ABOUT THE COMPANY

National instruments innovation center, center of excellence was established in 2015 under the department of electronic and communication Engineering in collaboration with National Instruments pvt. Ltd the world's largest company dedicated to virtual instrumentation technology.

This center has complete NI platform of both software and hardware as LabVIEW-2015 (Full Development System, Professional Development System, Vision Development Module, etc.), Multisim, NI MyRIO, NI MyDAQ, Wireless Sensor Network, ELVIS Board, NI Smart Camera, Robotic starter kit Mechatronics kits etc.

DURATION & ABOUT THE PROJECT

Duration:

The internship was for 6 weeks, from 23rd September 2022 to 18th November 2022 (As per in the certificate).

About the project:

The project was based on the analyzing the given data with the help of python libraries such as pandas, NumPy, matplotlib etc.

It consists of:

- **Data Collection:** The first step in data analytics is to collect or gather relevant data from multiple sources. Data can come from different databases, web servers, log files, social media, excel and CSV files, etc.
- **Data Preparation:** The next step in the process is to prepare the data. It involves cleaning the data to remove unwanted and redundant values, converting it into the right format, and making it ready for analysis. It also requires data wrangling.
- **Data Exploration:** After the data is ready, data exploration is done using various data visualization techniques to find unseen trends from the data.

- **Result interpretation:** The final step in any data analytics process is to derive meaningful results and check if the output is in line with your expected results.

By following all the above steps, we have analyzed and done the complete **EDA** (Exploratory Data Analysis) of the given datasets:

1. **Titanic** (with Target column '**Survived**')
2. **BigMart** (with Target column '**item outlet sales**')

The aim of this project is to represent statistics of the attrition trends and its causes by the help of graphs and charts.

IMPLEMENTATION

1. THE PROBLEM STATEMENT- Titanic:

Here are the highlights to note:

- On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. Translated 32% survival rate.
- One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew.
- Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others, such as women, children, and the upper-class.

Acquire data

The Python Pandas packages help us work with our dataset. We start by acquiring the titanic dataset into Pandas Data Frames.

```
In [1]: import pandas as pd
        import numpy as np
        import random as rnd
        import seaborn as sns
        import matplotlib.pyplot as plt
        %matplotlib inline
```

```
In [2]: tit=pd.read_csv('titanic.csv')
```

Analyze by describing data

Noting the feature names for directly manipulating or analyzing these.

```
In [3]: print(tit.columns.values)
['Survived' 'Pclass' 'Name' 'Sex' 'Age' 'Siblings/Spouses Aboard'
 'Parents/Children Aboard' 'Fare']
```

Which features are categorical?

These values classify the samples into sets of similar samples. Within categorical features are the values nominal, ordinal, ratio, or interval based? Among other things this helps us select the appropriate plots for visualization.

- Categorical: Survived, Sex, and Name. Ordinal: Pclass.

Which features are numerical?

These values change from sample to sample. Within numerical features are the values discrete, continuous, or time series based? Among other things this helps us select the appropriate plots for visualization.

- Continous: Age, Fare. Discrete: Siblings/Spouses Aboard, Parents/Children Aboard.

```
In [4]: tit.head()
```

Out[4]:

	Survived	Pclass	Name	Sex	Age	Siblings/Spouses Aboard	Parents/Children Aboard	Fare
0	0	3	Mr. Owen Harris Braund	male	22.0	1	0	7.2500
1	1	1	Mrs. John Bradley (Florence Briggs Thayer) Cum...	female	38.0	1	0	71.2833
2	1	3	Miss. Laina Heikkinen	female	26.0	0	0	7.9250
3	1	1	Mrs. Jacques Heath (Lily May Peel) Futrelle	female	35.0	1	0	53.1000
4	0	3	Mr. William Henry Allen	male	35.0	0	0	8.0500

```
In [5]: tit.tail()
```

Out[5]:

	Survived	Pclass	Name	Sex	Age	Siblings/Spouses Aboard	Parents/Children Aboard	Fare
882	0	2	Rev. Juozas Montvila	male	27.0	0	0	13.00
883	1	1	Miss. Margaret Edith Graham	female	19.0	0	0	30.00
884	0	3	Miss. Catherine Helen Johnston	female	7.0	1	2	23.45
885	1	1	Mr. Karl Howell Behr	male	26.0	0	0	30.00
886	0	3	Mr. Patrick Dooley	male	32.0	0	0	7.75

What is the distribution of numerical feature values across the samples?

This helps us determine, among other early insights, how representative is

the training dataset of the actual problem domain.

- Total samples are 891 or 40% of the actual number of passengers on board the Titanic (2,224).
- Survived is a categorical feature with 0 or 1 values.
- Around 38% samples survived representative of the actual survival rate at 32%.
- Most passengers (> 75%) did not travel with parents or children.
- Nearly 30% of the passengers had siblings and/or spouse aboard.
- Fares varied significantly with few passengers (<1%) paying as high as $512.
- Few elderly passengers (<1%) within age range 65-80.

```
In [6]: tit.describe()
```

Out[6]:

	Survived	Pclass	Age	Siblings/Spouses Aboard	Parents/Children Aboard	Fare
count	887.000000	887.000000	887.000000	887.000000	887.000000	887.00000
mean	0.385569	2.305524	29.471443	0.525366	0.383315	32.30542
std	0.487004	0.836662	14.121908	1.104669	0.807466	49.78204
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.00000
25%	0.000000	2.000000	20.250000	0.000000	0.000000	7.92500
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.45420
75%	1.000000	3.000000	38.000000	1.000000	0.000000	31.13750
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.32920

Analyze by pivoting features

To confirm some of our observations and assumptions, we can quickly analyze our feature correlations by pivoting features against each other. We can only do so at this stage for features which do not have any empty values. It also makes sense to do so only for features which are categorical (Sex), ordinal (Pclass) or discrete (Siblings/Spouses Aboard,

Parents/Children Aboard) type.

- **Pclass:** We observe a significant correlation (>0.5) among Pclass=1 and Survived. We decided to include this feature in our model.
- **Sex:** We confirm the observation during problem definition that Sex=female had a very high survival rate at 74%.
- **Siblings/Spouses Aboard, Parents/Children Aboard:** These features have zero correlation for certain values. It may be best to derive a feature or a set of features from these individual features.

```
In [7]: tit[['Pclass', 'Survived']].groupby(['Pclass'], as_index=False).mean().sort_values(by='Survived', ascending=False)
```

Out[7]:

	Pclass	Survived
0	1	0.629630
1	2	0.472826
2	3	0.244353

```
In [8]: tit[["Sex", "Survived"]].groupby(['Sex'], as_index=False).mean().sort_values(by='Survived', ascending=False)
```

Out[8]:

	Sex	Survived
0	female	0.742038
1	male	0.190227

```
In [9]: tit[["Siblings/Spouses Aboard", "Survived"]].groupby(['Siblings/Spouses Aboard'], as_index=False).mean().sort_valu
```

Out[9]:

	Siblings/Spouses Aboard	Survived
1	1	0.535885
2	2	0.464286
0	0	0.347682
3	3	0.250000
4	4	0.166667
5	5	0.000000
6	8	0.000000

```
In [10]: tit[["Parents/Children Aboard", "Survived"]].groupby(['Parents/Children Aboard'], as_index=False).mean().sort_value
```

Out[10]:

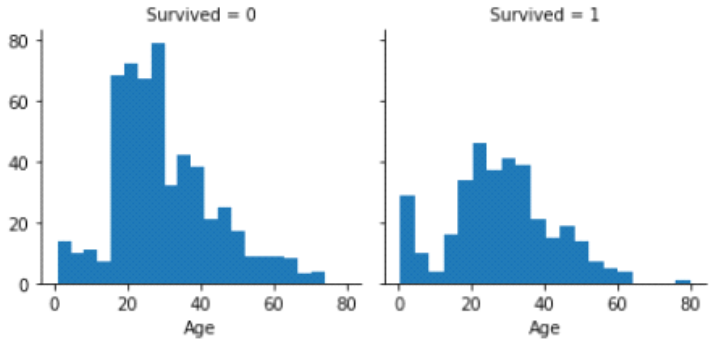
	Parents/Children Aboard	Survived
3	3	0.600000
1	1	0.550847
2	2	0.500000
0	0	0.345697
5	5	0.200000
4	4	0.000000
6	6	0.000000

Analyze by visualizing data

A histogram chart is useful for analyzing continuous numerical variables like Age where banding or ranges will help identify useful patterns.

Note that x-axis in histogram visualizations represents the count of samples or passengers.

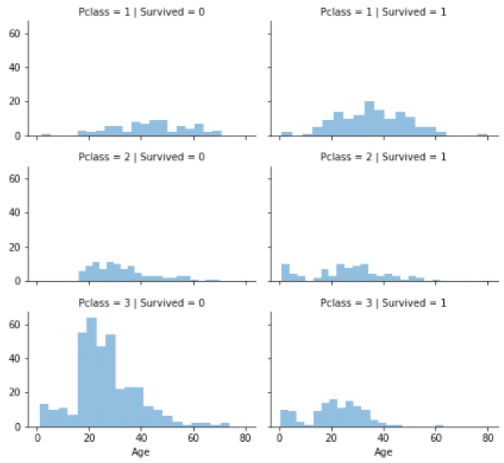
```
In [11]: g = sns.FacetGrid(tit, col='Survived')
         g.map(plt.hist, 'Age', bins=20)
Out[11]: <seaborn.axisgrid.FacetGrid at 0x6220dc0>
```



- Pclass=3 had most passengers, however most did not survive. Confirms our classifying assumption #2.
- Infant passengers in Pclass=2 and Pclass=3 mostly survived. Further qualifies our classifying assumption #2.
- Most passengers in Pclass=1 survived. Confirms our classifying assumption #3.
- Pclass varies in terms of Age distribution of passengers.

```
In [12]: grid = sns.FacetGrid(tit, col='Survived', row='Pclass', size=2.2, aspect=1.6)
         grid.map(plt.hist, 'Age', alpha=.5, bins=20)
         grid.add_legend();
```

```
C:\Users\hi\anaconda3\lib\site-packages\seaborn\axisgrid.py:337: UserWarning: The `size` parameter has been renamed
to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```



Let us create Age bands and determine correlations with Survived

```
In [13]: tit['AgeBand'] = pd.cut(tit['Age'], 5)
         tit[['AgeBand', 'Survived']].groupby(['AgeBand'], as_index=False).mean().sort_values(by='AgeBand', ascending=True)
```

Out[13]:

	AgeBand	Survived
0	(0.34, 16.336]	0.517544
1	(16.336, 32.252]	0.348786
2	(32.252, 48.168]	0.408696
3	(48.168, 64.084]	0.389610
4	(64.084, 80.0]	0.076923

This result is indicative while the competition is running.

2. THE PROBLEM STATEMENT- BigMart:

The data scientists at BigMart have collected sales data for 1559 products across 10 stores in different cities for the year 2013. Now each product has certain attributes that sets it apart from other products. Same is the case with each store.

The aim is to build a predictive model to find out the sales of each product at a particular store so that it would help the decision makers at BigMart to find out the properties of any product or store, which play a key role in

increasing the overall sales.

Loading Packages

In [1]:
```
import pandas as pd
import numpy as np                      # For mathematical calculations
import seaborn as sns                   # For data visualization
import matplotlib.pyplot as plt         # For plotting graphs
%matplotlib inline
import warnings   # To ignore any warnings
warnings.filterwarnings("ignore")
```

Reading Data

In [2]:
```
train = pd.read_csv('../input/bigmart-sales-data/Train.csv')
test = pd.read_csv('../input/bigmart-sales-data/Test.csv')
```

Let’s start with linking file path to the code as shown above. Basically, we gather file from the multiple resources and start the analysis of the dataset.

Dimensions of Data

In [3]:
```
train.shape,test.shape
```

Out[3]:
```
((8523, 12), (5681, 11))
```

Features of Data

In [5]:
```
train.info(),test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 12 columns):
Item_Identifier              8523 non-null object
Item_Weight                  7060 non-null float64
Item_Fat_Content             8523 non-null object
Item_Visibility              8523 non-null float64
Item_Type                    8523 non-null object
Item_MRP                     8523 non-null float64
Outlet_Identifier            8523 non-null object
Outlet_Establishment_Year    8523 non-null int64
Outlet_Size                  6113 non-null object
Outlet_Location_Type         8523 non-null object
Outlet_Type                  8523 non-null object
Item_Outlet_Sales            8523 non-null float64
dtypes: float64(4), int64(1), object(7)
memory usage: 799.2+ KB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5681 entries, 0 to 5680
Data columns (total 11 columns):
Item_Identifier              5681 non-null object
Item_Weight                  4705 non-null float64
Item_Fat_Content             5681 non-null object
Item_Visibility              5681 non-null float64
Item_Type                    5681 non-null object
Item_MRP                     5681 non-null float64
Outlet_Identifier            5681 non-null object
Outlet_Establishment_Year    5681 non-null int64
Outlet_Size                  4075 non-null object
Outlet_Location_Type         5681 non-null object
Outlet_Type                  5681 non-null object
dtypes: float64(3), int64(1), object(7)
memory usage: 488.3+ KB
```

DATA ANALYSIS OF DIFFERENT CATEGORY:

We can start the process by working on four levels: Store Level, Product Level, Customer Level and Macro Level.

Store Level Hypotheses

- **City type**: Stores located in urban or Tier 1 cities should have higher sales because of the higher income levels of people there.
- **Population Density**: Stores located in densely populated areas should have higher sales because of more demand.
- **Store Capacity**: Stores which are very big in size should have higher sales as they act like one-stop-shops and people would prefer getting everything from one place.
- **Competitors**: Stores having similar establishments nearby should have less sales because of more competition.
- **Marketing**: Stores which have a good marketing division should have higher sales as it will be able to attract customers through the right offers and advertising.
- **Location**: Stores located within popular marketplaces should have higher sales because of better access to customers
- **Ambiance**: Stores which are well-maintained and managed by polite and humble people are expected to have higher footfall and thus higher sales.

Product Level Hypotheses

- **Brand**: Branded products should have higher sales because of higher trust in the customer.
- **Packaging**: Products with good packaging can attract customers and sell more.
- **Utility**: Daily use products should have a higher tendency to sell as compared to the specific use products.
- **Display Area**: Products which are given bigger shelves in the store are likely to catch attention first and sell more. Visibility in Store: The location of product in a store will impact sales. Ones which are right at entrance will catch the eye of customer first rather than the ones in back.
- **Advertising**: Better advertising of products in the store will should higher sales in most cases. Promotional Offers: Products accompanied with attractive offers and discounts will sell more.

Customer Level Hypotheses

- **Customer Behavior**: Stores keeping the right set of products to meet the local needs of customers will have higher sales.
- **Job Profile**: Customer working at executive levels would have higher chances of purchasing high amount products as compared to customers working at entry or mid senior level.
- **Family Size**: More the number of family members, more amount

will be spent by a customer to buy products.

- **Annual Income**: Higher the annual income of a customer, customer is more likely to buy high cost products. Past Purchase History: Availablity of this information can help us to determine the frequency of a product being purchased by a user.

Macro Level Hypotheses

- **Environment**: If the environment is declared safe by government, customer would be more likely to purchase products without worrying if it's environment friendly or not.
- **Economic Growth**: If the current economy shows a consistent growth, per capita income will rise, therefore buying power of customers will increase.

We need to predict Item_Outlet_Sales for given test data

In [6]:
```
train['source'] = 'train'
# test['source'] = 'test'
test['Item_Outlet_Sales'] = 0
data = pd.concat([train, test], sort = False)
print(train.shape, test.shape, data.shape)
```

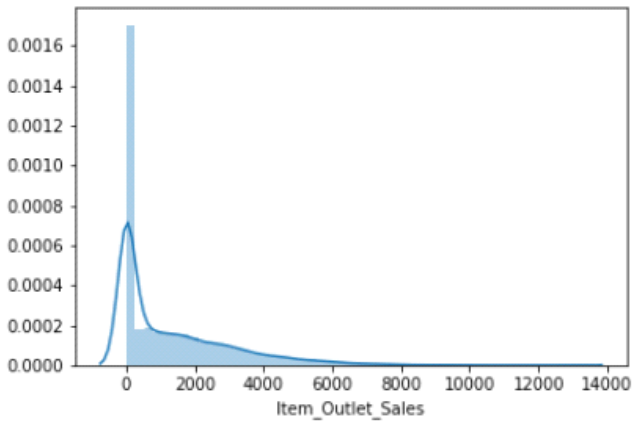
```
(8523, 13) (5681, 12) (14204, 13)
```

In [7]:
```
data['Item_Outlet_Sales'].describe()
```

Out[7]:
```
count    14204.000000
mean      1308.865489
std       1699.791423
min          0.000000
25%          0.000000
50%        559.272000
75%       2163.184200
max      13086.964800
Name: Item_Outlet_Sales, dtype: float64
```

In [8]:
```
sns.distplot(data['Item_Outlet_Sales'])
```

Out[8]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2e26c6e400>
```



In [9]:
```
print('Skewness: %f' % data['Item_Outlet_Sales'].skew())
print('Kurtsis: %f' %data['Item_Outlet_Sales'].kurt())
```

```
Skewness: 1.544684
Kurtsis: 2.419439
```

In [10]:
```
categorial_features = data.select_dtypes(include=[np.object])
categorial_features.head(2)
```

Out[10]:

	Item_Identifier	Item_Fat_Content	Item_Type	Outlet_Identifier	Outlet_Size	Outlet_Location_Type	Outlet_Type	source
0	FDA15	Low Fat	Dairy	OUT049	Medium	Tier 1	Supermarket Type1	train
1	DRC01	Regular	Soft Drinks	OUT018	Medium	Tier 3	Supermarket Type2	train

In [11]:
```
numerical_features = data.select_dtypes(include=[np.number])
numerical_features.head(2)
```

Out[11]:

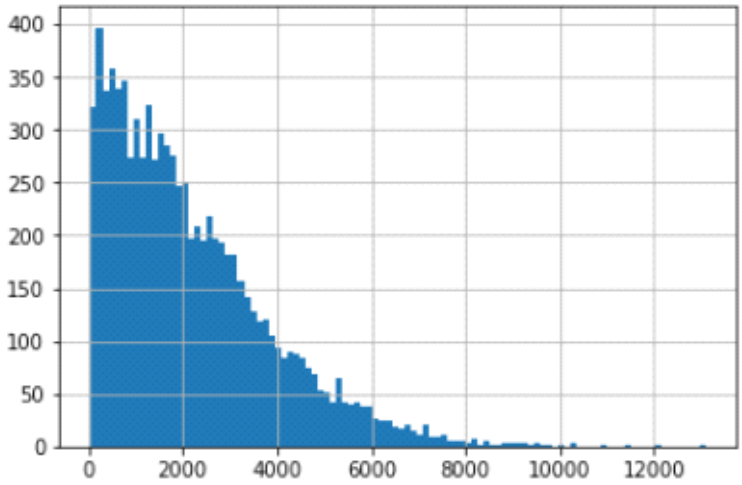
	Item_Weight	Item_Visibility	Item_MRP	Outlet_Establishment_Year	Item_Outlet_Sales
0	9.30	0.016047	249.8092	1999	3735.1380
1	5.92	0.019278	48.2692	2009	443.4228

In [12]:
```
data['Outlet_Establishment_Year'].value_counts()
```

Out[12]:
```
1985    2439
1987    1553
1999    1550
1997    1550
2004    1550
2002    1548
2009    1546
2007    1543
1998     925
Name: Outlet_Establishment_Year, dtype: int64
```

Let's visualize the continuous variables (i.e.,target variable-Item_Outlet_Sales) using histograms and categorical variables using bar plots.

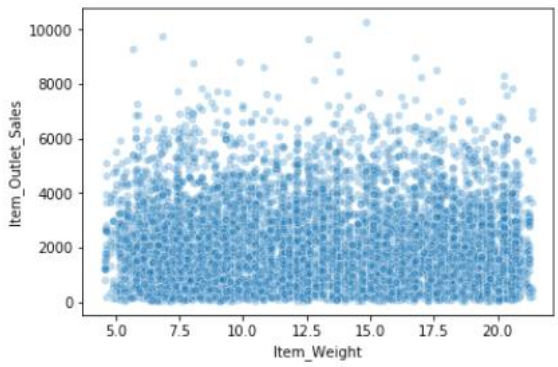
In [13]:
```
train['Item_Outlet_Sales'].hist(bins = 100);
```



Target Variable vs Independent Numerical Variables

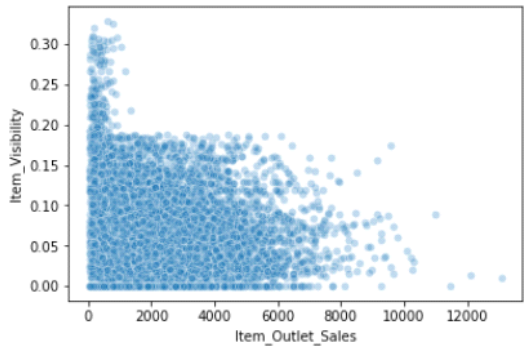
We can use of scatter plots for the continuous or numeric variables and violin plots for the categorical variables.

In [25]:
```
sns.scatterplot(x = 'Item_Weight',y = 'Item_Outlet_Sales',data = train,alpha = 0.3);
```



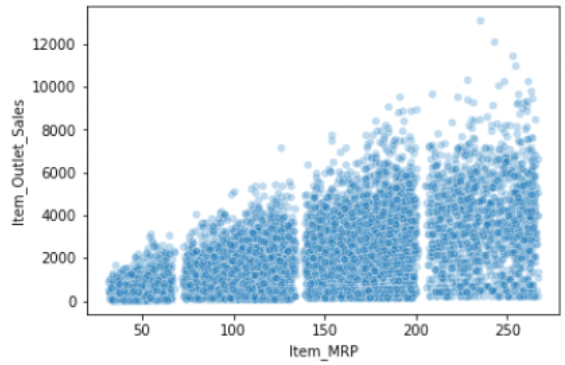
In [26]:
```
sns.scatterplot(x = 'Item_Outlet_Sales',y = 'Item_Visibility',data = train,alpha = 0.3)
```

Out[26]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2e26a53a90>
```



In [27]:
```
sns.scatterplot(x = 'Item_MRP',y = 'Item_Outlet_Sales',data = train,alpha = 0.3)
```

Out[27]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2e26aac588>
```

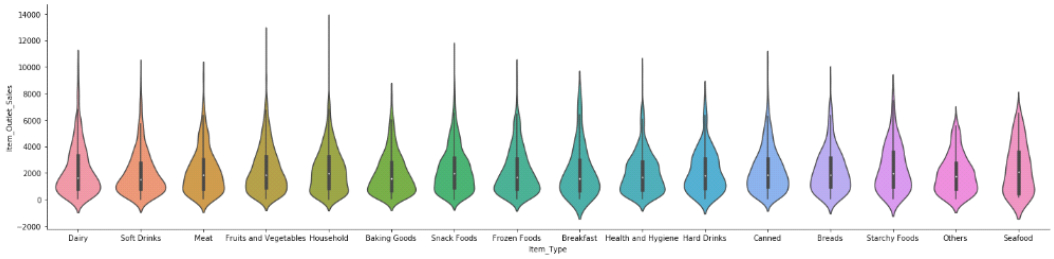


Target Variable vs Independent Categorical Variables

Here we'll use the violin plots as they show the full distribution of the data. The width of a violin plot at a particular level indicates the concentration or density of data at that level. The height of a violin tells us about the range of the target variable values.

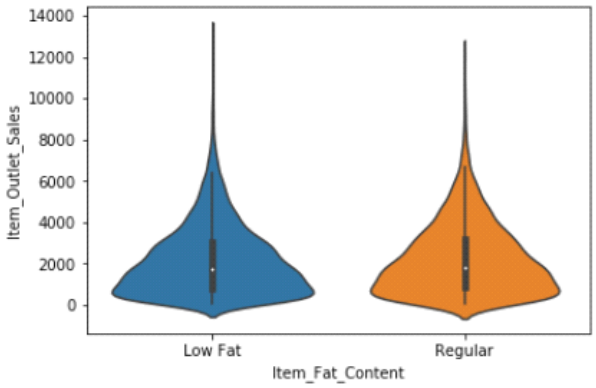
In [28]:
```
sns.catplot(x = 'Item_Type',y = 'Item_Outlet_Sales',kind = 'violin',data = train,aspect=4)
```

Out[28]:
```
<seaborn.axisgrid.FacetGrid at 0x7f2e264dbc50>
```



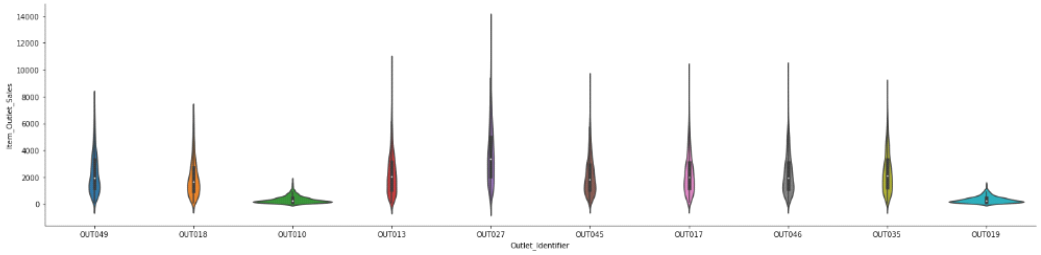
In [30]:
```
sns.violinplot(x = 'Item_Fat_Content',y = 'Item_Outlet_Sales',data = train)
```

Out[30]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2e263a37b8>
```



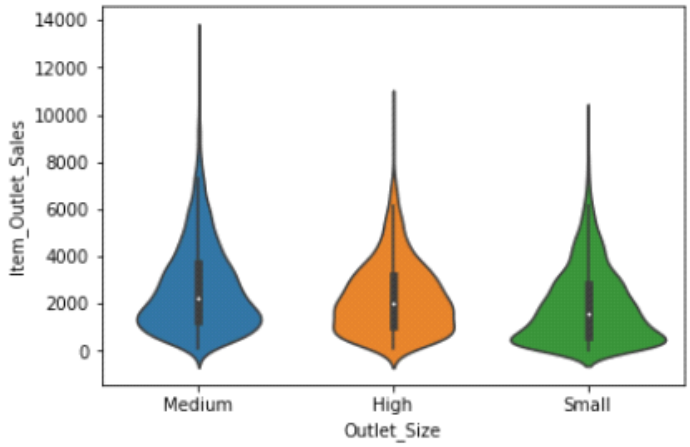
In [31]:
```
sns.catplot('Outlet_Identifier','Item_Outlet_Sales',kind = 'violin',data = train,aspect = 4)
```

Out[31]:
```
<seaborn.axisgrid.FacetGrid at 0x7f2e263fe080>
```



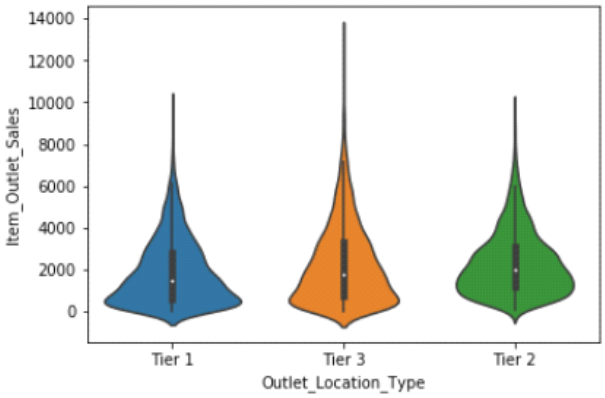
In [32]:
```
sns.violinplot('Outlet_Size','Item_Outlet_Sales',data = train)
```

Out[32]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2e262b5588>
```



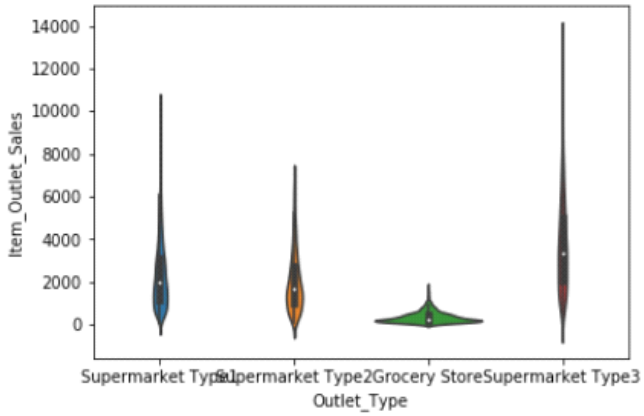
In [33]:
```
sns.violinplot('Outlet_Location_Type','Item_Outlet_Sales',data = train)
```

Out[33]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2e2480bc18>
```



In [34]:
```
sns.violinplot('Outlet_Type','Item_Outlet_Sales',data = train)
```

Out[34]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2e2482ae10>
```



These are the kind of insights that we can extract by visualizing our data. Hence, data visualization should be an important part of any kind data analysis.

CONCLUSION

Data analysis is an important process of research or simply discovering information related to any work. Data derived from the observation, experiment, and other primary and secondary data collection methods is large and cannot be taken as it is. Not all data is relevant, neither can it directly signify any trends, relations, facts, and associations within the data. To find out those required trends and relations, the data needs to be reconstructed in the relevant form and modified. This process is called data analysis. Data analysis and conclusion take forward the research.

The data need is to be clearly defined before the collection of data itself and the process is as follows:

Data Collection: It is gathering information based on research objectives and variables identified previously. The data gathered should be accurate and related to the research question. Data is collected from various sources using secondary collection techniques from organizational databases, previous surveys, and documents. Data is primarily collected through personal interactions and surveys. Then data is arranged and cleaned, removing insignificant information.

Data Processing: After arranging, data needs to be organized in tabular form with suitableanalysis tools. Data needs to be arranged in

spreadsheets and other statistical tools, thendata modelinghas to be created.

Data Analysis: Data needs to be cleaned of errors before analysis. Statistical tools provideanalysis like regression, correlation, averages, and others. After tools are applied, dataneeds to be understood andits findings are interpreted according to the research question.

Communicate results: Visualized data needs to be written in clearly and results should beshown in a classified and organized way. The data findings with diagrams and graphs maketheprocess of data interpretation and presentation complete.

Finally, the conclusion is the essential step in completing the data analysis process. Data analysis gives out certain results, but in big research studies, it is difficult to understand the essence or crux of the findings, relevant to the topicunder study. Theconclusion gives important inferences derived from the study and bind them together as a final summary of findings.

REFERENCE:

- https://www.kaggle.com/code/startupsci/titanic-data-science-solutions
- https://www.kaggle.com/code/pranavuikey/bigmart-eda-with-complete-explanation