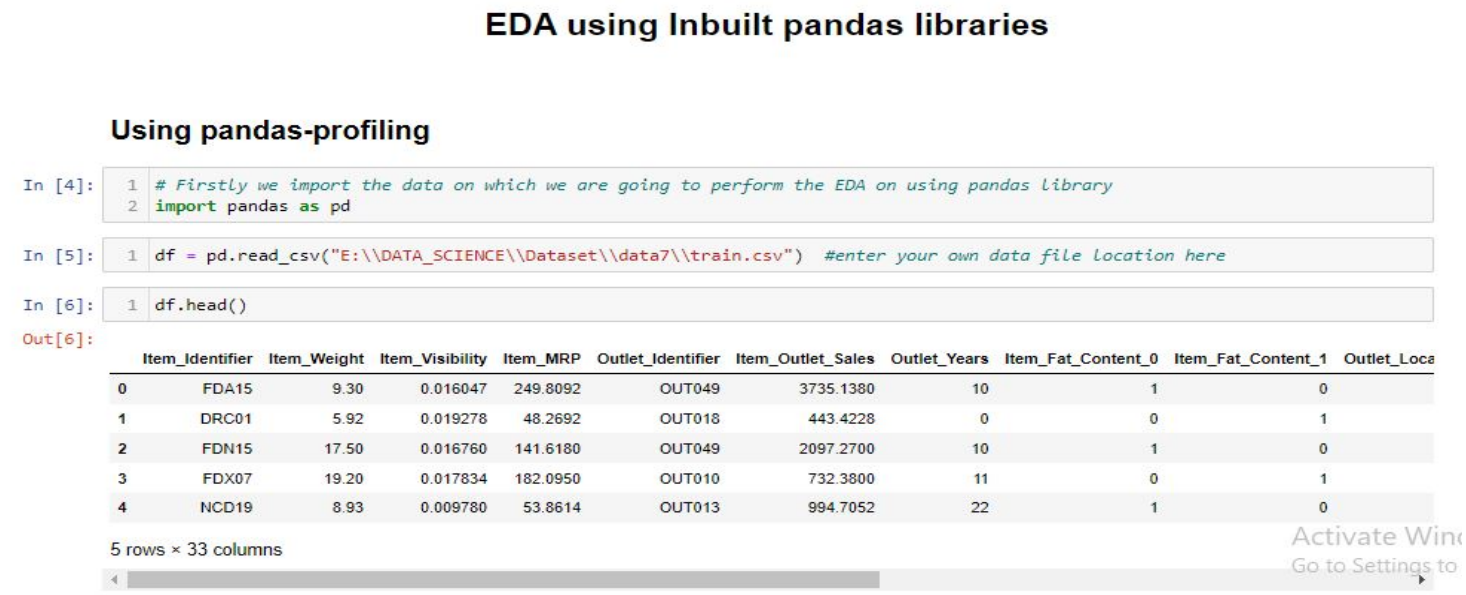


EDA using Inbuilt pandas libraries

Using pandas-profiling

In [4]:
```python
# Firstly we import the data on which we are going to perform the EDA on using pandas library
import pandas as pd
```

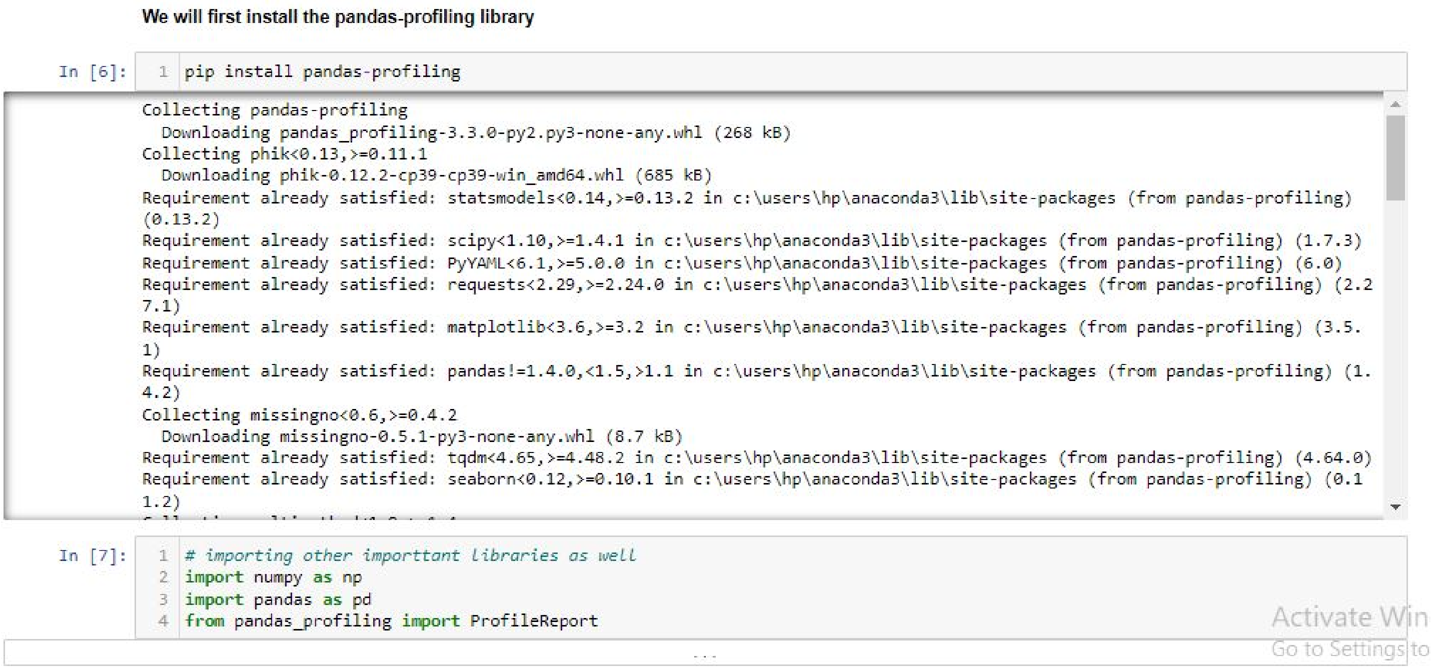
In [5]:
```python
df = pd.read_csv("E:\\DATA_SCIENCE\\Dataset\\data7\\train.csv")  #enter your own data file location here
```

In [6]:
```python
df.head()
```

Out[6]:

	Item_Identifier	Item_Weight	Item_Visibility	Item_MRP	Outlet_Identifier	Item_Outlet_Sales	Outlet_Years	Item_Fat_Content_0	Item_Fat_Content_1	Outlet_Loca
0	FDA15	9.30	0.016047	249.8092	OUT049	3735.1380	10	1	0	
1	DRC01	5.92	0.019278	48.2692	OUT018	443.4228	0	0	1	
2	FDN15	17.50	0.016760	141.6180	OUT049	2097.2700	10	1	0	
3	FDX07	19.20	0.017834	182.0950	OUT010	732.3800	11	0	1	
4	NCD19	8.93	0.009780	53.8614	OUT013	994.7052	22	1	0	

5 rows × 33 columns



We will first install the pandas-profiling library

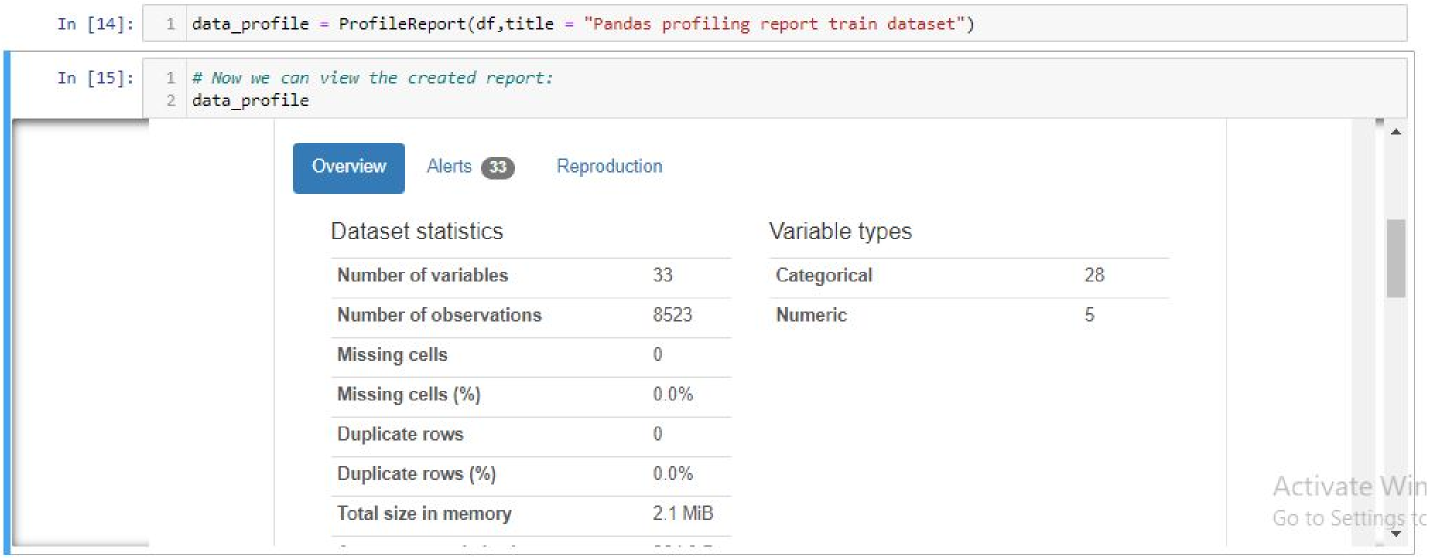
In [6]:
```
pip install pandas-profiling
```

```
Collecting pandas-profiling
  Downloading pandas_profiling-3.3.0-py2.py3-none-any.whl (268 kB)
Collecting phik<0.13,>=0.11.1
  Downloading phik-0.12.2-cp39-cp39-win_amd64.whl (685 kB)
Requirement already satisfied: statsmodels<0.14,>=0.13.2 in c:\users\hp\anaconda3\lib\site-packages (from pandas-profiling)
(0.13.2)
Requirement already satisfied: scipy<1.10,>=1.4.1 in c:\users\hp\anaconda3\lib\site-packages (from pandas-profiling) (1.7.3)
Requirement already satisfied: PyYAML<6.1,>=5.0.0 in c:\users\hp\anaconda3\lib\site-packages (from pandas-profiling) (6.0)
Requirement already satisfied: requests<2.29,>=2.24.0 in c:\users\hp\anaconda3\lib\site-packages (from pandas-profiling) (2.2
7.1)
Requirement already satisfied: matplotlib<3.6,>=3.2 in c:\users\hp\anaconda3\lib\site-packages (from pandas-profiling) (3.5.
1)
Requirement already satisfied: pandas!=1.4.0,<1.5,>1.1 in c:\users\hp\anaconda3\lib\site-packages (from pandas-profiling) (1.
4.2)
Collecting missingno<0.6,>=0.4.2
  Downloading missingno-0.5.1-py3-none-any.whl (8.7 kB)
Requirement already satisfied: tqdm<4.65,>=4.48.2 in c:\users\hp\anaconda3\lib\site-packages (from pandas-profiling) (4.64.0)
Requirement already satisfied: seaborn<0.12,>=0.10.1 in c:\users\hp\anaconda3\lib\site-packages (from pandas-profiling) (0.1
1.2)
```

In [7]:
```python
# importing other importtant libraries as well
import numpy as np
import pandas as pd
from pandas_profiling import ProfileReport
```

ProfileReport tool of the pandas_profiling library is used to create an eda report of the dataframe

After this we create a data_profile variable that contains the report and we give title to it.



In [14]:
```python
data_profile = ProfileReport(df,title = "Pandas profiling report train dataset")
```

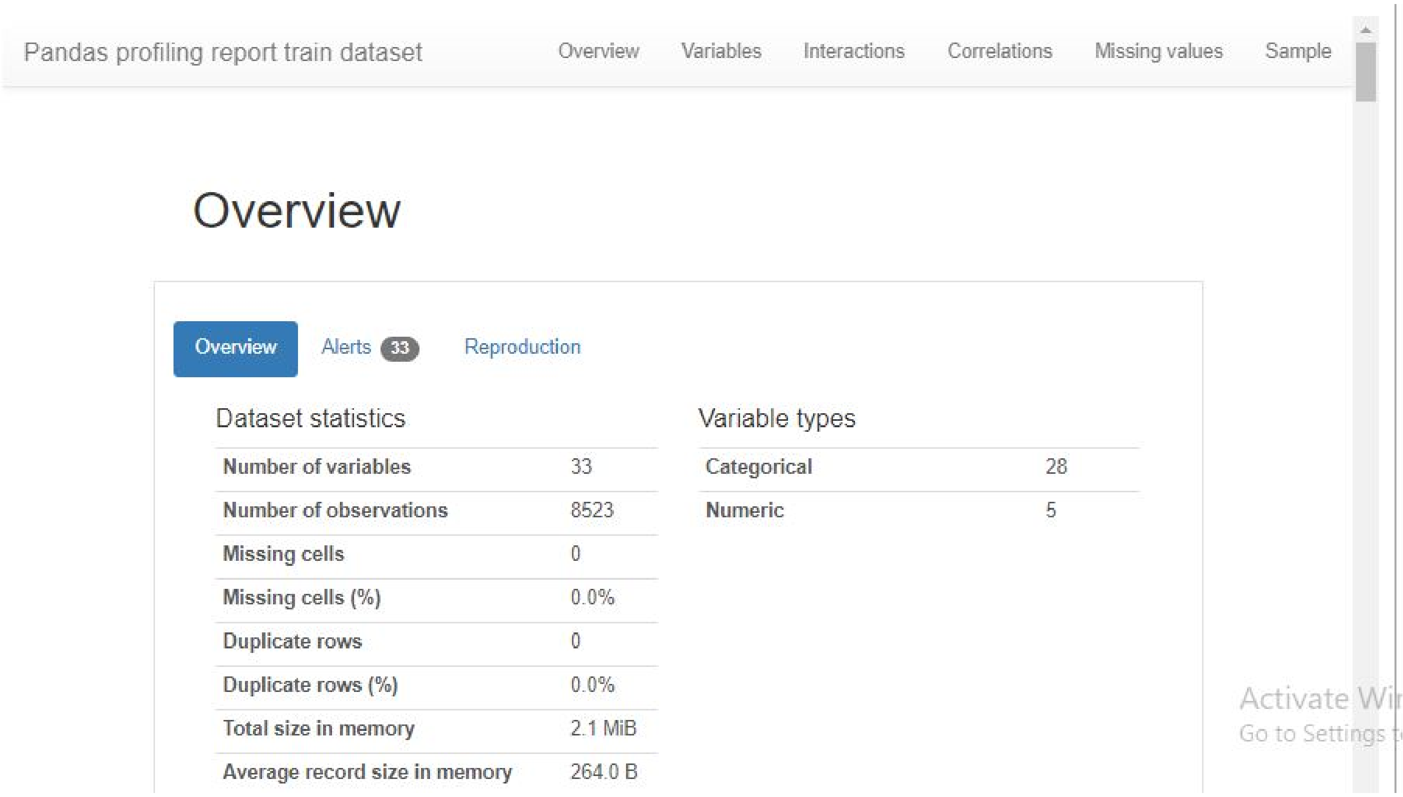
In [15]:
```python
# Now we can view the created report:
data_profile
```

Dataset statistics

Number of variables	33
Number of observations	8523
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	2.1 MiB

Variable types

Categorical	28
Numeric	5



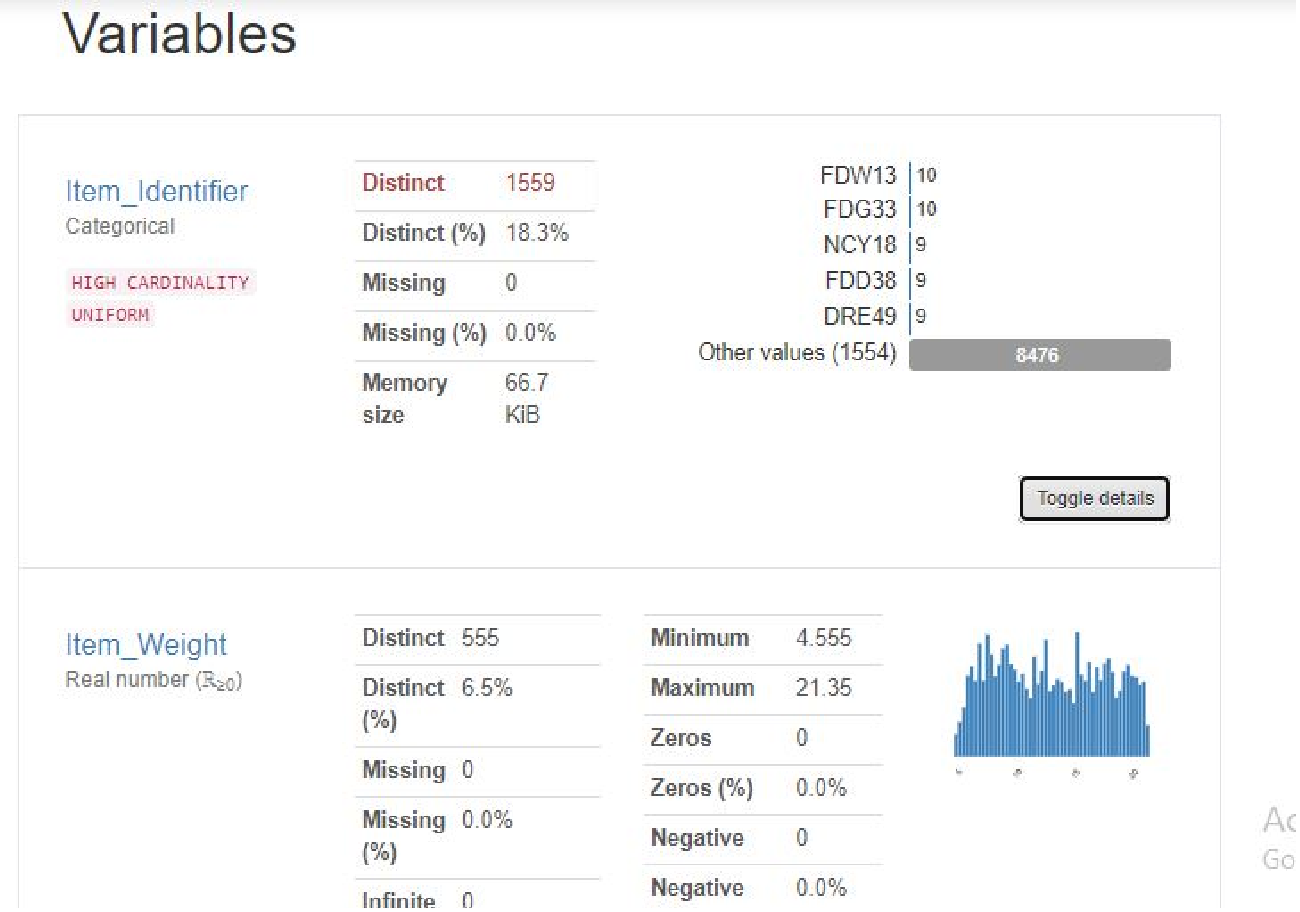
Overview

Dataset statistics

Number of variables	33
Number of observations	8523
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	2.1 MiB
Average record size in memory	264.0 B

Variable types

Categorical	28
Numeric	5



Variables

Item_Identifier

Categorical

HIGH CARDINALITY

UNIFORM

Distinct	1559
Distinct (%)	18.3%
Missing	0
Missing (%)	0.0%
Memory size	66.7 KiB

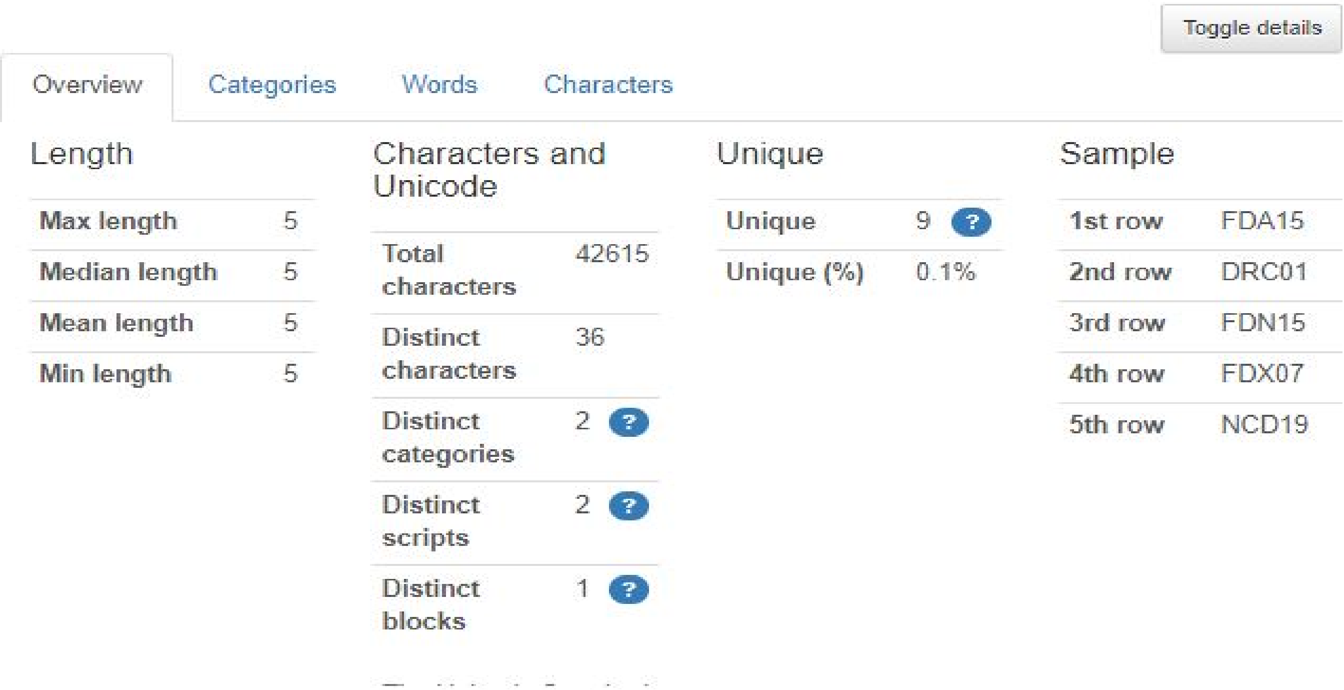
FDW13	10
FDG33	10
NCY18	9
FDD38	9
DRE49	9
Other values (1554)	8476

Item_Weight

Real number ($\mathbb{R}_{\geq 0}$)

Distinct	555
Distinct (%)	6.5%
Missing	0
Missing (%)	0.0%
Infinite	0

Minimum	4.555
Maximum	21.35
Zeros	0
Zeros (%)	0.0%
Negative	0
Negative	0.0%



Length

Max length	5
Median length	5
Mean length	5
Min length	5

Characters and Unicode

Total characters	42615
Distinct characters	36
Distinct categories	2 ?
Distinct scripts	2 ?
Distinct blocks	1 ?

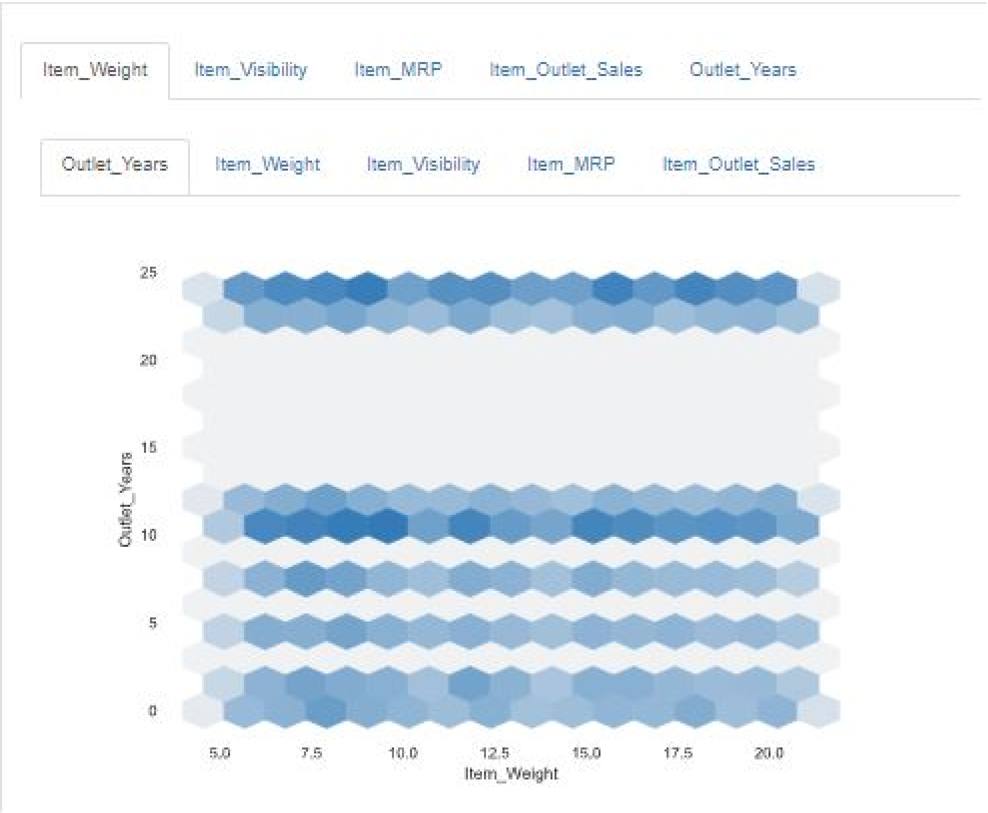
Unique

Unique	9 ?
Unique (%)	0.1%

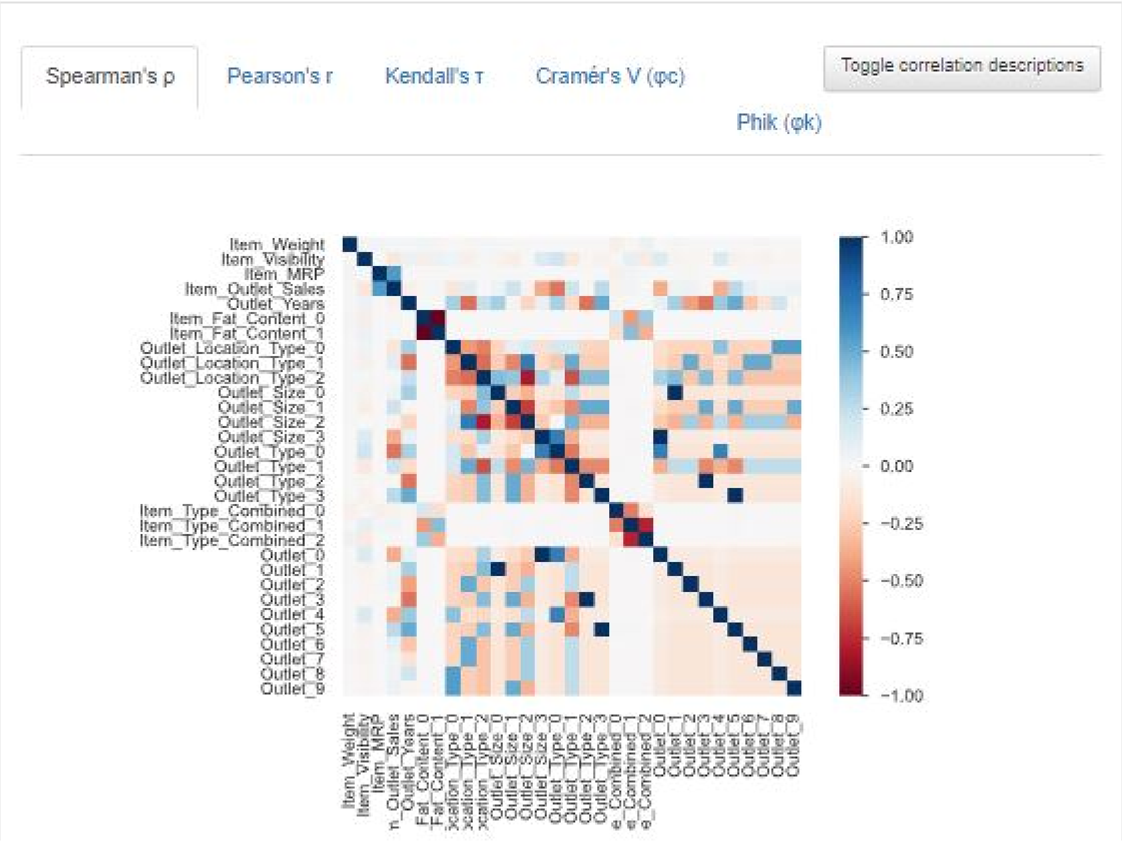
Sample

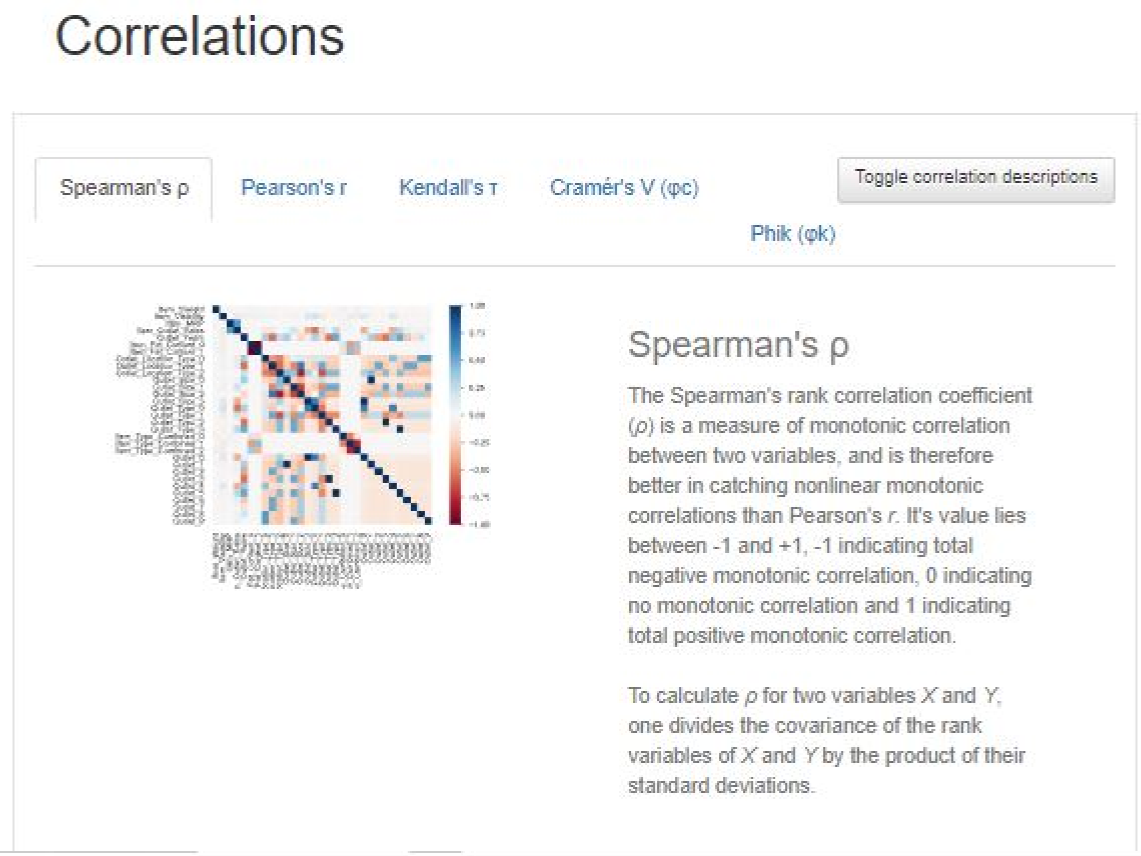
1st row	FDA15
2nd row	DRC01
3rd row	FDN15
4th row	FDX07
5th row	NCD19

Interactions



Correlations



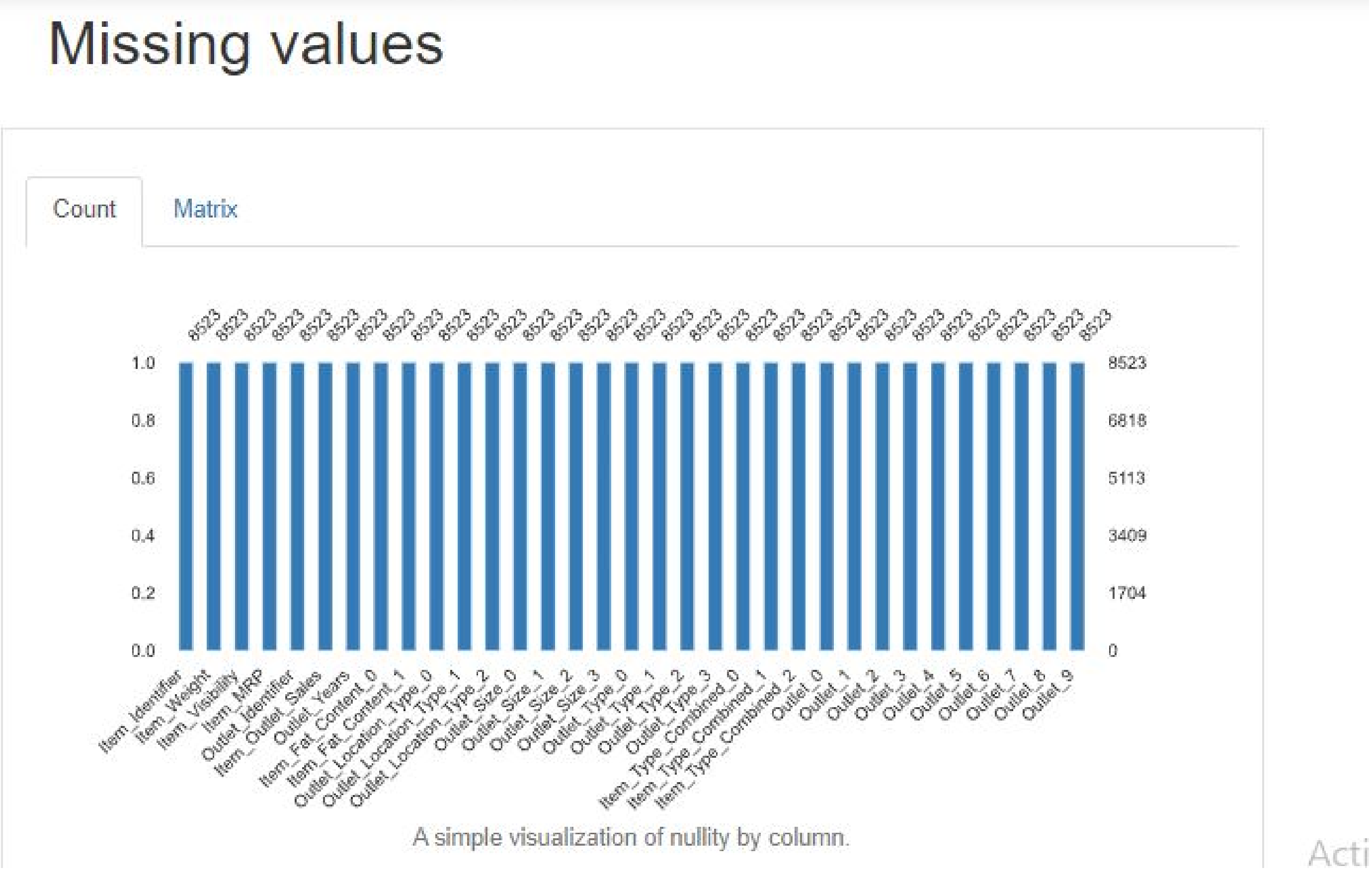


Correlations

Spearman's ρ

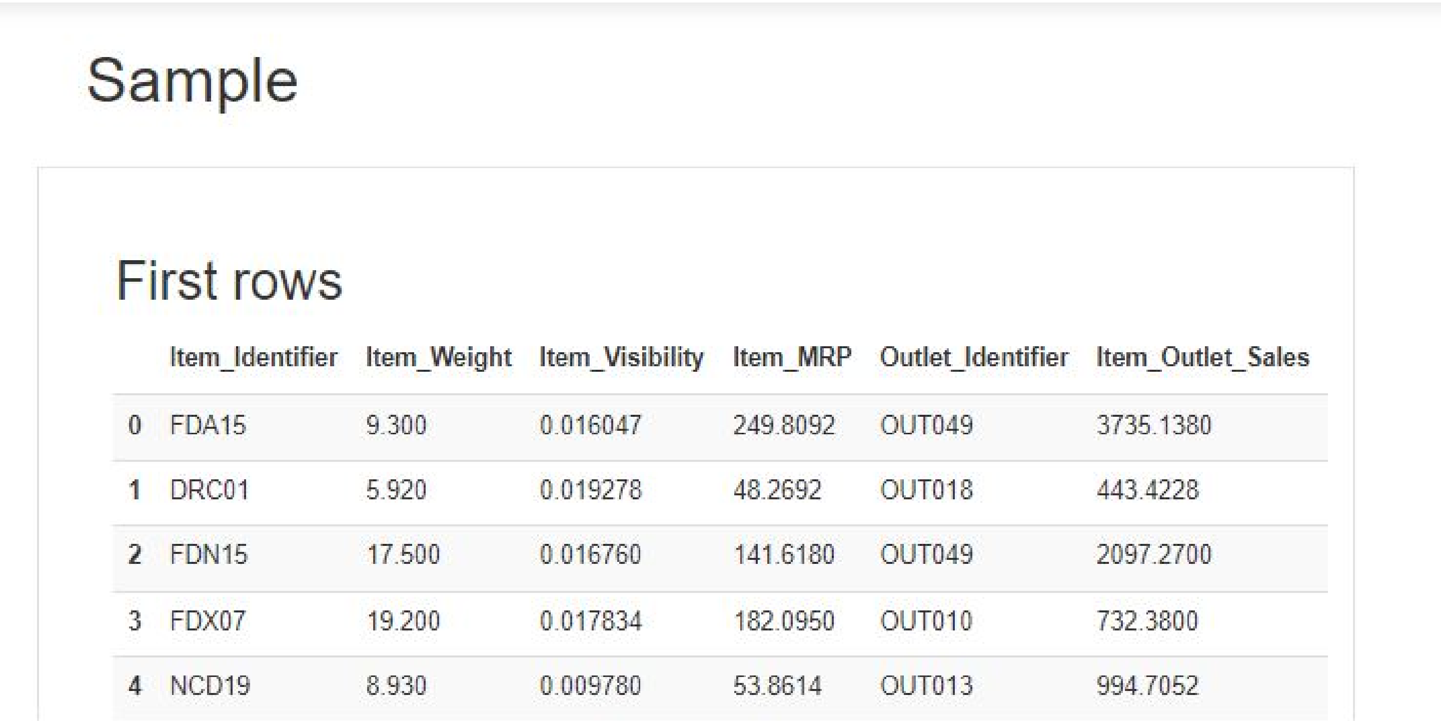
The Spearman's rank correlation coefficient (*ρ*) is a measure of monotonic correlation between two variables, and is therefore better in catching nonlinear monotonic correlations than Pearson's *r*. It's value lies between -1 and +1, -1 indicating total negative monotonic correlation, 0 indicating no monotonic correlation and 1 indicating total positive monotonic correlation.

To calculate *ρ* for two variables *X* and *Y*, one divides the covariance of the rank variables of *X* and *Y* by the product of their standard deviations.



Missing values

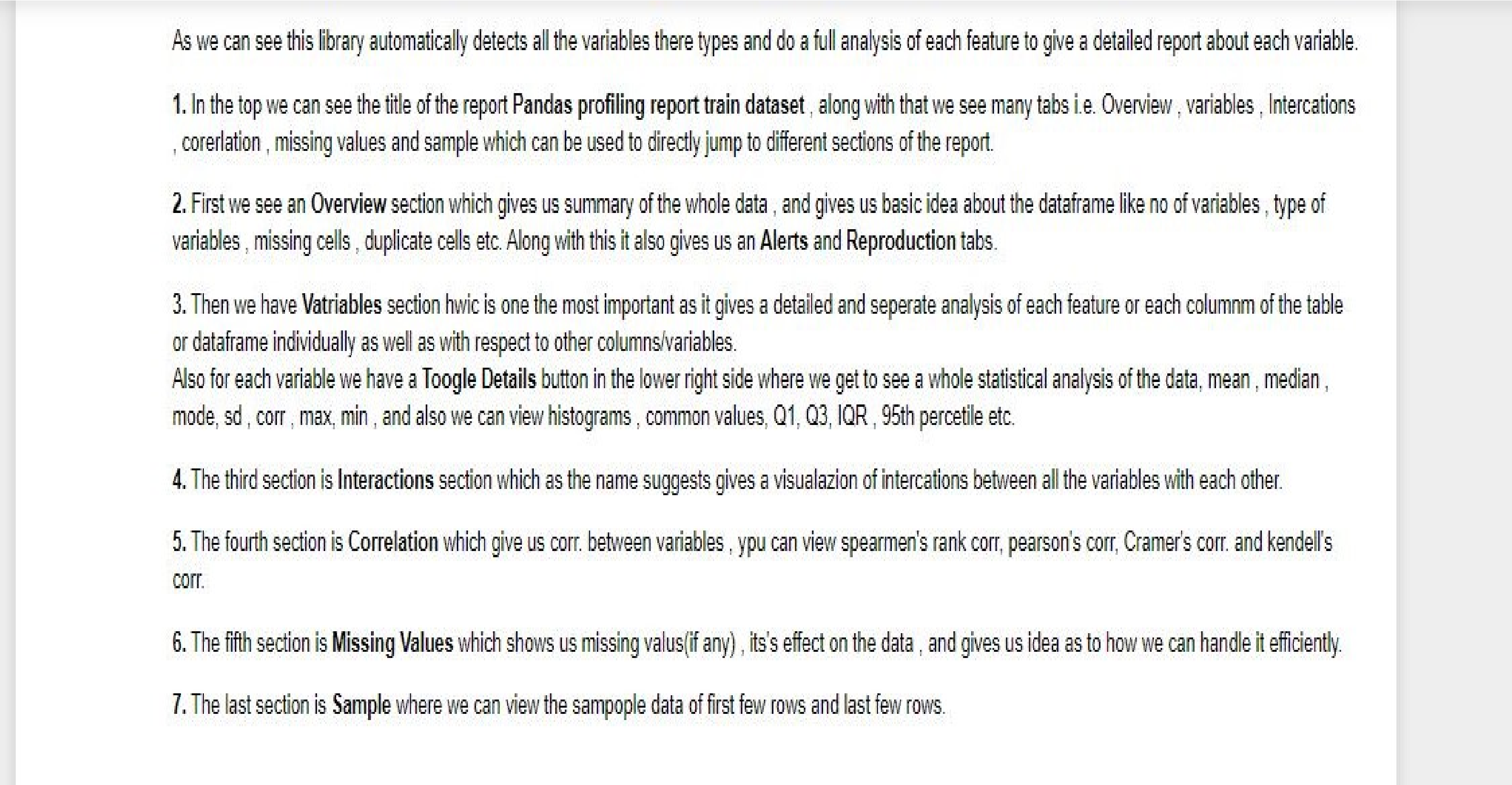
A simple visualization of nullity by column.



Sample

First rows

	Item_Identifier	Item_Weight	Item_Visibility	Item_MRP	Outlet_Identifier	Item_Outlet_Sales
0	FDA15	9.300	0.016047	249.8092	OUT049	3735.1380
1	DRC01	5.920	0.019278	48.2692	OUT018	443.4228
2	FDN15	17.500	0.016760	141.6180	OUT049	2097.2700
3	FDX07	19.200	0.017834	182.0950	OUT010	732.3800
4	NCD19	8.930	0.009780	53.8614	OUT013	994.7052



As we can see this library automatically detects all the variables there types and do a full analysis of each feature to give a detailed report about each variable.

1. In the top we can see the title of the report **Pandas profiling report train dataset** , along with that we see many tabs i.e. Overview , variables , Intercations , corerlation , missing values and sample which can be used to directly jump to different sections of the report.

2. First we see an **Overview** section which gives us summary of the whole data , and gives us basic idea about the dataframe like no of variables , type of variables , missing cells , duplicate cells etc. Along with this it also gives us an **Alerts** and **Reproduction** tabs.

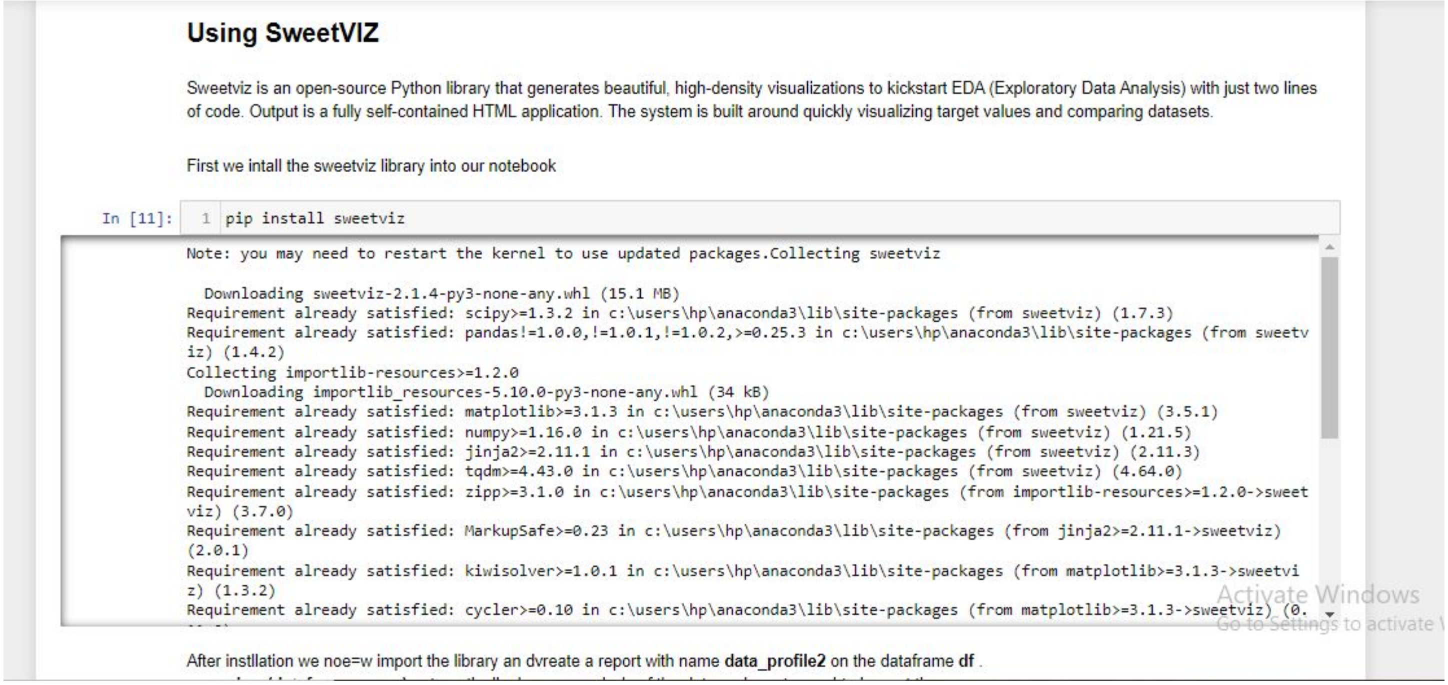
3. Then we have **Vatriables** section hwic is one the most important as it gives a detailed and seperate analysis of each feature or each columnm of the table or dataframe individually as well as with respect to other columns/variables.
Also for each variable we have a **Toogle Details** button in the lower right side where we get to see a whole statistical analysis of the data, mean , median , mode, sd , corr , max, min , and also we can view histograms , common values, Q1, Q3, IQR , 95th percetile etc.

4. The third section is **Interactions** section which as the name suggests gives a visualazion of intercations between all the variables with each other.

5. The fourth section is **Correlation** which give us corr. between variables , ypu can view spearmen's rank corr, pearson's corr, Cramer's corr. and kendell's corr.

6. The fifth section is **Missing Values** which shows us missing valus(if any) , its's effect on the data , and gives us idea as to how we can handle it efficiently.

7. The last section is **Sample** where we can view the sampople data of first few rows and last few rows.



Using SweetVIZ

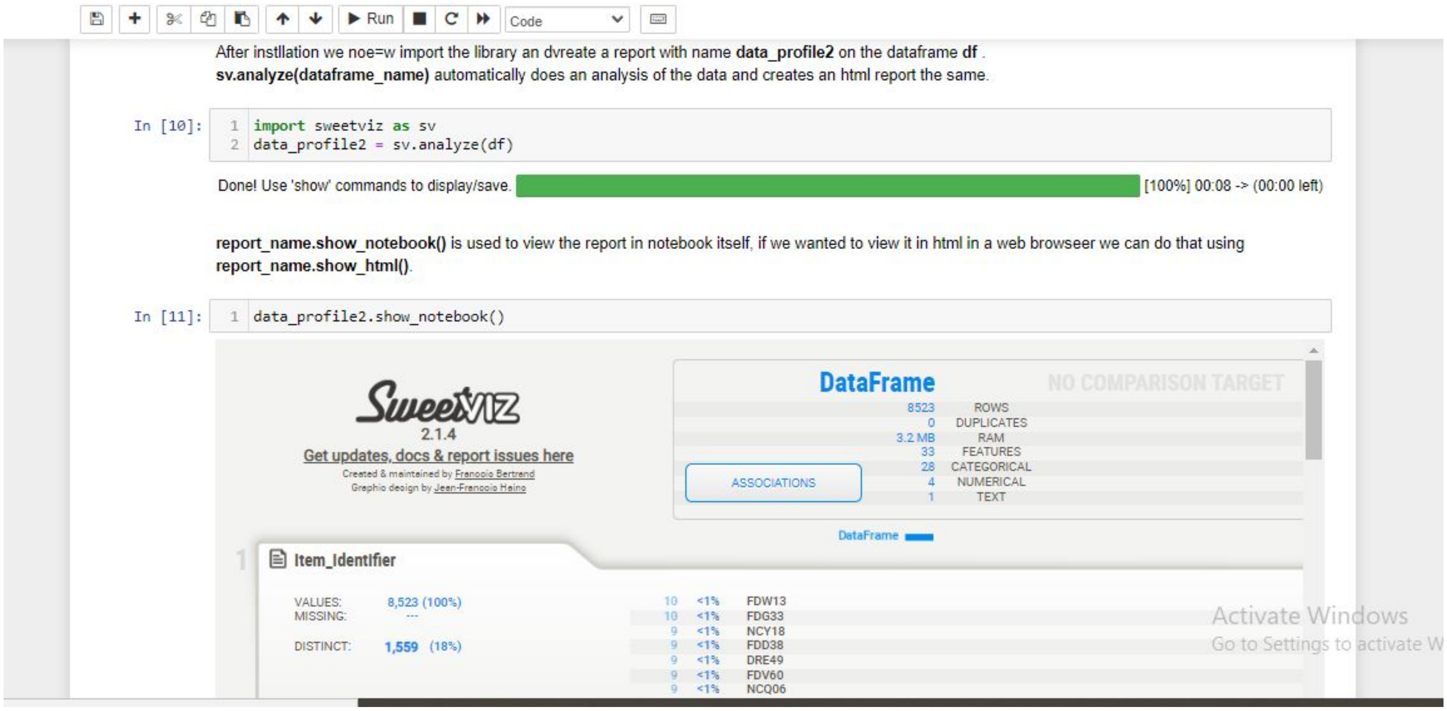
Sweetviz is an open-source Python library that generates beautiful, high-density visualizations to kickstart EDA (Exploratory Data Analysis) with just two lines of code. Output is a fully self-contained HTML application. The system is built around quickly visualizing target values and comparing datasets.

First we intall the sweetviz library into our notebook

```
In [11]: pip install sweetviz
```

```
Note: you may need to restart the kernel to use updated packages.Collecting sweetviz

  Downloading sweetviz-2.1.4-py3-none-any.whl (15.1 MB)
Requirement already satisfied: scipy>=1.3.2 in c:\users\hp\anaconda3\lib\site-packages (from sweetviz) (1.7.3)
Requirement already satisfied: pandas!=1.0.0,!=1.0.1,!=1.0.2,>=0.25.3 in c:\users\hp\anaconda3\lib\site-packages (from sweetv
iz) (1.4.2)
Collecting importlib-resources>=1.2.0
  Downloading importlib_resources-5.10.0-py3-none-any.whl (34 kB)
Requirement already satisfied: matplotlib>=3.1.3 in c:\users\hp\anaconda3\lib\site-packages (from sweetviz) (3.5.1)
Requirement already satisfied: numpy>=1.16.0 in c:\users\hp\anaconda3\lib\site-packages (from sweetviz) (1.21.5)
Requirement already satisfied: jinja2>=2.11.1 in c:\users\hp\anaconda3\lib\site-packages (from sweetviz) (2.11.3)
Requirement already satisfied: tqdm>=4.43.0 in c:\users\hp\anaconda3\lib\site-packages (from sweetviz) (4.64.0)
Requirement already satisfied: zipp>=3.1.0 in c:\users\hp\anaconda3\lib\site-packages (from importlib-resources>=1.2.0->sweet
viz) (3.7.0)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\hp\anaconda3\lib\site-packages (from jinja2>=2.11.1->sweetviz)
(2.0.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib>=3.1.3->sweetvi
z) (1.3.2)
Requirement already satisfied: cycler>=0.10 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib>=3.1.3->sweetviz) (0.
```

After instllation we noe=w import the library an dvreate a report with name **data_profile2** on the dataframe **df** .



After instllation we noe=w import the library an dvreate a report with name **data_profile2** on the dataframe **df** .
sv.analyze(dataframe_name) automatically does an analysis of the data and creates an html report the same.

```
In [10]: import sweetviz as sv
         data_profile2 = sv.analyze(df)
```

```
Done! Use 'show' commands to display/save. [100%] 00:08 -> (00:00 left)
```

report_name.show_notebook() is used to view the report in notebook itself, if we wanted to view it in html in a web browseer we can do that using **report_name.show_html()**.

```
In [11]: data_profile2.show_notebook()
```

SweetVIZ
2.1.4
Get updates, docs & report issues here
Created & maintained by Francois Bertrand
Graphic design by Jean-Francois Hains

DataFrame   NO COMPARISON TARGET

```
    8523   ROWS
       0   DUPLICATES
  3.2 MB   RAM
      33   FEATURES
      28   CATEGORICAL
       4   NUMERICAL
       1   TEXT
```

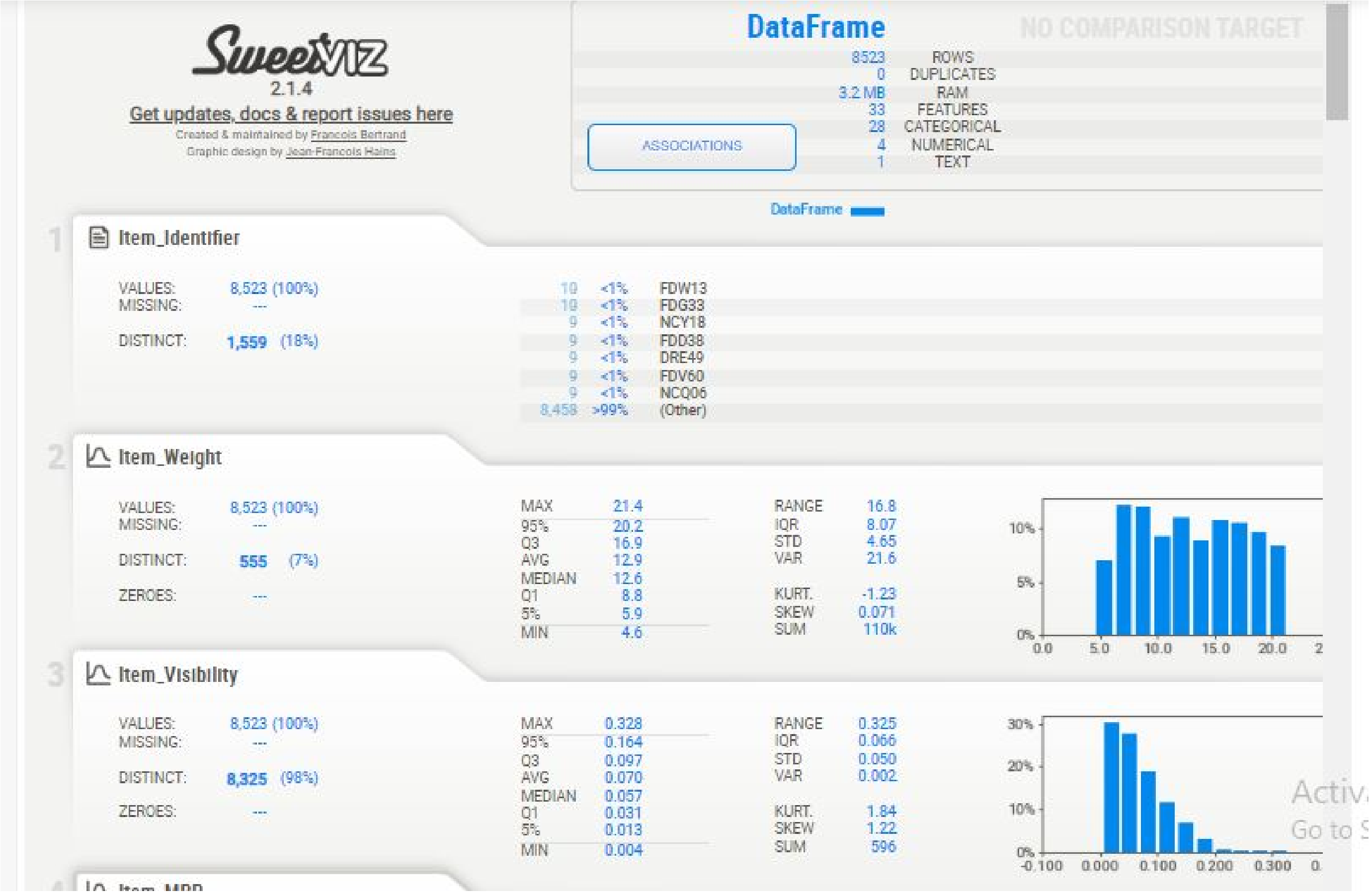
ASSOCIATIONS

DataFrame

1 Item_Identifier

```
VALUES:     8,523 (100%)
MISSING:    ---

DISTINCT:   1,559  (18%)
```

```
10   <1%   FDW13
10   <1%   FDG33
 9   <1%   NCY18
 9   <1%   FDD38
 9   <1%   DRE49
 9   <1%   FDV60
 9   <1%   NCQ06
```



Sweetviz
2.1.4
Get updates, docs & report issues here
Created & maintained by Francois Bertrand
Graphic design by Jean-Francois Hains

DataFrame NO COMPARISON TARGET

```
      8523   ROWS
         0   DUPLICATES
    3.2 MB   RAM
        33   FEATURES
        28   CATEGORICAL
         4   NUMERICAL
         1   TEXT
```

ASSOCIATIONS

DataFrame

1 Item_Identifier

```
VALUES:    8,523 (100%)          10     <1%    FDW13
MISSING:   ---                   10     <1%    FDG33
                                  9     <1%    NCY18
DISTINCT:  1,559  (18%)           9     <1%    FDD38
                                  9     <1%    DRE49
                                  9     <1%    FDV60
                                  9     <1%    NCQ06
                              8,458    >99%    (Other)
```

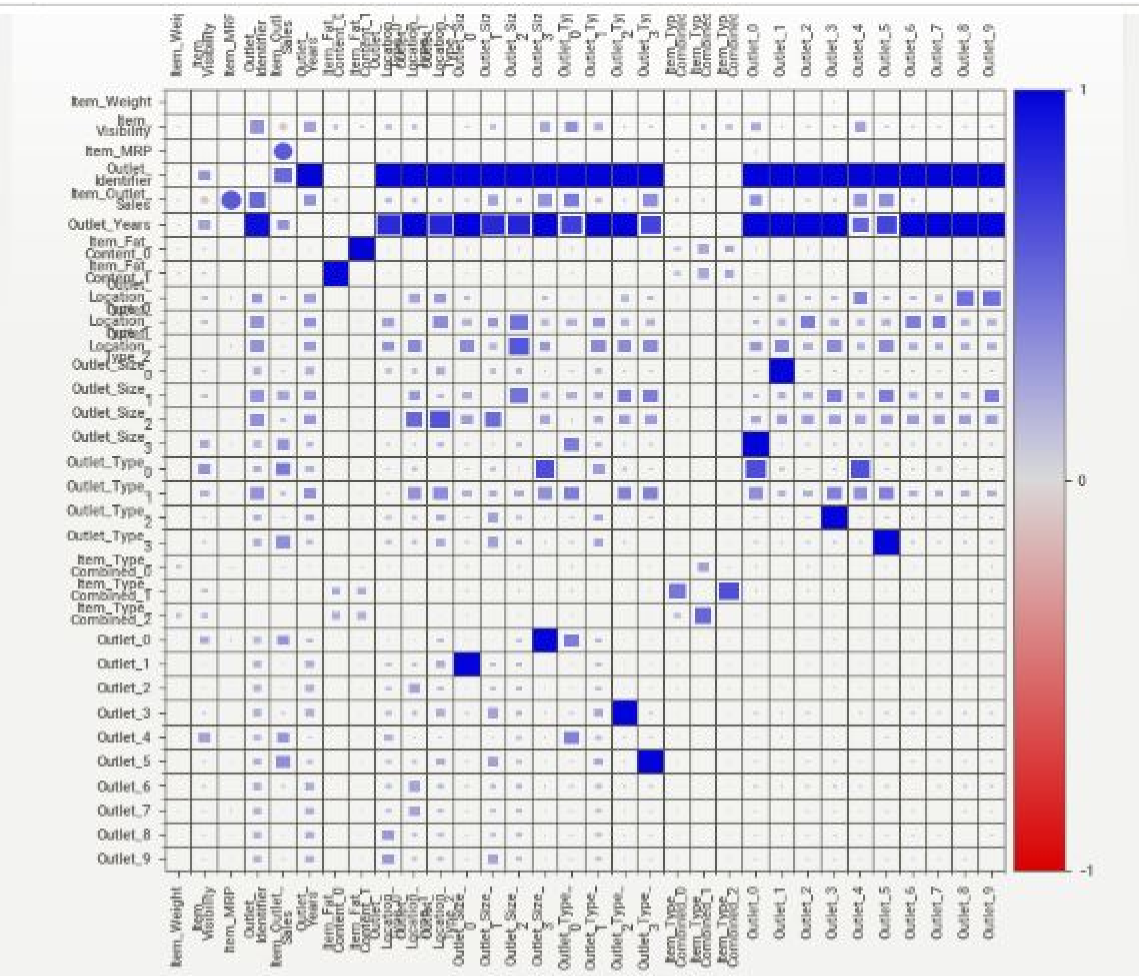
2 Item_Weight

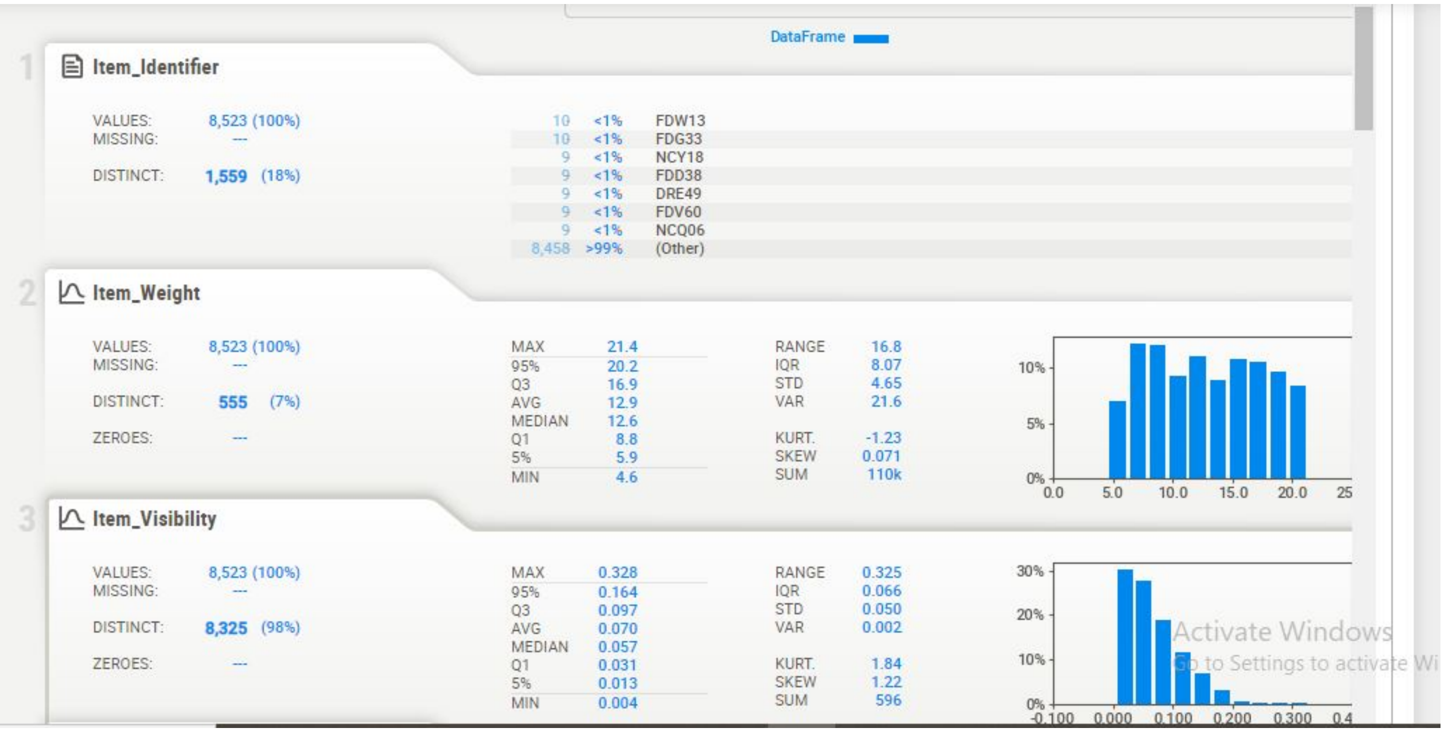
```
VALUES:    8,523 (100%)     MAX      21.4     RANGE   16.8
MISSING:   ---              95%      20.2     IQR     8.07
                            Q3       16.9     STD     4.65
DISTINCT:  555   (7%)       AVG      12.9     VAR     21.6
                            MEDIAN   12.6
ZEROES:    ---              Q1        8.8     KURT.   -1.23
                            5%        5.9     SKEW    0.071
                            MIN       4.6     SUM     110k
```

3 Item_Visibility

```
VALUES:    8,523 (100%)     MAX      0.328    RANGE   0.325
MISSING:   ---              95%      0.164    IQR     0.066
                            Q3       0.097    STD     0.050
DISTINCT:  8,325 (98%)      AVG      0.070    VAR     0.002
                            MEDIAN   0.057
ZEROES:    ---              Q1       0.031    KURT.   1.84
                            5%       0.013    SKEW    1.22
                            MIN      0.004    SUM     596
```

4 Item_MRP





DataFrame

1 Item_Identifier

VALUES:	8,523 (100%)
MISSING:	---
DISTINCT:	1,559 (18%)

10	<1%	FDW13
10	<1%	FDG33
9	<1%	NCY18
9	<1%	FDD38
9	<1%	DRE49
9	<1%	FDV60
9	<1%	NCQ06
8,458	>99%	(Other)

2 Item_Weight

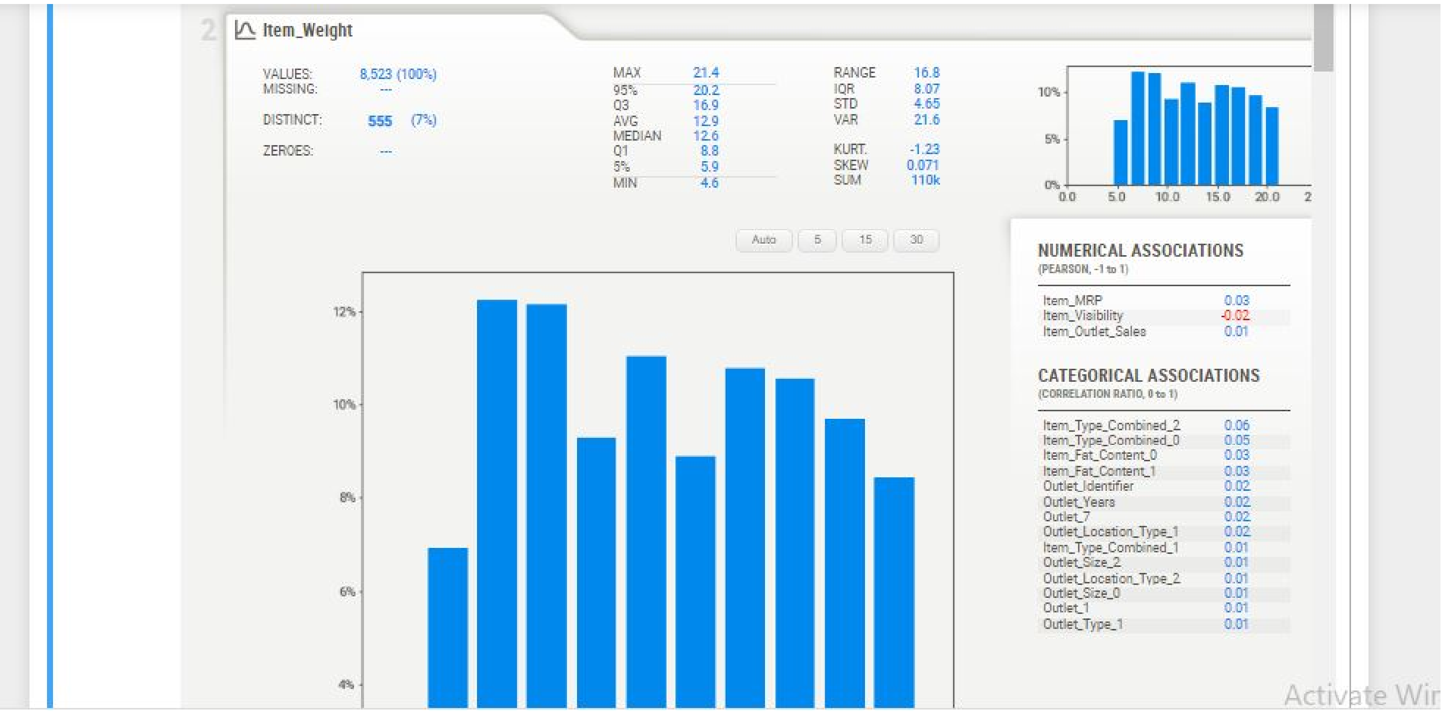
VALUES:	8,523 (100%)
MISSING:	---
DISTINCT:	555 (7%)
ZEROES:	---

MAX	21.4	RANGE	16.8
95%	20.2	IQR	8.07
Q3	16.9	STD	4.65
AVG	12.9	VAR	21.6
MEDIAN	12.6		
Q1	8.8	KURT.	-1.23
5%	5.9	SKEW	0.071
MIN	4.6	SUM	110k

3 Item_Visibility

VALUES:	8,523 (100%)
MISSING:	---
DISTINCT:	8,325 (98%)
ZEROES:	---

MAX	0.328	RANGE	0.325
95%	0.164	IQR	0.066
Q3	0.097	STD	0.050
AVG	0.070	VAR	0.002
MEDIAN	0.057		
Q1	0.031	KURT.	1.84
5%	0.013	SKEW	1.22
MIN	0.004	SUM	596



2 Item_Weight

VALUES:	8,523 (100%)
MISSING:	---
DISTINCT:	555 (7%)
ZEROES:	---

MAX	21.4	RANGE	16.8
95%	20.2	IQR	8.07
Q3	16.9	STD	4.65
AVG	12.9	VAR	21.6
MEDIAN	12.6		
Q1	8.8	KURT.	-1.23
5%	5.9	SKEW	0.071
MIN	4.6	SUM	110k

NUMERICAL ASSOCIATIONS

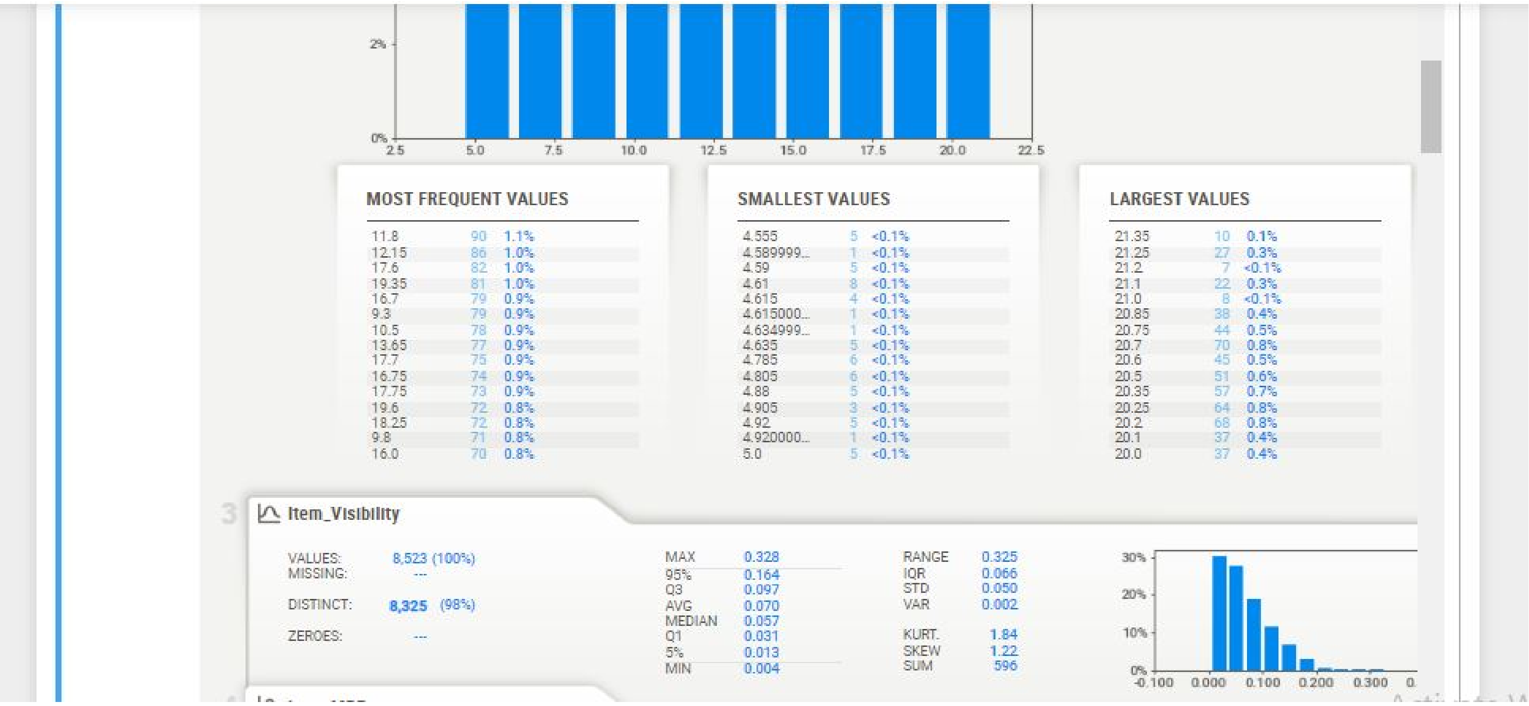
(PEARSON, -1 to 1)

Item_MRP	0.03
Item_Visibility	-0.02
Item_Outlet_Sales	0.01

CATEGORICAL ASSOCIATIONS

(CORRELATION RATIO, 0 to 1)

Item_Type_Combined_2	0.06
Item_Type_Combined_0	0.05
Item_Fat_Content_0	0.03
Item_Fat_Content_1	0.03
Outlet_Identifier	0.02
Outlet_Years	0.02
Outlet_7	0.02
Outlet_Location_Type_1	0.02
Item_Type_Combined_1	0.01
Outlet_Size_2	0.01
Outlet_Location_Type_2	0.01
Outlet_Size_0	0.01
Outlet_1	0.01
Outlet_Type_1	0.01



MOST FREQUENT VALUES

11.8	90	1.1%
12.15	86	1.0%
17.6	82	1.0%
19.35	81	1.0%
16.7	79	0.9%
9.3	79	0.9%
10.5	78	0.9%
13.65	77	0.9%
17.7	75	0.9%
16.75	74	0.9%
17.75	73	0.9%
19.6	72	0.8%
18.25	72	0.8%
9.8	71	0.8%
16.0	70	0.8%

SMALLEST VALUES

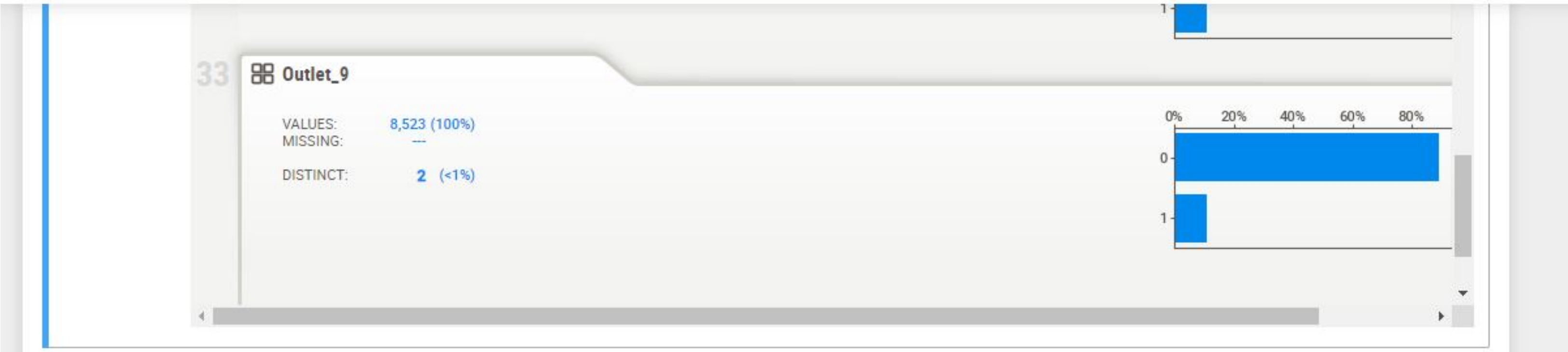
4.555	5	<0.1%
4.589999...	1	<0.1%
4.59	5	<0.1%
4.61	8	<0.1%
4.615	4	<0.1%
4.615000...	1	<0.1%
4.634999...	1	<0.1%
4.635	5	<0.1%
4.785	6	<0.1%
4.805	6	<0.1%
4.88	5	<0.1%
4.905	3	<0.1%
4.92	5	<0.1%
4.920000...	1	<0.1%
5.0	5	<0.1%

LARGEST VALUES

21.35	10	0.1%
21.25	27	0.3%
21.2	7	<0.1%
21.1	22	0.3%
21.0	8	<0.1%
20.85	38	0.4%
20.75	44	0.5%
20.7	70	0.8%
20.6	45	0.5%
20.5	51	0.6%
20.35	57	0.7%
20.25	64	0.8%
20.2	68	0.8%
20.1	37	0.4%
20.0	37	0.4%

3 Item_Visibility

VALUES:	8,523 (100%)	MAX	0.328	RANGE	0.325
MISSING:	---	95%	0.164	IQR	0.066
		Q3	0.097	STD	0.050
DISTINCT:	8,325 (98%)	AVG	0.070	VAR	0.002
		MEDIAN	0.057		
ZEROES:	---	Q1	0.031	KURT.	1.84
		5%	0.013	SKEW	1.22
		MIN	0.004	SUM	596

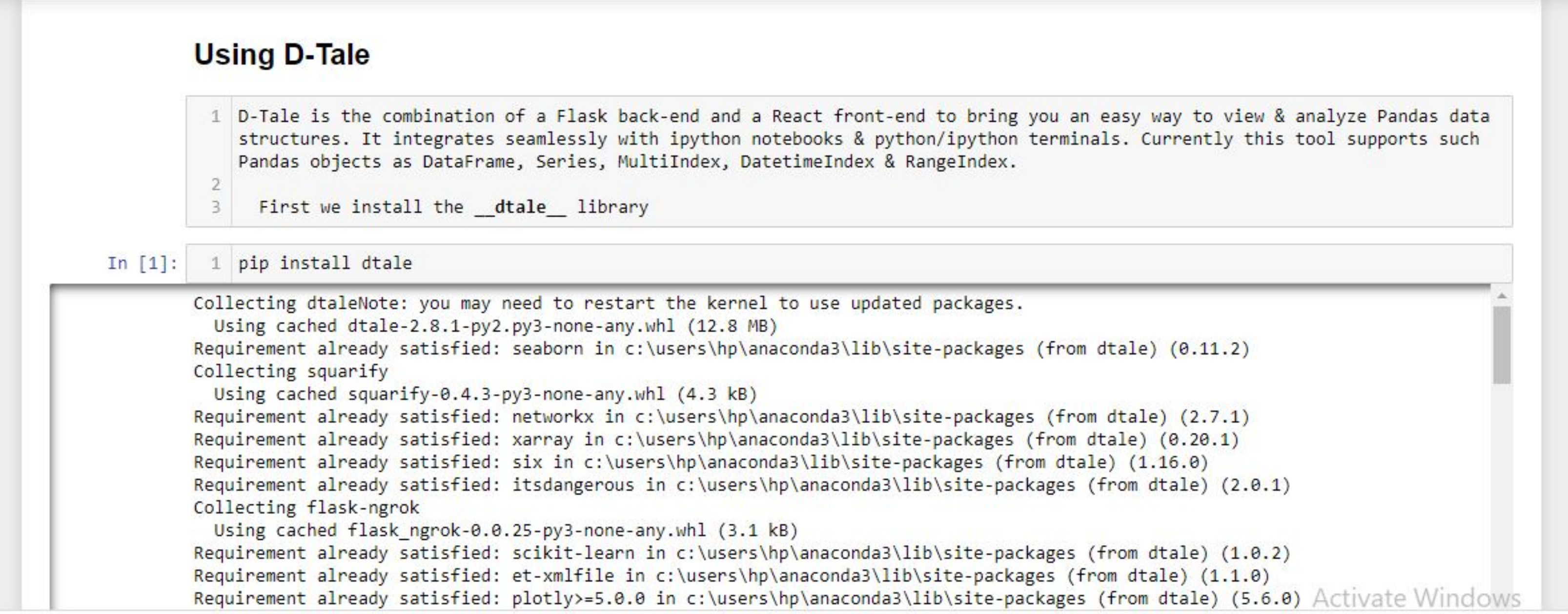


33 Outlet_9

VALUES:	8,523 (100%)
MISSING:	---
DISTINCT:	2 (<1%)

We can see that it creates a report which is simmilar in pattern to that of pandas profiling , the main difference is effects and visualization which is different . Here also we get an **Overview** on the top of the report which gives us a summary of the data , also there is an **Association** tab where we vcan see association between all the variables.

Then there is a list of all the features and on clicking on any feature , the section expands to give us all the information about the data , The visualization is **better** than pandas profiling as we get to see all the details very clearly and with nicely formed histograms and chars .

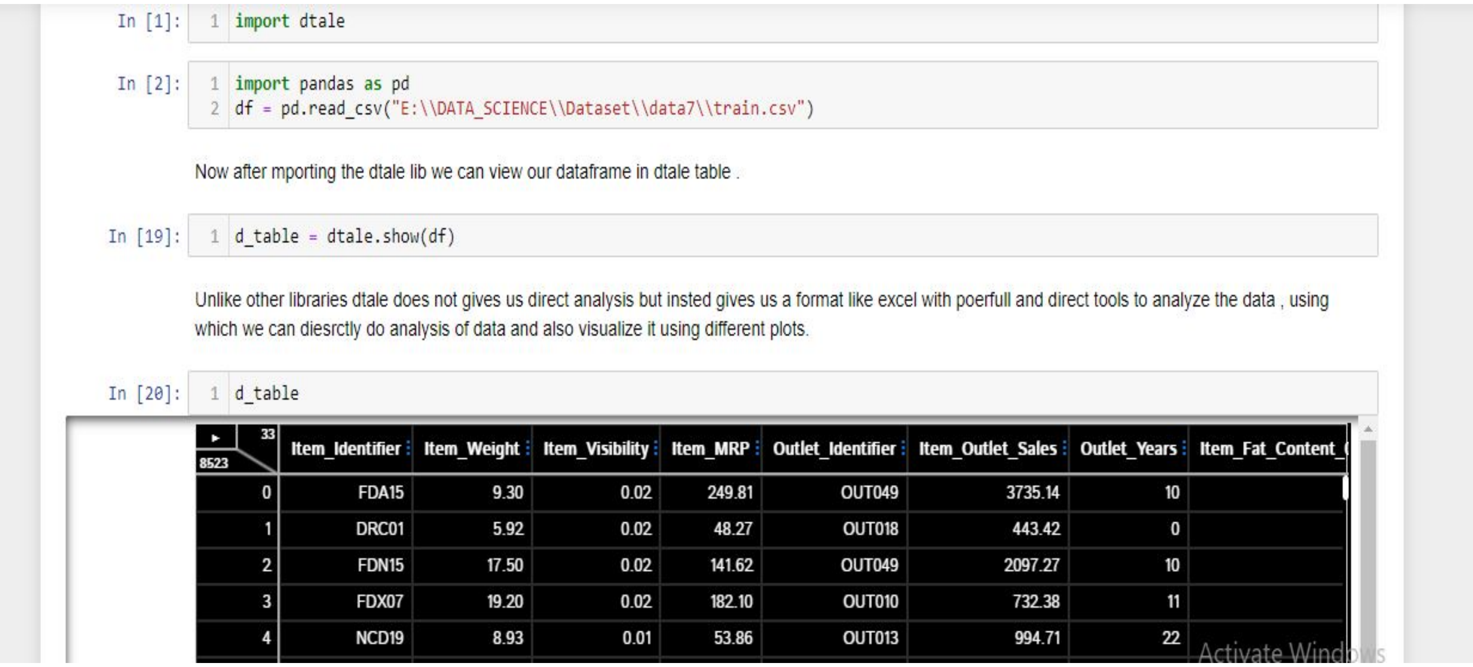


Using D-Tale

```
D-Tale is the combination of a Flask back-end and a React front-end to bring you an easy way to view & analyze Pandas data structures. It integrates seamlessly with ipython notebooks & python/ipython terminals. Currently this tool supports such Pandas objects as DataFrame, Series, MultiIndex, DatetimeIndex & RangeIndex.

 First we install the __dtale__ library
```

In [1]:
```
pip install dtale
```

```
Collecting dtaleNote: you may need to restart the kernel to use updated packages.
  Using cached dtale-2.8.1-py2.py3-none-any.whl (12.8 MB)
Requirement already satisfied: seaborn in c:\users\hp\anaconda3\lib\site-packages (from dtale) (0.11.2)
Collecting squarify
  Using cached squarify-0.4.3-py3-none-any.whl (4.3 kB)
Requirement already satisfied: networkx in c:\users\hp\anaconda3\lib\site-packages (from dtale) (2.7.1)
Requirement already satisfied: xarray in c:\users\hp\anaconda3\lib\site-packages (from dtale) (0.20.1)
Requirement already satisfied: six in c:\users\hp\anaconda3\lib\site-packages (from dtale) (1.16.0)
Requirement already satisfied: itsdangerous in c:\users\hp\anaconda3\lib\site-packages (from dtale) (2.0.1)
Collecting flask-ngrok
  Using cached flask_ngrok-0.0.25-py3-none-any.whl (3.1 kB)
Requirement already satisfied: scikit-learn in c:\users\hp\anaconda3\lib\site-packages (from dtale) (1.0.2)
Requirement already satisfied: et-xmlfile in c:\users\hp\anaconda3\lib\site-packages (from dtale) (1.1.0)
Requirement already satisfied: plotly>=5.0.0 in c:\users\hp\anaconda3\lib\site-packages (from dtale) (5.6.0)
```



In [1]:
```
import dtale
```

In [2]:
```
import pandas as pd
df = pd.read_csv("E:\\DATA_SCIENCE\\Dataset\\data7\\train.csv")
```

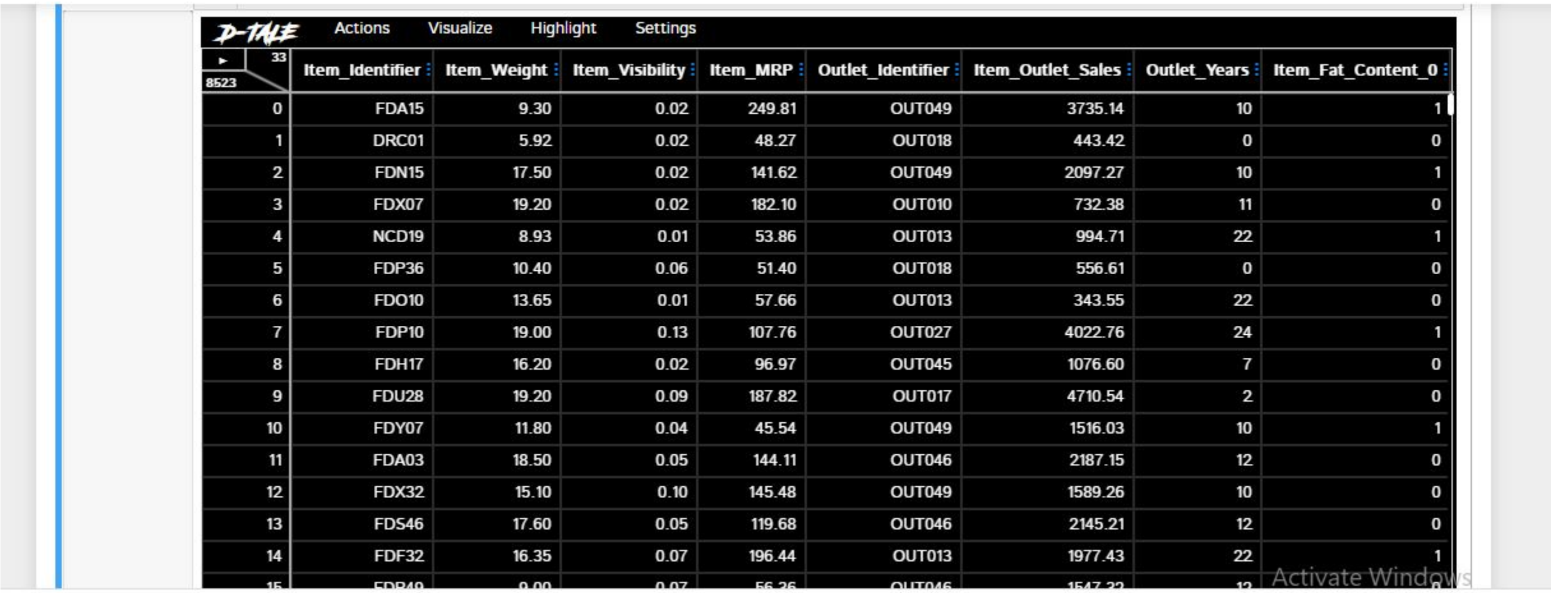
Now after mporting the dtale lib we can view our dataframe in dtale table .

In [19]:
```
d_table = dtale.show(df)
```

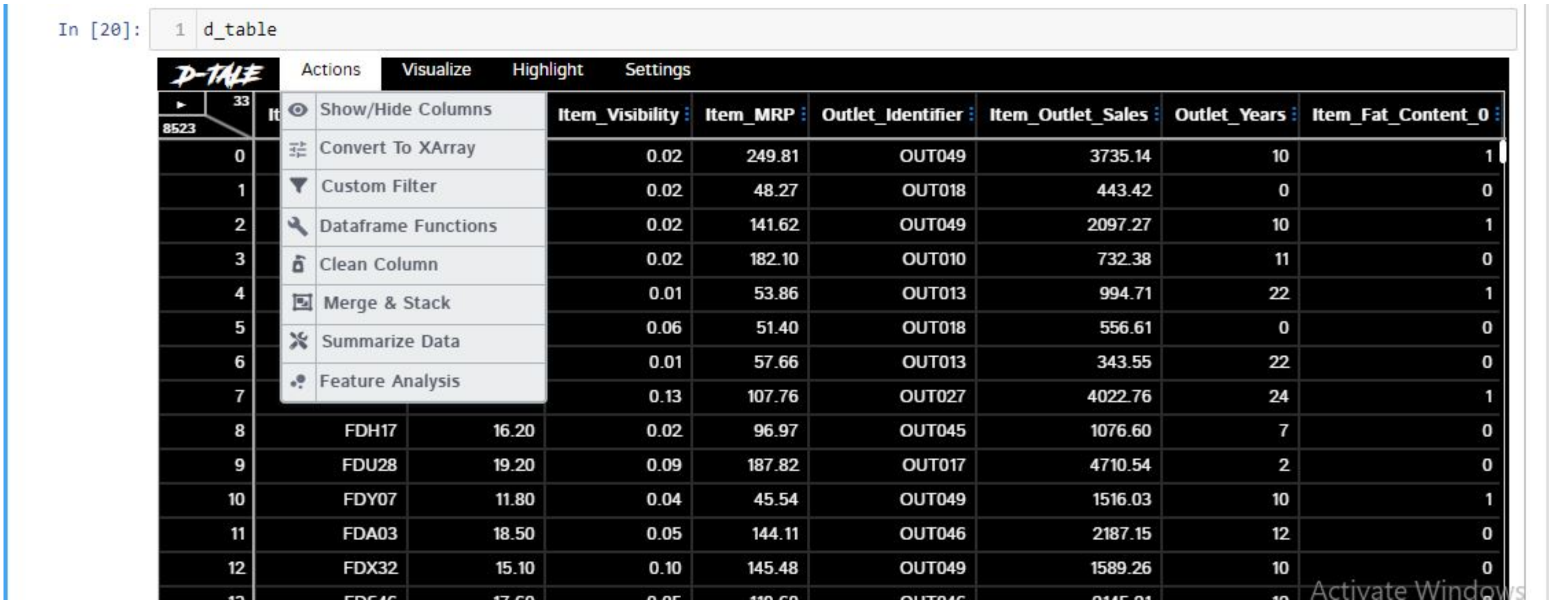
Unlike other libraries dtale does not gives us direct analysis but insted gives us a format like excel with poerfull and direct tools to analyze the data , using which we can diesrctly do analysis of data and also visualize it using different plots.

In [20]:
```
d_table
```

▸ 33 8523	Item_Identifier	Item_Weight	Item_Visibility	Item_MRP	Outlet_Identifier	Item_Outlet_Sales	Outlet_Years	Item_Fat_Content_(
0	FDA15	9.30	0.02	249.81	OUT049	3735.14	10	
1	DRC01	5.92	0.02	48.27	OUT018	443.42	0	
2	FDN15	17.50	0.02	141.62	OUT049	2097.27	10	
3	FDX07	19.20	0.02	182.10	OUT010	732.38	11	
4	NCD19	8.93	0.01	53.86	OUT013	994.71	22	

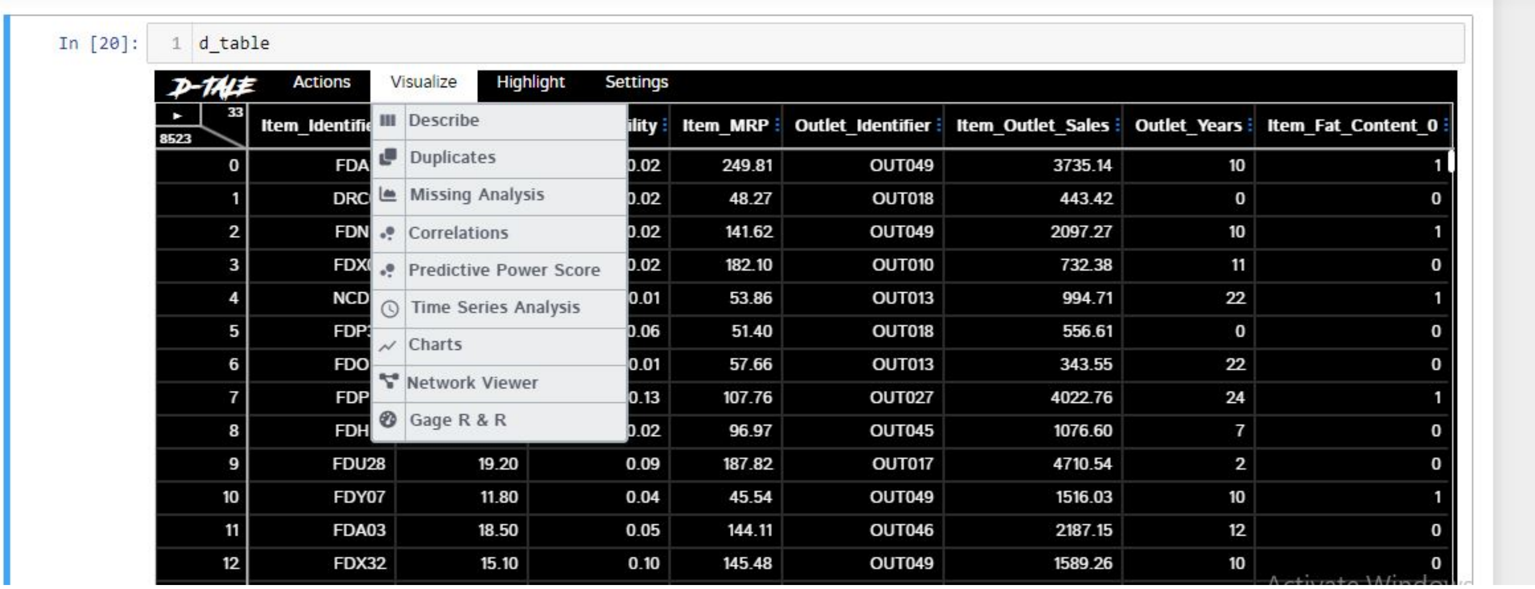


	Item_Identifier	Item_Weight	Item_Visibility	Item_MRP	Outlet_Identifier	Item_Outlet_Sales	Outlet_Years	Item_Fat_Content_0
0	FDA15	9.30	0.02	249.81	OUT049	3735.14	10	1
1	DRC01	5.92	0.02	48.27	OUT018	443.42	0	0
2	FDN15	17.50	0.02	141.62	OUT049	2097.27	10	1
3	FDX07	19.20	0.02	182.10	OUT010	732.38	11	0
4	NCD19	8.93	0.01	53.86	OUT013	994.71	22	1
5	FDP36	10.40	0.06	51.40	OUT018	556.61	0	0
6	FDO10	13.65	0.01	57.66	OUT013	343.55	22	0
7	FDP10	19.00	0.13	107.76	OUT027	4022.76	24	1
8	FDH17	16.20	0.02	96.97	OUT045	1076.60	7	0
9	FDU28	19.20	0.09	187.82	OUT017	4710.54	2	0
10	FDY07	11.80	0.04	45.54	OUT049	1516.03	10	1
11	FDA03	18.50	0.05	144.11	OUT046	2187.15	12	0
12	FDX32	15.10	0.10	145.48	OUT049	1589.26	10	0
13	FDS46	17.60	0.05	119.68	OUT046	2145.21	12	0
14	FDF32	16.35	0.07	196.44	OUT013	1977.43	22	1



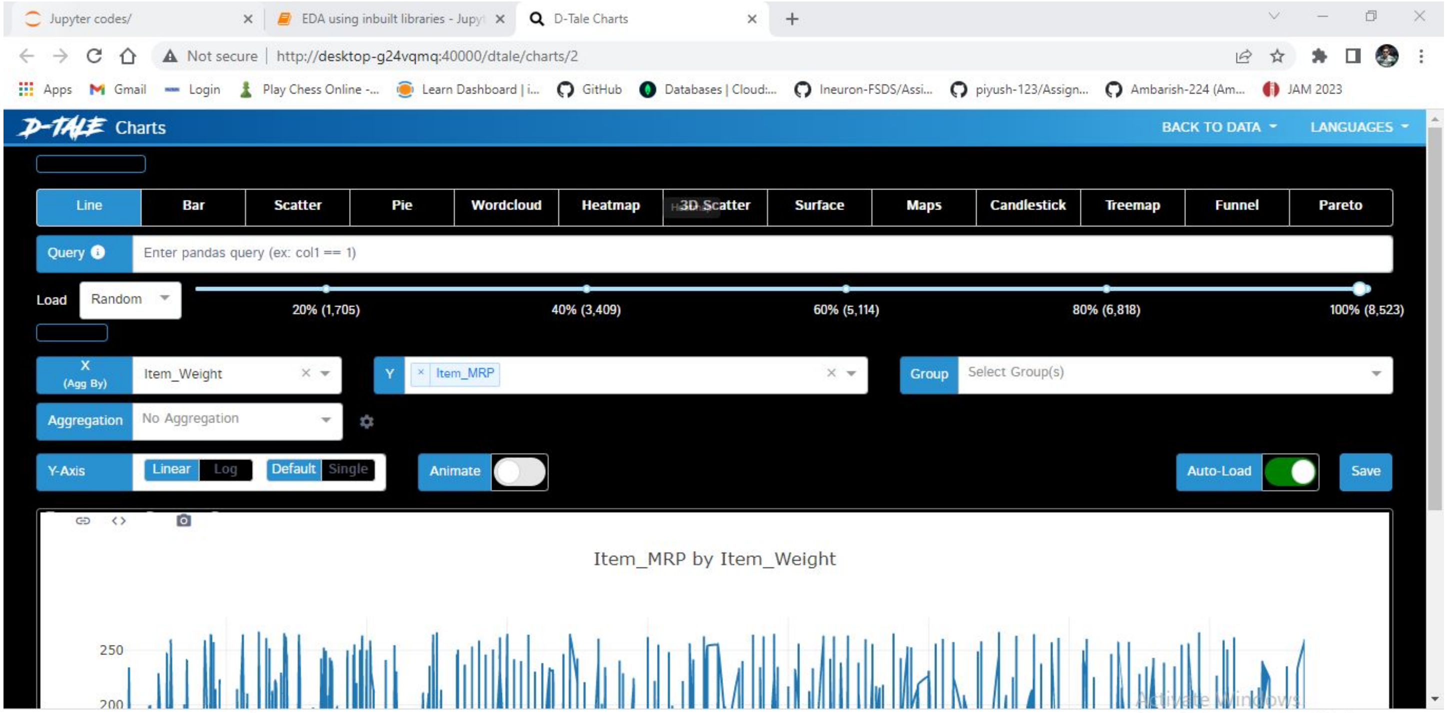
```
In [20]:  d_table
```

	It		Item_Visibility	Item_MRP	Outlet_Identifier	Item_Outlet_Sales	Outlet_Years	Item_Fat_Content_0
0			0.02	249.81	OUT049	3735.14	10	1
1			0.02	48.27	OUT018	443.42	0	0
2			0.02	141.62	OUT049	2097.27	10	1
3			0.02	182.10	OUT010	732.38	11	0
4			0.01	53.86	OUT013	994.71	22	1
5			0.06	51.40	OUT018	556.61	0	0
6			0.01	57.66	OUT013	343.55	22	0
7			0.13	107.76	OUT027	4022.76	24	1
8	FDH17	16.20	0.02	96.97	OUT045	1076.60	7	0
9	FDU28	19.20	0.09	187.82	OUT017	4710.54	2	0
10	FDY07	11.80	0.04	45.54	OUT049	1516.03	10	1
11	FDA03	18.50	0.05	144.11	OUT046	2187.15	12	0
12	FDX32	15.10	0.10	145.48	OUT049	1589.26	10	0



```
In [20]:  d_table
```

	Item_Identifie		ility	Item_MRP	Outlet_Identifier	Item_Outlet_Sales	Outlet_Years	Item_Fat_Content_0
0	FDA		0.02	249.81	OUT049	3735.14	10	1
1	DRC		0.02	48.27	OUT018	443.42	0	0
2	FDN		0.02	141.62	OUT049	2097.27	10	1
3	FDX		0.02	182.10	OUT010	732.38	11	0
4	NCD		0.01	53.86	OUT013	994.71	22	1
5	FDP		0.06	51.40	OUT018	556.61	0	0
6	FDO		0.01	57.66	OUT013	343.55	22	0
7	FDP		0.13	107.76	OUT027	4022.76	24	1
8	FDH		0.02	96.97	OUT045	1076.60	7	0
9	FDU28	19.20	0.09	187.82	OUT017	4710.54	2	0
10	FDY07	11.80	0.04	45.54	OUT049	1516.03	10	1
11	FDA03	18.50	0.05	144.11	OUT046	2187.15	12	0
12	FDX32	15.10	0.10	145.48	OUT049	1589.26	10	0



http://desktop-g24vqmq:40000/dtale/charts/2

Query: Enter pandas query (ex: col1 == 1)

Load: Random

X (Agg By): Item_Weight

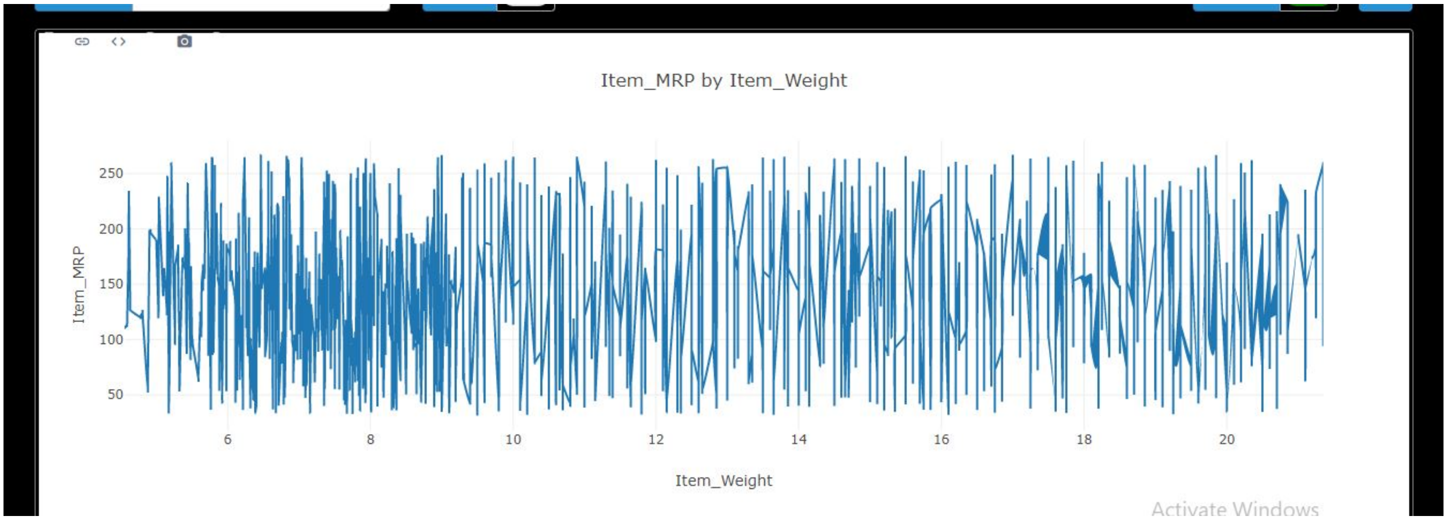
Y: Item_MRP

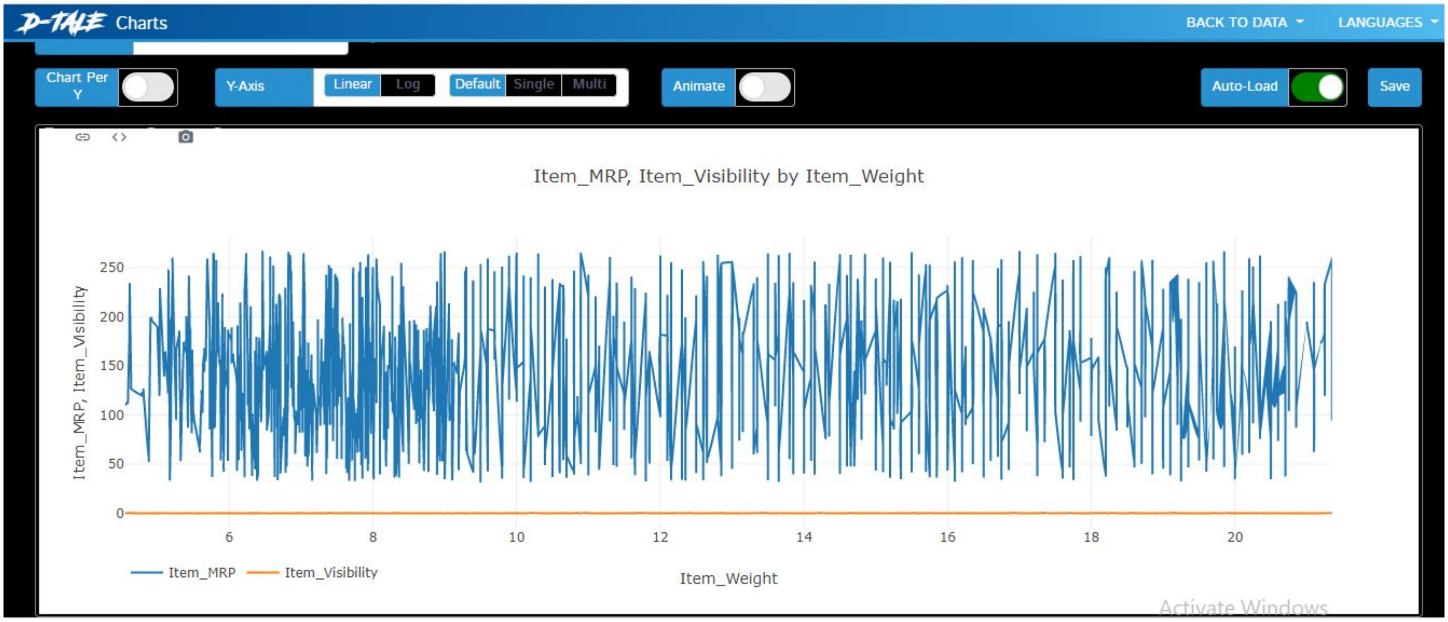
Group: Select Group(s)

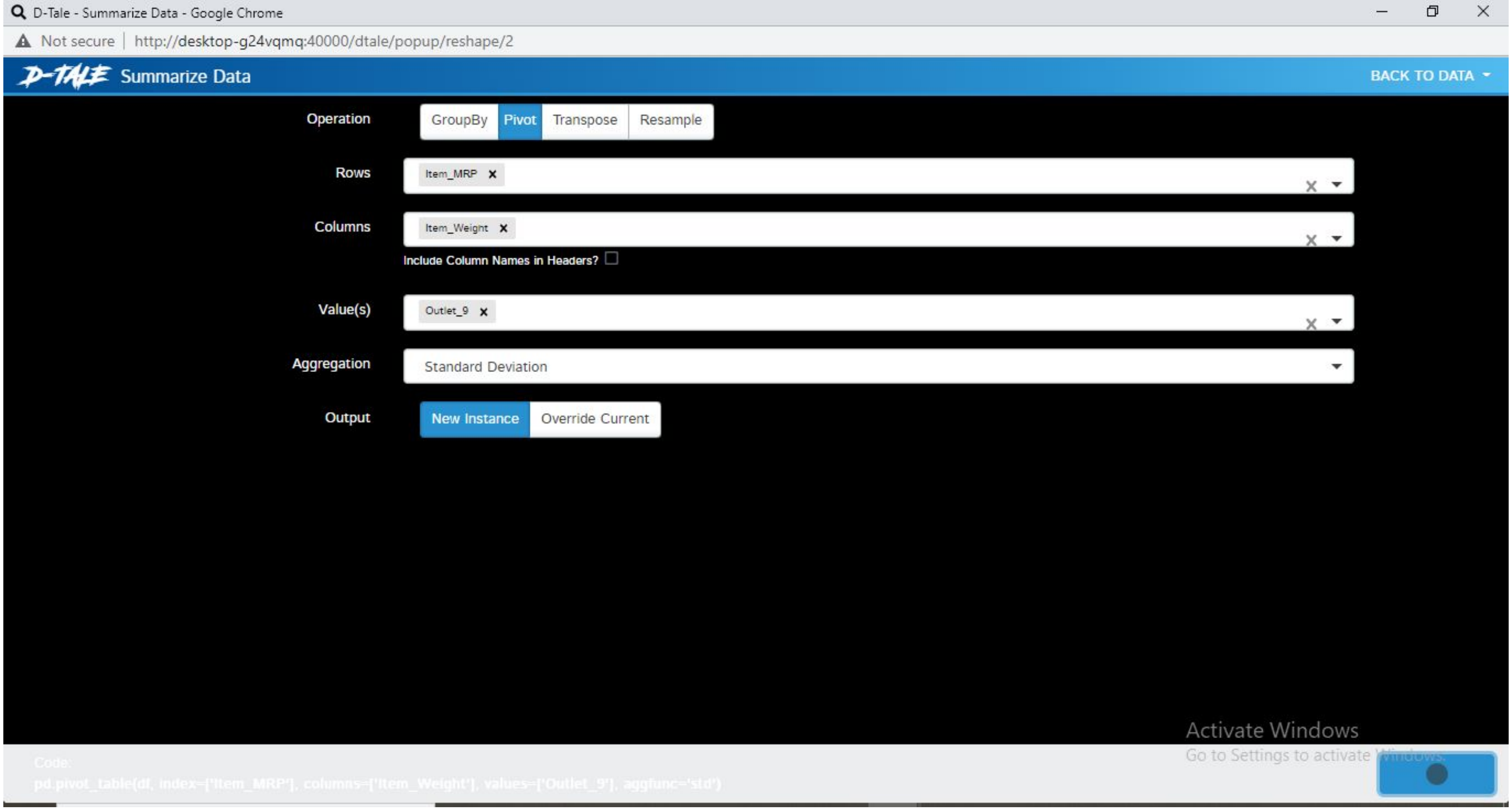
Aggregation: No Aggregation

Y-Axis: Linear Default

Item_MRP by Item_Weight


Item_MRP by Item_Weight
250
200
150
100
50
Item_MRP
6
8
10
12
14
16
18
20
Item_Weight
Activate Windows


D-TALE Charts
BACK TO DATA
LANGUAGES
Chart Per Y
Y-Axis
Linear
Log
Default
Single
Multi
Animate
Auto-Load
Save
Item_MRP, Item_Visibility by Item_Weight
250
200
150
100
50
0
Item_MRP, Item_Visibility
6
8
10
12
14
16
18
20
Item_MRP
Item_Visibility
Item_Weight
Activate Windows


D-Tale - Summarize Data - Google Chrome
Not secure | http://desktop-g24vqmq:40000/dtale/popup/reshape/2
D-TALE Summarize Data
BACK TO DATA
GroupBy
Transpose
Resample
Override Current
Activate Windows
Go to Settings to activate Windows.

Operation: Pivot

Rows: Item_MRP

Columns: Item_Weight

Include Column Names in Headers? □

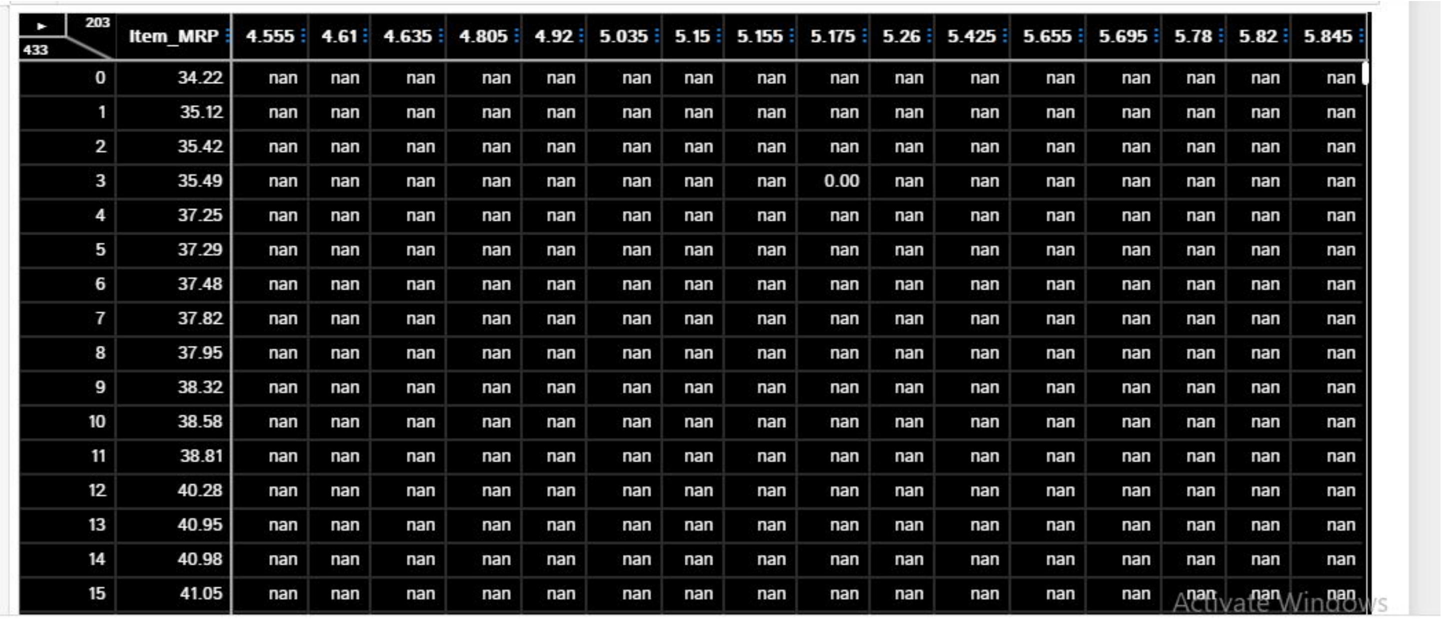
Value(s): Outlet_9

Aggregation: Standard Deviation

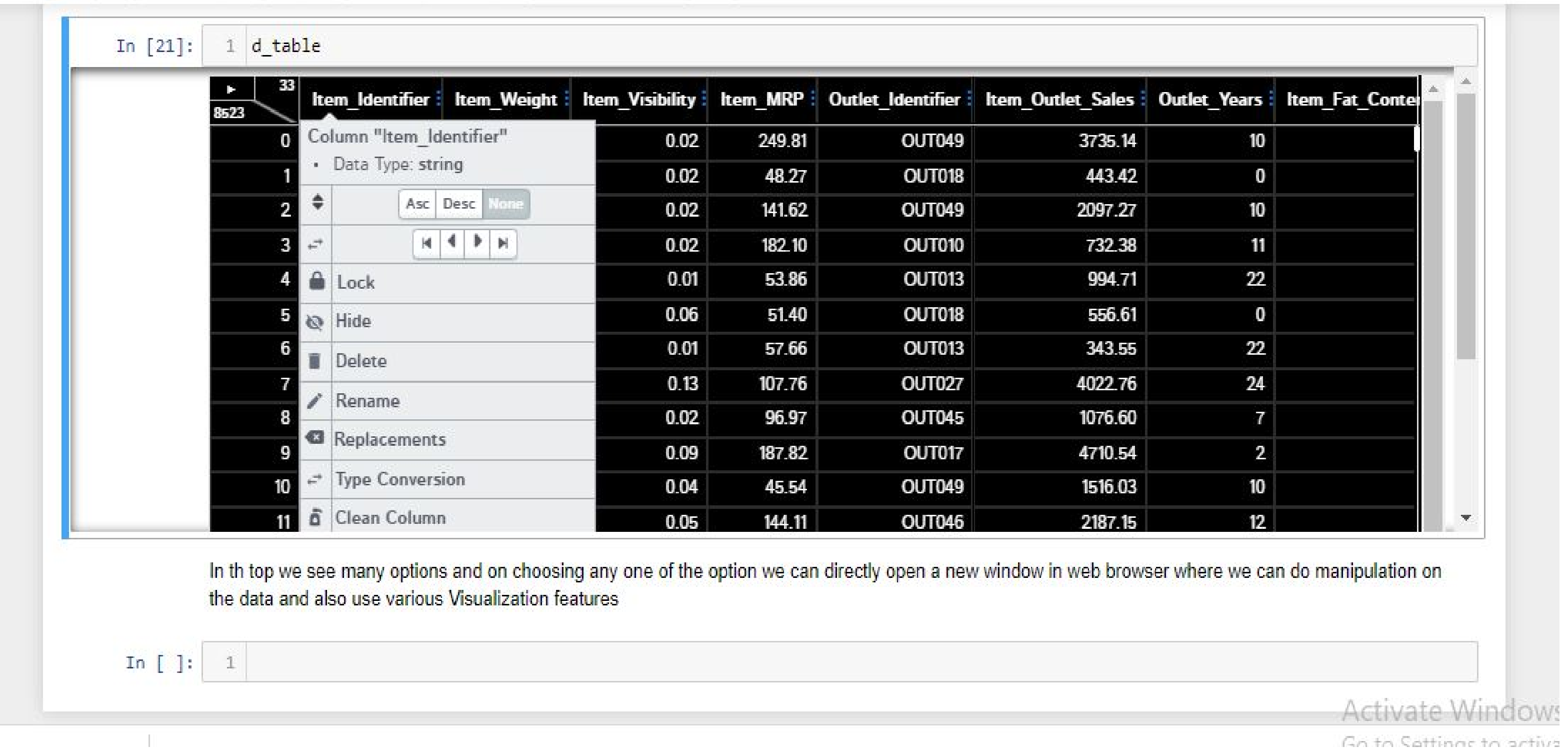
Output: New Instance

Code:

```
pd.pivot_table(df, index=['Item_MRP'], columns=['Item_Weight'], values=['Outlet_9'], aggfunc='std')
```



▸ 203 433	Item_MRP	4.555	4.61	4.635	4.805	4.92	5.035	5.15	5.155	5.175	5.26	5.425	5.655	5.695	5.78	5.82	5.845
0	34.22	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
1	35.12	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
2	35.42	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
3	35.49	nan	nan	nan	nan	nan	nan	nan	nan	0.00	nan	nan	nan	nan	nan	nan	nan
4	37.25	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
5	37.29	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
6	37.48	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
7	37.82	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
8	37.95	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
9	38.32	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
10	38.58	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
11	38.81	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
12	40.28	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
13	40.95	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
14	40.98	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
15	41.05	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan

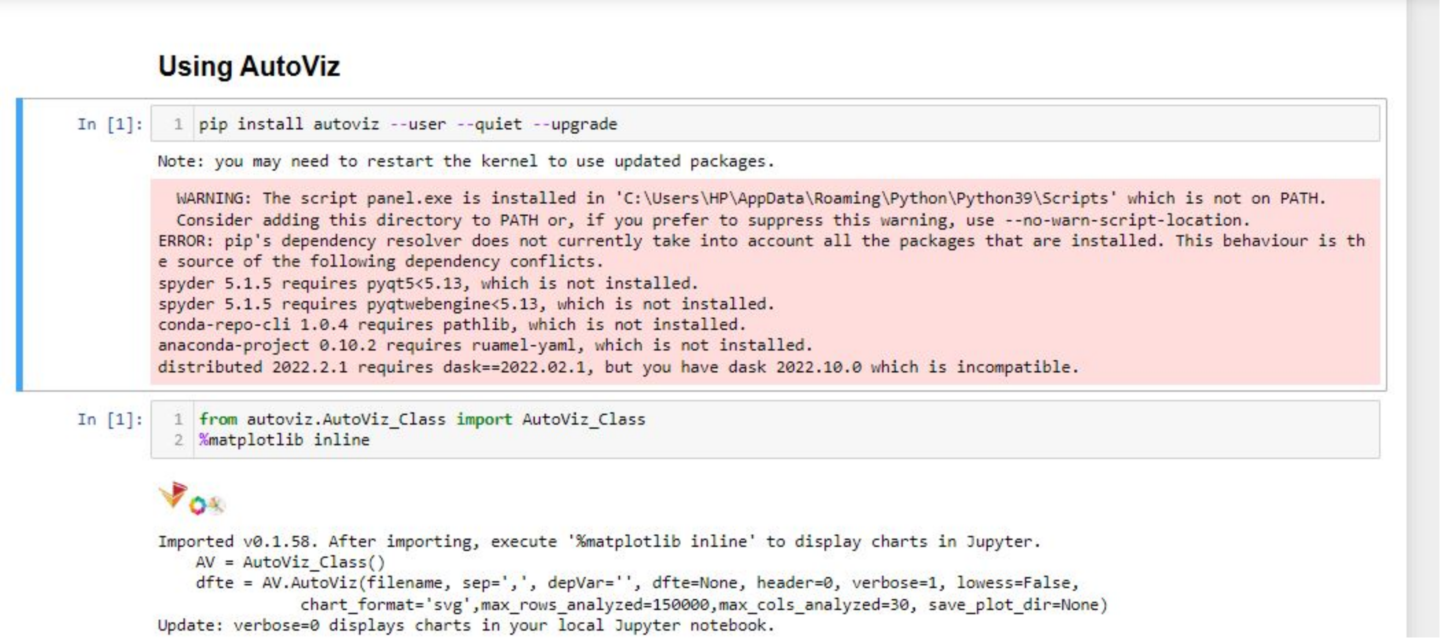


```
In [21]: d_table
```

	Item_Identifier	Item_Weight	Item_Visibility	Item_MRP	Outlet_Identifier	Item_Outlet_Sales	Outlet_Years	Item_Fat_Conte
0			0.02	249.81	OUT049	3735.14	10	
1			0.02	48.27	OUT018	443.42	0	
2			0.02	141.62	OUT049	2097.27	10	
3			0.02	182.10	OUT010	732.38	11	
4			0.01	53.86	OUT013	994.71	22	
5			0.06	51.40	OUT018	556.61	0	
6			0.01	57.66	OUT013	343.55	22	
7			0.13	107.76	OUT027	4022.76	24	
8			0.02	96.97	OUT045	1076.60	7	
9			0.09	187.82	OUT017	4710.54	2	
10			0.04	45.54	OUT049	1516.03	10	
11			0.05	144.11	OUT046	2187.15	12	

In th top we see many options and on choosing any one of the option we can directly open a new window in web browser where we can do manipulation on the data and also use various Visualization features

```
In [ ]:
```



Using AutoViz

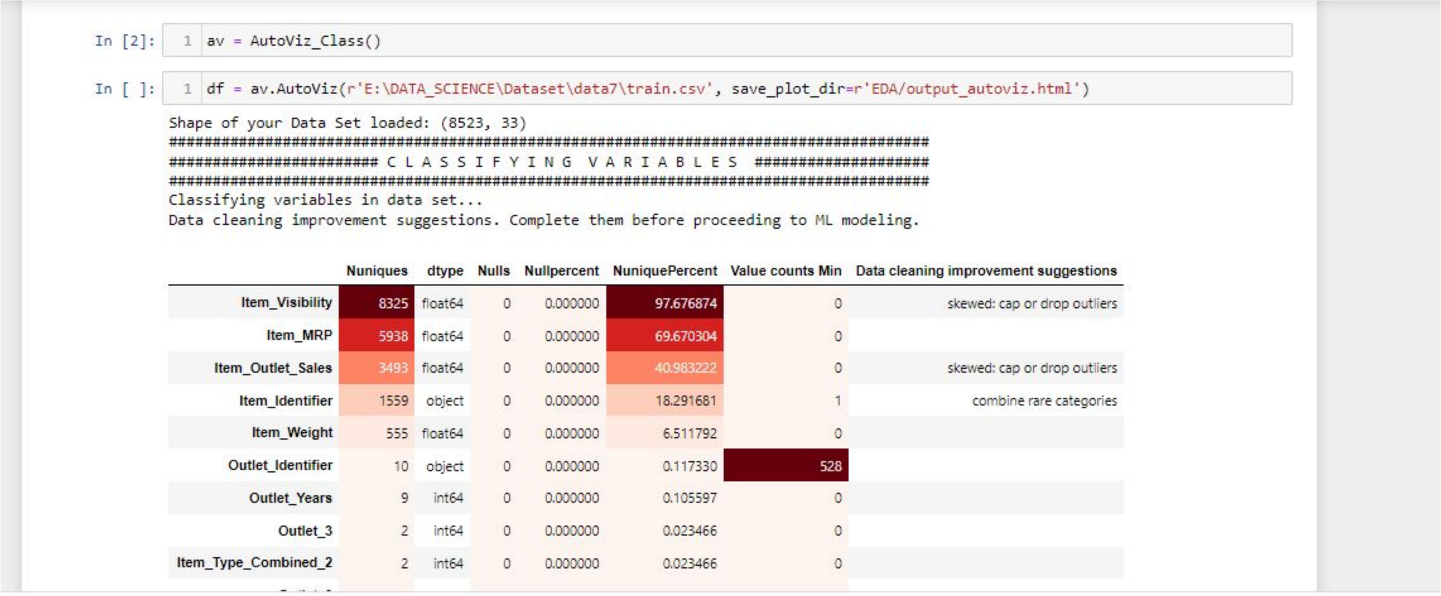
```
In [1]: pip install autoviz --user --quiet --upgrade
```

```
Note: you may need to restart the kernel to use updated packages.
```

```
  WARNING: The script panel.exe is installed in 'C:\Users\HP\AppData\Roaming\Python\Python39\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is th
e source of the following dependency conflicts.
spyder 5.1.5 requires pyqt5<5.13, which is not installed.
spyder 5.1.5 requires pyqtwebengine<5.13, which is not installed.
conda-repo-cli 1.0.4 requires pathlib, which is not installed.
anaconda-project 0.10.2 requires ruamel-yaml, which is not installed.
distributed 2022.2.1 requires dask==2022.02.1, but you have dask 2022.10.0 which is incompatible.
```

```
In [1]: from autoviz.AutoViz_Class import AutoViz_Class
        %matplotlib inline
```

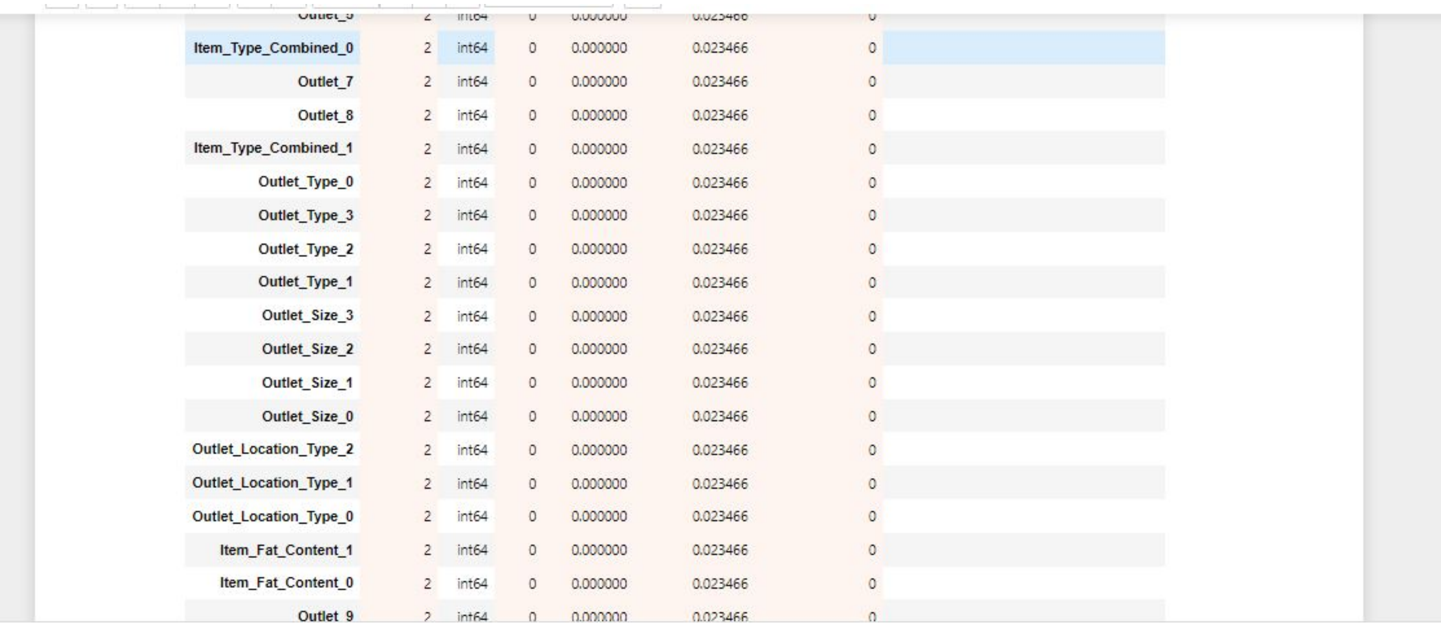
```
Imported v0.1.58. After importing, execute '%matplotlib inline' to display charts in Jupyter.
    AV = AutoViz_Class()
    dfte = AV.AutoViz(filename, sep=',', depVar='', dfte=None, header=0, verbose=1, lowess=False,
               chart_format='svg',max_rows_analyzed=150000,max_cols_analyzed=30, save_plot_dir=None)
Update: verbose=0 displays charts in your local Jupyter notebook.
```



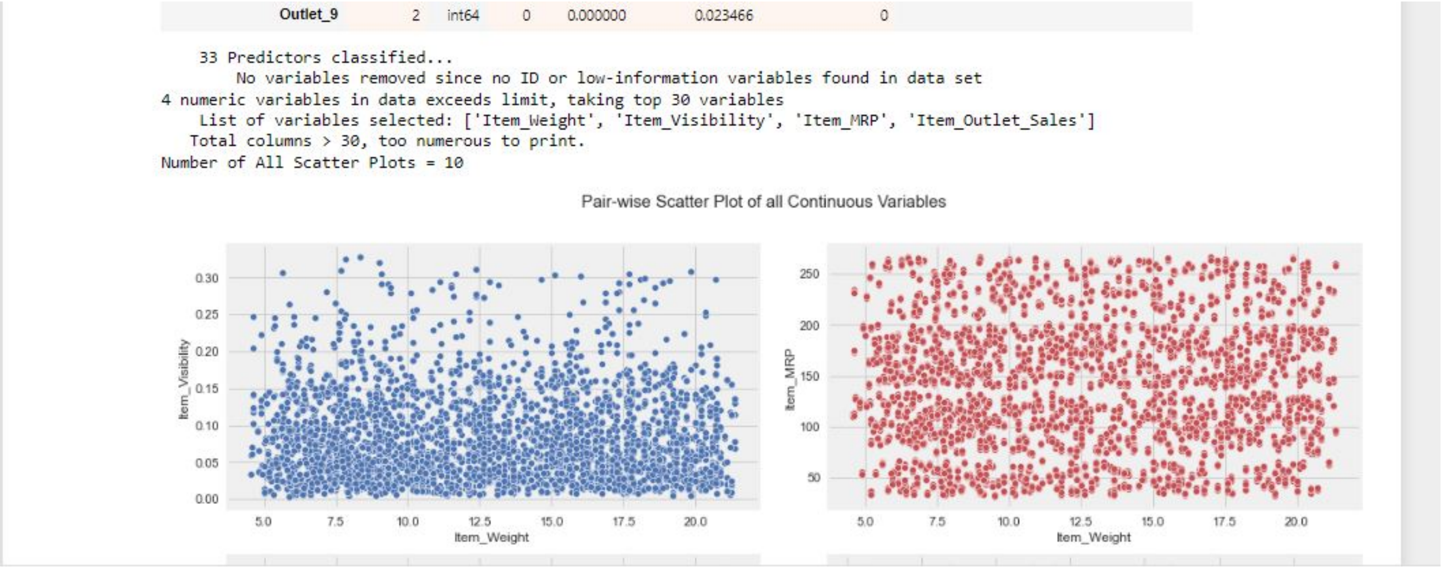
```
In [2]:  av = AutoViz_Class()

In [ ]:  df = av.AutoViz(r'E:\DATA_SCIENCE\Dataset\data7\train.csv', save_plot_dir=r'EDA/output_autoviz.html')
```

```
Shape of your Data Set loaded: (8523, 33)
#######################################################################################
######################## C L A S S I F Y I N G  V A R I A B L E S  ####################
#######################################################################################
Classifying variables in data set...
Data cleaning improvement suggestions. Complete them before proceeding to ML modeling.
```

	Nuniques	dtype	Nulls	Nullpercent	NuniquePercent	Value counts Min	Data cleaning improvement suggestions
Item_Visibility	8325	float64	0	0.000000	97.676874	0	skewed: cap or drop outliers
Item_MRP	5938	float64	0	0.000000	69.670304	0	
Item_Outlet_Sales	3493	float64	0	0.000000	40.983222	0	skewed: cap or drop outliers
Item_Identifier	1559	object	0	0.000000	18.291681	1	combine rare categories
Item_Weight	555	float64	0	0.000000	6.511792	0	
Outlet_Identifier	10	object	0	0.000000	0.117330	528	
Outlet_Years	9	int64	0	0.000000	0.105597	0	
Outlet_3	2	int64	0	0.000000	0.023466	0	
Item_Type_Combined_2	2	int64	0	0.000000	0.023466	0	



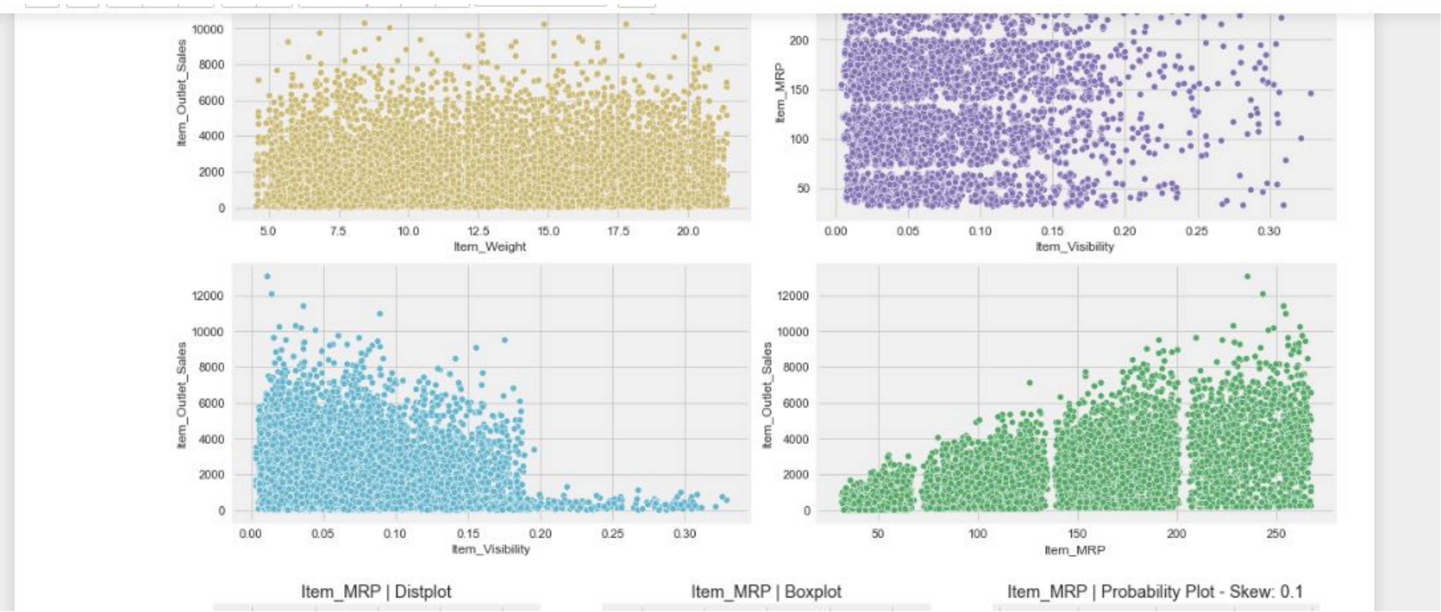
Outlet_5	2	int64	0	0.000000	0.023466	0	
Item_Type_Combined_0	2	int64	0	0.000000	0.023466	0	
Outlet_7	2	int64	0	0.000000	0.023466	0	
Outlet_8	2	int64	0	0.000000	0.023466	0	
Item_Type_Combined_1	2	int64	0	0.000000	0.023466	0	
Outlet_Type_0	2	int64	0	0.000000	0.023466	0	
Outlet_Type_3	2	int64	0	0.000000	0.023466	0	
Outlet_Type_2	2	int64	0	0.000000	0.023466	0	
Outlet_Type_1	2	int64	0	0.000000	0.023466	0	
Outlet_Size_3	2	int64	0	0.000000	0.023466	0	
Outlet_Size_2	2	int64	0	0.000000	0.023466	0	
Outlet_Size_1	2	int64	0	0.000000	0.023466	0	
Outlet_Size_0	2	int64	0	0.000000	0.023466	0	
Outlet_Location_Type_2	2	int64	0	0.000000	0.023466	0	
Outlet_Location_Type_1	2	int64	0	0.000000	0.023466	0	
Outlet_Location_Type_0	2	int64	0	0.000000	0.023466	0	
Item_Fat_Content_1	2	int64	0	0.000000	0.023466	0	
Item_Fat_Content_0	2	int64	0	0.000000	0.023466	0	
Outlet_9	2	int64	0	0.000000	0.023466	0	



Outlet_9	2	int64	0	0.000000	0.023466	0

```
    33 Predictors classified...
        No variables removed since no ID or low-information variables found in data set
4 numeric variables in data exceeds limit, taking top 30 variables
    List of variables selected: ['Item_Weight', 'Item_Visibility', 'Item_MRP', 'Item_Outlet_Sales']
   Total columns > 30, too numerous to print.
Number of All Scatter Plots = 10
```

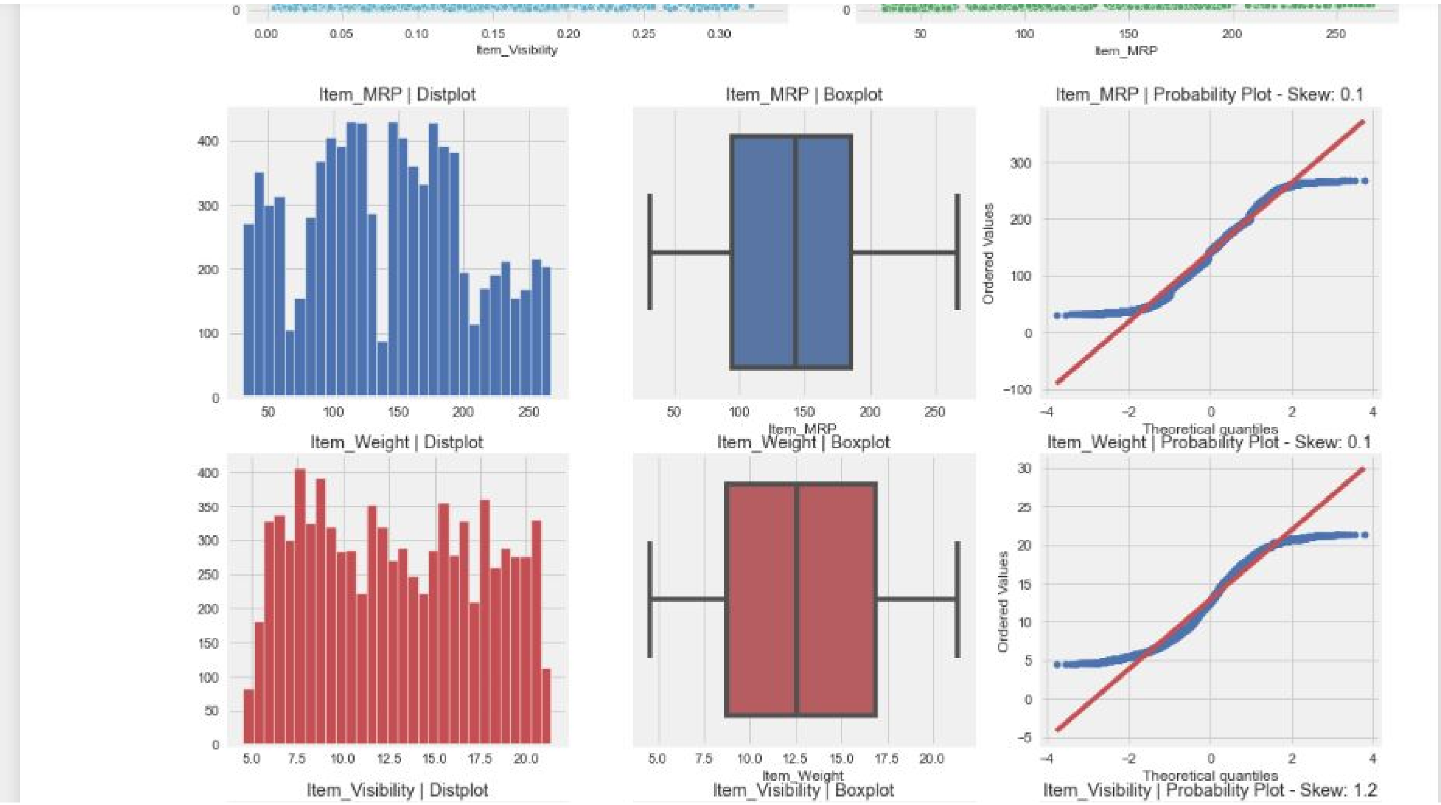
Pair-wise Scatter Plot of all Continuous Variables

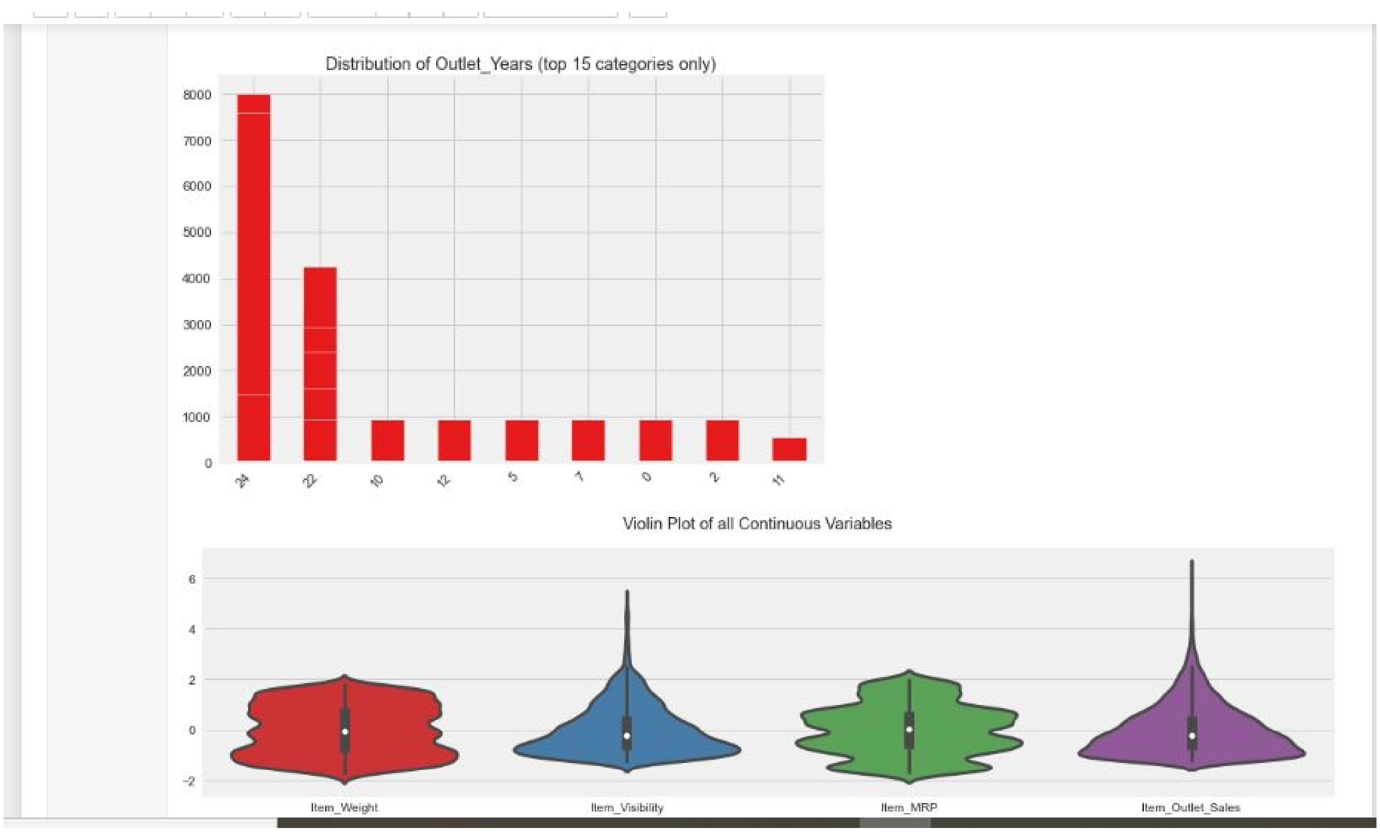


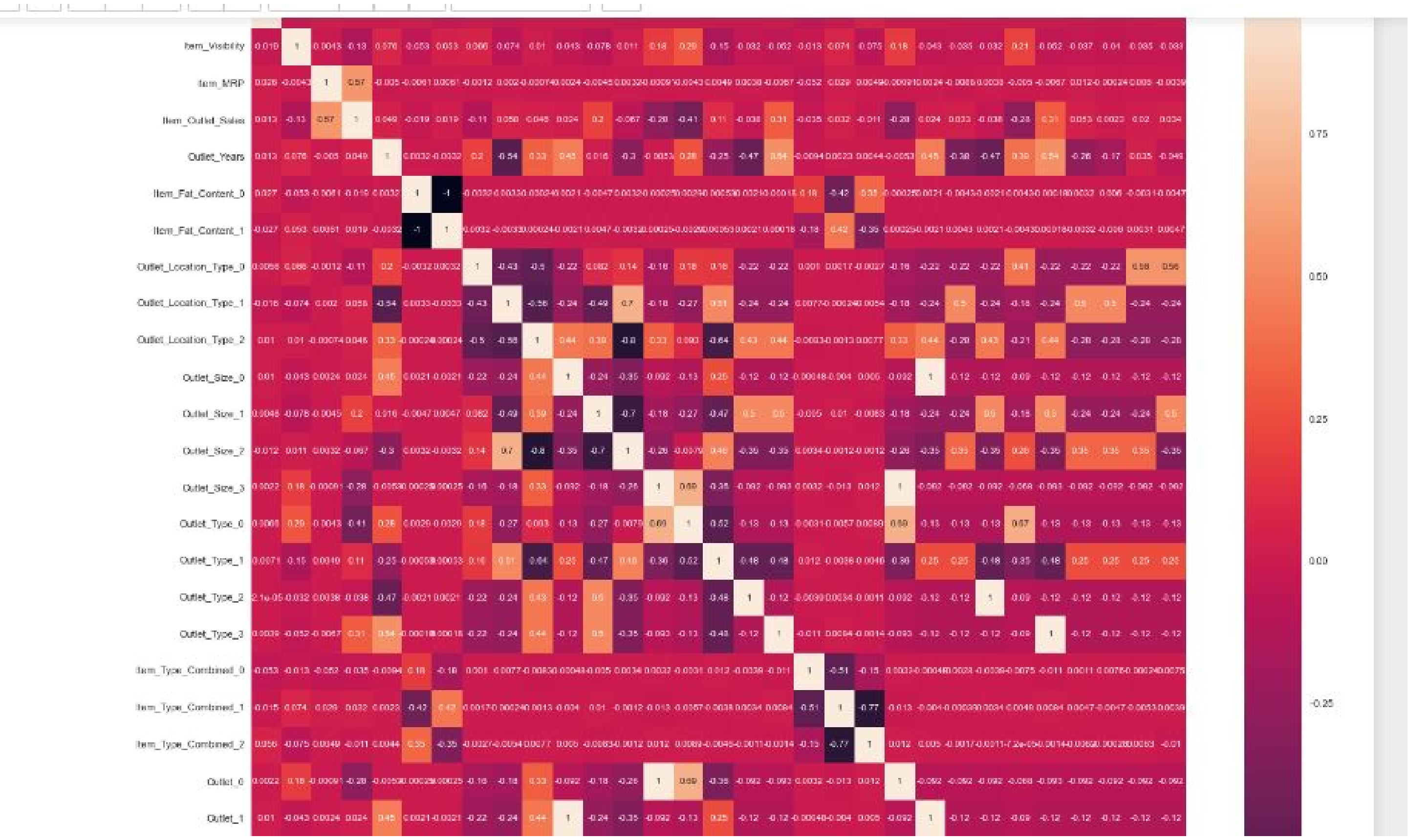
Item_MRP | Distplot

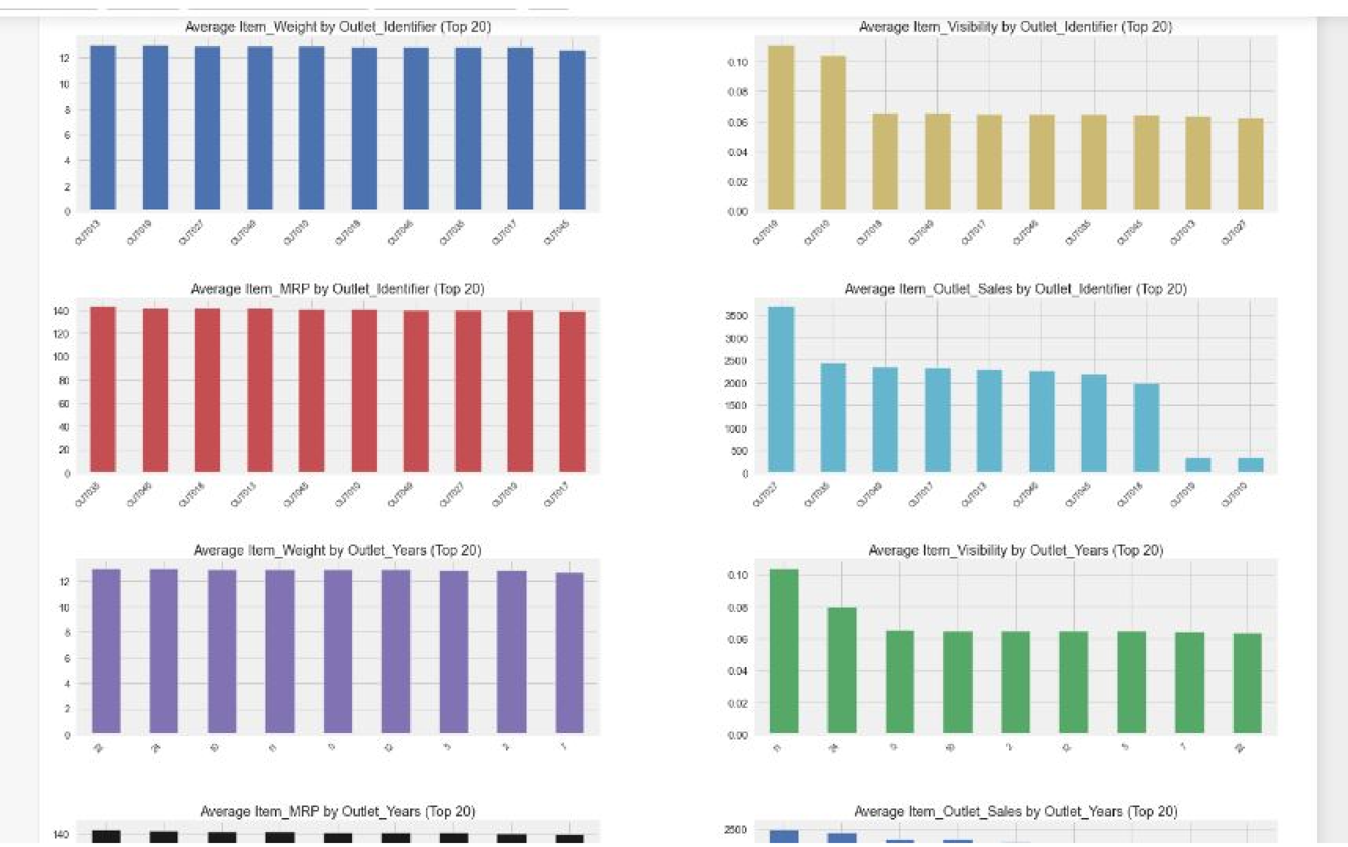
Item_MRP | Boxplot

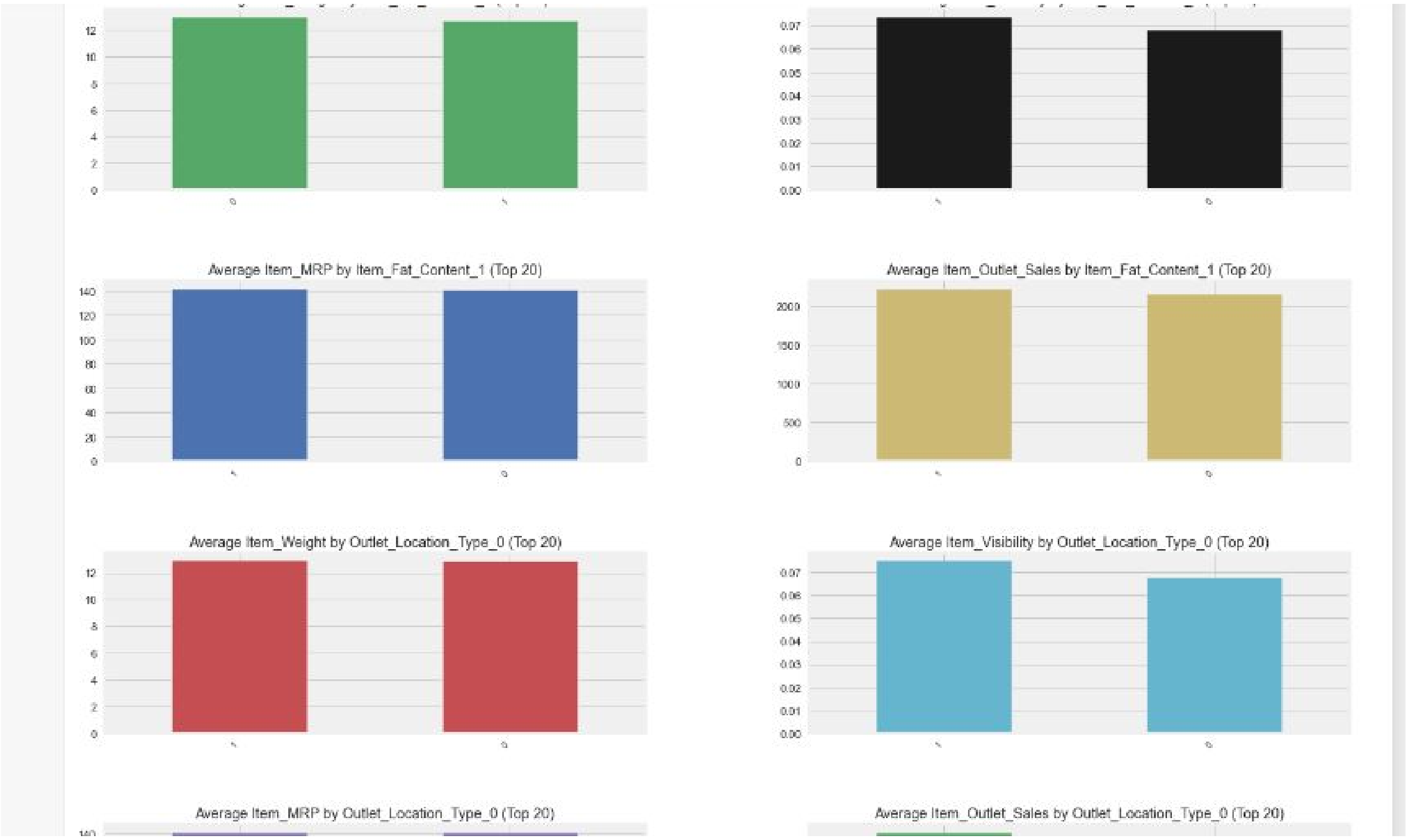
Item_MRP | Probability Plot - Skew: 0.1

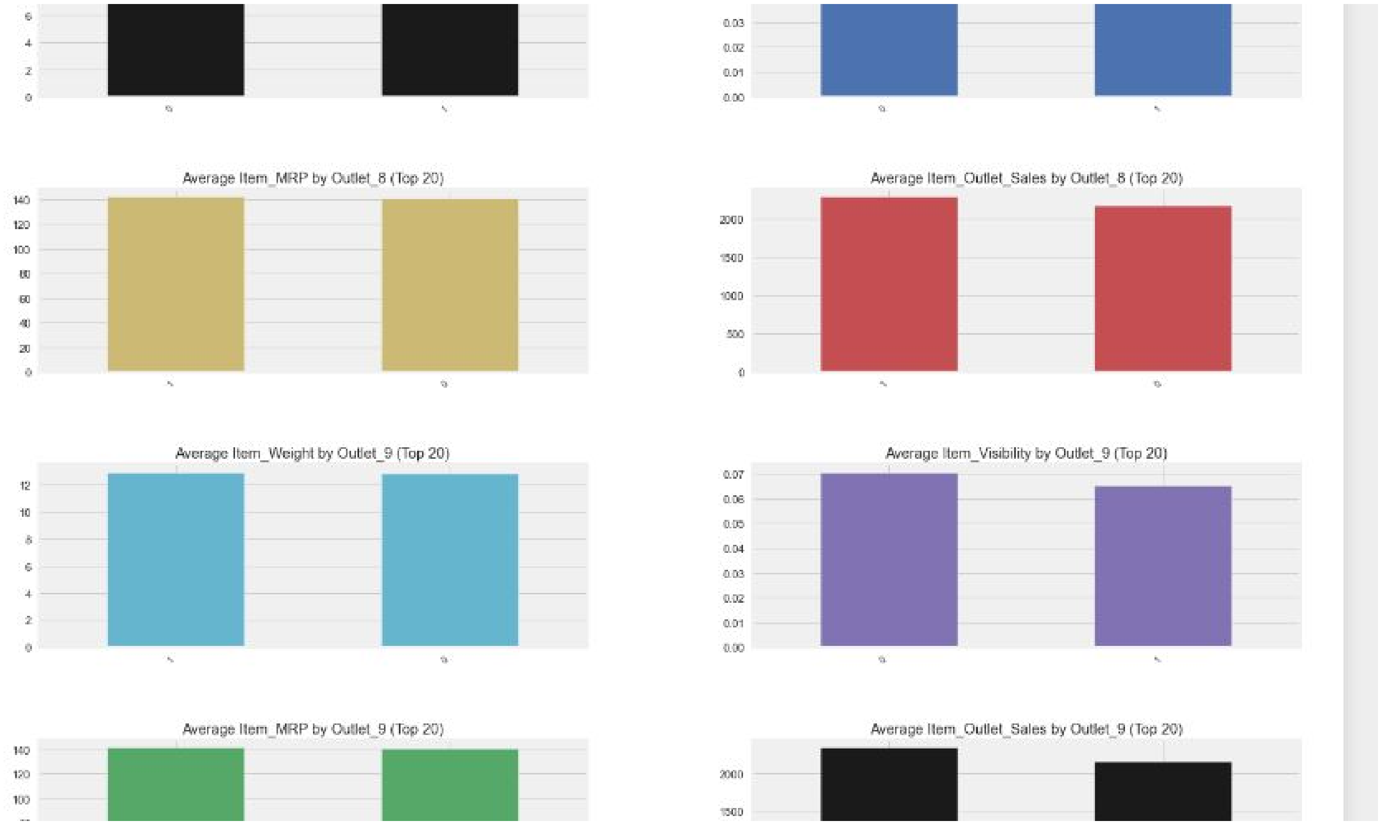

0
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Item_Visibility
0
50
100
150
200
250
Item_MRP
Item_MRP | Distplot
400
300
200
100
0
50
100
150
200
250
Item_MRP | Boxplot
50
100
150
200
250
Item_MRP
Item_MRP | Probability Plot - Skew: 0.1
300
200
100
0
−100
Ordered Values
−4
−2
0
2
4
Theoretical quantiles
Item_Weight | Distplot
400
350
300
250
200
150
100
50
0
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Item_Weight | Boxplot
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Item_Weight
Item_Weight | Probability Plot - Skew: 0.1
30
25
20
15
10
5
0
−5
Ordered Values
−4
−2
0
2
4
Theoretical quantiles
Item_Visibility | Distplot
Item_Visibility | Boxplot
Item_Visibility | Probability Plot - Skew: 1.2

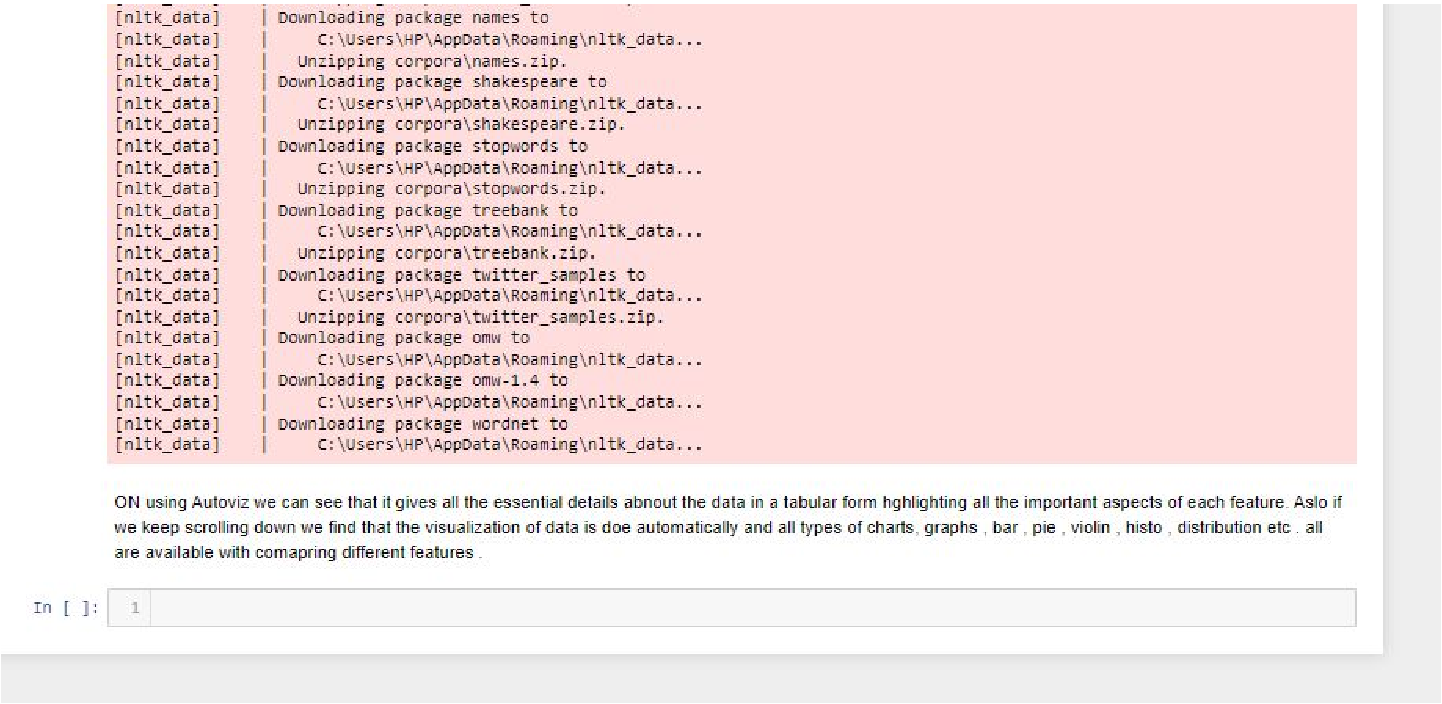

Distribution of Outlet_Years (top 15 categories only)
8000
7000
6000
5000
4000
3000
2000
1000
0
24
22
10
12
5
7
0
2
11
Violin Plot of all Continuous Variables
6
4
2
0
−2
Item_Weight
Item_Visibility
Item_MRP
Item_Outlet_Sales


Item_Visibility
Item_MRP
Item_Outlet_Sales
Outlet_Years
Item_Fat_Content_0
Item_Fat_Content_1
Outlet_Location_Type_0
Outlet_Location_Type_1
Outlet_Location_Type_2
Outlet_Size_0
Outlet_Size_1
Outlet_Size_2
Outlet_Size_3
Outlet_Type_0
Outlet_Type_1
Outlet_Type_2
Outlet_Type_3
Item_Type_Combined_0
Item_Type_Combined_1
Item_Type_Combined_2
Outlet_0
Outlet_1
0.75
0.50
0.25
0.00
-0.25


Average Item_Weight by Outlet_Identifier (Top 20)
0
2
4
6
8
10
12
Average Item_Visibility by Outlet_Identifier (Top 20)
0.00
0.02
0.04
0.06
0.08
0.10
Average Item_MRP by Outlet_Identifier (Top 20)
0
20
40
60
80
100
120
140
Average Item_Outlet_Sales by Outlet_Identifier (Top 20)
0
500
1000
1500
2000
2500
3000
3500
Average Item_Weight by Outlet_Years (Top 20)
0
2
4
6
8
10
12
Average Item_Visibility by Outlet_Years (Top 20)
0.00
0.02
0.04
0.06
0.08
0.10
Average Item_MRP by Outlet_Years (Top 20)
140
Average Item_Outlet_Sales by Outlet_Years (Top 20)
2500


12
10
8
6
4
2
0
0
1
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
1
0
Average Item_MRP by Item_Fat_Content_1 (Top 20)
140
120
100
80
60
40
20
0
1
0
Average Item_Outlet_Sales by Item_Fat_Content_1 (Top 20)
2000
1500
1000
500
0
1
0
Average Item_Weight by Outlet_Location_Type_0 (Top 20)
12
10
8
6
4
2
0
1
0
Average Item_Visibility by Outlet_Location_Type_0 (Top 20)
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
1
0
Average Item_MRP by Outlet_Location_Type_0 (Top 20)
140
Average Item_Outlet_Sales by Outlet_Location_Type_0 (Top 20)


6
4
2
0
0
1
0.03
0.02
0.01
0.00
0
1
Average Item_MRP by Outlet_8 (Top 20)
140
120
100
80
60
40
20
0
1
0
Average Item_Outlet_Sales by Outlet_8 (Top 20)
2000
1500
1000
500
0
1
0
Average Item_Weight by Outlet_9 (Top 20)
12
10
8
6
4
2
0
1
0
Average Item_Visibility by Outlet_9 (Top 20)
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0
1
Average Item_MRP by Outlet_9 (Top 20)
140
120
100
Average Item_Outlet_Sales by Outlet_9 (Top 20)
2000
1500



```
[nltk_data]    | Downloading package names to
[nltk_data]    |     C:\Users\HP\AppData\Roaming\nltk_data...
[nltk_data]    |   Unzipping corpora\names.zip.
[nltk_data]    | Downloading package shakespeare to
[nltk_data]    |     C:\Users\HP\AppData\Roaming\nltk_data...
[nltk_data]    |   Unzipping corpora\shakespeare.zip.
[nltk_data]    | Downloading package stopwords to
[nltk_data]    |     C:\Users\HP\AppData\Roaming\nltk_data...
[nltk_data]    |   Unzipping corpora\stopwords.zip.
[nltk_data]    | Downloading package treebank to
[nltk_data]    |     C:\Users\HP\AppData\Roaming\nltk_data...
[nltk_data]    |   Unzipping corpora\treebank.zip.
[nltk_data]    | Downloading package twitter_samples to
[nltk_data]    |     C:\Users\HP\AppData\Roaming\nltk_data...
[nltk_data]    |   Unzipping corpora\twitter_samples.zip.
[nltk_data]    | Downloading package omw to
[nltk_data]    |     C:\Users\HP\AppData\Roaming\nltk_data...
[nltk_data]    | Downloading package omw-1.4 to
[nltk_data]    |     C:\Users\HP\AppData\Roaming\nltk_data...
[nltk_data]    | Downloading package wordnet to
[nltk_data]    |     C:\Users\HP\AppData\Roaming\nltk_data...
```

ON using Autoviz we can see that it gives all the essential details abnout the data in a tabular form hghlighting all the important aspects of each feature. Aslo if we keep scrolling down we find that the visualization of data is doe automatically and all types of charts, graphs , bar , pie , violin , histo , distribution etc . all are available with comapring different features .

```
In [ ]:
```