Pandas by Mrittika Megaraj

```
Objects:
1 Series : 1D, Homogenous, indexes are explicit
2 DataFrame : collection of series, heterogenous, 2D
Inspection
```

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

In [2]:
```
print(dir(pd))
```

```
['ArrowDtype', 'BooleanDtype', 'Categorical', 'CategoricalDtype', 'Catego
ricalIndex', 'DataFrame', 'DateOffset', 'DatetimeIndex', 'DatetimeTZDtyp
e', 'ExcelFile', 'ExcelWriter', 'Flags', 'Float32Dtype', 'Float64Dtype',
'Float64Index', 'Grouper', 'HDFStore', 'Index', 'IndexSlice', 'Int16Dtyp
e', 'Int32Dtype', 'Int64Dtype', 'Int64Index', 'Int8Dtype', 'Interval', 'I
ntervalDtype', 'IntervalIndex', 'MultiIndex', 'NA', 'NaT', 'NamedAgg', 'P
eriod', 'PeriodDtype', 'PeriodIndex', 'RangeIndex', 'Series', 'SparseDtyp
e', 'StringDtype', 'Timedelta', 'TimedeltaIndex', 'Timestamp', 'UInt16Dty
pe', 'UInt32Dtype', 'UInt64Dtype', 'UInt64Index', 'UInt8Dtype', '__all_
_', '__builtins__', '__cached__', '__deprecated_num_index_names', '__dir_
_', '__doc__', '__docformat__', '__file__', '__getattr__', '__git_version
__', '__loader__', '__name__', '__package__', '__path__', '__spec__', '__
version__', '_config', '_is_numpy_dev', '_libs', '_testing', '_typing',
'_version', 'annotations', 'api', 'array', 'arrays', 'bdate_range', 'comp
at', 'concat', 'core', 'crosstab', 'cut', 'date_range', 'describe_optio
n', 'errors', 'eval', 'factorize', 'from_dummies', 'get_dummies', 'get_op
tion', 'infer_freq', 'interval_range', 'io', 'isna', 'isnull', 'json_norm
alize', 'lreshape', 'melt', 'merge', 'merge_asof', 'merge_ordered', 'notn
a', 'notnull', 'offsets', 'option_context', 'options', 'pandas', 'period_
range', 'pivot', 'pivot_table', 'plotting', 'qcut', 'read_clipboard', 're
ad_csv', 'read_excel', 'read_feather', 'read_fwf', 'read_gbq', 'read_hd
f', 'read_html', 'read_json', 'read_orc', 'read_parquet', 'read_pickle',
'read_sas', 'read_spss', 'read_sql', 'read_sql_query', 'read_sql_table',
'read_stata', 'read_table', 'read_xml', 'reset_option', 'set_eng_float_fo
rmat', 'set_option', 'show_versions', 'test', 'testing', 'timedelta_rang
e', 'to_datetime', 'to_numeric', 'to_pickle', 'to_timedelta', 'tseries',
'unique', 'util', 'value_counts', 'wide_to_long']
```

In [3]:
```
s1=pd.Series([100,200,300,400,500],index=["Store1","Store2","Store3","Store
```

In [4]:
```
type(s1)
```

Out[4]:
```
pandas.core.series.Series
```

In [5]:
```
s1
```

Out[5]:
```
Store1    100
Store2    200
Store3    300
Store4    400
Store5    500
dtype: int64
```

In [6]:
```
s2=pd.Series([12,10,25,32,4],index=["Store1","Store2","Store3","Store4","St
```

In [7]:
```
s2
```

Out[7]:
```
Store1    12
Store2    10
Store3    25
Store4    32
Store5     4
dtype: int64
```

In [8]:
```
df=pd.DataFrame({"Sales":s1,"Qty":s2})
```

In [9]:
```
type(df)
```

Out[9]:
```
pandas.core.frame.DataFrame
```

In [10]:
```
df
```

Out[10]:

	Sales	Qty
Store1	100	12
Store2	200	10
Store3	300	25
Store4	400	32
Store5	500	4

In [11]:
```
store=pd.read_csv("store.csv")
```

In [12]:
```
store
```

Out[12]:

	StoreCode	StoreName	StoreType	Location	OperatingCost	Staff_Cnt	TotalSales	To
0	STR101	Electronics Zone	Electronincs	California	21.0	60	160.0	
1	STR102	Apparel Zone	Apparel	California	21.0	60	160.0	
2	STR103	Super Bazar	Super Market	California	22.8	40	108.0	
3	STR104	Super Market	Super Market	California	21.4	60	258.0	
4	STR105	Central Store	Super Market	California	18.7	80	360.0	
5	STR106	Apparel Zone	Apparel	California	18.1	60	225.0	
6	STR107	Fashion Bazar	Apparel	California	14.3	80	360.0	
7	STR108	Digital Bazar	Electronincs	California	24.4	40	146.7	
8	STR109	Electronics Zone	Electronincs	Washington	22.8	40	140.8	
9	STR110	Apparel Zone	Apparel	Washington	19.2	60	167.6	
10	STR111	Super Bazar	Super Market	Washington	17.8	60	167.6	
11	STR112	Super Market	Super Market	Washington	16.4	80	275.8	
12	STR113	Central Store	Super Market	Washington	17.3	80	275.8	
13	STR114	Apparel Zone	Apparel	Washington	15.2	80	275.8	
14	STR115	Fashion Bazar	Apparel	Washington	10.4	80	472.0	
15	STR116	Digital Bazar	Electronincs	Washington	10.4	80	460.0	
16	STR117	Electronics Zone	Electronincs	Texas	14.7	80	440.0	
17	STR118	Apparel Zone	Apparel	Texas	32.4	40	78.7	
18	STR119	Super Bazar	Super Market	Texas	30.4	40	75.7	
19	STR120	Super Market	Super Market	Texas	33.9	40	71.1	
20	STR121	Central Store	Super Market	Texas	21.5	40	120.1	
21	STR122	Apparel Zone	Apparel	Texas	15.5	80	318.0	
22	STR123	Fashion Bazar	Apparel	Texas	15.2	80	304.0	
23	STR124	Digital Bazar	Electronincs	Texas	13.3	80	350.0	
24	STR125	Electronics Zone	Electronincs	Montana	19.2	80	400.0	

	StoreCode	StoreName	StoreType	Location	OperatingCost	Staff_Cnt	TotalSales	Tc
25	STR126	Apparel Zone	Apparel	Montana	27.3	40	79.0	
26	STR127	Super Bazar	Super Market	Montana	26.0	40	120.3	
27	STR128	Super Market	Super Market	Montana	30.4	40	95.1	
28	STR129	Central Store	Super Market	Montana	15.8	80	351.0	
29	STR130	Apparel Zone	Apparel	Montana	19.7	60	145.0	
30	STR131	Fashion Bazar	Apparel	Montana	15.0	80	301.0	
31	STR132	Digital Bazar	Electronincs	Montana	21.4	40	121.0	

In [13]:
```
store1=pd.read_csv("store.csv")
```

In [14]:
```
store1
```

Out[14]:

	StoreCode	StoreName	StoreType	Location	OperatingCost	Staff_Cnt	TotalSales	To
0	STR101	Electronics Zone	Electronincs	California	21.0	60	160.0	
1	STR102	Apparel Zone	Apparel	California	21.0	60	160.0	
2	STR103	Super Bazar	Super Market	California	22.8	40	108.0	
3	STR104	Super Market	Super Market	California	21.4	60	258.0	
4	STR105	Central Store	Super Market	California	18.7	80	360.0	
5	STR106	Apparel Zone	Apparel	California	18.1	60	225.0	
6	STR107	Fashion Bazar	Apparel	California	14.3	80	360.0	
7	STR108	Digital Bazar	Electronincs	California	24.4	40	146.7	
8	STR109	Electronics Zone	Electronincs	Washington	22.8	40	140.8	
9	STR110	Apparel Zone	Apparel	Washington	19.2	60	167.6	
10	STR111	Super Bazar	Super Market	Washington	17.8	60	167.6	
11	STR112	Super Market	Super Market	Washington	16.4	80	275.8	
12	STR113	Central Store	Super Market	Washington	17.3	80	275.8	
13	STR114	Apparel Zone	Apparel	Washington	15.2	80	275.8	
14	STR115	Fashion Bazar	Apparel	Washington	10.4	80	472.0	
15	STR116	Digital Bazar	Electronincs	Washington	10.4	80	460.0	
16	STR117	Electronics Zone	Electronincs	Texas	14.7	80	440.0	
17	STR118	Apparel Zone	Apparel	Texas	32.4	40	78.7	
18	STR119	Super Bazar	Super Market	Texas	30.4	40	75.7	
19	STR120	Super Market	Super Market	Texas	33.9	40	71.1	
20	STR121	Central Store	Super Market	Texas	21.5	40	120.1	
21	STR122	Apparel Zone	Apparel	Texas	15.5	80	318.0	
22	STR123	Fashion Bazar	Apparel	Texas	15.2	80	304.0	
23	STR124	Digital Bazar	Electronincs	Texas	13.3	80	350.0	
24	STR125	Electronics Zone	Electronincs	Montana	19.2	80	400.0	

	StoreCode	StoreName	StoreType	Location	OperatingCost	Staff_Cnt	TotalSales	To
25	STR126	Apparel Zone	Apparel	Montana	27.3	40	79.0	
26	STR127	Super Bazar	Super Market	Montana	26.0	40	120.3	
27	STR128	Super Market	Super Market	Montana	30.4	40	95.1	
28	STR129	Central Store	Super Market	Montana	15.8	80	351.0	
29	STR130	Apparel Zone	Apparel	Montana	19.7	60	145.0	
30	STR131	Fashion Bazar	Apparel	Montana	15.0	80	301.0	
31	STR132	Digital Bazar	Electronincs	Montana	21.4	40	121.0	

EDA : Exploratory Data Analysis

In [15]:
```
store.head()
```

Out[15]:

	StoreCode	StoreName	StoreType	Location	OperatingCost	Staff_Cnt	TotalSales	Total
0	STR101	Electronics Zone	Electronincs	California	21.0	60	160.0	
1	STR102	Apparel Zone	Apparel	California	21.0	60	160.0	
2	STR103	Super Bazar	Super Market	California	22.8	40	108.0	
3	STR104	Super Market	Super Market	California	21.4	60	258.0	
4	STR105	Central Store	Super Market	California	18.7	80	360.0	

In [16]:
```
store.tail()
```

Out[16]:

	StoreCode	StoreName	StoreType	Location	OperatingCost	Staff_Cnt	TotalSales	Tota
27	STR128	Super Market	Super Market	Montana	30.4	40	95.1	
28	STR129	Central Store	Super Market	Montana	15.8	80	351.0	
29	STR130	Apparel Zone	Apparel	Montana	19.7	60	145.0	
30	STR131	Fashion Bazar	Apparel	Montana	15.0	80	301.0	
31	STR132	Digital Bazar	Electronincs	Montana	21.4	40	121.0	

In [17]:
```
store.head(10)
```

Out[17]:

	StoreCode	StoreName	StoreType	Location	OperatingCost	Staff_Cnt	TotalSales	Tot
0	STR101	Electronics Zone	Electronincs	California	21.0	60	160.0	
1	STR102	Apparel Zone	Apparel	California	21.0	60	160.0	
2	STR103	Super Bazar	Super Market	California	22.8	40	108.0	
3	STR104	Super Market	Super Market	California	21.4	60	258.0	
4	STR105	Central Store	Super Market	California	18.7	80	360.0	
5	STR106	Apparel Zone	Apparel	California	18.1	60	225.0	
6	STR107	Fashion Bazar	Apparel	California	14.3	80	360.0	
7	STR108	Digital Bazar	Electronincs	California	24.4	40	146.7	
8	STR109	Electronics Zone	Electronincs	Washington	22.8	40	140.8	
9	STR110	Apparel Zone	Apparel	Washington	19.2	60	167.6	

In [18]:
```
store.tail(10)
```

Out[18]:

	StoreCode	StoreName	StoreType	Location	OperatingCost	Staff_Cnt	TotalSales	Tota
22	STR123	Fashion Bazar	Apparel	Texas	15.2	80	304.0	
23	STR124	Digital Bazar	Electronincs	Texas	13.3	80	350.0	
24	STR125	Electronics Zone	Electronincs	Montana	19.2	80	400.0	
25	STR126	Apparel Zone	Apparel	Montana	27.3	40	79.0	
26	STR127	Super Bazar	Super Market	Montana	26.0	40	120.3	
27	STR128	Super Market	Super Market	Montana	30.4	40	95.1	
28	STR129	Central Store	Super Market	Montana	15.8	80	351.0	
29	STR130	Apparel Zone	Apparel	Montana	19.7	60	145.0	
30	STR131	Fashion Bazar	Apparel	Montana	15.0	80	301.0	
31	STR132	Digital Bazar	Electronincs	Montana	21.4	40	121.0	

In [19]:
```
# to check no of rows and columns
store.shape
```

Out[19]: (32, 15)

In [20]:
```
store.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   StoreCode        32 non-null     object 
 1   StoreName        32 non-null     object 
 2   StoreType        32 non-null     object 
 3   Location         32 non-null     object 
 4   OperatingCost    32 non-null     float64
 5   Staff_Cnt        32 non-null     int64  
 6   TotalSales       32 non-null     float64
 7   Total_Customers  32 non-null     int64  
 8   AcqCostPercust   29 non-null     float64
 9   BasketSize       32 non-null     float64
 10  ProfitPercust    32 non-null     float64
 11  OwnStore         32 non-null     int64  
 12  OnlinePresence   32 non-null     int64  
 13  Tenure           32 non-null     int64  
 14  StoreSegment     32 non-null     int64  
dtypes: float64(5), int64(6), object(4)
memory usage: 3.9+ KB
```

In [21]:
```
store.index
```

Out[21]:
```
RangeIndex(start=0, stop=32, step=1)
```

In [22]:
```
store.columns
```

Out[22]:
```
Index(['StoreCode', 'StoreName', 'StoreType', 'Location', 'OperatingCos
t',
       'Staff_Cnt', 'TotalSales', 'Total_Customers', 'AcqCostPercust',
       'BasketSize', 'ProfitPercust', 'OwnStore', 'OnlinePresence', 'Tenu
re',
       'StoreSegment'],
      dtype='object')
```

In [23]:
```
store.dtypes
```

Out[23]:
```
StoreCode          object
StoreName          object
StoreType          object
Location           object
OperatingCost     float64
Staff_Cnt           int64
TotalSales        float64
Total_Customers     int64
AcqCostPercust    float64
BasketSize        float64
ProfitPercust     float64
OwnStore            int64
OnlinePresence      int64
Tenure              int64
StoreSegment        int64
dtype: object
```

In [24]:
```
# to show no of dimensions
store.ndim
```

Out[24]: 2

In [25]:
```
# it shows the statistical summary of the data
store.describe().T
```

Out[25]:

	count	mean	std	min	25%	50%	75%	max
OperatingCost	32.0	20.090625	6.026948	10.400	15.42500	19.200	22.80	33.900
Staff_Cnt	32.0	61.875000	17.859216	40.000	40.00000	60.000	80.00	80.000
TotalSales	32.0	230.721875	123.938694	71.100	120.82500	196.300	326.00	472.000
Total_Customers	32.0	146.687500	68.562868	52.000	96.50000	123.000	180.00	335.000
AcqCostPercust	29.0	3.651034	0.532664	2.760	3.15000	3.730	3.92	4.930
BasketSize	32.0	3.217250	0.978457	1.513	2.58125	3.325	3.61	5.424
ProfitPercust	32.0	17.848750	1.786943	14.500	16.89250	17.710	18.90	22.900
OwnStore	32.0	0.437500	0.504016	0.000	0.00000	0.000	1.00	1.000
OnlinePresence	32.0	0.406250	0.498991	0.000	0.00000	0.000	1.00	1.000
Tenure	32.0	3.687500	0.737804	3.000	3.00000	4.000	4.00	5.000
StoreSegment	32.0	2.625000	1.211504	1.000	2.00000	2.000	4.00	4.000

Accessing a column

In [26]:
```
store.head()
```

Out[26]:

	StoreCode	StoreName	StoreType	Location	OperatingCost	Staff_Cnt	TotalSales	Total_
0	STR101	Electronics Zone	Electronincs	California	21.0	60	160.0	
1	STR102	Apparel Zone	Apparel	California	21.0	60	160.0	
2	STR103	Super Bazar	Super Market	California	22.8	40	108.0	
3	STR104	Super Market	Super Market	California	21.4	60	258.0	
4	STR105	Central Store	Super Market	California	18.7	80	360.0	

In [27]:
```
store[['StoreName','TotalSales']]
```

Out[27]:

	StoreName	TotalSales
0	Electronics Zone	160.0
1	Apparel Zone	160.0
2	Super Bazar	108.0
3	Super Market	258.0
4	Central Store	360.0
5	Apparel Zone	225.0
6	Fashion Bazar	360.0
7	Digital Bazar	146.7
8	Electronics Zone	140.8
9	Apparel Zone	167.6
10	Super Bazar	167.6
11	Super Market	275.8
12	Central Store	275.8
13	Apparel Zone	275.8
14	Fashion Bazar	472.0
15	Digital Bazar	460.0
16	Electronics Zone	440.0
17	Apparel Zone	78.7
18	Super Bazar	75.7
19	Super Market	71.1
20	Central Store	120.1
21	Apparel Zone	318.0
22	Fashion Bazar	304.0
23	Digital Bazar	350.0
24	Electronics Zone	400.0
25	Apparel Zone	79.0
26	Super Bazar	120.3
27	Super Market	95.1
28	Central Store	351.0
29	Apparel Zone	145.0
30	Fashion Bazar	301.0
31	Digital Bazar	121.0

In [28]:
```
store.StoreName
```

Out[28]:
```
0     Electronics Zone
1         Apparel Zone
2          Super Bazar
3         Super Market
4        Central Store
5         Apparel Zone
6        Fashion Bazar
7        Digital Bazar
8     Electronics Zone
9         Apparel Zone
10         Super Bazar
11        Super Market
12       Central Store
13        Apparel Zone
14       Fashion Bazar
15       Digital Bazar
16    Electronics Zone
17        Apparel Zone
18         Super Bazar
19        Super Market
20       Central Store
21        Apparel Zone
22       Fashion Bazar
23       Digital Bazar
24    Electronics Zone
25        Apparel Zone
26         Super Bazar
27        Super Market
28       Central Store
29        Apparel Zone
30       Fashion Bazar
31       Digital Bazar
Name: StoreName, dtype: object
```

Accessing rows

```
.iloc() : integer based indexing
.loc : Label based indexing
```

In [29]:
```
store.head()
```

Out[29]:

	StoreCode	StoreName	StoreType	Location	OperatingCost	Staff_Cnt	TotalSales	Total
0	STR101	Electronics Zone	Electronincs	California	21.0	60	160.0	
1	STR102	Apparel Zone	Apparel	California	21.0	60	160.0	
2	STR103	Super Bazar	Super Market	California	22.8	40	108.0	
3	STR104	Super Market	Super Market	California	21.4	60	258.0	
4	STR105	Central Store	Super Market	California	18.7	80	360.0	

In [30]:
```
store.iloc[4:9]
```

Out[30]:

	StoreCode	StoreName	StoreType	Location	OperatingCost	Staff_Cnt	TotalSales	Tot
4	STR105	Central Store	Super Market	California	18.7	80	360.0	
5	STR106	Apparel Zone	Apparel	California	18.1	60	225.0	
6	STR107	Fashion Bazar	Apparel	California	14.3	80	360.0	
7	STR108	Digital Bazar	Electronincs	California	24.4	40	146.7	
8	STR109	Electronics Zone	Electronincs	Washington	22.8	40	140.8	

In [31]:
```
store.iloc[4:9,0:5]
```

Out[31]:

	StoreCode	StoreName	StoreType	Location	OperatingCost
4	STR105	Central Store	Super Market	California	18.7
5	STR106	Apparel Zone	Apparel	California	18.1
6	STR107	Fashion Bazar	Apparel	California	14.3
7	STR108	Digital Bazar	Electronincs	California	24.4
8	STR109	Electronics Zone	Electronincs	Washington	22.8

In [32]:
```
store2=store.set_index(store.StoreCode)
```

In [33]:
```
store2.head()
```

Out[33]:

StoreCode	StoreCode	StoreName	StoreType	Location	OperatingCost	Staff_Cnt	TotalSal
STR101	STR101	Electronics Zone	Electronincs	California	21.0	60	160
STR102	STR102	Apparel Zone	Apparel	California	21.0	60	160
STR103	STR103	Super Bazar	Super Market	California	22.8	40	108
STR104	STR104	Super Market	Super Market	California	21.4	60	258
STR105	STR105	Central Store	Super Market	California	18.7	80	360

In [34]:
```
store2.index
```

Out[34]:
```
Index(['STR101', 'STR102', 'STR103', 'STR104', 'STR105', 'STR106', 'STR10
7',
       'STR108', 'STR109', 'STR110', 'STR111', 'STR112', 'STR113', 'STR11
4',
       'STR115', 'STR116', 'STR117', 'STR118', 'STR119', 'STR120', 'STR12
1',
       'STR122', 'STR123', 'STR124', 'STR125', 'STR126', 'STR127', 'STR12
8',
       'STR129', 'STR130', 'STR131', 'STR132'],
      dtype='object', name='StoreCode')
```

In [35]:
```
store2.loc['STR109','OperatingCost']
```

Out[35]:
```
22.8
```

Calculated Column

In [36]:
```
store.head()
```

Out[36]:

	StoreCode	StoreName	StoreType	Location	OperatingCost	Staff_Cnt	TotalSales	Total
0	STR101	Electronics Zone	Electronincs	California	21.0	60	160.0	
1	STR102	Apparel Zone	Apparel	California	21.0	60	160.0	
2	STR103	Super Bazar	Super Market	California	22.8	40	108.0	
3	STR104	Super Market	Super Market	California	21.4	60	258.0	
4	STR105	Central Store	Super Market	California	18.7	80	360.0	

In [37]:
```
# Calculated Column : Column which has been added in the dataframe using sc
```

In [38]:
```
store['Total_Cost']=store.OperatingCost+store.AcqCostPercust*store.Total_Cu
```

In [39]:
```
store.head()
```

Out[39]:

	StoreCode	StoreName	StoreType	Location	OperatingCost	Staff_Cnt	TotalSales	Total
0	STR101	Electronics Zone	Electronincs	California	21.0	60	160.0	
1	STR102	Apparel Zone	Apparel	California	21.0	60	160.0	
2	STR103	Super Bazar	Super Market	California	22.8	40	108.0	
3	STR104	Super Market	Super Market	California	21.4	60	258.0	
4	STR105	Central Store	Super Market	California	18.7	80	360.0	

In [40]:
```
store['Region']=store.Location.str.upper()
```

In [41]:
```
store.head()
```

Out[41]:

	StoreCode	StoreName	StoreType	Location	OperatingCost	Staff_Cnt	TotalSales	Total
0	STR101	Electronics Zone	Electronincs	California	21.0	60	160.0	
1	STR102	Apparel Zone	Apparel	California	21.0	60	160.0	
2	STR103	Super Bazar	Super Market	California	22.8	40	108.0	
3	STR104	Super Market	Super Market	California	21.4	60	258.0	
4	STR105	Central Store	Super Market	California	18.7	80	360.0	

Rename

In [42]:
```
store.rename(columns={"Location":"State"},inplace=True)
```

In [43]:
```
store.head()
```

Out[43]:

	StoreCode	StoreName	StoreType	State	OperatingCost	Staff_Cnt	TotalSales	Total
0	STR101	Electronics Zone	Electronincs	California	21.0	60	160.0	
1	STR102	Apparel Zone	Apparel	California	21.0	60	160.0	
2	STR103	Super Bazar	Super Market	California	22.8	40	108.0	
3	STR104	Super Market	Super Market	California	21.4	60	258.0	
4	STR105	Central Store	Super Market	California	18.7	80	360.0	

Delete a variable

In [44]:
```
store.drop(columns=['Region'],inplace=True)
```

In [45]:
```
store.head()
```

Out[45]:

	StoreCode	StoreName	StoreType	State	OperatingCost	Staff_Cnt	TotalSales	Total
0	STR101	Electronics Zone	Electronincs	California	21.0	60	160.0	
1	STR102	Apparel Zone	Apparel	California	21.0	60	160.0	
2	STR103	Super Bazar	Super Market	California	22.8	40	108.0	
3	STR104	Super Market	Super Market	California	21.4	60	258.0	
4	STR105	Central Store	Super Market	California	18.7	80	360.0	

Handling duplicates

In [46]:
```
score=pd.read_csv("score.csv")
```

In [47]:
```
score
```

Out[47]:

	Student	Section	Test1	Test2	Final
0	Capalleti	1	94	91	87
1	Dubose	2	51	65	91
2	Engles	1	95	97	97
3	Grant	2	63	75	80
4	Krupski	2	80	76	71
5	Lundsford	1	92	40	86
6	Mcbane	1	75	78	72
7	Capalleti	1	94	65	87
8	Dubose	2	51	65	91
9	Engles	1	95	97	97
10	Grant	2	63	75	80
11	Krupski	2	80	76	71
12	Lundsford	1	92	40	86
13	Mcbane	1	75	78	72

In [48]:
```
# duplicate detection
score[score.duplicated()]
```

Out[48]:

	Student	Section	Test1	Test2	Final
8	Dubose	2	51	65	91
9	Engles	1	95	97	97
10	Grant	2	63	75	80
11	Krupski	2	80	76	71
12	Lundsford	1	92	40	86
13	Mcbane	1	75	78	72

In [49]:
```
# dropping duplicates
score1=score.drop_duplicates()
```

In [50]:
```
score1.shape
```

Out[50]: (8, 5)

In [51]:
```
score.shape
```

Out[51]: (14, 5)

Handling Missing Values

In [52]:
```
# NaN : Missing value
weather=pd.read_csv("weather_data1.csv")
```

In [53]:
```
weather
```

Out[53]:

	day	temperature	windspeed	event
0	1/1/2017	32.0	6.0	Rain
1	1/4/2017	NaN	9.0	Sunny
2	1/5/2017	28.0	NaN	Snow
3	1/6/2017	NaN	7.0	NaN
4	1/7/2017	32.0	NaN	Rain
5	1/8/2017	NaN	NaN	Sunny
6	1/9/2017	NaN	NaN	NaN
7	1/10/2017	34.0	8.0	Cloudy
8	1/11/2017	40.0	12.0	Sunny

In [54]:
```
# Identying missing values
weather.isnull().sum()
```

Out[54]:
```
day            0
temperature    4
windspeed      4
event          2
dtype: int64
```

In [55]:
```
weather.notnull().sum()
```

Out[55]:
```
day            9
temperature    5
windspeed      5
event          7
dtype: int64
```

In [56]:
```
# Missing value imputation : Replacing missing values with sensible informa
weather.fillna(0)
```

Out[56]:

	day	temperature	windspeed	event
0	1/1/2017	32.0	6.0	Rain
1	1/4/2017	0.0	9.0	Sunny
2	1/5/2017	28.0	0.0	Snow
3	1/6/2017	0.0	7.0	0
4	1/7/2017	32.0	0.0	Rain
5	1/8/2017	0.0	0.0	Sunny
6	1/9/2017	0.0	0.0	0
7	1/10/2017	34.0	8.0	Cloudy
8	1/11/2017	40.0	12.0	Sunny

In [57]:
```
weather.fillna({"temperature":0,"windspeed":0,"event":"No Event"})
```

Out[57]:

	day	temperature	windspeed	event
0	1/1/2017	32.0	6.0	Rain
1	1/4/2017	0.0	9.0	Sunny
2	1/5/2017	28.0	0.0	Snow
3	1/6/2017	0.0	7.0	No Event
4	1/7/2017	32.0	0.0	Rain
5	1/8/2017	0.0	0.0	Sunny
6	1/9/2017	0.0	0.0	No Event
7	1/10/2017	34.0	8.0	Cloudy
8	1/11/2017	40.0	12.0	Sunny

In [58]:
```
weather.fillna({"temperature":weather.temperature.mean(),"windspeed":weathe
```

Out[58]:

	day	temperature	windspeed	event
0	1/1/2017	32.0	6.0	Rain
1	1/4/2017	33.2	9.0	Sunny
2	1/5/2017	28.0	8.4	Snow
3	1/6/2017	33.2	7.0	No Event
4	1/7/2017	32.0	8.4	Rain
5	1/8/2017	33.2	8.4	Sunny
6	1/9/2017	33.2	8.4	No Event
7	1/10/2017	34.0	8.0	Cloudy
8	1/11/2017	40.0	12.0	Sunny

In [59]:
```
weather.fillna(method="ffill")
```

Out[59]:

	day	temperature	windspeed	event
0	1/1/2017	32.0	6.0	Rain
1	1/4/2017	32.0	9.0	Sunny
2	1/5/2017	28.0	9.0	Snow
3	1/6/2017	28.0	7.0	Snow
4	1/7/2017	32.0	7.0	Rain
5	1/8/2017	32.0	7.0	Sunny
6	1/9/2017	32.0	7.0	Sunny
7	1/10/2017	34.0	8.0	Cloudy
8	1/11/2017	40.0	12.0	Sunny

In [60]:
```
weather.fillna(method="bfill")
```

Out[60]:

	day	temperature	windspeed	event
0	1/1/2017	32.0	6.0	Rain
1	1/4/2017	28.0	9.0	Sunny
2	1/5/2017	28.0	7.0	Snow
3	1/6/2017	32.0	7.0	Rain
4	1/7/2017	32.0	8.0	Rain
5	1/8/2017	34.0	8.0	Sunny
6	1/9/2017	34.0	8.0	Cloudy
7	1/10/2017	34.0	8.0	Cloudy
8	1/11/2017	40.0	12.0	Sunny

In [61]:
```
weather
```

Out[61]:

	day	temperature	windspeed	event
0	1/1/2017	32.0	6.0	Rain
1	1/4/2017	NaN	9.0	Sunny
2	1/5/2017	28.0	NaN	Snow
3	1/6/2017	NaN	7.0	NaN
4	1/7/2017	32.0	NaN	Rain
5	1/8/2017	NaN	NaN	Sunny
6	1/9/2017	NaN	NaN	NaN
7	1/10/2017	34.0	8.0	Cloudy
8	1/11/2017	40.0	12.0	Sunny

In [62]:
```
weather1=weather.set_index(weather.day)
```

In [63]:
```
weather1
```

Out[63]:

	day	temperature	windspeed	event
day				
1/1/2017	1/1/2017	32.0	6.0	Rain
1/4/2017	1/4/2017	NaN	9.0	Sunny
1/5/2017	1/5/2017	28.0	NaN	Snow
1/6/2017	1/6/2017	NaN	7.0	NaN
1/7/2017	1/7/2017	32.0	NaN	Rain
1/8/2017	1/8/2017	NaN	NaN	Sunny
1/9/2017	1/9/2017	NaN	NaN	NaN
1/10/2017	1/10/2017	34.0	8.0	Cloudy
1/11/2017	1/11/2017	40.0	12.0	Sunny

In [64]:
```
weather.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9 entries, 0 to 8
Data columns (total 4 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   day          9 non-null      object 
 1   temperature  5 non-null      float64
 2   windspeed    5 non-null      float64
 3   event        7 non-null      object 
dtypes: float64(2), object(2)
memory usage: 420.0+ bytes
```

In [65]:
```
weather['day']=pd.to_datetime(weather.day)
```

In [66]:
```
weather.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9 entries, 0 to 8
Data columns (total 4 columns):
 #   Column       Non-Null Count  Dtype         
---  ------       --------------  -----         
 0   day          9 non-null      datetime64[ns]
 1   temperature  5 non-null      float64       
 2   windspeed    5 non-null      float64       
 3   event        7 non-null      object        
dtypes: datetime64[ns](1), float64(2), object(1)
memory usage: 420.0+ bytes
```

In [67]:
```
weather.set_index(weather.day,inplace=True)
```

In [68]:
```
weather.index
```

Out[68]:
```
DatetimeIndex(['2017-01-01', '2017-01-04', '2017-01-05', '2017-01-06',
               '2017-01-07', '2017-01-08', '2017-01-09', '2017-01-10',
               '2017-01-11'],
              dtype='datetime64[ns]', name='day', freq=None)
```

In [69]:
```
weather.interpolate(method='time')
```

```
---------------------------------------------------------------------
--
ValueError                                Traceback (most recent call las
t)
Cell In[69], line 1
----> 1 weather.interpolate(method='time')

File ~\anaconda3\Lib\site-packages\pandas\util\_decorators.py:331, in dep
recate_nonkeyword_arguments.<locals>.decorate.<locals>.wrapper(*args, **k
wargs)
    325 if len(args) > num_allow_args:
    326     warnings.warn(
    327         msg.format(arguments=_format_argument_list(allow_args)),
    328         FutureWarning,
    329         stacklevel=find_stack_level(),
    330     )
--> 331 return func(*args, **kwargs)

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:11855, in DataFra
me.interpolate(self, method, axis, limit, inplace, limit_direction, limit
_area, downcast, **kwargs)
  11843 @deprecate_nonkeyword_arguments(version=None, allowed_args=["sel
f", "method"])
  11844 def interpolate(
  11845     self: DataFrame,
   (...)
  11853     **kwargs,
  11854 ) -> DataFrame | None:
> 11855     return super().interpolate(
  11856         method,
  11857         axis,
  11858         limit,
  11859         inplace,
  11860         limit_direction,
  11861         limit_area,
  11862         downcast,
  11863         **kwargs,
  11864     )

File ~\anaconda3\Lib\site-packages\pandas\core\generic.py:7568, in NDFram
e.interpolate(self, method, axis, limit, inplace, limit_direction, limit_
area, downcast, **kwargs)
   7562 if isna(index).any():
   7563     raise NotImplementedError(
   7564         "Interpolation with NaNs in the index "
   7565         "has not been implemented. Try filling "
   7566         "those NaNs before interpolating."
   7567     )
-> 7568 new_data = obj._mgr.interpolate(
   7569     method=method,
   7570     axis=axis,
   7571     index=index,
   7572     limit=limit,
   7573     limit_direction=limit_direction,
   7574     limit_area=limit_area,
   7575     inplace=inplace,
   7576     downcast=downcast,
   7577     **kwargs,
   7578 )
   7580 result = self._constructor(new_data)
   7581 if should_transpose:
```

```
File ~\anaconda3\Lib\site-packages\pandas\core\internals\managers.py:422,
in BaseBlockManager.interpolate(self, **kwargs)
    421 def interpolate(self: T, **kwargs) -> T:
--> 422     return self.apply("interpolate", **kwargs)

File ~\anaconda3\Lib\site-packages\pandas\core\internals\managers.py:352,
in BaseBlockManager.apply(self, f, align_keys, ignore_failures, **kwargs)
    350         applied = b.apply(f, **kwargs)
    351     else:
--> 352         applied = getattr(b, f)(**kwargs)
    353 except (TypeError, NotImplementedError):
    354     if not ignore_failures:

File ~\anaconda3\Lib\site-packages\pandas\core\internals\blocks.py:1619,
in EABackedBlock.interpolate(self, method, axis, inplace, limit, fill_val
ue, **kwargs)
   1617     new_values = values.T.fillna(value=fill_value, method=method,
limit=limit).T
   1618 else:
-> 1619     new_values = values.fillna(value=fill_value, method=method, l
imit=limit)
   1620 return self.make_block_same_class(new_values)

File ~\anaconda3\Lib\site-packages\pandas\core\arrays\_mixins.py:317, in
NDArrayBackedExtensionArray.fillna(self, value, method, limit)
    313 @doc(ExtensionArray.fillna)
    314 def fillna(
    315     self: NDArrayBackedExtensionArrayT, value=None, method=None,
limit=None
    316 ) -> NDArrayBackedExtensionArrayT:
--> 317     value, method = validate_fillna_kwargs(
    318         value, method, validate_scalar_dict_value=False
    319     )
    321     mask = self.isna()
    322     # error: Argument 2 to "check_value_size" has incompatible ty
pe
    323     # "ExtensionArray"; expected "ndarray"

File ~\anaconda3\Lib\site-packages\pandas\util\_validators.py:390, in val
idate_fillna_kwargs(value, method, validate_scalar_dict_value)
    388     raise ValueError("Must specify a fill 'value' or 'method'.")
    389 elif value is None and method is not None:
--> 390     method = clean_fill_method(method)
    392 elif value is not None and method is None:
    393     if validate_scalar_dict_value and isinstance(value, (list, tu
ple)):

File ~\anaconda3\Lib\site-packages\pandas\core\missing.py:125, in clean_f
ill_method(method, allow_nearest)
    123     expecting = "pad (ffill), backfill (bfill) or nearest"
    124 if method not in valid_methods:
--> 125     raise ValueError(f"Invalid fill method. Expecting {expectin
g}. Got {method}")
    126 return method

ValueError: Invalid fill method. Expecting pad (ffill) or backfill (bfil
l). Got time
```

In [ ]:
```
weather
```

In [ ]:
```
weather.dropna(axis=1,how="all")
```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```