

```
In [1]: import pandas as pd
```

```
In [3]: #read the files #step=1 #using file path
file_path=r"C:\Users\Mrityunjay\Desktop\Data science naresh it\Self practice\bank.csv"
bank_df=pd.read_csv(file_path,sep=";")
```

Out[3]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	poutcome	previous	emp.var.rate	unemployment	avg.balance	avg.poutcome	total.cnt.poutcome
0	30	unemployed	married	primary	no	1787	no	no	cellular	...	...	...	...	...	...	...	...	...	...	
1	33	services	married	secondary	no	4789	yes	yes	cellular	...	...	...	...	...	...	...	...	...	...	
2	35	management	single	tertiary	no	1350	yes	no	cellular	...	...	...	...	...	...	...	...	...	...	
3	30	management	married	tertiary	no	1476	yes	yes	unknown	...	...	...	...	...	...	...	...	...	...	
4	59	blue-collar	married	secondary	no	0	yes	no	unknown	...	...	...	...	...	...	...	...	...	...	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4516	33	services	married	secondary	no	-333	yes	no	cellular	...	...	...	...	...	...	...	...	...	...	
4517	57	self-employed	married	tertiary	yes	-3313	yes	yes	unknown	...	...	...	...	...	...	...	...	...	...	
4518	57	technician	married	secondary	no	295	no	no	cellular	...	...	...	...	...	...	...	...	...	...	
4519	28	blue-collar	married	secondary	no	1137	no	no	cellular	...	...	...	...	...	...	...	...	...	...	
4520	44	entrepreneur	single	tertiary	no	1136	yes	yes	cellular	...	...	...	...	...	...	...	...	...	...	

4521 rows × 17 columns



```
In [5]: #step=2 #using filename
bank_df=pd.read_csv("bank.csv",sep=";")
bank_df
```

Out[5]:

	age	job	marital	education	default	balance	housing	loan	contact	day
0	30	unemployed	married	primary	no	1787	no	no	cellular	1
1	33	services	married	secondary	no	4789	yes	yes	cellular	11
2	35	management	single	tertiary	no	1350	yes	no	cellular	16
3	30	management	married	tertiary	no	1476	yes	yes	unknown	3
4	59	blue-collar	married	secondary	no	0	yes	no	unknown	5
...	...	...	...	...	...	...	...	...	...	...
4516	33	services	married	secondary	no	-333	yes	no	cellular	1
4517	57	self-employed	married	tertiary	yes	-3313	yes	yes	unknown	11
4518	57	technician	married	secondary	no	295	no	no	cellular	16
4519	28	blue-collar	married	secondary	no	1137	no	no	cellular	3
4520	44	entrepreneur	single	tertiary	no	1136	yes	yes	cellular	5

4521 rows × 17 columns



In [9]: `#checking the datasets  
bank_df.head() #head of datasets`

Out[9]:

	age	job	marital	education	default	balance	housing	loan	contact	day
0	30	unemployed	married	primary	no	1787	no	no	cellular	19
1	33	services	married	secondary	no	4789	yes	yes	cellular	11
2	35	management	single	tertiary	no	1350	yes	no	cellular	16
3	30	management	married	tertiary	no	1476	yes	yes	unknown	3
4	59	blue-collar	married	secondary	no	0	yes	no	unknown	5



In [30]: `bank_df.head(10) #head of datasets providing numbers`

Out[30]:

	age	job	marital	education	default	balance	housing	loan	contact	day
0	30	unemployed	married	primary	no	1787	no	no	cellular	19
1	33	services	married	secondary	no	4789	yes	yes	cellular	11
2	35	management	single	tertiary	no	1350	yes	no	cellular	16
3	30	management	married	tertiary	no	1476	yes	yes	unknown	3
4	59	blue-collar	married	secondary	no	0	yes	no	unknown	5
5	35	management	single	tertiary	no	747	no	no	cellular	23
6	36	self-employed	married	tertiary	no	307	yes	no	cellular	14
7	39	technician	married	secondary	no	147	yes	no	cellular	6
8	41	entrepreneur	married	tertiary	no	221	yes	no	unknown	14
9	43	services	married	primary	no	-88	yes	yes	cellular	17

◀ ▶

In [38]: #tail of datasets  
bank\_df.tail()

Out[38]:

	age	job	marital	education	default	balance	housing	loan	contact	da
4516	33	services	married	secondary	no	-333	yes	no	cellular	3
4517	57	self-employed	married	tertiary	yes	-3313	yes	yes	unknown	1
4518	57	technician	married	secondary	no	295	no	no	cellular	1
4519	28	blue-collar	married	secondary	no	1137	no	no	cellular	1
4520	44	entrepreneur	single	tertiary	no	1136	yes	yes	cellular	1

◀ ▶

In [40]: #tail of datasets prviding number  
bank\_df.tail(10)

Out[40]:

	age	job	marital	education	default	balance	housing	loan	contact	da
4511	46	blue-collar	married	secondary	no	668	yes	no	unknown	
4512	40	blue-collar	married	secondary	no	1100	yes	no	unknown	
4513	49	blue-collar	married	secondary	no	322	no	no	cellular	
4514	38	blue-collar	married	secondary	no	1205	yes	no	cellular	
4515	32	services	single	secondary	no	473	yes	no	cellular	
4516	33	services	married	secondary	no	-333	yes	no	cellular	
4517	57	self-employed	married	tertiary	yes	-3313	yes	yes	unknown	
4518	57	technician	married	secondary	no	295	no	no	cellular	
4519	28	blue-collar	married	secondary	no	1137	no	no	cellular	
4520	44	entrepreneur	single	tertiary	no	1136	yes	yes	cellular	

◀ ▶

In [52]: #shape of datasets  
bank\_df.shape

Out[52]: (4521, 17)

In [54]: #rows in datasets  
bank\_df.shape[0]

Out[54]: 4521

In [56]: #columns in datasets  
bank\_df.shape[1]

Out[56]: 17

In [58]: #size of datasets  
bank\_df.shape[0] \*bank\_df.shape[1] #using rows and columns counts

Out[58]: 76857

In [60]: #size of datasets  
bank\_df.size #using method

Out[60]: 76857

In [62]: #checking columns in datasets  
bank\_df.columns

```
Out[62]: Index(['age', 'job', 'marital', 'education', 'default', 'balance', 'housing',
   'loan', 'contact', 'day', 'month', 'duration', 'campaign', 'pdays',
   'previous', 'poutcome', 'y'],
  dtype='object')
```

```
In [64]: #columns datatypes
bank_df.dtypes
```

```
Out[64]: age           int64
          job            object
          marital         object
          education       object
          default          object
          balance          int64
          housing          object
          loan             object
          contact          object
          day              int64
          month            object
          duration         int64
          campaign         int64
          pdays            int64
          previous          int64
          poutcome          object
          y                 object
          dtype: object
```

```
In [71]: #information of datasets
bank_df.info()      #columns,datatype,nullvalue,columns count..etc.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4521 entries, 0 to 4520
Data columns (total 17 columns):
 #   Column    Non-Null Count  Dtype  
 --- 
 0   age        4521 non-null   int64  
 1   job        4521 non-null   object  
 2   marital    4521 non-null   object  
 3   education  4521 non-null   object  
 4   default    4521 non-null   object  
 5   balance    4521 non-null   int64  
 6   housing    4521 non-null   object  
 7   loan       4521 non-null   object  
 8   contact    4521 non-null   object  
 9   day        4521 non-null   int64  
 10  month      4521 non-null   object  
 11  duration   4521 non-null   int64  
 12  campaign   4521 non-null   int64  
 13  pdays      4521 non-null   int64  
 14  previous   4521 non-null   int64  
 15  poutcome   4521 non-null   object  
 16  y          4521 non-null   object  
dtypes: int64(7), object(10)
memory usage: 600.6+ KB
```

In [90]: `#isnull(), isna(), isnull().sum()  
bank_df.isnull() #null value in columns get trues`

Out[90]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month
0	False	False	False	False	False	False	False	False	False	False	F
1	False	False	False	False	False	False	False	False	False	False	F
2	False	False	False	False	False	False	False	False	False	False	F
3	False	False	False	False	False	False	False	False	False	False	F
4	False	False	False	False	False	False	False	False	False	False	F
...	...	...	...	...	...	...	...	...	...	...	...
4516	False	False	False	False	False	False	False	False	False	False	F
4517	False	False	False	False	False	False	False	False	False	False	F
4518	False	False	False	False	False	False	False	False	False	False	F
4519	False	False	False	False	False	False	False	False	False	False	F
4520	False	False	False	False	False	False	False	False	False	False	F

4521 rows × 17 columns



In [88]: `bank_df.isnull().sum() #sum of all null-box in each columns`

Out[88]:

age	0
job	0
marital	0
education	0
default	0
balance	0
housing	0
loan	0
contact	0
day	0
month	0
duration	0
campaign	0
pdays	0
previous	0
poutcome	0
y	0
dtype:	int64

In [82]: `bank_df.isna() #null value in columns get trues`

Out[82]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month
0	False	False	False	False	False	False	False	False	False	False	F
1	False	False	False	False	False	False	False	False	False	False	F
2	False	False	False	False	False	False	False	False	False	False	F
3	False	False	False	False	False	False	False	False	False	False	F
4	False	False	False	False	False	False	False	False	False	False	F
...	...	...	...	...	...	...	...	...	...	...	...
4516	False	False	False	False	False	False	False	False	False	False	F
4517	False	False	False	False	False	False	False	False	False	False	F
4518	False	False	False	False	False	False	False	False	False	False	F
4519	False	False	False	False	False	False	False	False	False	False	F
4520	False	False	False	False	False	False	False	False	False	False	F

4521 rows × 17 columns



In [97]:

```
#isnull().count(), isna().count()
bank_df.isnull().count() #if not null count it : false.
```

Out[97]:

age	4521
job	4521
marital	4521
education	4521
default	4521
balance	4521
housing	4521
loan	4521
contact	4521
day	4521
month	4521
duration	4521
campaign	4521
pdays	4521
previous	4521
poutcome	4521
y	4521
dtype: int64	

In [101...]

```
bank_df.isna().count() #if not null count it : false.
```

```
Out[101... age      4521
job      4521
marital   4521
education 4521
default    4521
balance    4521
housing    4521
loan      4521
contact    4521
day       4521
month     4521
duration   4521
campaign   4521
pdays     4521
previous   4521
poutcome   4521
y         4521
dtype: int64
```

```
In [197... bank_df.columns
```

```
Out[197... Index(['age', 'job', 'marital', 'education', 'default', 'balance', 'housing',
      'loan', 'contact', 'day', 'month', 'duration', 'campaign', 'pdays',
      'previous', 'poutcome', 'y'],
      dtype='object')
```

```
In [109... #take method
bank_df.take([100,200,250]) #by-default indexing rows
```

	age	job	marital	education	default	balance	housing	loan	contact	day
<b>100</b>	36	blue-collar	married	secondary	no	0	yes	no	unknown	6
<b>200</b>	34	technician	single	tertiary	no	992	yes	no	cellular	2
<b>250</b>	31	management	married	tertiary	no	395	yes	no	cellular	5



```
In [121... #rows indexing
bank_df.take([140,230,240],axis=0)
```

	age	job	marital	education	default	balance	housing	loan	contact	day
<b>140</b>	32	management	divorced	tertiary	no	948	no	no	cellular	12
<b>230</b>	52	unknown	married	secondary	no	142	no	no	cellular	25
<b>240</b>	34	admin.	single	secondary	no	1	yes	no	cellular	22



```
In [119... #columns indexing
bank_df.take([1,3,5],axis=1)
```

Out[119...]

	job	education	balance
<b>0</b>	unemployed	primary	1787
<b>1</b>	services	secondary	4789
<b>2</b>	management	tertiary	1350
<b>3</b>	management	tertiary	1476
<b>4</b>	blue-collar	secondary	0
...	...	...	...
<b>4516</b>	services	secondary	-333
<b>4517</b>	self-employed	tertiary	-3313
<b>4518</b>	technician	secondary	295
<b>4519</b>	blue-collar	secondary	1137
<b>4520</b>	entrepreneur	tertiary	1136

4521 rows × 3 columns

In [125...]

```
#rows and columns
bank_df.take([12,100,130],axis=0).take([2,3,6],axis=1)
```

Out[125...]

	marital	education	housing
<b>12</b>	married	tertiary	no
<b>100</b>	married	secondary	yes
<b>130</b>	married	primary	yes

In [133...]

```
#use iloc indexing method
bank_df.iloc[[2,100,250],[2,3,6]]
```

Out[133...]

	marital	education	housing
<b>2</b>	single	tertiary	yes
<b>100</b>	married	secondary	yes
<b>250</b>	married	tertiary	yes

In [137...]

```
bank_df.iloc[2:10,[2,3,6]]
```

Out[137...]

	marital	education	housing
2	single	tertiary	yes
3	married	tertiary	yes
4	married	secondary	yes
5	single	tertiary	no
6	married	tertiary	yes
7	married	secondary	yes
8	married	tertiary	yes
9	married	primary	yes

In [139...]

bank\_df.iloc[[2, 100, 250], 2:6]

Out[139...]

	marital	education	default	balance
2	single	tertiary	no	1350
100	married	secondary	no	0
250	married	tertiary	no	395

In [147...]

bank\_df.iloc[120:130, 2:6]

Out[147...]

	marital	education	default	balance
120	divorced	secondary	no	1890
121	married	primary	no	419
122	married	secondary	no	2693
123	married	secondary	no	-231
124	single	tertiary	yes	-55
125	single	tertiary	no	462
126	single	secondary	no	119
127	single	secondary	no	586
128	single	secondary	no	228
129	divorced	tertiary	no	4659

In [161...]

bank\_df.iloc[[2, 100, 250], :]

Out[161...]

	age	job	marital	education	default	balance	housing	loan	contact	day
2	35	management	single	tertiary	no	1350	yes	no	cellular	16
100	36	blue-collar	married	secondary	no	0	yes	no	unknown	6
250	31	management	married	tertiary	no	395	yes	no	cellular	9

◀ ▶

In [163...]

bank\_df.iloc[[2,100,250],3:]

Out[163...]

	education	default	balance	housing	loan	contact	day	month	duration	campaign
2	tertiary	no	1350	yes	no	cellular	16	apr	185	
100	secondary	no	0	yes	no	unknown	6	may	337	
250	tertiary	no	395	yes	no	cellular	9	feb	155	

◀ ▶

In [165...]

bank\_df.iloc[:,[2,5,8]]

Out[165...]

	marital	balance	contact
0	married	1787	cellular
1	married	4789	cellular
2	single	1350	cellular
3	married	1476	unknown
4	married	0	unknown
...	...	...	...
4516	married	-333	cellular
4517	married	-3313	unknown
4518	married	295	cellular
4519	married	1137	cellular
4520	single	1136	cellular

4521 rows × 3 columns

In [169...]

bank\_df.iloc[:,2:8]

Out[169...]

	marital	education	default	balance	housing	loan
<b>0</b>	married	primary	no	1787	no	no
<b>1</b>	married	secondary	no	4789	yes	yes
<b>2</b>	single	tertiary	no	1350	yes	no
<b>3</b>	married	tertiary	no	1476	yes	yes
<b>4</b>	married	secondary	no	0	yes	no
...	...	...	...	...	...	...
<b>4516</b>	married	secondary	no	-333	yes	no
<b>4517</b>	married	tertiary	yes	-3313	yes	yes
<b>4518</b>	married	secondary	no	295	no	no
<b>4519</b>	married	secondary	no	1137	no	no
<b>4520</b>	single	tertiary	no	1136	yes	yes

4521 rows × 6 columns

In [201...]

bank\_df.columns

Out[201...]

```
Index(['age', 'job', 'marital', 'education', 'default', 'balance', 'housing',
       'loan', 'contact', 'day', 'month', 'duration', 'campaign', 'pdays',
       'previous', 'poutcome', 'y'],
      dtype='object')
```

In [187...]

#using loc method

bank\_df.loc[:, ["age", "job"]]

Out[187...]

	age	job
<b>0</b>	30	unemployed
<b>1</b>	33	services
<b>2</b>	35	management
<b>3</b>	30	management
<b>4</b>	59	blue-collar
...	...	...
<b>4516</b>	33	services
<b>4517</b>	57	self-employed
<b>4518</b>	57	technician
<b>4519</b>	28	blue-collar
<b>4520</b>	44	entrepreneur

4521 rows × 2 columns

In [189...]

bank\_df.loc[100:105, ["age", "job"]]

Out[189...]

	age	job
<b>100</b>	36	blue-collar
<b>101</b>	38	management
<b>102</b>	49	technician
<b>103</b>	57	services
<b>104</b>	34	management
<b>105</b>	32	management

In [203...]

bank\_df.loc[100:105, ['age', 'job', 'marital', 'education', 'housing', 'loan', 'contact', 'campaign', 'pdays', 'previous', 'y']]

Out[203...]

	age	job	marital	education	housing	loan	contact	campaign	pdays	p
100	36	blue-collar	married	secondary		yes	no	unknown	1	-1
101	38	management	single	tertiary		yes	no	cellular	1	367
102	49	technician	married	secondary		no	no	telephone	2	-1
103	57	services	single	primary		yes	no	telephone	2	-1
104	34	management	single	secondary		no	no	unknown	1	-1
105	32	management	single	tertiary		yes	no	cellular	2	145



In [ ]:

- dict to data sets
- take dtypes
- convert into dictionary
- create two list
- create two dataframes

In [15]:

```
#create dataframe using lists
name=["ramesh","suresh","manish","mohit"]
goal=["doctor","engineer","accountant","CS"]
age=[20,18,20,21]
pd.DataFrame(zip(name,goal,age),columns=["name","goal","age"],index=[1,2,3,4])
```

Out[15]:

	name	goal	age
1	ramesh	doctor	20
2	suresh	engineer	18
3	manish	accountant	20
4	mohit	CS	21

In [19]:

```
#create dataframe using dictionary
dict1={"name":["ramesh","suresh","manish","mohit"],
       "goal":["doctor","engineer","accountant","CS"],
       "age":[20,18,20,21]}
pd.DataFrame(dict1)
```

	name	goal	age
0	ramesh	doctor	20
1	suresh	engineer	18
2	manish	accountant	20
3	mohit	CS	21

```
In [50]: #list to dictionary to dataframes
dict2={}
name=["ramesh","suresh","manish","mohit"]
goal=["doctor","engineer","accountant","CS"]
for key,value in zip(name,goal):
    dict2[key]=value
dict2
keys=[]
value=[]
for i,j in dict2.items():
    keys.append(i)
    value.append(j)
pd.DataFrame(zip(keys,value),columns=["name","goals"])
```

	name	goals
0	ramesh	doctor
1	suresh	engineer
2	manish	accountant
3	mohit	CS

```
In [66]: #tuple list to dataframe
tuple_list=[("manish","engineer"),("mrityunjay","manager"),("ashish","CEO"),("mohit",
dict1={}
for i in tuple_list:
    dict1[i[0]]=i[1]
name=[]
post=[]
for i,j in dict1.items():
    name.append(i)
    post.append(j)
pd.DataFrame(zip(name,post),columns=["name","post"])
```

	name	post
0	manish	engineer
1	mrityunjay	manager
2	ashish	CEO
3	mohit	doctor

In [ ]: