

```
import pandas as pd
df=pd.read_csv("Churn_Modelling.csv")
```

```
-----
FileNotFoundError                                Traceback (most recent call last)
<ipython-input-8-1888dddf0a4f> in <cell line: 2>()
      1 import pandas as pd
----> 2 df=pd.read_csv("Churn_Modelling.csv")
```

4 frames

```
/usr/local/lib/python3.10/dist-packages/pandas/io/common.py in
get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text, errors,
storage_options)
    857         if ioargs.encoding and "b" not in ioargs.mode:
    858             # Encoding
--> 859             handle = open(
    860                 handle,
    861                 ioargs.mode,
```

```
FileNotFoundError: [Errno 2] No such file or directory: 'Churn_Modelling.csv'
```

```
import pandas as pd
df=pd.read_csv('Churn_Modelling.csv')
df.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance
0	1	15634602	Hargrave	619	France	Female	42	2	
1	2	15647311	Hill	608	Spain	Female	41	1	8380
2	3	15619304	Onio	502	France	Female	42	8	15966
3	4	15701354	Boni	699	France	Female	39	1	
4	5	15737888	Mitchell	850	Spain	Female	43	2	12551

Next steps: [Generate code with df](#) [View recommended plots](#)

```
df.drop(['Age'],axis=1)
#axis=1 means the age column still exist to perm delete inplace=true
df.drop(['Age'],axis=1,inplace=True)#to delete perm
```

```
#to add elements to the dataset add new columns as newcredutscore
df['New_CreditScore']=df['CreditScore']+10
df.head(10)#to print the table
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Tenure	Balance
0	1	15634602	Hargrave	619	France	Female	2	0.00
1	2	15647311	Hill	608	Spain	Female	1	83807.86
2	3	15619304	Onio	502	France	Female	8	159660.80
3	4	15701354	Boni	699	France	Female	1	0.00
4	5	15737888	Mitchell	850	Spain	Female	2	125510.82
5	6	15574012	Chu	645	Spain	Male	8	113755.78
6	7	15592531	Bartlett	822	France	Male	7	0.00
7	8	15656148	Obinna	376	Germany	Female	4	115046.74
8	9	15792365	He	501	France	Male	4	142051.07
9	10	15592389	H?	684	France	Male	2	134603.88

Next steps: [Generate code with df](#) [View recommended plots](#)

```
#to convert any column into index use index function
df.set_index('Surname') #changes are not permammetly
```

	RowNumber	CustomerId	CreditScore	Geography	Gender	Tenure	Balance	
Surname								
Hargrave	1	15634602	619	France	Female	2	0.00	
Hill	2	15647311	608	Spain	Female	1	83807.86	
Onio	3	15619304	502	France	Female	8	159660.80	
Boni	4	15701354	699	France	Female	1	0.00	
Mitchell	5	15737888	850	Spain	Female	2	125510.82	
...
Obijaku	9996	15606229	771	France	Male	5	0.00	
Johnstone	9997	15569892	516	France	Male	10	57369.61	
Liu	9998	15584532	709	France	Female	7	0.00	
Sabbatini	9999	15682355	772	Germany	Male	3	75075.31	
Walker	10000	15628319	792	France	Female	4	130142.79	

10000 rows × 13 columns

```
df.iloc[2] #row number 2 ka data de rha hai
df.iloc[15:21]#row 15 se 20 tak ka data dega
df.iloc[15:21:4] #15 se 20 row ka data with 4 ki steping
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Tenure	Balance
15	16	15643966	Goforth	616	Germany	Male	3	143129.41
19	20	15568982	Hao	726	France	Female	6	0.00

```
df.loc[2] #loc can take character name also as indexing
```

```
RowNumber      3
CustomerId     15619304
Surname        Onio
CreditScore    502
Geography      France
Gender         Female
Tenure         8
Balance       159660.8
NumOfProducts  3
HasCrCard      1
IsActiveMember 0
EstimatedSalary 113931.57
Exited         1
New_CreditScore 512
Name: 2, dtype: object
```

```
df.index
```

```
RangeIndex(start=0, stop=10000, step=1)
```

```
df['Geography'].value_counts()
```

```
Geography
France    5014
Germany   2509
Spain     2477
Name: count, dtype: int64
```

```
df['IsActiveMember'].value_counts() # 0 matlab active ni hai and 1 matlab active h
```

```
IsActiveMember
1      5151
0     4849
Name: count, dtype: int64
```

```
df['IsActiveMember'].value_counts()
```

```
df.sample()#it randomly pick any row
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Tenure	Balance
674	675	15642394	He	529	Spain	Male	5	0.0

df.describe() #categorical data is not included bydefault

	RowNumber	CustomerId	CreditScore	Tenure	Balance	NumOfProducts
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000000
mean	5000.50000	1.569094e+07	650.528800	5.012800	76485.889288	1.530000
std	2886.89568	7.193619e+04	96.653299	2.892174	62397.405202	0.581000
min	1.00000	1.556570e+07	350.000000	0.000000	0.000000	1.000000
25%	2500.75000	1.562853e+07	584.000000	3.000000	0.000000	1.000000
50%	5000.50000	1.569074e+07	652.000000	5.000000	97198.540000	1.000000
75%	7500.25000	1.575323e+07	718.000000	7.000000	127644.240000	2.000000
max	10000.00000	1.581569e+07	850.000000	10.000000	250898.090000	4.000000

df.describe(include="all")#include the categorical data also

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Tenure
count	10000.00000	1.000000e+04	10000	10000.000000	10000	10000	10000.000000
unique	NaN	NaN	2932	NaN	3	2	NaN
top	NaN	NaN	Smith	NaN	France	Male	NaN
freq	NaN	NaN	32	NaN	5014	5457	NaN
mean	5000.50000	1.569094e+07	NaN	650.528800	NaN	NaN	5.012800
std	2886.89568	7.193619e+04	NaN	96.653299	NaN	NaN	2.892174
min	1.00000	1.556570e+07	NaN	350.000000	NaN	NaN	0.000000
25%	2500.75000	1.562853e+07	NaN	584.000000	NaN	NaN	3.000000
50%	5000.50000	1.569074e+07	NaN	652.000000	NaN	NaN	5.000000
75%	7500.25000	1.575323e+07	NaN	718.000000	NaN	NaN	7.000000
max	10000.00000	1.581569e+07	NaN	850.000000	NaN	NaN	10.000000

df.isnull() #give false if no values are null

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Tenure	Balance
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...
9995	False	False	False	False	False	False	False	False
9996	False	False	False	False	False	False	False	False
9997	False	False	False	False	False	False	False	False
9998	False	False	False	False	False	False	False	False
9999	False	False	False	False	False	False	False	False

10000 rows × 9 columns

df.isnull().sum() #gives us the no. of values which are null in each row

RowNumber	0
CustomerId	0
Surname	0
CreditScore	0

```
Geography      0
Gender         0
Tenure         0
Balance        0
NumOfProducts 0
HasCrCard      0
IsActiveMember 0
EstimatedSalary 0
Exited         0
New_CreditScore 0
dtype: int64

df.duplicated().sum() #duplictions in a row is there or not

0

df.rename(columns={'Surname':'Name','Balance':'lpa'}) #temporary changes
df.head(5)
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Tenure	Balance
0	1	15634602	Hargrave	619	France	Female	2	0.00
1	2	15647311	Hill	608	Spain	Female	1	83807.86
2	3	15619304	Onio	502	France	Female	8	159660.80
3	4	15701354	Boni	699	France	Female	1	0.00
4	5	15737888	Mitchell	850	Spain	Female	2	125510.82

Next steps:

Generate code with df

 View recommended plots

```
df.iloc[[1,1]] #pahla row 2times
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Tenure	Balance	M
	1	2	15647311	Hill	608	Spain	Female	1	83807.86
	1	2	15647311	Hill	608	Spain	Female	1	83807.86

```
df.iloc[[1,2,3]] #1st row then 2nd row then 3rd row
```

RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Tenure	Balance	
1	2	15647311	Hill	608	Spain	Female	1	83807.86
2	3	15619304	Onio	502	France	Female	8	159660.80
3	4	15701354	Boni	699	France	Female	1	0.00

```
df.iloc[[1,1],[0,1]] #1st row then again 1st row and 0 th column and the 1st column
```

RowNumber	CustomerId	
1	2	15647311
1	2	15647311

```
df.loc[1]

RowNumber      2
CustomerId    15647311
Surname        Hill
CreditScore    608
Geography      Spain
Gender         Female
Tenure         1
Balance       83807.86
NumOfProducts  1
HasCrCard      0
IsActiveMember 1
EstimatedSalary 112542.58
Exited         0
New_CreditScore 618
Name: 1, dtype: object
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Tenure	Balance
2	3	15619304	Onio	502	France	Female	8	159660.80
3	4	15701354	Boni	699	France	Female	1	0.00
4	5	15737888	Mitchell	850	Spain	Female	2	125510.82
5	6	15574012	Chu	645	Spain	Male	8	113755.78
6	7	15592531	Bartlett	822	France	Male	7	0.00
7	8	15656148	Obinna	376	Germany	Female	4	115046.74

```
x=df.set_index('Surname') #change the index
x.head()
x.loc['Hill'] #gives the data of the person having surname hill
```

RowNumber	CustomerId	CreditScore	Geography	Gender	Tenure	Balance	Num
Surname							
Hill	2	15647311	608	Spain	Female	1	83807.86
Hill	900	15632210	657	Germany	Male	2	171770.55
Hill	1124	15596800	779	Germany	Male	1	158456.76
Hill	2113	15631201	472	Spain	Female	4	0.00
Hill	2367	15623430	672	France	Male	9	0.00
Hill	3904	15678129	643	Spain	Female	9	150840.03
Hill	4071	15611551	676	Spain	Male	1	131659.59
Hill	4116	15584505	580	France	Female	5	113923.81
Hill	4389	15801152	553	Spain	Female	1	142876.98
Hill	6557	15612207	840	Germany	Female	1	87779.83
Hill	7413	15631693	697	France	Male	7	0.00
Hill	7430	15644878	685	Spain	Female	6	117302.62
Hill	8600	15708713	633	France	Male	3	0.00
Hill	8742	15762855	622	Spain	Female	8	0.00
Hill	8805	15773973	765	France	Male	2	0.00
Hill	9073	15777315	529	France	Male	6	93616.35
Hill	9431	15591980	753	France	Male	5	122568.05

 Generate

randomly select 5 items from a list



Close

```
x=df.set_index('Surname') #change the index
x.head()
x.loc[['Boni','Hill','Onio']]
```

Boni	8292	15742879	668	Spain	Male	1	147904.31
Boni	8624	15666197	430	Germany	Female	8	153058.64
Boni	8840	15774328	606	Germany	Male	1	144757.97
Boni	9012	15717498	775	France	Male	6	133970.22
Boni	9088	15791316	714	France	Male	3	0.00
Boni	9603	15603135	634	Germany	Female	3	95727.05
Hill	2	15647311	608	Spain	Female	1	83807.86
Hill	900	15632210	657	Germany	Male	2	171770.55
Hill	1124	15596800	779	Germany	Male	1	158456.76
Hill	2113	15631201	472	Spain	Female	4	0.00
Hill	2367	15623430	672	France	Male	9	0.00
Hill	3904	15678129	643	Spain	Female	9	150840.03
Hill	4071	15611551	676	Spain	Male	1	131659.59
Hill	4116	15584505	580	France	Female	5	113923.81
Hill	4389	15801152	553	Spain	Female	1	142876.98
Hill	6557	15612207	840	Germany	Female	1	87779.83
Hill	7413	15631693	697	France	Male	7	0.00
Hill	7430	15644878	685	Spain	Female	6	117302.62
Hill	8600	15708713	633	France	Male	3	0.00
Hill	8742	15762855	622	Spain	Female	8	0.00
Hill	8805	15773973	765	France	Male	2	0.00
Hill	9073	15777315	529	France	Male	6	93616.35
Hill	9431	15591980	753	France	Male	5	122568.05
Onio	3	15619304	502	France	Female	8	159660.80
Onio	1737	15663489	633	Germany	Female	0	138577.34
Onio	3811	15746802	477	France	Female	6	131286.46
Onio	4399	15707007	743	France	Female	8	0.00
Onio	5089	15734610	543	France	Male	4	89838.71
Onio	7429	15750055	503	Spain	Male	9	100262.88

```
df.loc[df['Balance']>100000]
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Tenure	Balance
2	3	15619304	Onio	502	France	Female	8	159660.80
4	5	15737888	Mitchell	850	Spain	Female	2	125510.80
5	6	15574012	Chu	645	Spain	Male	8	113755.70
7	8	15656148	Obinna	376	Germany	Female	4	115046.70
8	9	15792365	He	501	France	Male	4	142051.00
...
9985	9986	15586914	Nepean	659	France	Male	6	123841.40
9986	9987	15581736	Bartlett	673	Germany	Male	1	183579.50
9987	9988	15588839	Mancini	606	Spain	Male	8	180307.70
9993	9994	15569266	Rahman	644	France	Male	7	155060.40
9999	10000	15628319	Walker	792	France	Female	4	130142.70

4799 rows × 14 columns

```
df.loc[(df['Balance']>10000) &(df['Gender']=='Male')] # and operator (intersection)
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Tenure	Balance
	5	6	15574012	Chu	645	Spain	Male	8 113
	8	9	15792365	He	501	France	Male	4 142
	9	10	15592389	H?	684	France	Male	2 134
	10	11	15767821	Bearce	528	France	Male	6 102
	15	16	15643966	Goforth	616	Germany	Male	3 143

	9987	9988	15588839	Mancini	606	Spain	Male	8 180
	9990	9991	15798964	Nkemakonam	714	Germany	Male	3 35
	9993	9994	15569266	Rahman	644	France	Male	7 155
	9996	9997	15569892	Johnstone	516	France	Male	10 57
	9998	9999	15682355	Sabbatini	772	Germany	Male	3 75

3494 rows × 14 columns

```
df.loc[(df['Balance']>10000) | (df['Gender']=='Male')] #or operator(union)
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Tenure	Balance
	1	2	15647311	Hill	608	Spain	Female	1 83807
	2	3	15619304	Onio	502	France	Female	8 159660
	4	5	15737888	Mitchell	850	Spain	Female	2 125510
	5	6	15574012	Chu	645	Spain	Male	8 113755
	6	7	15592531	Bartlett	822	France	Male	7 1

	9993	9994	15569266	Rahman	644	France	Male	7 155060
	9995	9996	15606229	Obijaku	771	France	Male	5 1
	9996	9997	15569892	Johnstone	516	France	Male	10 57365
	9998	9999	15682355	Sabbatini	772	Germany	Male	3 75075
	9999	10000	15628319	Walker	792	France	Female	4 130142

8345 rows × 14 columns

```
df.loc[(df['Balance']>10000) &(df['Gender']=='Male') ,['Surname']]
```

	Surname
5	Chu
8	He
9	H?
10	Bearce
15	Goforth
...	...
9987	Mancini
9990	Nkemakonam
9993	Rahman
9996	Johnstone
9998	Sabbatini

3494 rows × 1 columns

```
#to find the unique values in the column
df['Tenure'].unique()

array([ 2,  1,  8,  7,  4,  6,  3, 10,  5,  9,  0])
```


```
df['NumOfProducts'].unique()

array([1, 3, 2, 4])

df['NumOfProducts'].value_counts() #to know kaunsi value kitni baar arhi hai

NumOfProducts
1    5084
2    4590
3     266
4      60
Name: count, dtype: int64

df1=df #copy of df in df1
pd.concat([df,df1]) #concating df1 and df
```



	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	
0	1	15634602	Hargrave	619	France	Female	2	0.00	1	1	1	
1	2	15647311	Hill	608	Spain	Female	1	83807.86	1	0	1	
2	3	15619304	Onio	502	France	Female	8	159660.80	3	1	0	
3	4	15701354	Boni	699	France	Female	1	0.00	2	0	0	
4	5	15737888	Mitchell	850	Spain	Female	2	125510.82	1	1	1	
...	
9995	9996	15606229	Obijaku	771	France	Male	5	0.00	2	1	0	
9996	9997	15569892	Johnstone	516	France	Male	10	57369.61	1	1	1	
9997	9998	15584532	Liu	709	France	Female	7	0.00	1	0	1	
9998	9999	15682355	Sabbatini	772	Germany	Male	3	75075.31	2	1	0	
9999	10000	15628319	Walker	792	France	Female	4	130142.79	1	1	0	

20000 rows × 14 columns