

How To Perform Exploratory Data Analysis -A Guide for Beginners

[BEGINNER](#)[DATA EXPLORATION](#)[DATA SCIENCE](#)[PYTHON](#)[STATISTICS](#)[STRUCTURED DATA](#)

This article was published as a part of the [Data Science Blogathon](#)

Introduction

Exploratory Data Analysis is a set of techniques that were developed by Tukey, John Wilder in 1970. The philosophy behind this approach was to examine the data before building a model. John Tukey encouraged statisticians to explore the data, and possibly formulate hypotheses that could lead to new data collection and experiments. Today Data scientists and analysts spend most of their time in Data Wrangling and Exploratory Data Analysis also known as EDA. But what is this EDA and why it is so important? This article explains what is EDA and how to apply EDA techniques to a dataset.

Table of Contents

1. What is Exploratory Data Analysis?
2. Why EDA is important?
3. How to perform EDA?
4. Endnote

What is Exploratory Data Analysis?

Exploratory Data Analysis or EDA is used to take insights from the data. Data Scientists and Analysts try to find different patterns, relations, and anomalies in the data using some statistical graphs and other visualization techniques. Following things are part of EDA :

1. Get maximum insights from a data set
2. Uncover underlying structure
3. Extract important variables from the dataset
4. Detect outliers and anomalies(if any)
5. Test underlying assumptions
6. Determine the optimal factor settings

Why EDA is important?

The main purpose of EDA is to detect any errors, outliers as well as to understand different patterns in the data. It allows Analysts to understand the data better before making any assumptions. The outcomes of EDA helps businesses to know their customers, expand their business and take decisions accordingly.

How to perform EDA?

To understand EDA better let us take an example. We will be using [Automobile Dataset](#) for analysis.

1. Import libraries and load dataset

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

```
auto=pd.read_csv('Automobile dataset.data')
```

```
auto.head()
```

3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.60	...	130	mpfi	3.47	2.68	9.00	111	5000	21	27	13495		
0	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47	2.68	9.0	111	5000	21	27	16500	
1	1	?	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	...	152	mpfi	2.68	3.47	9.0	154	5000	19	26	16500	
2	2	164		audi	gas	std	four	sedan	fwd	front	99.8	...	109	mpfi	3.19	3.40	10.0	102	5500	24	30	13950
3	2	164		audi	gas	std	four	sedan	4wd	front	99.4	...	136	mpfi	3.19	3.40	8.0	115	5500	18	22	17450
4	2	?		audi	gas	std	two	sedan	fwd	front	99.8	...	136	mpfi	3.19	3.40	8.5	110	5500	19	25	15250

5 rows x 26 columns

Image Source: Jupyter Notebook Screenshot

We can see that the dataset has 26 attributes and column names are missing. We can also observe that there are '?' at some places which means our data has missing value also. We will fill in column names first.

```
cols=[ 'symboling', 'normalized_losses', 'make', 'fuel_type', 'aspiration', 'num_of_doors', 'body_style', 'drive_wheels_engine', '1
```

```
auto.columns=cols
```

```
auto.head()
```

	symboling	normalized_losses	make	fuel_type	aspiration	num_of_doors	body_style	drive_wheels_engine	location	wheel_base	...	engine
0	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	...	
1	1	?	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	...	
2	2	164	audi	gas	std	four	sedan	fwd	front	99.8	...	
3	2	164	audi	gas	std	four	sedan	4wd	front	99.4	...	
4	2	?	audi	gas	std	two	sedan	fwd	front	99.8	...	

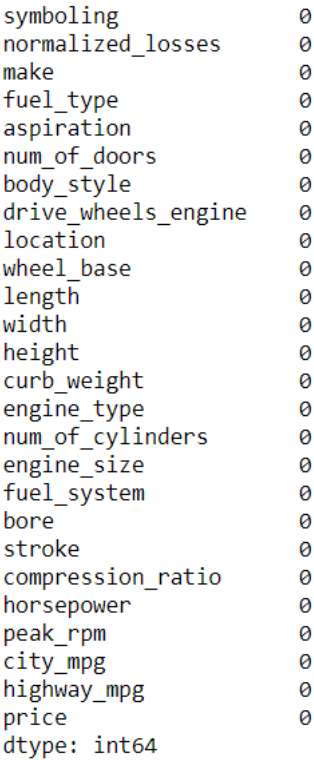
5 rows x 26 columns

Image Source: Jupyter Notebook Screenshot

We got our column names. The price column is our target variable.

2.Check for missing values

```
auto.isnull().sum()
```



symboling	0
normalized_losses	0
make	0
fuel_type	0
aspiration	0
num_of_doors	0
body_style	0
drive_wheels_engine	0
location	0
wheel_base	0
length	0
width	0
height	0
curb_weight	0
engine_type	0
num_of_cylinders	0
engine_size	0
fuel_system	0
bore	0
stroke	0
compression_ratio	0
horsepower	0
peak_rpm	0
city_mpg	0
highway_mpg	0
price	0
dtype: int64	

Image Source: Jupyter Notebook Screenshot

It is showing that we don't have any null values in our dataset but we have observed earlier that there were '?' symbols in the dataset, which means that these symbols are in the form of an object. Let us now check the data types of each attribute.

```
auto.info()
```

Image Source: Jupyter Notebook Screenshot

We can observe that those columns that have symbols are in object form as well as some columns should be of an integer type but are of an object type. Now let us detect which columns have symbols and if there are any other symbols too.

```
#Checking for wrong entries like symbols -,?,#,*,etc. for col in auto.columns: print('{} :  
{}`.format(col,auto[col].unique()))
```

Image Source: Jupyter Notebook Screenshot

There are null values in our dataset in form of '?' only but pandas are not reading them so we will replace them into *np.nan* form.

```
for col in auto.columns: auto[col].replace({'?':np.nan},inplace=True)
```

```
auto.head()
```

Image Source: Jupyter Notebook Screenshot

Now we can observe that the ‘?’ symbols have been converted into *NaN* form. Let us check for missing values again.

```
auto.isnull().sum()
```

Image Source: Jupyter Notebook Screenshot

We can observe that now there are missing values in some columns.

3. Visualizing the missing values

With the help of heatmap, we can see the amount of data that is missing from the attribute. With this, we can make decisions whether to drop these missing values or to replace them. Usually dropping the missing

values is not advisable but sometimes it may be helpful too.

```
sns.heatmap(auto.isnull(),cbar=False,cmap='viridis')
```

Image Source: Jupyter Notebook Screenshot

Now observe that there are many missing values in *normalized_losses* while other columns have fewer missing values. We can't drop the *normalized_losses* column as it may be important for our prediction.

4.Replacing the missing values

We will be replacing these missing values with mean because the number of missing values is less(we can use median too).

```
num_col = ['normalized_losses', 'bore', 'stroke', 'horsepower', 'peak_rpm','price'] for col in num_col:
auto[col]=pd.to_numeric(auto[col]) auto[col].fillna(auto[col].mean(), inplace=True) auto.head()
```

Image Source: Jupyter Notebook Screenshot

We can observe that now our missing values are replaced with mean.

5. Asking Analytical Questions and Visualizations

This is the most important step in EDA. This step will decide how much can you think as an Analyst. This step varies from person to person in terms of their questioning ability. Try to ask questions related to independent variables and the target variable. For example – how fuel_type will affect the price of the car?

Before this let us check the correlation between different variables, this will give us a roadmap on how to proceed further.

```
plt.figure(figsize=(10,10)) sns.heatmap(auto.corr(),cbar=True,annot=True,cmap='Blues')
```

Image Source: Jupyter Notebook Screenshot

Positive Correlation

- *Price – wheel_base, length, width, curb_weight, engine_size, bore, horsepower*
- *wheelbase – length, width, height, curb_weight, engine_size, price*
- *horsepower – length, width, curb_weight, engine_size, bore, price*
- *Highway mpg – city mpg*

Negative Correlation

- *Price – highway_mpg, city_mpg*
- *highway_mpg – wheel base, length, width, curb_weight, engine_size, bore, horsepower, price*
- *city – wheel base, length, width, curb_weight, engine_size, bore, horsepower, price*

This heatmap has given us great insights into the data.

Now let us apply domain knowledge and ask the questions which will affect the price of the automobile.

1. How does the horsepower affect the price?

```
plt.figure(figsize=(10,10))      plt.scatter(x='horsepower',y='price',data=auto)      plt.xlabel('Horsepower')  
plt.ylabel('Price')
```

Image Source: Jupyter Notebook Screenshot

We can see that most of the horsepower value lies between 50-150 has price mostly between 5000-25000, there are outliers also(between 200-300).

Let's see a count between 50-100 i.e univariate analysis of horsepower.

```
sns.histplot(auto.horsepower,bins=10)
```


The average count between 50-100 is 50 and it is positively skewed.

2. What is the relation between engine_size and price?

```
plt.figure(figsize=(10,10))    plt.scatter(x='engine_size',y='price',data=auto)    plt.xlabel('Engine    size')  
plt.ylabel('Price')
```

We can observe that the pattern is similar to horsepower vs price.

3. How does highway_mpg affects price?

```
plt.figure(figsize=(10,10))    plt.scatter(x='highway_mpg',y='price',data=auto)    plt.xlabel('Higway    mpg')  
plt.ylabel('Price')
```

Image Source: Jupyter Notebook Screenshot

We can see price decreases with an increase in highway_mpg.

Let us check the number of doors.

```
#Unique values in num_of_doors auto.num_of_doors.value_counts()
```

```
four    114
two      88
Name: num_of_doors, dtype: int64
```

Image Source: Jupyter Notebook Screenshot

4. Relation between no. of doors and price

We will use a boxplot for this analysis.

```
sns.boxplot(x='price',y='num_of_doors',data=auto)
```

Image Source: Jupyter Notebook Screenshot

With this boxplot, we can conclude that the average price of a vehicle with two doors is 10000, and the average price of a vehicle with four doors is 12000.

With this plot, we have gained enough insights from data and our data is ready to build a model.

Endnote

Thank you for reading this article.

I hope this article has helped you to understand Exploratory Data Analysis and how to apply different EDA techniques to a dataset.

Please give feedback for this article in the comment section below.

About Me

I am a final year undergrad student of Btech in Computer Science with a specialization in Artificial Intelligence. I am a self-motivated learner who is eager to help the Data Science community as much as I can.

You can contact me on [LinkedIn](#) or by [Email](#) for any queries or project collaboration.

The media shown in this article are not owned by Analytics Vidhya and are used at the Author's discretion.

Article Url - <https://www.analyticsvidhya.com/blog/2021/08/how-to-perform-exploratory-data-analysis-a-guide-for-beginners/>



[nimit2](#)