Pandas_opn

August 19, 2022

[1]:
```
import pandas as pd
import numpy as np
#tab is used to read get hints on command
```

[2]:
```
datadf=pd.read_csv("k_circle_sales (1).csv")
datadf
```

[2]:

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	\
0	FDA15	9.300	Low Fat	0.016047	
1	DRC01	5.920	Regular	0.019278	
2	FDN15	17.500	Low Fat	0.016760	
3	FDX07	19.200	Regular	0.000000	
4	NCD19	8.930	Low Fat	0.000000	
…	…	…	…	…	
8518	FDF22	6.865	Low Fat	0.056783	
8519	FDS36	8.380	Regular	0.046982	
8520	NCJ29	10.600	Low Fat	0.035186	
8521	FDN46	7.210	Regular	0.145221	
8522	DRG01	14.800	Low Fat	0.044878	

	Item_Type	Item_MRP	Outlet_Identifier	\
0	Dairy	249.8	OUT049	
1	Soft Drinks	48.3	OUT018	
2	Meat	141.6	OUT049	
3	Fruits and Vegetables	182.1	OUT010	
4	Household	53.9	OUT013	
…	…	…	…	
8518	Snack Foods	214.5	OUT013	
8519	Baking Goods	108.2	OUT045	
8520	Health and Hygiene	85.1	OUT035	
8521	Snack Foods	103.1	OUT018	
8522	Soft Drinks	75.5	OUT046	

	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Type	\
0	1999	Medium	Tier 2	
1	2009	Medium	Tier 2	
2	1999	Medium	Tier 2	

```
3                          1998         NaN                  NaN
4                          1987        High              Tier 3
…                           …           …                    …
8518                       1987        High              Tier 3
8519                       2002         NaN                 NaN
8520                       2004       Small               Tier1
8521                       2009      Medium              Tier 2
8522                       1997       Small               Tier1

            Outlet_Type  Item_Outlet_Sales  Profit
0     Supermarket Type1          3735.1380    11.5
1     Supermarket Type2           443.4228    14.3
2     Supermarket Type1          2097.2700    14.5
3         Grocery Store           732.3800    13.6
4     Supermarket Type1           994.7052    14.1
…                  …               …     …
8518  Supermarket Type1          2778.3834    14.1
8519  Supermarket Type1           549.2850    14.2
8520  Supermarket Type1          1193.1136     9.5
8521  Supermarket Type2          1845.5976    14.2
8522  Supermarket Type1           765.6700    14.6

[8523 rows x 13 columns]
```

[3]:
```
datadf.head() # gives us the first 5 rows
```

[3]:
```
  Item_Identifier  Item_Weight Item_Fat_Content  Item_Visibility  \
0           FDA15         9.30          Low Fat         0.016047
1           DRC01         5.92          Regular         0.019278
2           FDN15        17.50          Low Fat         0.016760
3           FDX07        19.20          Regular         0.000000
4           NCD19         8.93          Low Fat         0.000000

              Item_Type  Item_MRP Outlet_Identifier  \
0                 Dairy     249.8            OUT049
1           Soft Drinks      48.3            OUT018
2                  Meat     141.6            OUT049
3  Fruits and Vegetables     182.1            OUT010
4             Household      53.9            OUT013

   Outlet_Establishment_Year Outlet_Size Outlet_Location_Type  \
0                       1999      Medium               Tier 2
1                       2009      Medium               Tier 2
2                       1999      Medium               Tier 2
3                       1998         NaN                  NaN
4                       1987        High              Tier 3
```

```
         Outlet_Type  Item_Outlet_Sales  Profit
0  Supermarket Type1          3735.1380    11.5
1  Supermarket Type2           443.4228    14.3
2  Supermarket Type1          2097.2700    14.5
3      Grocery Store           732.3800    13.6
4  Supermarket Type1           994.7052    14.1
```

[4]:
```
datadf.head(2)
```

[4]:
```
  Item_Identifier  Item_Weight Item_Fat_Content  Item_Visibility    Item_Type  \
0           FDA15         9.30          Low Fat         0.016047        Dairy
1           DRC01         5.92          Regular         0.019278  Soft Drinks

   Item_MRP Outlet_Identifier  Outlet_Establishment_Year Outlet_Size  \
0     249.8            OUT049                       1999      Medium
1      48.3            OUT018                       2009      Medium

  Outlet_Location_Type        Outlet_Type  Item_Outlet_Sales  Profit
0               Tier 2  Supermarket Type1          3735.1380    11.5
1               Tier 2  Supermarket Type2           443.4228    14.3
```

[5]:
```
datadf.tail() #gives us lat 5 values
```

[5]:
```
     Item_Identifier  Item_Weight Item_Fat_Content  Item_Visibility  \
8518           FDF22        6.865          Low Fat         0.056783
8519           FDS36        8.380          Regular         0.046982
8520           NCJ29       10.600          Low Fat         0.035186
8521           FDN46        7.210          Regular         0.145221
8522           DRG01       14.800          Low Fat         0.044878

                Item_Type  Item_MRP Outlet_Identifier  \
8518         Snack Foods     214.5            OUT013
8519        Baking Goods     108.2            OUT045
8520  Health and Hygiene      85.1            OUT035
8521         Snack Foods     103.1            OUT018
8522         Soft Drinks      75.5            OUT046

      Outlet_Establishment_Year Outlet_Size Outlet_Location_Type  \
8518                       1987        High               Tier 3
8519                       2002         NaN                  NaN
8520                       2004       Small               Tier1
8521                       2009      Medium               Tier 2
8522                       1997       Small               Tier1

            Outlet_Type  Item_Outlet_Sales  Profit
8518  Supermarket Type1          2778.3834    14.1
8519  Supermarket Type1           549.2850    14.2
```

```
8520  Supermarket Type1          1193.1136     9.5
8521  Supermarket Type2          1845.5976    14.2
8522  Supermarket Type1           765.6700    14.6
```

[6]:
```
datadf.tail(2)
```

[6]:
```
     Item_Identifier  Item_Weight Item_Fat_Content  Item_Visibility  \
8521           FDN46         7.21          Regular         0.145221
8522           DRG01        14.80          Low Fat         0.044878

        Item_Type  Item_MRP Outlet_Identifier  Outlet_Establishment_Year  \
8521  Snack Foods     103.1            OUT018                       2009
8522  Soft Drinks      75.5            OUT046                       1997

     Outlet_Size Outlet_Location_Type        Outlet_Type  Item_Outlet_Sales  \
8521      Medium               Tier 2  Supermarket Type2          1845.5976
8522       Small                Tier1  Supermarket Type1           765.6700

      Profit
8521    14.2
8522    14.6
```

[7]:
```
datadf.info() #gives complete info about the data from this we can infer if␣
↪there is any missing data
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 13 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Item_Identifier            8523 non-null   object
 1   Item_Weight                7775 non-null   float64
 2   Item_Fat_Content           8523 non-null   object
 3   Item_Visibility            8523 non-null   float64
 4   Item_Type                  8523 non-null   object
 5   Item_MRP                   8523 non-null   float64
 6   Outlet_Identifier          8523 non-null   object
 7   Outlet_Establishment_Year  8523 non-null   int64
 8   Outlet_Size                6113 non-null   object
 9   Outlet_Location_Type       6473 non-null   object
 10  Outlet_Type                8523 non-null   object
 11  Item_Outlet_Sales          8523 non-null   float64
 12  Profit                     8523 non-null   float64
dtypes: float64(5), int64(1), object(7)
memory usage: 865.7+ KB
```

[8]:
```
datadf.isnull() # gives table with f and t f--data is available t--data not␣
↪available
```

[8]:

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	\
0	False	False	False	False	
1	False	False	False	False	
2	False	False	False	False	
3	False	False	False	False	
4	False	False	False	False	
…	…	…	…	…	
8518	False	False	False	False	
8519	False	False	False	False	
8520	False	False	False	False	
8521	False	False	False	False	
8522	False	False	False	False	

	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	\
0	False	False	False	False	
1	False	False	False	False	
2	False	False	False	False	
3	False	False	False	False	
4	False	False	False	False	
…	…	…	…	…	
8518	False	False	False	False	
8519	False	False	False	False	
8520	False	False	False	False	
8521	False	False	False	False	
8522	False	False	False	False	

	Outlet_Size	Outlet_Location_Type	Outlet_Type	Item_Outlet_Sales	\
0	False	False	False	False	
1	False	False	False	False	
2	False	False	False	False	
3	True	True	False	False	
4	False	False	False	False	
…	…	…	…	…	
8518	False	False	False	False	
8519	True	True	False	False	
8520	False	False	False	False	
8521	False	False	False	False	
8522	False	False	False	False	

	Profit
0	False
1	False
2	False
3	False

```
4       False
…       …
8518    False
8519    False
8520    False
8521    False
8522    False

[8523 rows x 13 columns]
```

[9]:
```
datadf.isnull().sum() #-- toget sum of total null values in that column
```

[9]:
```
Item_Identifier               0
Item_Weight                 748
Item_Fat_Content              0
Item_Visibility               0
Item_Type                     0
Item_MRP                      0
Outlet_Identifier             0
Outlet_Establishment_Year     0
Outlet_Size                2410
Outlet_Location_Type       2050
Outlet_Type                   0
Item_Outlet_Sales             0
Profit                        0
dtype: int64
```

[10]:
```
#how to handle null values
x=datadf.copy() #--copying whole df into another variable
#drop null values
x.dropna() #--drops all the rows with null value but it will not permanent␣
 ↪change in df we need to dave it in another variable
```

[10]:
```
     Item_Identifier  Item_Weight Item_Fat_Content  Item_Visibility  \
0              FDA15        9.300          Low Fat         0.016047
1              DRC01        5.920          Regular         0.019278
2              FDN15       17.500          Low Fat         0.016760
4              NCD19        8.930          Low Fat         0.000000
5              FDP36       10.395          Regular         0.000000
…              …           …               …               …
8517           FDF53       20.750              reg         0.083607
8518           FDF22        6.865          Low Fat         0.056783
8520           NCJ29       10.600          Low Fat         0.035186
8521           FDN46        7.210          Regular         0.145221
8522           DRG01       14.800          Low Fat         0.044878

               Item_Type  Item_MRP Outlet_Identifier  \
```

```
0                  Dairy     249.8            OUT049
1            Soft Drinks      48.3            OUT018
2                   Meat     141.6            OUT049
4              Household      53.9            OUT013
5           Baking Goods      51.4            OUT018
…                      …       …               …
8517        Frozen Foods     178.8            OUT046
8518         Snack Foods     214.5            OUT013
8520  Health and Hygiene      85.1            OUT035
8521         Snack Foods     103.1            OUT018
8522         Soft Drinks      75.5            OUT046

      Outlet_Establishment_Year Outlet_Size Outlet_Location_Type  \
0                          1999      Medium               Tier 2
1                          2009      Medium               Tier 2
2                          1999      Medium               Tier 2
4                          1987        High               Tier 3
5                          2009      Medium               Tier 2
…                           …          …                    …
8517                       1997       Small                Tier1
8518                       1987        High               Tier 3
8520                       2004       Small                Tier1
8521                       2009      Medium               Tier 2
8522                       1997       Small                Tier1

            Outlet_Type  Item_Outlet_Sales  Profit
0     Supermarket Type1          3735.1380    11.5
1     Supermarket Type2           443.4228    14.3
2     Supermarket Type1          2097.2700    14.5
4     Supermarket Type1           994.7052    14.1
5     Supermarket Type2           556.6088     9.5
…                    …                …     …
8517  Supermarket Type1          3608.6360    13.2
8518  Supermarket Type1          2778.3834    14.1
8520  Supermarket Type1          1193.1136     9.5
8521  Supermarket Type2          1845.5976    14.2
8522  Supermarket Type1           765.6700    14.6

[5365 rows x 13 columns]
```

[11]:
```
x.isnull().sum()
```

[11]:
```
Item_Identifier               0
Item_Weight                 748
Item_Fat_Content              0
Item_Visibility               0
Item_Type                     0
```

```
Item_MRP                        0
Outlet_Identifier               0
Outlet_Establishment_Year       0
Outlet_Size                  2410
Outlet_Location_Type         2050
Outlet_Type                     0
Item_Outlet_Sales               0
Profit                          0
dtype: int64
```

[12]:
```
x.dropna(inplace=True) # to permanently drop null value and store it in same␣
↪variable
```

[13]:
```
x.isnull().sum()
```

[13]:
```
Item_Identifier              0
Item_Weight                  0
Item_Fat_Content             0
Item_Visibility              0
Item_Type                    0
Item_MRP                     0
Outlet_Identifier            0
Outlet_Establishment_Year    0
Outlet_Size                  0
Outlet_Location_Type         0
Outlet_Type                  0
Item_Outlet_Sales            0
Profit                       0
dtype: int64
```

[14]:
```
x.shape
```

[14]:
```
(5365, 13)
```

[15]:
```
y=datadf.copy()
```

[16]:
```
#fillna method to fill values with some constant
y.fillna(value=0,inplace=True)
```

[17]:
```
y.isnull().sum()
```

[17]:
```
Item_Identifier              0
Item_Weight                  0
Item_Fat_Content             0
Item_Visibility              0
Item_Type                    0
Item_MRP                     0
```

```
Outlet_Identifier            0
Outlet_Establishment_Year    0
Outlet_Size                  0
Outlet_Location_Type         0
Outlet_Type                  0
Item_Outlet_Sales            0
Profit                       0
dtype: int64
```

[18]:
```
y.shape
```

[18]:
```
(8523, 13)
```

[19]:
```
datadf.isnull().sum()
```

[19]:
```
Item_Identifier               0
Item_Weight                 748
Item_Fat_Content              0
Item_Visibility               0
Item_Type                     0
Item_MRP                      0
Outlet_Identifier             0
Outlet_Establishment_Year     0
Outlet_Size                2410
Outlet_Location_Type       2050
Outlet_Type                   0
Item_Outlet_Sales             0
Profit                        0
dtype: int64
```

[20]:
```
#replace with mean
datadf['Item_Weight'].fillna(datadf['Item_Weight'].mean(),inplace=True)
```

[21]:
```
datadf.isnull().sum()
```

[21]:
```
Item_Identifier               0
Item_Weight                   0
Item_Fat_Content              0
Item_Visibility               0
Item_Type                     0
Item_MRP                      0
Outlet_Identifier             0
Outlet_Establishment_Year     0
Outlet_Size                2410
Outlet_Location_Type       2050
Outlet_Type                   0
Item_Outlet_Sales             0
```

```
Profit                          0
dtype: int64
```

[22]:
```
datadf['Outlet_Size'].mode() #--mode will always return with index and while␣
↪using it with fillna, always check the mode() op
```

[22]:
```
0    Medium
dtype: object
```

[23]:
```
#replace with mode
datadf['Outlet_Size'].fillna(datadf['Outlet_Size'].mode()[0],inplace=True)␣
↪#mode()[0] to access a particular mode if there are equal no of accurences␣
↪ie 2 or 3 diff modes, cause the mode() will return all the modes but we need␣
↪to choose only one to fill our Nan
```

[24]:
```
datadf.isnull().sum()
```

[24]:
```
Item_Identifier                 0
Item_Weight                     0
Item_Fat_Content                0
Item_Visibility                 0
Item_Type                       0
Item_MRP                        0
Outlet_Identifier               0
Outlet_Establishment_Year       0
Outlet_Size                     0
Outlet_Location_Type         2050
Outlet_Type                     0
Item_Outlet_Sales               0
Profit                          0
dtype: int64
```

[25]:
```
datadf.columns
```

[25]:
```
Index(['Item_Identifier', 'Item_Weight', 'Item_Fat_Content', 'Item_Visibility',
       'Item_Type', 'Item_MRP', 'Outlet_Identifier',
       'Outlet_Establishment_Year', 'Outlet_Size', 'Outlet_Location_Type',
       'Outlet_Type', 'Item_Outlet_Sales', 'Profit'],
      dtype='object')
```

[26]:
```
#to find number of unique values(categories) inside a column
datadf['Item_Fat_Content'].nunique()   #-- gives no of unique categories of data
```

[26]:
```
5
```

[27]:
```
# to get different categorical values inside column
datadf['Item_Fat_Content'].unique()  #-- gives value of unique categories
```

[27]:
```
array(['Low Fat', 'Regular', 'low fat', 'LF', 'reg'], dtype=object)
```

[28]:
```
# to get no of unique values in a column
datadf['Item_Fat_Content'].value_counts() #--gives us everything individually
```

[28]:
```
Low Fat    5089
Regular    2889
LF          316
reg         117
low fat     112
Name: Item_Fat_Content, dtype: int64
```

[29]:
```
#access elements, display item fat content, item time, mrp
datadf.loc[:,['Item_Fat_Content','Item_Type','Item_MRP']]
```

[29]:

	Item_Fat_Content	Item_Type	Item_MRP
0	Low Fat	Dairy	249.8
1	Regular	Soft Drinks	48.3
2	Low Fat	Meat	141.6
3	Regular	Fruits and Vegetables	182.1
4	Low Fat	Household	53.9
…	…	…	…
8518	Low Fat	Snack Foods	214.5
8519	Regular	Baking Goods	108.2
8520	Low Fat	Health and Hygiene	85.1
8521	Regular	Snack Foods	103.1
8522	Low Fat	Soft Drinks	75.5

[8523 rows x 3 columns]

[30]:
```
#display 25 rows of item type, mrp,profit
datadf.loc[0:25,['Item_Type','Item_MRP','Profit']]
```

[30]:

	Item_Type	Item_MRP	Profit
0	Dairy	249.8	11.5
1	Soft Drinks	48.3	14.3
2	Meat	141.6	14.5
3	Fruits and Vegetables	182.1	13.6
4	Household	53.9	14.1
5	Baking Goods	51.4	9.5
6	Snack Foods	57.7	8.8
7	Snack Foods	107.8	13.3
8	Frozen Foods	97.0	13.0
9	Frozen Foods	187.8	13.6
10	Fruits and Vegetables	45.5	14.3
11	Dairy	144.1	14.3
12	Fruits and Vegetables	145.5	14.6

```
13            Snack Foods     119.7    14.1
14  Fruits and Vegetables     196.4    13.5
15              Breakfast      56.4    13.0
16     Health and Hygiene     115.3     8.6
17              Breakfast      54.4    14.8
18            Hard Drinks     113.3    14.4
19                  Dairy     230.5    14.0
20            Snack Foods     250.9    14.6
21           Baking Goods     144.5    14.7
22              Household     196.5    14.4
23           Baking Goods     107.7    13.7
24           Frozen Foods     165.0    14.3
25              Household      45.9    14.3
```

[31]:
```
#display 25 rows of item type, mrp,profit
datadf[['Item_Type','Item_MRP','Profit']].head(25)
```

[31]:
```
                Item_Type  Item_MRP  Profit
0                   Dairy     249.8    11.5
1             Soft Drinks      48.3    14.3
2                    Meat     141.6    14.5
3   Fruits and Vegetables     182.1    13.6
4               Household      53.9    14.1
5            Baking Goods      51.4     9.5
6             Snack Foods      57.7     8.8
7             Snack Foods     107.8    13.3
8            Frozen Foods      97.0    13.0
9            Frozen Foods     187.8    13.6
10  Fruits and Vegetables      45.5    14.3
11                  Dairy     144.1    14.3
12  Fruits and Vegetables     145.5    14.6
13            Snack Foods     119.7    14.1
14  Fruits and Vegetables     196.4    13.5
15              Breakfast      56.4    13.0
16     Health and Hygiene     115.3     8.6
17              Breakfast      54.4    14.8
18            Hard Drinks     113.3    14.4
19                  Dairy     230.5    14.0
20            Snack Foods     250.9    14.6
21           Baking Goods     144.5    14.7
22              Household     196.5    14.4
23           Baking Goods     107.7    13.7
24           Frozen Foods     165.0    14.3
```

[32]:
```
datadf.loc[[28,50,89,90,200],['Item_Type','Item_MRP','Profit']]
```

[32]:

	Item_Type	Item_MRP	Profit
28	Dairy	45.5	14.0
50	Health and Hygiene	143.9	7.4
89	Meat	117.3	14.0
90	Fruits and Vegetables	122.2	14.2
200	Health and Hygiene	104.6	14.5

[33]:
```
#sorting
#display data high to los using item_mrp
datadf.sort_values(by=['Item_MRP'],ascending=False)
```

[33]:

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	\
4802	FDS13	6.4650	Low Fat	0.125210	
5884	FDR25	17.0000	Regular	0.139522	
4159	FDK51	19.8500	Low Fat	0.008763	
5254	NCS29	9.0000	Low Fat	0.069488	
5558	NCS29	9.0000	Low Fat	0.069532	
…	…	…	…	…	
2242	FDG40	11.6881	Low Fat	0.039631	
2011	FDX59	10.1950	Low Fat	0.051618	
2135	DRK12	9.5000	Low Fat	0.042057	
6958	DRK12	9.5000	Low Fat	0.041851	
153	DRK12	11.6881	Low Fat	0.041683	

	Item_Type	Item_MRP	Outlet_Identifier	\
4802	Canned	266.9	OUT017	
5884	Canned	266.9	OUT046	
4159	Dairy	266.7	OUT010	
5254	Health and Hygiene	266.7	OUT013	
5558	Health and Hygiene	266.6	OUT035	
…	…	…	…	
2242	Frozen Foods	32.0	OUT027	
2011	Breads	32.0	OUT013	
2135	Soft Drinks	31.9	OUT018	
6958	Soft Drinks	31.5	OUT013	
153	Soft Drinks	31.3	OUT027	

	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Type	\
4802	2007	Medium	NaN	
5884	1997	Small	Tier1	
4159	1998	Medium	?	
5254	1987	High	Tier 3	
5558	2004	Small	Tier1	
…	…	…	…	
2242	1985	Medium	Tier 2	
2011	1987	High	Tier 3	
2135	2009	Medium	Tier 2	

```
6958                      1987        High              Tier 3
153                       1985      Medium              Tier 2

            Outlet_Type  Item_Outlet_Sales  Profit
4802  Supermarket Type1          1059.9536    13.2
5884  Supermarket Type1          5034.7796    13.2
4159      Grocery Store           264.9884    14.1
5254  Supermarket Type1          3974.8260    14.2
5558  Supermarket Type1          6624.7100    14.1
…                   …                …     …
2242  Supermarket Type3           984.7182    14.0
2011  Supermarket Type1           373.5138    14.3
2135  Supermarket Type2           366.1900    14.4
6958  Supermarket Type1           466.0600    12.2
153   Supermarket Type3           898.8300    11.7

[8523 rows x 13 columns]
```

[34]:
```
datadf.sort_values(by=['Item_MRP'],ascending=True).head()
```

[34]:
```
     Item_Identifier  Item_Weight Item_Fat_Content  Item_Visibility  \
153            DRK12      11.6881          Low Fat         0.041683
6958           DRK12       9.5000          Low Fat         0.041851
2135           DRK12       9.5000          Low Fat         0.042057
2242           FDG40      11.6881          Low Fat         0.039631
2011           FDX59      10.1950          Low Fat         0.051618

         Item_Type  Item_MRP Outlet_Identifier  Outlet_Establishment_Year  \
153    Soft Drinks      31.3            OUT027                       1985
6958   Soft Drinks      31.5            OUT013                       1987
2135   Soft Drinks      31.9            OUT018                       2009
2242  Frozen Foods      32.0            OUT027                       1985
2011        Breads      32.0            OUT013                       1987

     Outlet_Size Outlet_Location_Type        Outlet_Type  Item_Outlet_Sales  \
153       Medium               Tier 2  Supermarket Type3           898.8300
6958        High               Tier 3  Supermarket Type1           466.0600
2135      Medium               Tier 2  Supermarket Type2           366.1900
2242      Medium               Tier 2  Supermarket Type3           984.7182
2011        High               Tier 3  Supermarket Type1           373.5138

      Profit
153     11.7
6958    12.2
2135    14.4
2242    14.0
2011    14.3
```

[35]:
```
datadf.sort_values(by=['Item_Weight','Item_MRP'],ascending=[False,False]).
↪head(5) #for multiple column sorting pririty is given to the first var then␣
↪if all first var all eq then it looks for second var, also use two valus for␣
↪ascending
```

[35]:

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	\
229	FDX25	100.00	Low Fat	0.101562	Canned	
2802	FDC02	21.35	Low Fat	0.068765	Canned	
43	FDC02	21.35	Low Fat	0.069103	Canned	
2368	FDC02	21.35	Low Fat	0.068809	Canned	
483	FDC02	21.35	Low Fat	0.115195	Canned	

	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	Outlet_Size	\
229	181.9	OUT027	1985	Medium	
2802	260.4	OUT013	1987	High	
43	259.9	OUT018	2009	Medium	
2368	258.5	OUT035	2004	Small	
483	258.3	OUT010	1998	Medium	

	Outlet_Location_Type	Outlet_Type	Item_Outlet_Sales	Profit
229	Tier 2	Supermarket Type3	3101.2964	13.3
2802	Tier 3	Supermarket Type1	3644.5892	14.0
43	Tier 2	Supermarket Type2	6768.5228	12.4
2368	Tier1	Supermarket Type1	5206.5560	12.7
483	--	Grocery Store	520.6556	14.1

[36]:
```
datadf.sort_values(by=['Item_Weight','Item_MRP'],ascending=[False,True]).head(5)
```

[36]:

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	\
229	FDX25	100.00	Low Fat	0.101562	
6389	FDR07	21.35	Low Fat	0.078061	
4257	FDR07	21.35	Low Fat	0.130127	
483	FDC02	21.35	Low Fat	0.115195	
4468	FDC02	21.35	Low Fat	0.068822	

	Item_Type	Item_MRP	Outlet_Identifier	\
229	Canned	181.9	OUT027	
6389	Fruits and Vegetables	96.0	OUT018	
4257	Fruits and Vegetables	96.2	OUT010	
483	Canned	258.3	OUT010	
4468	Canned	258.3	OUT046	

	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Type	\
229	1985	Medium	Tier 2	
6389	2009	Medium	Tier 2	
4257	1998	Medium	NaN	
483	1998	Medium	--	

```
4468                      1997       Small                Tier1

            Outlet_Type  Item_Outlet_Sales  Profit
229   Supermarket Type3          3101.2964    13.3
6389  Supermarket Type2           380.8376    14.0
4257      Grocery Store           190.4188    14.7
483       Grocery Store           520.6556    14.1
4468  Supermarket Type1          7028.8506    13.8
```

[37]:
```
#Filtering
#want to know sales greater than 3000
datadf[datadf['Item_Outlet_Sales']>3000]
```

[37]:
```
     Item_Identifier  Item_Weight Item_Fat_Content  Item_Visibility  \
0              FDA15       9.3000          Low Fat         0.016047
7              FDP10      11.6881          Low Fat         0.127470
9              FDU28      19.2000          Regular         0.094450
20             FDN22      18.8500          Regular         0.138190
21             FDW12      11.6881          Regular         0.035400
…                …            …                …                …
8504           NCN18       0.0000          Low Fat         0.124111
8506           DRF37      17.2500          Low Fat         0.084676
8510           FDN58      13.8000          Regular         0.056862
8511           FDF05      17.5000          Low Fat         0.026980
8517           FDF53      20.7500              reg         0.083607

         Item_Type  Item_MRP Outlet_Identifier  Outlet_Establishment_Year  \
0            Dairy     249.8            OUT049                       1999
7      Snack Foods     107.8            OUT027                       1985
9     Frozen Foods     187.8            OUT017                       2007
20     Snack Foods     250.9            OUT013                       1987
21    Baking Goods     144.5            OUT027                       1985
…              …       …                …                            …
8504     Household     111.8            OUT027                       1985
8506   Soft Drinks     263.2            OUT018                       2009
8510   Snack Foods     231.6            OUT035                       2004
8511  Frozen Foods     262.6            OUT018                       2009
8517  Frozen Foods     178.8            OUT046                       1997

     Outlet_Size Outlet_Location_Type        Outlet_Type  Item_Outlet_Sales  \
0         Medium               Tier 2  Supermarket Type1          3735.1380
7         Medium               Tier 2  Supermarket Type3          4022.7636
9         Medium                   --  Supermarket Type1          4710.5350
20          High               Tier 3  Supermarket Type1          3775.0860
21        Medium               Tier 2  Supermarket Type3          4064.0432
…           …                     …                …                …
8504      Medium               Tier 2  Supermarket Type3          4138.6128
```

```
8506      Medium              Tier 2  Supermarket Type2          3944.8650
8510       Small               Tier1  Supermarket Type1          7182.6504
8511      Medium              Tier 2  Supermarket Type2          4207.8560
8517       Small               Tier1  Supermarket Type1          3608.6360

      Profit
0       11.5
7       13.3
9       13.6
20      14.6
21      14.7
…       …
8504    14.1
8506    14.3
8510    11.8
8511    14.8
8517    13.2

[2266 rows x 13 columns]
```

[38]:
```
#display medium outlet size data only
datadf[datadf['Outlet_Size']=='Medium']
```

[38]:
```
     Item_Identifier  Item_Weight Item_Fat_Content  Item_Visibility  \
0              FDA15        9.300          Low Fat         0.016047
1              DRC01        5.920          Regular         0.019278
2              FDN15       17.500          Low Fat         0.016760
3              FDX07       19.200          Regular         0.000000
5              FDP36       10.395          Regular         0.000000
…              …           …               …               …
8514           FDA01       15.000          Regular         0.054489
8515           FDH24       20.700          Low Fat         0.021518
8516           NCJ19       18.600          Low Fat         0.118661
8519           FDS36        8.380          Regular         0.046982
8521           FDN46        7.210          Regular         0.145221

                   Item_Type  Item_MRP Outlet_Identifier  \
0                      Dairy     249.8            OUT049
1                Soft Drinks      48.3            OUT018
2                       Meat     141.6            OUT049
3     Fruits and Vegetables     182.1            OUT010
5              Baking Goods      51.4            OUT018
…                      …       …               …
8514                 Canned      57.6            OUT045
8515           Baking Goods     157.5            OUT018
8516                 Others      58.8            OUT018
8519           Baking Goods     108.2            OUT045
```

```
8521            Snack Foods     103.1            OUT018

      Outlet_Establishment_Year Outlet_Size Outlet_Location_Type  \
0                          1999      Medium               Tier 2
1                          2009      Medium               Tier 2
2                          1999      Medium               Tier 2
3                          1998      Medium                  NaN
5                          2009      Medium               Tier 2
…                           …         …                    …
8514                       2002      Medium                  NaN
8515                       2009      Medium               Tier 2
8516                       2009      Medium               Tier 2
8519                       2002      Medium                  NaN
8521                       2009      Medium               Tier 2

            Outlet_Type  Item_Outlet_Sales  Profit
0     Supermarket Type1          3735.1380    11.5
1     Supermarket Type2           443.4228    14.3
2     Supermarket Type1          2097.2700    14.5
3         Grocery Store           732.3800    13.6
5     Supermarket Type2           556.6088     9.5
…                   …                …     …
8514  Supermarket Type1           468.7232    14.0
8515  Supermarket Type2          1571.2880    14.2
8516  Supermarket Type2           858.8820    14.6
8519  Supermarket Type1           549.2850    14.2
8521  Supermarket Type2          1845.5976    14.2

[5203 rows x 13 columns]
```

[39]:
```
#display transaction sale>3000 and profit is>13
datadf[(datadf['Item_Outlet_Sales']>3000)&(datadf['Profit']>13)]
```

[39]:
```
     Item_Identifier  Item_Weight Item_Fat_Content  Item_Visibility  \
7              FDP10      11.6881          Low Fat         0.127470
9              FDU28      19.2000          Regular         0.094450
20             FDN22      18.8500          Regular         0.138190
21             FDW12      11.6881          Regular         0.035400
24             FDR28      13.8500          Regular         0.025896
…              …          …                …               …
8503           FDQ44      20.5000          Low Fat         0.036133
8504           NCN18       0.0000          Low Fat         0.124111
8506           DRF37      17.2500          Low Fat         0.084676
8511           FDF05      17.5000          Low Fat         0.026980
8517           FDF53      20.7500              reg         0.083607

                 Item_Type  Item_MRP Outlet_Identifier  \
```

```
7                Snack Foods     107.8           OUT027
9              Frozen Foods      187.8           OUT017
20               Snack Foods     250.9           OUT013
21             Baking Goods      144.5           OUT027
24             Frozen Foods      165.0           OUT046
…                       …       …               …
8503  Fruits and Vegetables      120.2           OUT035
8504              Household      111.8           OUT027
8506            Soft Drinks      263.2           OUT018
8511           Frozen Foods      262.6           OUT018
8517           Frozen Foods      178.8           OUT046

      Outlet_Establishment_Year Outlet_Size Outlet_Location_Type  \
7                          1985      Medium               Tier 2
9                          2007      Medium                   --
20                         1987        High               Tier 3
21                         1985      Medium               Tier 2
24                         1997       Small                Tier1
…                          …         …                    …
8503                       2004       Small                Tier1
8504                       1985      Medium               Tier 2
8506                       2009      Medium               Tier 2
8511                       2009      Medium               Tier 2
8517                       1997       Small                Tier1

            Outlet_Type  Item_Outlet_Sales  Profit
7     Supermarket Type3          4022.7636    13.3
9     Supermarket Type1          4710.5350    13.6
20    Supermarket Type1          3775.0860    14.6
21    Supermarket Type3          4064.0432    14.7
24    Supermarket Type1          4078.0250    14.3
…                 …                …     …
8503  Supermarket Type1          3392.9168    14.3
8504  Supermarket Type3          4138.6128    14.1
8506  Supermarket Type2          3944.8650    14.3
8511  Supermarket Type2          4207.8560    14.8
8517  Supermarket Type1          3608.6360    13.2

[1735 rows x 13 columns]
```

[40]:
```
#display transaction sale>3000 and profit is>13 mrp>120
datadf[(datadf['Item_Outlet_Sales']>3000)&(datadf['Profit']>13)&(datadf['Item_MRP']>120)]
```

[40]:
```
     Item_Identifier  Item_Weight Item_Fat_Content  Item_Visibility  \
9              FDU28      19.2000          Regular         0.094450
20             FDN22      18.8500          Regular         0.138190
21             FDW12      11.6881          Regular         0.035400
```

```
24             FDR28      13.8500          Regular         0.025896
32             FDP33      18.7000          Low Fat         0.000000
…              …          …              …               …
8496           FDJ57       7.4200          Regular         0.021696
8503           FDQ44      20.5000          Low Fat         0.036133
8506           DRF37      17.2500          Low Fat         0.084676
8511           FDF05      17.5000          Low Fat         0.026980
8517           FDF53      20.7500              reg         0.083607

                  Item_Type  Item_MRP Outlet_Identifier  \
9              Frozen Foods     187.8            OUT017
20              Snack Foods     250.9            OUT013
21             Baking Goods     144.5            OUT027
24             Frozen Foods     165.0            OUT046
32              Snack Foods     256.7            OUT018
…                      …       …                …
8496                Seafood     185.4            OUT017
8503  Fruits and Vegetables     120.2            OUT035
8506            Soft Drinks     263.2            OUT018
8511           Frozen Foods     262.6            OUT018
8517           Frozen Foods     178.8            OUT046

      Outlet_Establishment_Year Outlet_Size Outlet_Location_Type  \
9                          2007      Medium                   --
20                         1987        High               Tier 3
21                         1985      Medium               Tier 2
24                         1997       Small                Tier1
32                         2009      Medium               Tier 2
…                          …         …                  …
8496                       2007      Medium                  NaN
8503                       2004       Small                Tier1
8506                       2009      Medium               Tier 2
8511                       2009      Medium               Tier 2
8517                       1997       Small                Tier1

            Outlet_Type  Item_Outlet_Sales  Profit
9     Supermarket Type1          4710.5350    13.6
20    Supermarket Type1          3775.0860    14.6
21    Supermarket Type3          4064.0432    14.7
24    Supermarket Type1          4078.0250    14.3
32    Supermarket Type2          3068.0064    14.3
…                 …                …     …
8496  Supermarket Type1          3715.1640    14.1
8503  Supermarket Type1          3392.9168    14.3
8506  Supermarket Type2          3944.8650    14.3
8511  Supermarket Type2          4207.8560    14.8
8517  Supermarket Type1          3608.6360    13.2
```

```
[1615 rows x 13 columns]
```

[41]:
```
#Grouping
#usecase:avg age of male or female, avg sales of tier1 or tier2
data=pd.DataFrame({'Gender':['M','F','M','M','F'],'age':
 ↪[32,35,36,40,42],'height':[6.2,3.4,5,5.5,6]})
data
```

[41]:
```
  Gender  age  height
0      M   32     6.2
1      F   35     3.4
2      M   36     5.0
3      M   40     5.5
4      F   42     6.0
```

[ ]:
```

```

[42]:
```
data.groupby('Gender')['age'].mean()
```

[42]:
```
Gender
F    38.5
M    36.0
Name: age, dtype: float64
```

[43]:
```
data.groupby('Gender')['height'].mean()
```

[43]:
```
Gender
F    4.700000
M    5.566667
Name: height, dtype: float64
```

[44]:
```
data.groupby('Gender').mean()
```

[44]:
```
          age    height
Gender
F       38.5  4.700000
M       36.0  5.566667
```

[45]:
```
#what is avg profit in every outlet size
datadf.groupby('Outlet_Size')['Profit'].mean()
```

[45]:
```
Outlet_Size
High      13.428541
Medium    13.410340
Small     13.418132
Name: Profit, dtype: float64
```

[46]:
```
#what is sum of profit and avg profit in every outlet size
datadf.groupby('Outlet_Size').agg({'Profit':['mean','sum']})
```

[46]:

Outlet_Size	Profit	
	mean	sum
High	13.428541	12515.4
Medium	13.410340	69774.0
Small	13.418132	32042.5

[48]:
```
#avg sales and profit of each item type on every outlet size
datadf.groupby(['Item_Type','Outlet_Size']).agg({'Item_Outlet_Sales':
 ↪['mean','sum'],'Profit':['mean','sum']})
```

[48]:

		Item_Outlet_Sales		Profit
		mean	sum	mean
Item_Type	Outlet_Size			
Baking Goods	High	2050.901134	1.497158e+05	13.306849
	Medium	2044.572273	7.932940e+05	13.408505
	Small	1724.681911	3.225155e+05	13.318717
Breads	High	2080.731528	5.201829e+04	13.480000
	Medium	2244.802689	3.479444e+05	13.606452
	Small	2158.795546	1.532745e+05	13.440845
Breakfast	High	2104.286508	2.735572e+04	13.938462
	Medium	2290.501060	1.534636e+05	13.100000
	Small	1715.988533	5.147966e+04	13.520000
Canned	High	2211.265203	1.437322e+05	13.221538
	Medium	2316.837355	9.151508e+05	13.487848
	Small	2038.457667	3.852685e+05	13.284127
Dairy	High	2453.181713	1.962545e+05	13.398750
	Medium	2351.587472	9.500413e+05	13.320050
	Small	1900.495836	3.762982e+05	13.644444
Frozen Foods	High	2214.096189	2.036968e+05	13.427174
	Medium	2233.128106	1.150061e+06	13.375146
	Small	1895.489818	4.719770e+05	13.359036
Fruits and Vegetables	High	2405.118103	3.415268e+05	13.528169
	Medium	2421.365188	1.845080e+06	13.462598
	Small	1931.258454	6.334528e+05	13.367988
Hard Drinks	High	2363.590000	5.436257e+04	13.495652
	Medium	2178.049011	3.071049e+05	13.270922
	Small	1926.518932	9.632595e+04	12.876000
Health and Hygiene	High	1953.042439	1.191356e+05	13.498361
	Medium	2054.796907	6.636994e+05	13.449536
	Small	1929.155500	2.623651e+05	13.691176
Household	High	2408.217992	2.480465e+05	13.589320
	Medium	2363.485893	1.299917e+06	13.506182
	Small	1974.824976	5.075300e+05	13.449027

Meat	High	2321.953141	9.520008e+04	13.256098
	Medium	2294.731205	6.081038e+05	13.364528
	Small	1800.519024	2.142618e+05	13.201681
Others	High	2202.383175	3.523813e+04	14.056250
	Medium	2103.764947	2.061690e+05	13.260204
	Small	1529.282073	8.411051e+04	13.463636
Seafood	High	1629.345760	8.146729e+03	12.260000
	Medium	2300.287785	8.971122e+04	13.030769
	Small	2550.513350	5.101027e+04	13.785000
Snack Foods	High	2473.968987	3.092461e+05	13.300000
	Medium	2388.512514	1.767499e+06	13.334865
	Small	1958.330459	6.560407e+05	13.509851
Soft Drinks	High	2351.727890	1.152347e+05	13.463265
	Medium	2132.347796	5.757339e+05	13.491852
	Small	1602.612305	2.019292e+05	13.369048
Starchy Foods	High	2302.791947	4.375305e+04	13.305263
	Medium	2299.731732	2.092756e+05	13.465934
	Small	2588.753047	9.837262e+04	13.392105

		sum
Item_Type	Outlet_Size	
Baking Goods	High	971.4
	Medium	5202.5
	Small	2490.6
Breads	High	337.0
	Medium	2109.0
	Small	954.3
Breakfast	High	181.2
	Medium	877.7
	Small	405.6
Canned	High	859.4
	Medium	5327.7
	Small	2510.7
Dairy	High	1071.9
	Medium	5381.3
	Small	2701.6
Frozen Foods	High	1235.3
	Medium	6888.2
	Small	3326.4
Fruits and Vegetables	High	1921.0
	Medium	10258.5
	Small	4384.7
Hard Drinks	High	310.4
	Medium	1871.2
	Small	643.8
Health and Hygiene	High	823.4

```
                        Medium        4344.2
                        Small         1862.0
Household               High          1399.7
                        Medium        7428.4
                        Small         3456.4
Meat                    High           543.5
                        Medium        3541.6
                        Small         1571.0
Others                  High           224.9
                        Medium        1299.5
                        Small          740.5
Seafood                 High            61.3
                        Medium         508.2
                        Small          275.7
Snack Foods             High          1662.5
                        Medium        9867.8
                        Small         4525.8
Soft Drinks             High           659.7
                        Medium        3642.8
                        Small         1684.5
Starchy Foods           High           252.8
                        Medium        1225.4
                        Small          508.9
```

[51]:
```
#concat and merging---------->Joining different dataframes
df1=pd.DataFrame({'Name':['Har','Swe','Gaye','Yuva'],'age':
 ↪['23','21','41','52']})
df1
```

[51]:
```
   Name age
0   Har  23
1   Swe  21
2  Gaye  41
3  Yuva  52
```

[58]:
```
df2=pd.DataFrame({'Name':['Swe','Swe','Gaye','Yuva'],'height':['5.5','5.3','5.
 ↪3','5.7']})
df2
```

[58]:
```
   Name height
0   Swe    5.5
1   Swe    5.3
2  Gaye    5.3
3  Yuva    5.7
```

[59]:
```
d=pd.concat([df1,df2])  #default adds in row
d
```

[59]:
```
   Name  age height
0   Har   23    NaN
1   Swe   21    NaN
2  Gaye   41    NaN
3  Yuva   52    NaN
0   Swe  NaN    5.5
1   Swe  NaN    5.3
2  Gaye  NaN    5.3
3  Yuva  NaN    5.7
```

[62]:
```
e=pd.concat([df1,df2],axis=1) #default axis=0(row) axis=1(column) doesnot check␣
 ↪any condition it will simply add but in below case you can see height of har␣
 ↪is missing but concat just simply added it
e
```

[62]:
```
   Name age  Name height
0   Har  23   Swe    5.5
1   Swe  21   Swe    5.3
2  Gaye  41  Gaye    5.3
3  Yuva  52  Yuva    5.7
```

[71]:
```
#merging
#inner---default inner
df1=pd.DataFrame({'Name':['Har','Swe','Gaye','Yuva'],'age':
 ↪['23','21','41','52']})
df1
df2=pd.DataFrame({'Name':['Har','George','Kennedy','Yuva'],'age':
 ↪['23','21','41','53']})
df2
f=pd.merge(df1,df2) #by default innerjoin so we get only common elements checks␣
 ↪both age and name
f
```

[71]:
```
  Name age
0  Har  23
```

[70]:
```
g=pd.merge(df1,df2,on='Name') #checks common elements only based on age
g
```

[70]:
```
   Name age_x age_y
0   Har    23    23
1  Yuva    52    53
```

[72]:
```
h=pd.merge(df1,df2,on='age') #checks common elements only based on age
h
```

[72]:
```
  Name_x age   Name_y
0    Har  23      Har
1    Swe  21   George
2   Gaye  41  Kennedy
```

[73]:
```
i=pd.merge(df1,df2,on='Name',how='left') #left join--alll left element plus␣
 ↪common on right
i
```

[73]:
```
   Name age_x age_y
0   Har    23    23
1   Swe    21   NaN
2  Gaye    41   NaN
3  Yuva    52    53
```

[74]:
```
j=pd.merge(df1,df2,on='Name',how='right') #right join--alll right element plus␣
 ↪common on left
j
```

[74]:
```
       Name age_x age_y
0       Har    23    23
1    George   NaN    21
2  Kennedy   NaN    41
3      Yuva    52    53
```

[76]:
```
k=pd.merge(df1,df2,on='Name',how='outer') #outer join--alll right element plus␣
 ↪common on left plus all elem on left
k
```

[76]:
```
       Name age_x age_y
0       Har    23    23
1       Swe    21   NaN
2      Gaye    41   NaN
3      Yuva    52    53
4    George   NaN    21
5  Kennedy   NaN    41
```

[77]:
```
#add DF in rows
df1.append(df2)
```

[77]:
```
       Name age
0       Har  23
1       Swe  21
2      Gaye  41
3      Yuva  52
0       Har  23
1    George  21
```

```
2  Kennedy  41
3     Yuva  53
```

[78]:
```
df1.append(df2,ignore_index=True) #Default--->ignore_index=False
```

[78]:
```
      Name age
0      Har  23
1      Swe  21
2     Gaye  41
3     Yuva  52
4      Har  23
5   George  21
6  Kennedy  41
7     Yuva  53
```