Univariate Analysis

Why do we need Exploratory Data Analysis (EDA)?

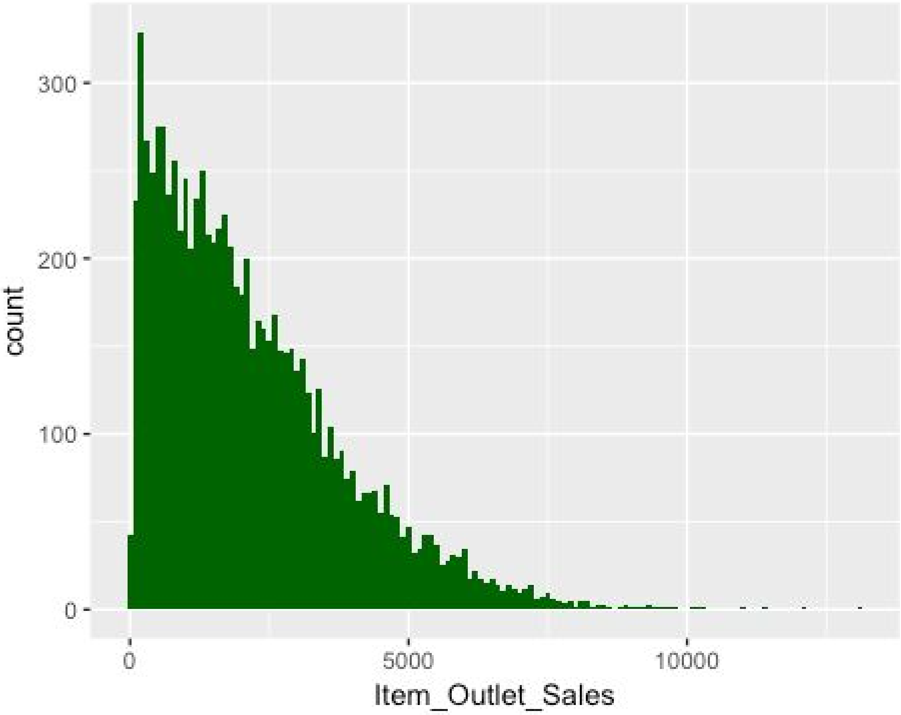
After understanding the dimensions and properties of data, we have to deep dive and explore the data visually. It helps us in understanding the nature of data in terms of distribution of the individual variables/features, finding missing values, relationship with other variables and many other things.

Let’s start with univariate EDA. It involves exploring variables individually. We will try to visualize the continuous variables using histograms and categorical variables using bar plots.

Target Variable - Item_Outlet_Sales

Since our target variable is continuous, we can visualise it by plotting its histogram.

```
ggplot(train) + geom_histogram(aes(train$Item_Outlet_Sales), binwidth = 100, fill = "darkgreen") +
xlab("Item_Outlet_Sales")
```



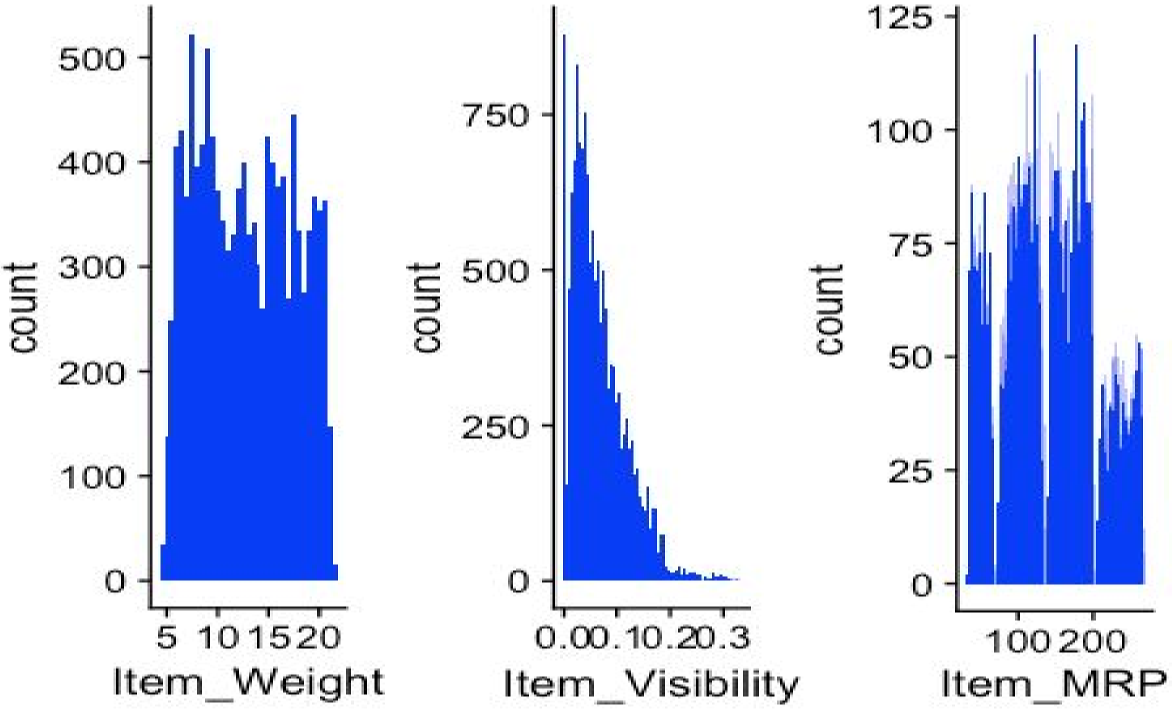
It is a right skewed variable and would need some data transformation to treat its skewness.

Independent Variables (Numeric variables)

Now let’s check the numeric independent variables. We’ll again use the histograms for visualizations because that will help us in visualizing the distribution of the variables.

```
p1 = ggplot(combi) + geom_histogram(aes(Item_Weight), binwidth = 0.5, fill = "blue")
p2 = ggplot(combi) + geom_histogram(aes(Item_Visibility), binwidth = 0.005, fill = "blue")
p3 = ggplot(combi) + geom_histogram(aes(Item_MRP), binwidth = 1, fill = "blue")
plot_grid(p1, p2, p3, nrow = 1) # plot_grid() from cowplot package
```

```
Warning : Removed 2439 rows containing non-finite values (stat_bin).
```

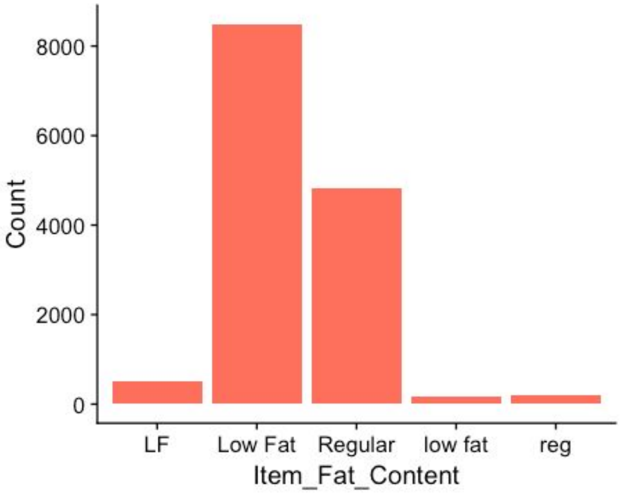


Observations

- There seems to be no clear-cut pattern in Item_Weight.
- Item_Visibility is right-skewed and should be transformed to curb its skewness.
- We can clearly see 4 different distributions for Item_MRP. It is an interesting insight.

Independent Variables (categorical variables)

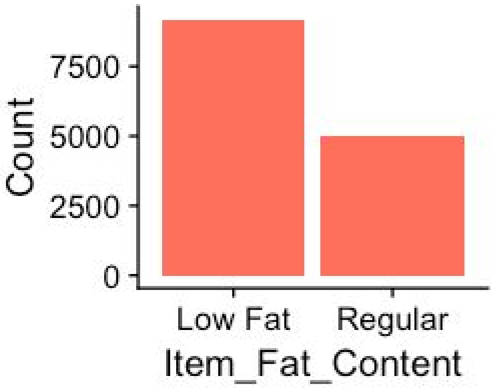
Now we'll try to explore and gain some insights from the categorical variables. A categorical variable or feature can have only a finite set of values.



Let's first plot **Item_Fat_Content**.

```
ggplot(combi %>% group_by(Item_Fat_Content) %>%
summarise(Count = n())) +
  geom_bar(aes(Item_Fat_Content, Count), stat =
"identity", fill = "coral1")
```

In the figure above, 'LF', 'low fat', and 'Low Fat' are the same category and can be combined into one. Similarly we can be done for 'reg' and 'Regular' into one. After making these corrections we'll plot the same figure again.



```
combi$Item_Fat_Content[combi$Item_Fat_Content == "LF"] = "Low
Fat"
combi$Item_Fat_Content[combi$Item_Fat_Content == "low fat"] =
"Low Fat"
combi$Item_Fat_Content[combi$Item_Fat_Content == "reg"] =
"Regular"
ggplot(combi %>% group_by(Item_Fat_Content) %>%
summarise(Count = n())) + geom_bar(aes(Item_Fat_Content,
Count), stat = "identity", fill = "coral1")
```

Now let's check the other categorical variables.

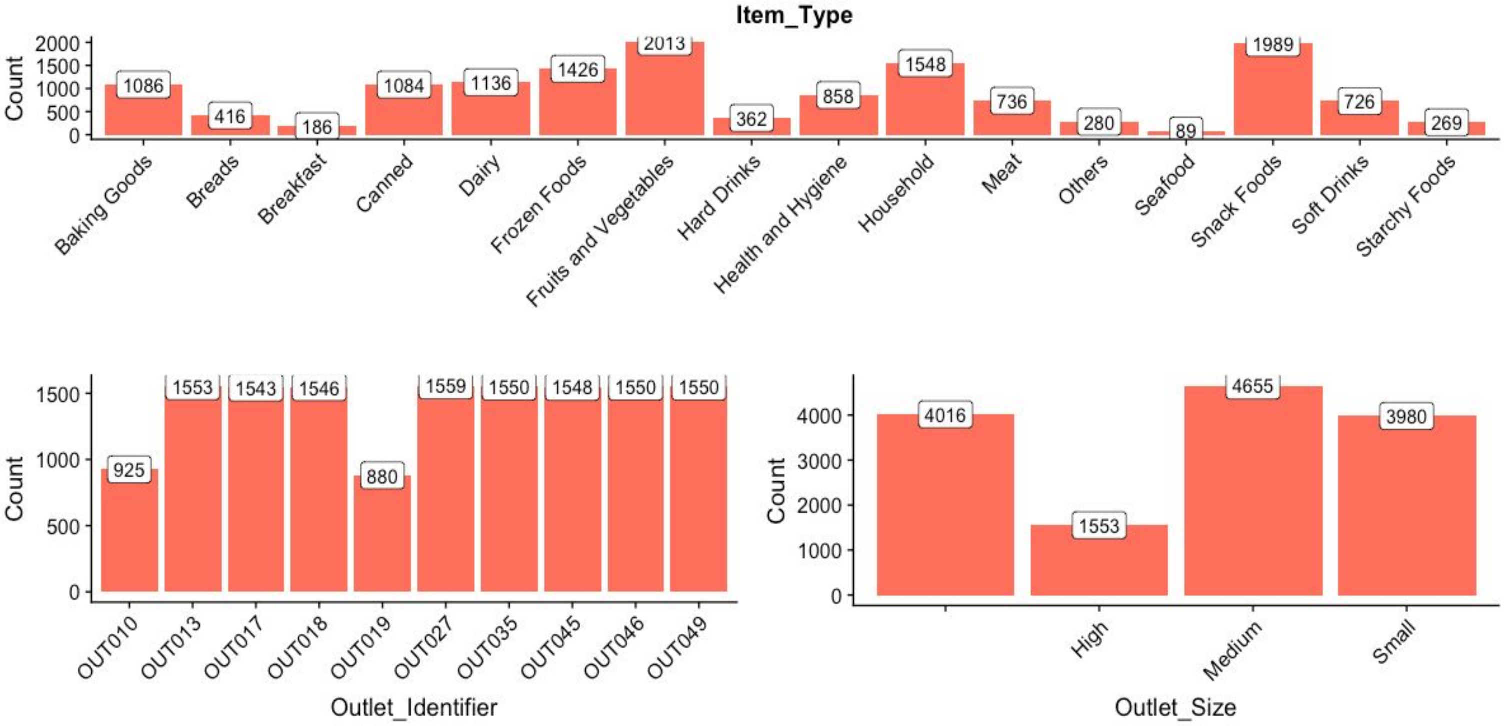
# plot for **Item_Type**

```
p4 = ggplot(combi %>% group_by(Item_Type) %>% summarise(Count = n())) +
  geom_bar(aes(Item_Type, Count), stat = "identity", fill = "coral1") +
  xlab("") +
  geom_label(aes(Item_Type, Count, label = Count), vjust = 0.5) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  ggtitle("Item_Type")
```

```
# plot for Outlet_Identifier
p5 = ggplot(combi %>% group_by(Outlet_Identifier) %>% summarise(Count = n())) +
  geom_bar(aes(Outlet_Identifier, Count), stat = "identity", fill = "coral1") +
  geom_label(aes(Outlet_Identifier, Count, label = Count), vjust = 0.5) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# plot for Outlet_Size
p6 = ggplot(combi %>% group_by(Outlet_Size) %>% summarise(Count = n())) +
  geom_bar(aes(Outlet_Size, Count), stat = "identity", fill = "coral1") +
  geom_label(aes(Outlet_Size, Count, label = Count), vjust = 0.5) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

second_row = plot_grid(p5, p6, nrow = 1)
# plotting both plots together  plot_grid(p4, second_row, ncol = 1)
```



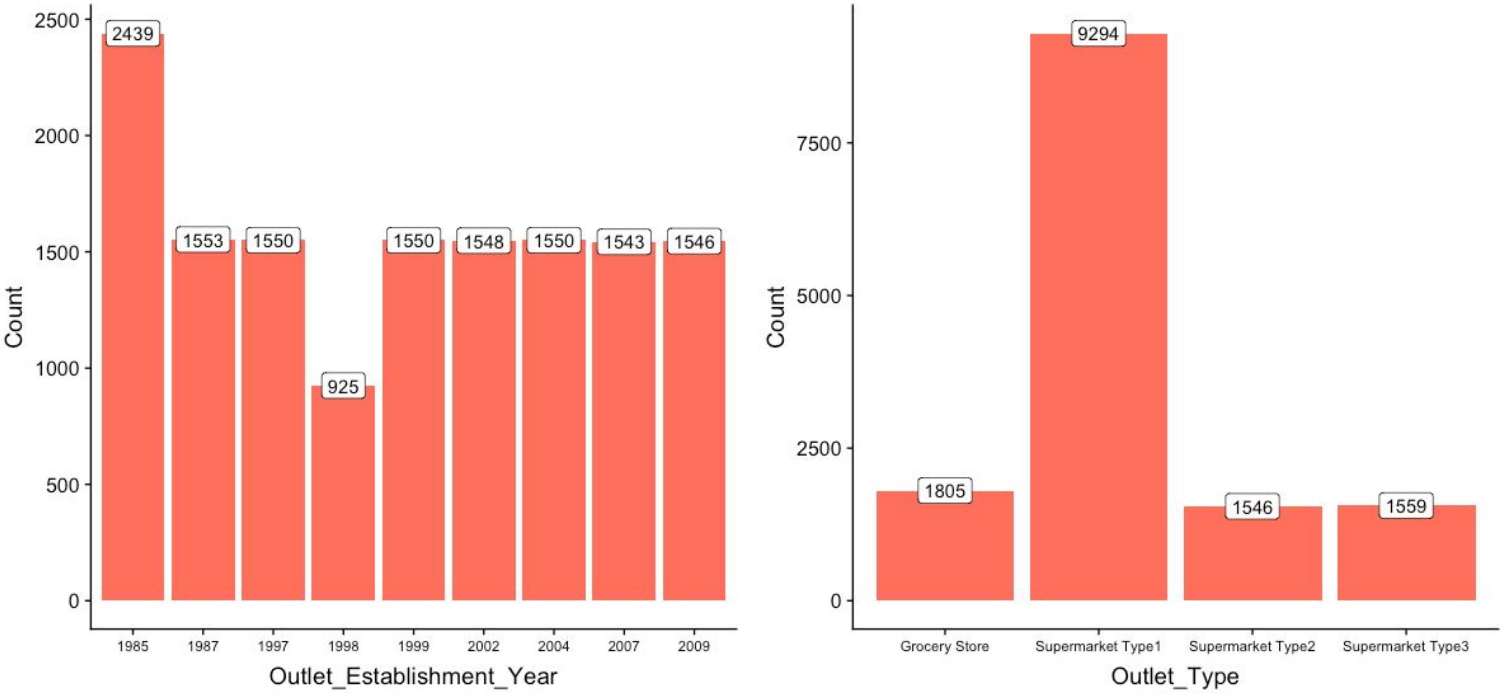
In Outlet_Size's plot, for 4016 observations, Outlet_Size is blank or missing.We will check for this in the bivariate analysis to substitute the missing values in the Outlet_Size.

We'll also check the remaining categorical variables.

```
# plot for Outlet_Establishment_Year
p7 = ggplot(combi %>% group_by(Outlet_Establishment_Year) %>% summarise(Count = n())) +
  geom_bar(aes(factor(Outlet_Establishment_Year), Count), stat = "identity", fill = "coral1") +
  geom_label(aes(factor(Outlet_Establishment_Year), Count, label = Count), vjust = 0.5) +
  xlab("Outlet_Establishment_Year") +
  theme(axis.text.x = element_text(size = 8.5))
```

```
# plot for Outlet_Type
p8 = ggplot(combi %>% group_by(Outlet_Type) %>% summarise(Count = n())) +
  geom_bar(aes(Outlet_Type, Count), stat = "identity", fill = "coral1") +
  geom_label(aes(factor(Outlet_Type), Count, label = Count), vjust = 0.5) +
  theme(axis.text.x = element_text(size = 8.5))

# plotting both plots together
plot_grid(p7, p8, ncol = 2)
```



Observations

- Lesser number of observations in the data for the outlets established in the year 1998 as compared to the other years.
- Supermarket Type 1 seems to be the most popular category of Outlet_Type.

After looking at every feature individually, let’s now do some bivariate analysis. Here we’ll explore the independent variables with respect to the target variable. The objective is to discover hidden relationships between the independent variable and the target variable and use those findings in missing data imputation and feature engineering in the next module.

We will make use of **scatter plots** for the continuous or numeric variables and **violin plots** for the categorical variables.

```
train = combi[1:nrow(train)] # extracting train data from the combined data
```

Target Variable vs Independent Numerical Variables

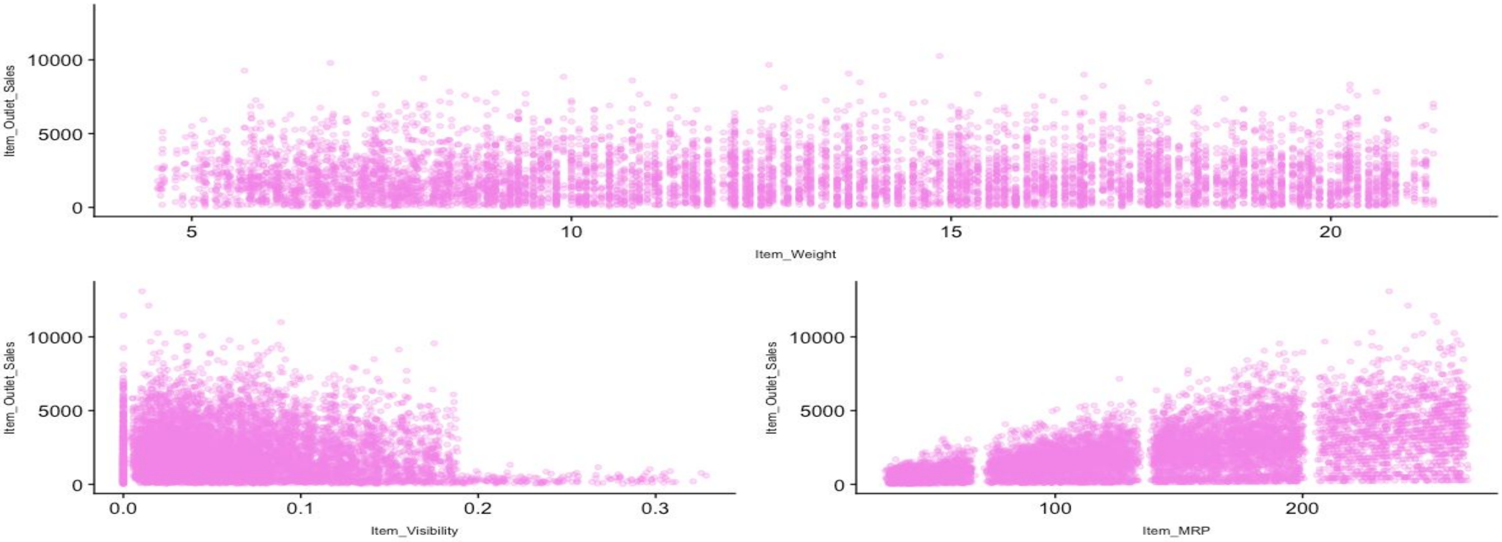
Let’s explore the numerical variables first.

```
# Item_Weight vs Item_Outlet_Sales
p9 = ggplot(train) +  geom_point(aes(Item_Weight, Item_Outlet_Sales), colour = "violet", alpha = 0.3) +
    theme(axis.title = element_text(size = 8.5))

# Item_Visibility vs Item_Outlet_Sales
p10 = ggplot(train) + geom_point(aes(Item_Visibility, Item_Outlet_Sales), colour = "violet", alpha = 0.3) +
     theme(axis.title = element_text(size = 8.5))

# Item_MRP vs Item_Outlet_Sales
p11 = ggplot(train) + geom_point(aes(Item_MRP, Item_Outlet_Sales), colour = "violet", alpha = 0.3) +
     theme(axis.title = element_text(size = 8.5))

second_row_2 = plot_grid(p10, p11, ncol = 2)
plot_grid(p9, second_row_2, nrow = 2)
```



```
Removed 1463 rows containing missing values (geom_point).
```

Observations

- Item_Outlet_Sales is spread well across the entire range of the Item_Weight without any obvious pattern.
- In Item_Visibility vs Item_Outlet_Sales, there is a string of points at Item_Visibility = 0.0 which seems strange as item visibility cannot be completely zero. We will take note of this issue and deal with it in the later stages.
- In the third plot of Item_MRP vs Item_Outlet_Sales, we can clearly see 4 segments of prices that can be used in feature engineering to create a new variable.