	A
6	Hassan	V	9.9	9.0	9.2	A

```
df2.isnull().head(7)
```

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

```
print(df2.isnull().sum())
```

```
Names       0
Initials    0
SEM1        4
```

```
SEM2        0
SEM3        0
Grade       0
dtype: int64
```

```
print(df2.isnull().sum().sum())
```

```
4
```

```
#extracting column
print(df['SEM1'])
```

```
0     9.8
1     8.9
2     9.9
3     7.7
4     9.8
5     8.9
6     9.9
7     7.7
8     9.8
9     7.0
10    9.9
11    7.7
12    9.8
13    8.0
14    9.9
15    7.7
Name: SEM1, dtype: float64
```

```
print(df2[['SEM1','Names']])
```

```
    SEM1   Names
0    9.8     Joe
1    8.9  Rajesh
2    NaN  Kissan
3    NaN    Mary
4    NaN    Jeen
5    NaN     Raj
6    9.9  Hassan
7    7.7    Mari
8    9.8    Jess
9    7.0  Rajini
10   9.9   Kiran
11   7.7    Maya
12   9.8   Jolin
13   8.0    Riya
14   9.9    Sana
15   7.7    Mark
```

```
#printing one row
print(df[df.index==1])
```

```
    Names Initials  SEM1  SEM2  SEM3 Grade
1  Rajesh        M   8.9   9.1   9.3     A
```

```
#printing multiple rows
print(df[df.index.isin(range(2,5))])
```

```
    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+
```

```
#printing one row as  series
print(df2.loc[1])
```

```
Names       Rajesh
Initials         M
SEM1           8.9
SEM2           9.1
SEM3           9.3
Grade            A
Name: 1, dtype: object
```

```
df2.loc[6:11]
```

```
     Names Initials  SEM1  SEM2  SEM3 Grade
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
```

```
df2.loc[[10, 2, 7]]
```

```
     Names Initials  SEM1  SEM2  SEM3 Grade
10   Kiran        V   9.9   9.3   9.2     A
2   Kissan        V   NaN   9.3   9.2     A
7     Mari        N   7.7   8.0   7.1     B
```

```
df2.loc[10:15, ['Names', 'Grade',]]
```

```
    Names Grade
10  Kiran     A
11   Maya     B
12  Jolin    A+
13   Riya     A
14   Sana     A
15   Mark     B
```

```
df2.iloc[10:14, :3] #[row,column]
```

```
    Names Initials  SEM1
10  Kiran        V   9.9
11   Maya        N   7.7
12  Jolin        K   9.8
13   Riya        M   8.0
```

```
df2.loc[:7, ["Names"]] #[row,column]
```

```
    Names
0     Joe
1  Rajesh
2  Kissan
3    Mary
4    Jeen
5     Raj
6  Hassan
7    Mari
```

```
df2.loc[df['Names']=='Rajesh']
```

```
    Names Initials  SEM1  SEM2  SEM3 Grade
1  Rajesh        M   8.9   9.1   9.3     A
```

```
df[df.Grade == 'A']
df[df.SEM2 == 10]
```

```
  Names Initials  SEM1  SEM2  SEM3 Grade
0   Joe        K   9.8  10.0   9.9    A+
```

```
df.loc[df['SEM1'] > 9.8, ['Names']] #printing names of students who
scored more than 9.8 in SEM1
```

```
     Names
2   Kissan
6   Hassan
10   Kiran
14    Sana
```

```
df3 = df2.copy()
#df3 = df3.dropna()
df3
```

```
     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   NaN   9.3   9.2     A
3     Mary        N   NaN   8.0   7.1     B
4     Jeen        K   NaN   9.1   9.9    A+
5      Raj        M   NaN   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
```

```
df3.dropna(inplace=True, axis=1) #inplace says not to reflect the
changes in the df
df3
```

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

```
df3.dropna(inplace=True, how='all') #drops both rows and columns with
NaN
df3
```

```
     Names Initials  SEM2  SEM3 Grade
0      Joe         K  10.0   9.9    A+
1   Rajesh         M   9.1   9.3     A
2   Kissan         V   9.3   9.2     A
3     Mary         N   8.0   7.1     B
4     Jeen         K   9.1   9.9    A+
5      Raj         M   9.1   9.3     A
6   Hassan         V   9.0   9.2     A
7     Mari         N   8.0   7.1     B
8     Jess         K   9.1   9.9    A+
9   Rajini         M   9.1   9.3     A
10   Kiran         V   9.3   9.2     A
11    Maya         N   8.0   7.1     B
12   Jolin         K   9.1   9.9    A+
```

```
13    Riya        M   9.1   9.3     A
14    Sana        V   9.3   9.2     A
15    Mark        N   8.0   7.0     B
```

```
dfn = pd.read_csv("D:\Jwalitha\Datasets\grades_withnulls.csv")
dfn
```

```
     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
```

```
mv = dfn['Sem1'].mean()
dfn = dfn.fillna(mv)
print(dfn)
```

```
     Names Initials      Sem1  Sem2       Sem3 Grade  Placed
0      Joe        K  9.800000  10.0   9.900000    A+       1
1   Rajesh        M  8.900000   9.1   9.300000     A       1
2   Kissan        V  9.900000   9.8  10.000000     A       0
3     Mary        N  7.700000   8.0   8.992857     B       0
4     Jeen        K  9.800000   9.1   9.900000    A+       1
5      Raj        M  8.900000   9.1   9.300000     A       1
6   Hassan        V  9.900000   9.0   9.200000     A       1
7     Mari        N  7.700000   8.0   7.100000     B       1
8     Jess        K  8.992857   9.1   9.900000    A+       1
9   Rajini        M  8.992857   9.1   9.300000     A       0
10   Kiran        V  8.992857   9.3   9.200000     A       0
11    Maya        N  7.700000   8.0   7.100000     B       0
12   Jolin        K  9.800000   9.1   9.900000    A+       1
13  Rajesh        M  8.900000   9.1   9.300000     A       1
14    Riya        M  9.300000   9.9  10.000000     A       1
15    Sana        V  9.900000   9.3   9.200000     A       0
16    Mark        N  7.700000   8.0   7.000000     B       0
```

```
dfn = dfn.drop_duplicates()
print(dfn)
```

```
     Names Initials      Sem1  Sem2       Sem3 Grade  Placed
0      Joe        K  9.800000  10.0   9.900000    A+       1
1   Rajesh        M  8.900000   9.1   9.300000     A       1
2   Kissan        V  9.900000   9.8  10.000000     A       0
3     Mary        N  7.700000   8.0   8.992857     B       0
4     Jeen        K  9.800000   9.1   9.900000    A+       1
5      Raj        M  8.900000   9.1   9.300000     A       1
6   Hassan        V  9.900000   9.0   9.200000     A       1
7     Mari        N  7.700000   8.0   7.100000     B       1
8     Jess        K  8.992857   9.1   9.900000    A+       1
9   Rajini        M  8.992857   9.1   9.300000     A       0
10   Kiran        V  8.992857   9.3   9.200000     A       0
11    Maya        N  7.700000   8.0   7.100000     B       0
12   Jolin        K  9.800000   9.1   9.900000    A+       1
14    Riya        M  9.300000   9.9  10.000000     A       1
15    Sana        V  9.900000   9.3   9.200000     A       0
16    Mark        N  7.700000   8.0   7.000000     B       0
```

```
dfn.rename(columns = {'Grade':'GPA'}, inplace = True)
dfn.head()
```

```
    Names Initials  Sem1  Sem2       Sem3 GPA  Placed
0     Joe        K   9.8  10.0   9.900000  A+       1
1  Rajesh        M   8.9   9.1   9.300000   A       1
2  Kissan        V   9.9   9.8  10.000000   A       0
3    Mary        N   7.7   8.0   8.992857   B       0
4    Jeen        K   9.8   9.1   9.900000  A+       1
```

```
dfn.columns = ['Student_name', 'Mr/Ms', '1stsem', '2ndsem','3rdsem',
'F_Outcome','Placed?']
dfn.head()
```

```
  Student_name Mr/Ms  1stsem  2ndsem     3rdsem F_Outcome  Placed?
0          Joe     K     9.8    10.0   9.900000        A+        1
1       Rajesh     M     8.9     9.1   9.300000         A        1
2       Kissan     V     9.9     9.8  10.000000         A        0
3         Mary     N     7.7     8.0   8.992857         B        0
4         Jeen     K     9.8     9.1   9.900000        A+        1
```

```
dfn['Avg_score'] = (dfn['1stsem']+dfn['2ndsem']+dfn['3rdsem'])/3
dfn.head()
```

```
  Student_name Mr/Ms  1stsem  2ndsem     3rdsem F_Outcome  Placed?
Avg_score
0          Joe     K     9.8    10.0   9.900000        A+        1
9.900000
1       Rajesh     M     8.9     9.1   9.300000         A        1
9.100000
```

```
2       Kissan     V     9.9     9.8  10.000000          A         0
9.900000
3         Mary     N     7.7     8.0   8.992857          B         0
8.230952
4         Jeen     K     9.8     9.1   9.900000         A+         1
9.600000
```

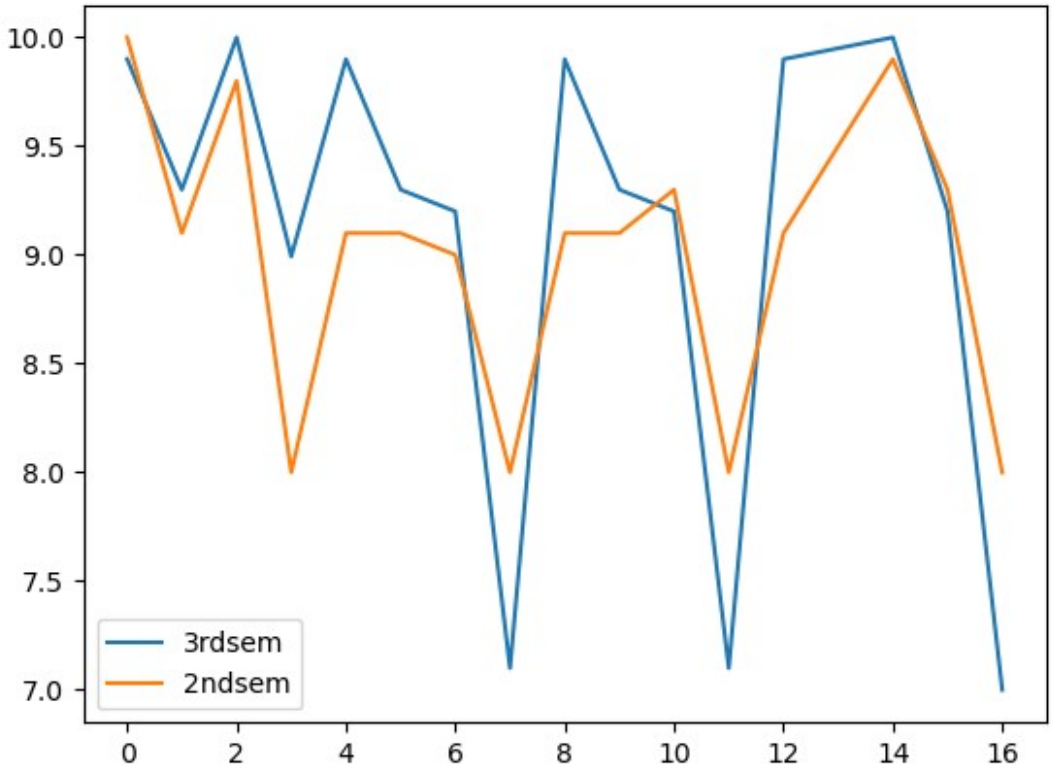
```
dfn.groupby('Placed?').min()
```

```
        Student_name Mr/Ms  1stsem  2ndsem  3rdsem F_Outcome
Avg_score
Placed?

0               Kiran     M     7.7     8.0     7.0         A
7.566667
1             Hassan     K     7.7     8.0     7.1         A
7.600000
```

```
dfn[['3rdsem', '2ndsem']].plot.line()
```

```
<Axes: >
```




```
#customize the color
#df[['BMI', 'Glucose']].plot.line(figsize=(20, 10), color={"BMI":
"red", "Glucose": "blue"})
```

```
dfn.plot.line(subplots=True)
```

```
array([<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >],
dtype=object)
```