

```
import pandas as pd

df = pd.read_csv('dataset_Facebook.csv', sep=';')

# Display first n rows of data set the returns first 5 rows
df.head()
```

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	Lifetime Post Consumers	Lifetime Post Consumptions	Lifetime Post Impressions by peop. who ha liked yo Pa	
0	139441	Photo		2	12	4	3	0.0	2752	5091	178	109	159	30
1	139441	Status		2	12	3	10	0.0	10460	19057	1457	1361	1674	117
2	139441	Photo		3	12	3	3	0.0	2413	4373	177	113	154	28
3	139441	Photo		2	12	2	10	1.0	50128	87991	2211	790	1119	610
4	139441	Photo		2	12	2	3	0.0	7244	13594	671	410	580	62

```
# print data set information
# The info() method prints information about the DataFrame.
# The information contains the number of columns, column labels, column data types, memory usage, range index,
# and the number of cells in each column
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 19 columns):
#   Column                                                                 Non-Null Count  Dtype
---  -
0   Page total likes                                                    500 non-null   int64
1   Type                                                                500 non-null   object
2   Category                                                            500 non-null   int64
3   Post Month                                                          500 non-null   int64
4   Post Weekday                                                        500 non-null   int64
5   Post Hour                                                           500 non-null   int64
6   Paid                                                                499 non-null   float64
7   Lifetime Post Total Reach                                           500 non-null   int64
8   Lifetime Post Total Impressions                                     500 non-null   int64
9   Lifetime Engaged Users                                              500 non-null   int64
10  Lifetime Post Consumers                                              500 non-null   int64
11  Lifetime Post Consumptions                                           500 non-null   int64
12  Lifetime Post Impressions by people who have liked your Page       500 non-null   int64
13  Lifetime Post reach by people who like your Page                   500 non-null   int64
14  Lifetime People who have liked your Page and engaged with your post 500 non-null   int64
15  comment                                                             500 non-null   int64
16  like                                                                499 non-null   float64
17  share                                                                496 non-null   float64
18  Total Interactions                                                  500 non-null   int64
dtypes: float64(3), int64(15), object(1)
memory usage: 74.3+ KB
```

▼ a. Create data subsets

```
df_subset1 = df[df[['like', 'share']]]
print(df_subset1)
```

	like	share
0	79.0	17.0
1	130.0	29.0
2	66.0	14.0
3	1572.0	147.0
4	325.0	49.0
..
495	53.0	26.0

```
496    53.0    22.0
497    93.0    18.0
498    91.0    38.0
499    91.0    28.0

[500 rows x 2 columns]
```

```
df[['Type','like']]
```

	Type	like
0	Photo	79.0
1	Status	130.0
2	Photo	66.0
3	Photo	1572.0
4	Photo	325.0
...
495	Photo	53.0
496	Photo	53.0
497	Photo	93.0
498	Photo	91.0
499	Photo	91.0

500 rows x 2 columns

```
df_subset2=df[df['like']>100]
print(df_subset2)
```

3	58	1572.0	147.0	1777
4	19	325.0	49.0	393
5	1	152.0	33.0	186
6	3	249.0	27.0	279
...
488	21	277.0	80.0	378
491	1	105.0	46.0	152
492	0	128.0	9.0	137
493	17	185.0	55.0	257
494	10	125.0	41.0	176

[252 rows x 19 columns]

```
df.loc[[1,3,7]]
```

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users
1	139441	Status	2	12	3	10	0.0	10460	19057	1457
3	139441	Photo	2	12	2	10	1.0	50128	87991	2211
7	139441	Photo	3	12	7	9	1.0	13720	24137	537

```
# df.iloc[row_start: row_end , column_start : column_end]
```

```
df.iloc[1:3,0:2]
```

	Page total likes	Type
1	139441	Status
2	139441	Photo

```
df.loc[5:8]
```

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	Lifetime Post Consumers	Lifetime Post Consumptions	Lifetime Post Impressions by people who have liked your Page
5	139441	Status	2	12	1	9	0.0	10472	20849	1191	1073	1389	1600
6	139441	Photo	3	12	1	3	1.0	11692	19479	481	265	364	1540
7	139441	Photo	3	12	7	9	1.0	13720	24137	537	232	305	1970
8	139441	Status	2	12	7	3	0.0	11844	22538	1530	1407	1692	1520

```
df.loc[1:7,['like','share']]
```

like share

▼ b. Merge Data

```

# Create two subsets of a data sets and merge data from these two subsets.

# Display data set information

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 19 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0    Page total likes                          500 non-null    int64
1    Type                                       500 non-null    object
2    Category                                  500 non-null    int64
3    Post Month                               500 non-null    int64
4    Post Weekday                             500 non-null    int64
5    Post Hour                                500 non-null    int64
6    Paid                                       499 non-null    float64
7    Lifetime Post Total Reach                500 non-null    int64
8    Lifetime Post Total Impressions          500 non-null    int64
9    Lifetime Engaged Users                  500 non-null    int64
10   Lifetime Post Consumers                  500 non-null    int64
11   Lifetime Post Consumptions              500 non-null    int64
12   Lifetime Post Impressions by people who have liked your Page  500 non-null    int64
13   Lifetime Post reach by people who like your Page              500 non-null    int64
14   Lifetime People who have liked your Page and engaged with your post  500 non-null    int64
15   comment                                  500 non-null    int64
16   like                                     499 non-null    float64
17   share                                    496 non-null    float64
18   Total Interactions                      500 non-null    int64
dtypes: float64(3), int64(15), object(1)
memory usage: 74.3+ KB

# create subset1 with data from row 1 to row 5 and columns 'Category','like','share','Type' of Facebook metric data set
subset1 = df.loc[1:5,['Category','like','share','Type']]
subset1
```

	Category	like	share	Type
1	2	130.0	29.0	Status
2	3	66.0	14.0	Photo
3	2	1572.0	147.0	Photo
4	2	325.0	49.0	Photo
5	2	152.0	33.0	Status

```

# create subset2 with data from row 11 to row 13 and in columns 'Category','like','share','Type' of Facebook metric data set
subset2 = df.loc[11:13,['Category','like','share','Type']]
subset2
```

	Category	like	share	Type
11	2	88.0	18.0	Photo
12	2	90.0	14.0	Photo
13	2	137.0	10.0	Photo

```
print(subset1, '\n\n', subset2)
```

	Category	like	share	Type
1	2	130.0	29.0	Status
2	3	66.0	14.0	Photo
3	2	1572.0	147.0	Photo
4	2	325.0	49.0	Photo
5	2	152.0	33.0	Status

```
Category  like  share  Type
11        2   88.0   18.0  Photo
12        2   90.0   14.0  Photo
13        2  137.0   10.0  Photo

# merge subset1 and subset 2  along axis 0
# ( Axis 0 means row wise,axis 1 means column wise )
merge_set = pd.concat([subset1, subset2], axis=0)
print(merge_set)
```

```
Category  like  share  Type
1         2  130.0   29.0  Status
2         3   66.0   14.0  Photo
3         2  1572.0  147.0  Photo
4         2   325.0   49.0  Photo
5         2   152.0   33.0  Status
11        2   88.0   18.0  Photo
12        2   90.0   14.0  Photo
13        2  137.0   10.0  Photo

# Display the size of merge set.  i.e after combining two sets
merge_set.shape

(8, 4)
```

▼ Sorting Data

```
# Sort data in ascending order of 'like'

df.sort_values(by='like',ascending=True)
```

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	Lifetime Post Consumers	Lifetime Post Consumptions	Lifet F Impressi by pec who t liked y F
21	138414	Photo	1	12	7	10	0.0	1384	2467	15	15	20	2
417	104070	Photo	1	3	3	10	0.0	1874	2474	25	25	31	1
100	137020	Photo	1	10	4	9	1.0	1357	2453	37	37	55	2
76	137893	Photo	1	11	3	2	0.0	1228	2392	17	17	19	2
441	98195	Photo	1	3	5	4	1.0	1845	2670	9	9	9	1
...
168	135428	Photo	1	9	3	10	0.0	41984	68290	3370	2420	4074	34
349	117764	Photo	3	5	5	13	0.0	81856	124753	3000	1637	2718	52
379	111620	Photo	3	4	1	14	1.0	105632	147918	3984	2254	3391	48
244	130791	Photo	2	7	3	5	1.0	180480	319133	8072	4010	6242	108
111	136736	Photo	1	10	6	8	0.0	1261	2158	37	37	49	1

500 rows × 19 columns

```
# Sort data in descending order of 'like' column in Facebbok metric data set and use 'mergesort' method for sorting

df.sort_values(by ='like',ascending=False, kind='mergesort')
```

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	Lifetime Post Consumers	Lifetime Post Consumptions	Impressi by pec who h liked y F	
244	130791	Photo		2	7	3	5	1.0	180480	319133	8072	4010	6242	108
379	111620	Photo		3	4	1	14	1.0	105632	147918	3984	2254	3391	48
349	117764	Photo		3	5	5	13	0.0	81856	124753	3000	1637	2718	52
168	135428	Photo		1	9	3	10	0.0	41984	68290	3370	2420	4074	34
3	139441	Photo		2	12	2	10	1.0	50128	87991	2211	790	1119	61
...
76	137893	Photo		1	11	3	2	0.0	1228	2392	17	17	19	2
100	137020	Photo		1	10	4	9	1.0	1357	2453	37	37	55	2
417	104070	Photo		1	3	3	10	0.0	1874	2474	25	25	31	1
441	98195	Photo		1	3	5	4	1.0	1845	2670	9	9	9	1
111	136736	Photo		1	10	6	8	0.0	1261	2158	37	37	49	1

500 rows x 14 columns

▼ d. Transposing Data

Get transpose of data. DataFrame.transpose() converts rows into column

```
result= df.transpose()
print(result)
```

	0	1	2	\
Page total likes	139441	139441	139441	
Type	Photo	Status	Photo	
Category	2	2	3	
Post Month	12	12	12	
Post Weekday	4	3	3	
Post Hour	3	10	3	
Paid	0.0	0.0	0.0	
Lifetime Post Total Reach	2752	10460	2413	
Lifetime Post Total Impressions	5091	19057	4373	
Lifetime Engaged Users	178	1457	177	
Lifetime Post Consumers	109	1361	113	
Lifetime Post Consumptions	159	1674	154	
Lifetime Post Impressions by people who have li...	3078	11710	2812	
Lifetime Post reach by people who like your Page	1640	6112	1503	
Lifetime People who have liked your Page and en...	119	1108	132	
comment	4	5	0	
like	79.0	130.0	66.0	
share	17.0	29.0	14.0	
Total Interactions	100	164	80	

	3	4	5	\
Page total likes	139441	139441	139441	
Type	Photo	Photo	Status	
Category	2	2	2	
Post Month	12	12	12	
Post Weekday	2	2	1	
Post Hour	10	3	9	
Paid	1.0	0.0	0.0	
Lifetime Post Total Reach	50128	7244	10472	
Lifetime Post Total Impressions	87991	13594	20849	
Lifetime Engaged Users	2211	671	1191	
Lifetime Post Consumers	790	410	1073	
Lifetime Post Consumptions	1119	580	1389	
Lifetime Post Impressions by people who have li...	61027	6228	16034	
Lifetime Post reach by people who like your Page	32048	3200	7852	
Lifetime People who have liked your Page and en...	1386	396	1016	
comment	58	19	1	
like	1572.0	325.0	152.0	
share	147.0	49.0	33.0	
Total Interactions	1777	393	186	

	6	7	8	\
Page total likes	139441	139441	139441	
Type	Photo	Photo	Status	
Category	3	3	2	
Post Month	12	12	12	
Post Weekday	1	7	7	
Post Hour	3	9	3	
Paid	1.0	1.0	0.0	
Lifetime Post Total Reach	11692	13720	11844	
Lifetime Post Total Impressions	19479	24137	22538	
Lifetime Engaged Users	481	537	1530	
Lifetime Post Consumers	265	232	1407	
Lifetime Post Consumptions	364	305	1692	
Lifetime Post Impressions by people who have li...	15432	19728	15220	
Lifetime Post reach by people who like your Page	9328	11056	7912	
Lifetime People who have liked your Page and on	370	422	1250	

```
df.shape
```

```
(500, 19)
```

```
result.shape
```

```
(19, 500)
```

```
# df.loc[0:4,['like','share','Category','Type']]
```

```
selective_df=pd.DataFrame(df.iloc[0:3],columns=['like','share','Category','Type'])
print(selective_df.head(5))
```

	like	share	Category	Type
0	79.0	17.0	2	Photo
1	130.0	29.0	2	Status
2	66.0	14.0	3	Photo

▼ e. Shape and reshape Data

1. melt()

2. pivot_table()

```
# create a subset with first 3 rows and 'category','post.month','post.Hour', 'paid'
sub1 = df.loc[0:3,['Category', 'Post Month','Post Hour', 'Paid']]
sub1.melt(id_vars=['Category'])
```

	Category	variable	value
0	2	Post Month	12.0
1	2	Post Month	12.0
2	3	Post Month	12.0
3	2	Post Month	12.0
4	2	Post Hour	3.0
5	2	Post Hour	10.0
6	3	Post Hour	3.0
7	2	Post Hour	10.0
8	2	Paid	0.0
9	2	Paid	0.0
10	3	Paid	0.0
11	2	Paid	1.0

```
# A PivotTable is an interactive way to quickly summarize large amounts of data
# pivot() is the complete opposite of melt()
```

```
pivote_table=pd.pivot_table(selective_df,index=[ 'Category'])
print(pivote_table)
```

	like	share
Category		
2	104.5	23.0
3	66.0	14.0

```
pivote_table.shape
```

```
(2, 2)
```

```
pivote_table.reset_index(inplace=True)
print(pivote_table)
```

	Category	like	share
0	2	104.5	23.0
1	3	66.0	14.0

```
pivote_table.melt(id_vars=['like','share'])
```

	like	share	variable	value
0	104.5	23.0	Category	2
1	66.0	14.0	Category	3

```
# Example :
```

```
home_expendature = { "Month": [ 'Jan'      , 'Jan'      , 'Jan'      , 'feb'      , 'feb'      , 'march'    , 'march' ],
                    "category": [ 'transport','grocery','household','entertainment','transport','grocery','household'],
                    "amount"  : [74,          235,          175,          100,          115,          240,          225]
                    }
```

```
home_expendaturedf=pd.DataFrame(home_expendature)
print(home_expendaturedf)
```

	Month	category	amount
0	Jan	transport	74
1	Jan	grocery	235
2	Jan	household	175
3	feb	entertainment	100
4	feb	transport	115
5	march	grocery	240
6	march	household	225

```
pd.pivot_table(home_expendaturedf,index=[ 'category','Month'])
```

		amount
	category	Month
entertainment	feb	100
	Jan	235
	march	240
household	Jan	175
	march	225
transport	Jan	74
	feb	115