

# ASSIGNMENT NO :- 2

## 1)Creating Pandas dataframe using

### A) Two-dimensional list

```
import pandas as pd
d=[['Aniket',25],['Rohit',30],['Suhas',35]]
df = pd.DataFrame(d, columns=['Name','Age'])
print(df)
```

#### OUTPUT:-

	Name	Age
0	Aniket	25
1	Rohit	30
2	Suhas	35

### B) dict of narray

```
import pandas as pd
import numpy as np
d={'Name': np.array(['Rohit', 'Aniket']),
  'Age': np.array([25, 30])}
df = pd.DataFrame(d)
print(df)
```

#### OUTPUT:-

	Name	Age
0	Rohit	25
1	Aniket	30

### C) List of lists

```
import pandas as pd
d=[['Aniket', 'Rohit'],[25, 30]]
df = pd.DataFrame(d,columns=['Name','Age'])
print(df)
```

**OUTPUT:-**

	Name	Age
0	Aniket	Rohit
1	25	30

### D) List of tuples

```
import pandas as pd
d=[('Aniket', 'Rohit'),(25, 30)]
df = pd.DataFrame(d,columns=['Name','Age'])
print(df)
```

**OUTPUT:-**

	Name	Age
--	------	-----

```
0 Aniket Rohit
```

```
1 25 30
```

## E) List of Dicts

```
import pandas as pd
```

```
d= [{'Name': 'Aniket', 'Age': 25}, {'Name': 'Rohit', 'Age': 30}]
```

```
df = pd.DataFrame(d)
```

```
print(df)
```

**OUTPUT:-**

```
   Name  Age
```

```
0 Aniket  25
```

```
1  Rohit  30
```

## F) List of nested dictionary

```
data = {'X': {'A': 1, 'B': 2}, 'Y': {'A': 3, 'B': 4}}
```

```
d= pd.DataFrame(data)
```

```
print(d)
```

**OUTPUT:-**

```
   X  Y
```

```
A  1  3
```

```
B  2  4
```

## G) Pandas series

```
import pandas as pd  
s1= pd.Series(['Aniket', 'Rohit'], name='Name')  
s2= pd.Series([25, 30], name='Age')  
df = pd.DataFrame({'Name': s1, 'Age': s2})  
print(df)
```

**OUTPUT:-**

```
   Name  Age  
0  Aniket  25  
1   Rohit  30
```

## H) String data

```
import pandas as pd  
d={'col1': ['a', 'b', 'c'], 'col2': ['x', 'y', 'z']}  
df= pd.DataFrame(d)  
print(df)
```

**OUTPUT:-**

```
col1 col2  
0    a    x  
1    b    y  
2    c    z
```

## 2) Reindexing in Pandas DataFrame

```
import pandas as pd
df=pd.DataFrame({'A':[1,2,3],'B':[4,5,6]})
ri=df.reindex([2, 0, 1])
print("\nReindexed DataFrame:\n",ri)
```

### OUTPUT:-

Reindexed DataFrame:

	A	B
2	3	6
0	1	4
1	2	5

## 3) Add or replace some of the values of the dataframe with some defined external values.

```
import pandas as pd
df=pd.DataFrame({'A': [1, 2, 3],'B': [4, 5, 6],'C': [7, 8, 9]})
ext_val={2: 10,5: 50,8: 100}
df.replace(ext_val,inplace=True)
print(df)
```

### OUTPUT:-

	A	B	C
--	---	---	---

```
0  1  4  7
1 10 50 100
2  3  6  9
```

#### 4) Reset Index in Pandas Dataframe

```
import pandas as pd
df=pd.DataFrame({'A': [1, 2, 3],'B': [4, 5, 6]},index=['x', 'y', 'z'])
r=df.reset_index()
print("DataFrame after resetting index:\n",r)
```

#### OUTPUT:-

DataFrame after resetting index:

```
index A B
0  x  1  4
1  y  2  5
2  z  3  6
```

#### 5) Setting custom Index to column and row of DataFrame

```
import pandas as pd
df = pd.DataFrame({'Name': ['Alice', 'Bob'],
                  'Age': [25, 30],
                  'City': ['New York', 'Los Angeles']})
df.set_index('Name', inplace=True)
df.columns = ['Age','City']
print(df)
```

## OUTPUT:-

	Age	City
Name		
Alice	25	New York
Bob	30	Los Angeles

## 6) Sorting a Dataframe by Index

```
import pandas as pd
df = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6]}, index=['b', 'a', 'c'])
print("Original DataFrame:\n",df)
sort= df.sort_index()
print("\nDataFrame sorted by index:\n",sort)
```

## OUTPUT:-

Original DataFrame:

	A	B
b	1	4
a	2	5
c	3	6

DataFrame sorted by index:

	A	B
a	2	5
b	1	4
c	3	6

## 7) Sorting a Dataframe by Multiple Columns

```
import pandas as pd

df = pd.DataFrame({
    'Name': ['Aniket', 'Rohit', 'Suhas', 'Pratik'],
    'Age': [24, 30, 22, 35],
    'Score': [85, 90, 88, 90]})

print("Original DataFrame:\n",df)

sort=df.sort_values(by=['Age', 'Score'])

print("\nDataFrame sorted by 'Age' and 'Score':\n",sort)
```

### OUTPUT:-

Original DataFrame:

	Name	Age	Score
0	Aniket	24	85
1	Rohit	30	90
2	Suhas	22	88
3	Pratik	35	90

DataFrame sorted by 'Age' and 'Score':

	Name	Age	Score
2	Suhas	22	88
0	Aniket	24	85
1	Rohit	30	90
3	Pratik	35	90

## 8) To iterate over rows in Pandas Dataframe



```
import pandas as pd

df = pd.DataFrame({
    'Name': ['Alice', 'Bob', 'Charlie'],
    'Age': [24, 30, 22],
    'Score': [85, 90, 88]})

for index, row in df.iterrows():
    print(f"Index: {index}, Name: {row['Name']}, Age: {row['Age']}, Score: {row['Score']}")
```

### OUTPUT:-

Index: 0, Name: Alice, Age: 24, Score: 85

Index: 1, Name: Bob, Age: 30, Score: 90

Index: 2, Name: Charlie, Age: 22, Score: 88

## 9) To import only specified columns from a csv file

```
import pandas as pd

d=pd.read_csv("D:\\nba.csv",index_col="Name")
f=d["Age"]
print(f)
```

### OUTPUT:-

Name	Age
Aniket	18
Rohit	18
Suhas	18
Pratik	18
Yash	18

Vikki 18  
Sanket 18  
Harshad 19  
Onkar 18  
Name: Age, dtype: int64

## 10) Select any row from a Dataframe using `iloc[]` and `iat[]` in Pandas

```
import pandas as pd
d=pd.DataFrame({
    'Name': ['Aniket', 'Rohit', 'Suhas', 'Pratik'],
    'Age': [24, 30, 22, 35],
    'Score': [85, 90, 88, 90]})
print("Original DataFrame:\n",d)
iloc=d.iloc[2]
print("\nRow selected using iloc:\n",iloc)
iat=d.iat[2, 1]
print("\nValue selected using iat:\n",iat)
```

### OUTPUT:-

Original DataFrame:

	Name	Age	Score
0	Aniket	24	85
1	Rohit	30	90
2	Suhas	22	88
3	Pratik	35	90

Row selected using `iloc`:

```
Name    Suhas
Age      22
Score    88
Name: 2, dtype: object
```

Value selected using iat:

22

11) To drop those rows in which specific columns have missing values.

```
import pandas as pd
d=pd.DataFrame({
    'Name': ['Aniket', 'Rohit', 'Suhas',None],
    'Age': [24, None , 22, 35],
    'Score': [85, 90, None, 90]})
print("Original DataFrame:\n",d)
dr=d.dropna(subset=['Name','Age'])
print("\nDrop values of Missing Values:\n",dr)
```

## OUTPUT:-

Original DataFrame:

```
   Name  Age  Score
0 Aniket 24.0  85.0
1  Rohit  NaN  90.0
2  Suhas 22.0   NaN
3   None 35.0  90.0
```

Drop values of Missing Values:

	Name	Age	Score
0	Aniket	24.0	85.0
2	Suhas	22.0	NaN

## 12) Insert row at given position in Pandas Dataframe

```
import pandas as pd

df = pd.DataFrame({
    'Name': ['Alice', 'Bob'],
    'Age': [24, 30],
    'Score': [85, 90]})

print("Original DataFrame:\n",df)

r={'Name': 'David', 'Age': 35, 'Score': 92}

position=2

df.loc[position]=r

df=df.sort_index()

print("\nDataFrame after inserting a new row at position 1:\n",df)
```

### OUTPUT:-

Original DataFrame:

	Name	Age	Score
0	Alice	24	85
1	Bob	30	90

DataFrame after inserting a new row at position 1:

	Name	Age	Score
0	Alice	24	85
1	Bob	30	90

### 13) Create a list from rows in Pandas dataframe

```
import pandas as pd

df = pd.DataFrame({
    'Name': ['Alice', 'Bob', 'Charlie'],
    'Age': [24, 30, 22],
    'Score': [85, 90, 88]})

print("Original DataFrame:\n",df)
l= df.values.tolist()
print("\nList is:\n",l)
```

#### OUTPUT:-

Original DataFrame:

	Name	Age	Score
0	Alice	24	85
1	Bob	30	90
2	Charlie	22	88

List is:

```
[['Alice', 24, 85], ['Bob', 30, 90], ['Charlie', 22, 88]]
```

### 14) Ranking Rows of DataFrame

```
import pandas as pd
```

```
d=pd.DataFrame({"name":["aniket","rohit","suhas","pratik"],"age":[25,35,35,48],'salary':[10000,96000,54000,52000]})
d['rank']=d['age'].rank(method='average')
print(d)
```

### OUTPUT:-

```
   name age salary rank
0 aniket  25  10000  1.0
1 rohit   35  96000  2.5
2 suhas   35  54000  2.5
3 pratik  48  52000  4.0
```

## 15)Sorting rows in DataFrame

```
import pandas as pd
d=pd.DataFrame({"name":["aniket","rohit"],"age":[25,50],'salary':[10000,96000]})
sd=d.sort_values(by='age', ascending=False)
print(sd)
```

### OUTPUT:-

```
   name age salary
1 rohit  50  96000
0 aniket  25  10000
```

## 16)Select row with maximum and minimum value in dataframe

```
import pandas as pd
```

```

d=pd.DataFrame({
    'Name': ['Aniket', 'Rohit', 'Suhas'],
    'Age': [24, 30, 22],
    'Score': [85, 90, 80]})
print("Original DataFrame:\n",d)
max= d[d['Score'] == d['Score'].max()]
min= d[d['Score'] == d['Score'].min()]
print("\nRow with maximum value in 'Score':\n",max)
print("\nRow with minimum value in 'Score':\n",min)

```

## OUTPUT:-

Original DataFrame:

	Name	Age	Score
0	Aniket	24	85
1	Rohit	30	90
2	Suhas	22	80

Row with maximum value in 'Score':

	Name	Age	Score
1	Rohit	30	90

Row with minimum value in 'Score':

	Name	Age	Score
2	Suhas	22	80

## 17)Convert a column to row name/index in Pandas

```
import pandas as pd
```

```

d=pd.DataFrame({
    'Name': ['Aniket', 'Rohit', 'Suhas'],
    'Age': [24, 30, 22],
    'Score': [85, 90, 80]})

print("Original DataFrame:\n",d)
d.set_index('Name', inplace=True)
print("\nDataFrame with 'Name' as the index:\n",d)

```

## OUTPUT:-

Original DataFrame:

	Name	Age	Score
0	Aniket	24	85
1	Rohit	30	90
2	Suhas	22	80

DataFrame with 'Name' as the index:

	Age	Score
Name		
Aniket	24	85
Rohit	30	90
Suhas	22	80

## 18)Randomly select rows from DataFrame

```

import pandas as pd
d=pd.DataFrame({
    'Name': ['Aniket', 'Rohit', 'Suhas'],
    'Age': [24, 30, 22],

```



```
'Score': [85, 90, 80]})  
rr= d.sample(n=2, random_state=42)  
print(rr)
```

### OUTPUT:-

```
   Name  Age  Score  
0  Aniket  24    85  
1   Rohit  30    90
```

## 19)To get column names in dataframe

```
import pandas as pd  
d=pd.DataFrame({  
    'Name': ['Aniket', 'Rohit', 'Suhas'],  
    'Age': [24, 30, 22],  
    'Score': [85, 90, 80]})  
c=d.columns.tolist()  
print(c)
```

### OUTPUT:-

```
['Name', 'Age', 'Score']
```

## 20)To rename a specific columns in a dataframe

```
import pandas as pd  
d=pd.DataFrame({
```

```

'Name': ['Aniket', 'Rohit', 'Suhas'],
'Age': [24, 30, 22],
'Score': [85, 90, 80]})
print("Original DataFrame:\n",d)
d.rename(columns={'Score': 'Test Score'}, inplace=True)
print("\nDataFrame after renaming columns:\n",d)

```

## OUTPUT:-

Original DataFrame:

	Name	Age	Score
0	Aniket	24	85
1	Rohit	30	90
2	Suhas	22	80

DataFrame after renaming columns:

	Name	Age	Test Score
0	Aniket	24	85
1	Rohit	30	90
2	Suhas	22	80

## 21)Get unique values from a column in DataFrame

```

import pandas as pd
d=pd.DataFrame({
    'Name': ['Aniket', 'Rohit', 'Suhas'],
    'Age': [24, 30, 22],
    'Score': [85, 90, 80]})
u=d['Name'].unique()

```

```
print("Unique values is:\n",u)
```

### OUTPUT:-

Unique values is:

```
['Aniket' 'Rohit' 'Suhas']
```

22) To count, find and replace missing values with 0 from DataFrame.

```
import pandas as pd
```

```
import numpy as np
```

```
df = pd.DataFrame({  
    'Name': ['Aniket', 'Rohit', np.nan, 'Pratik'],  
    'Age': [24, np.nan, 22, np.nan],  
    'Score': [85, 90, 88, np.nan]})  
cnt= df.isnull().sum()  
print("\nCount of missing values in each column:\n",cnt)  
df.fillna(0, inplace=True)  
print("\nDataFrame after replacing missing values with 0:\n",df)
```

### OUTPUT:-

Count of missing values in each column:

```
Name    1
```

```
Age      2
```

```
Score    1
```

```
dtype: int64
```

DataFrame after replacing missing values with 0:

	Name	Age	Score
0	Aniket	24.0	85.0
1	Rohit	0.0	90.0
2	0	22.0	88.0
3	Pratik	0.0	0.0

23)To identify the columns of a given DataFrame which have at least one missing value.

```
import pandas as pd
import numpy as np
df = pd.DataFrame({
    'Name': ['Aniket', 'Rohit', np.nan, 'Pratik'],
    'Age': [24, np.nan, 22, np.nan],
    'Score': [85, 90, 88, np.nan]})
missing_col= df.columns[df.isnull().any()].tolist()
print("Columns missing value:")
print(missing_col)
```

**OUTPUT:-**

```
Columns missing value:
['Name', 'Age', 'Score']
```

24)To join two dataframes along rows and merge with common records.

```
import pandas as pd
```

```

df1 = pd.DataFrame({'ID': [1, 2, 3, 4], 'Name': ['A', 'B', 'C', 'D']})
df2 = pd.DataFrame({'ID': [3, 4, 5, 6], 'Name': ['C', 'D', 'E', 'F']})
cd= pd.concat([df1, df2], axis=0, ignore_index=True)
md= pd.merge(df1, df2, on='ID', how='inner', suffixes=('_df1', '_df2'))
print("Concatenated DataFrame:\n",cd)
print("\nMerged DataFrame:\n",md)

```

## OUTPUT:-

Concatenated DataFrame:

	ID	Name
0	1	A
1	2	B
2	3	C
3	4	D
4	3	C
5	4	D
6	5	E
7	6	F

Merged DataFrame:

	ID	Name_df1	Name_df2
0	3	C	C
1	4	D	D

## 25)To add rows to an existing DataFrame

```

import pandas as pd
df = pd.DataFrame({'ID': [1, 2, 3],

```

```

        'Name': ['Aniket', 'Rohit', 'Suhas'],
        'Age': [24, 30, 22])

new_row = pd.DataFrame({'ID': [4], 'Name': ['Pratik'], 'Age': [35]})
df = pd.concat([df, new_row], ignore_index=True)
print("DataFrame after adding a row using append:\n",df)

```

## OUTPUT:-

DataFrame after adding a row using append:

```

   ID  Name  Age
0  1  Aniket  24
1  2   Rohit  30
2  3   Suhas  22
3  4   Pratik  35

```

## 26)Getting frequency counts of a columns in Pandas DataFrame

```

import pandas as pd

idx=pd.Index(['Harry','Mick','Arther','Nick','Harry','Arther'],name='student')
print(idx.value_counts())

```

## OUTPUT:-

```

student
Harry    2
Arther    2
Mick      1
Nick      1
Name: count, dtype: int64

```

## 27) Split a text column into two columns in Pandas DataFrame

```
import pandas as pd

df= pd.DataFrame({'col': ['Alice,24', 'Bob,27', 'Charlie,22']})

df[['Name', 'Age']] = df['col'].str.split(',', expand=True)

print("\nDataFrame After Splitting Column:\n",df)
```

### OUTPUT:-

DataFrame After Splitting Column:

	col	Name	Age
0	Alice,24	Alice	24
1	Bob,27	Bob	27
2	Charlie,22	Charlie	22

## 28) Get the index of minimum value in DataFrame column

```
import pandas as pd

df=pd.DataFrame({
    'Name': ['Aniket', 'Rohit', 'Suhas', 'Pratik'],
    'Age': [24, 30, 22, 35]})

m=df['Age'].idxmin()

print("Index of the minimum value in the 'Age' column:",m)
```

### OUTPUT:-

Index of the minimum value in the 'Age' column: 2

## 29)Get the index of maximum value in DataFrame column

```
import pandas as pd
df=pd.DataFrame({
    'Name': ['Aniket', 'Rohit', 'Suhas', 'Pratik'],
    'Age': [24, 30, 22, 35]})
m=df['Age'].idxmax()
print("Index of the maximum value in the 'Age' column:",m)
```

### OUTPUT:-

Index of the maximum value in the 'Age' column: 3

## 30)Difference of two columns in Pandas dataframe

```
import pandas as pd
df = pd.DataFrame({'c1': [10, 20, 30, 40],
                   'c2': [5, 15, 25, 35]})
df['dif'] = df['c1'] - df['c2']
print(df)
```

### OUTPUT:-

	c1	c2	dif
0	10	5	5
1	20	15	5
2	30	25	5



3 40 35 5

### 31)Get n-largest values from a particular column in Pandas DataFrame

```
import pandas as pd
df=pd.DataFrame({
    'Name': ['Aniket', 'Rohit', 'Suhas', 'Pratik'],
    'Score': [85, 92, 88, 95]})
l=df.nlargest(2, 'Score')
print(l)
```

**OUTPUT:-**

	Name	Score
3	Pratik	95
1	Rohit	92

### 32)Get n-smallest values from a particular column in Pandas DataFrame

```
import pandas as pd
df=pd.DataFrame({
    'Name': ['Aniket', 'Rohit', 'Suhas', 'Pratik'],
    'Score': [85, 92, 88, 95]})
s=df.nsmallest(2, 'Score')
print(s)
```

**OUTPUT:-**

	Name	Score
0	Aniket	85
2	Suhas	88

### 33) Drop one or multiple columns in Pandas Dataframe

```
import pandas as pd
df = pd.DataFrame({
    'A': [1, 2, 3],
    'B': [4, 5, 6],
    'C': [7, 8, 9]})
ds= df.drop('B', axis=1)
dm= df.drop(['A', 'C'], axis=1)
print("Drop single column:\n",ds)
print("\nDrop multiple columns:\n",dm)
```

#### OUTPUT:-

Drop single column:

	A	C
0	1	7
1	2	8
2	3	9

Drop multiple columns:

	B
0	4
1	5

### 34)Capitalize first letter of a column in Pandas dataframe

```
import pandas as pd
df = pd.DataFrame({'Name': ['aniket', 'rohit', 'suhas', 'pratik']})
df=df['Name'].str.capitalize()
print(df)
```

#### OUTPUT:-

```
0  Aniket
1  Rohit
2  Suhas
3  Pratik
Name: Name, dtype: object
```

### 35)To import excel data into a Pandas dataframe and find a list of employees where hire\_date between JAN-2013 to DEC-2020.

```
import pandas as pd
d= pd.read_excel('D:\\employees.xlsx')
df= d[(d['HIRE_DATE'] >= '2013-01-01') & (d['HIRE_DATE'] <= '2020-12-31')]
print(df)
```

#### OUTPUT:-

```
IMP_ID  NAME  SALARY DEPARTMENT HIRE_DATE
```

0	101.0	ROHIT	120412.0	ADMIN	2013-01-01
3	104.0	PRATIK	84947.0	SELS	2013-01-01
6	107.0	VIKKI	453567.0	ADMIN	2013-01-01