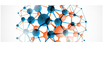
Exploratory Data Analysis Project (Writeup)



for summer internship in Shree Cement

Name - *Anshul Barjatya*

Roll NO. - *18145015*

Branch - *Metallurgical Engineering*

What is Exploratory Data Analysis

> *Exploratory Data Analysis, or EDA, is essentially a type of storytelling for statisticians. It is nothing but a data exploration technique to understand the various aspects of the Data.*

What is The Objective of EDA

> It is basically used to filter the data from redundancies.

The objectives of EDA are to:

- Suggest hypotheses about the causes of observed phenomena
- Assess assumptions on which statistical inference will be based
- Support the selection of appropriate statistical tools and techniques
- Provide a basis for further data collection through surveys or experiments

What are The Steps involved in EDA

> It follows a systematic set of steps to explore the data in most efficient way possible.

Basic Steps in Data Exploration and Preprocessing:

- Identification of variables and data types
- Analyzing the basic metrics
- Non-Graphical Univariate Analysis
- Graphical Univariate Analysis
- Bivariate Analysis
- Variable transformations
- Missing value treatment
- Outlier treatment
- Correlation Analysis
- Dimensionality Reduction

Here We discuss :

So Now I am bringing a dataset from Kaggle and performing EDA on it

> *Source*

In [4]:
```
import numpy as np
import pandas as pd
import seaborn as sns
```

In [4]:
```
data=pd.read_csv("C:/Users/ANSHUL JAIN/Desktop/Python Data Analysis/StudentsPerformance.csv")
```

Step#1. understanding the data

In [22]:
```
# for getting the head part of Dataset for our reference.
data.head()
```

Out[22]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

In [7]:
```
# for describing the stats
data.describe()
```

Out[7]:

	math score	reading score	writing score
count	1000.000000	1000.000000	1000.000000
mean	66.089000	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.000000	17.000000	10.000000
25%	57.000000	59.000000	57.750000
50%	66.000000	70.000000	69.000000
75%	77.000000	79.000000	79.000000
max	100.000000	100.000000	100.000000

In [23]:
```
# no. of rows and columns
data.shape
```

Out[23]:
```
(1000, 8)
```

In [13]:
```
# To get the names of diffrenct columns in dataset
data.columns
```

Out[13]:
```
Index(['gender', 'race/ethnicity', 'parental level of education', 'lunch',
       'test preparation course', 'math score', 'reading score',
       'writing score'],
      dtype='object')
```

In [12]:
```
# To know how many unique data posses by particular column
data.nunique()
```

Out[12]:
```
gender                         2
race/ethnicity                 5
parental level of education    6
lunch                          2
test preparation course        2
math score                    81
reading score                 72
writing score                 77
dtype: int64
```

In [10]:
```
data["gender"].unique()
```

Out[10]:
```
array(['female', 'male'], dtype=object)
```

In [10]:
```
data["race/ethnicity"].unique()
```

Out[10]:
```
array(['group B', 'group C', 'group A', 'group D', 'group E'],
      dtype=object)
```

In [28]:
```
data["parental level of education"].unique()
```

Out[28]:
```
array(["bachelor's degree", 'some college', "master's degree",
       "associate's degree", 'high school', 'some high school'],
      dtype=object)
```

Step#2. Cleaning the data

In [25]:
```
# checking that is there any null value in dataset
data.isnull().sum()
```

Out[25]:
```
gender                       0
race/ethnicity               0
parental level of education  0
lunch                        0
test preparation course      0
math score                   0
reading score                0
writing score                0
dtype: int64
```

In [24]:
```
# Now we would remove the unnecessary columns in our dataset which is called dropping the redundant data
# which does not influence our dataset.
student= data.drop(['race/ethnicity', 'parental level of education'], axis=1)
```

In [26]:
```
student.head()
```

Out[26]:

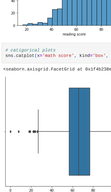
	gender	lunch	test preparation course	math score	reading score	writing score
0	female	standard	none	72	72	74
1	female	standard	completed	69	90	88
2	female	standard	none	90	95	93
3	male	free/reduced	none	47	57	44
4	male	standard	none	76	78	75

Step#3. Relationship Analysis

In [26]:
```
# correlation matrix
# Correlation is a statistic that measures the degree to which two variables move in relation to each other.
corelation=student.corr()
```

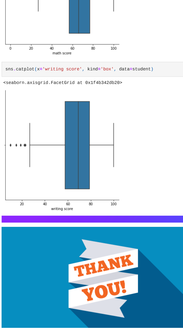
In [27]:
```
sns.heatmap(corelation, xticklabels=corelation.columns, yticklabels=corelation.columns, annot=True )
```

Out[27]:
```
<AxesSubplot:>
```



In [28]:
```
sns.pairplot(student)
```

Out[28]:
```
<seaborn.axisgrid.PairGrid at 0x1f4aea8be60>
```



In [30]:
```
# Scatter Plot
# A scatter plot is a type of plot or mathematical diagram using Cartesian coordinates
# to display values for typically two variables for a set of data.
# Scatter plots are important in statistics because they can show the extent of correlation,
# if any, between the values of observed quantities or phenomena (called variables).
sns.relplot(x='math score', y='reading score', hue='gender', data=student)
sns.relplot(x='math score', y='reading score', hue='lunch', data=student)
```

Out[30]:
```
<seaborn.axisgrid.FacetGrid at 0x1f4afee98b0>
```


In [36]:
```
# Histogram
# A histogram is an approximate representation of the distribution of numerical data.
sns.histplot(student['math score'])
```

Out[36]:
```
<AxesSubplot:xlabel='math score', ylabel='Count'>
```


In [37]:
```
sns.histplot(student['reading score'])
```

Out[37]:
```
<AxesSubplot:xlabel='reading score', ylabel='Count'>
```


In [42]:
```
# catigorical plots
sns.catplot(x='math score', kind='box', data=student)
```

Out[42]:
```
<seaborn.axisgrid.FacetGrid at 0x1f4b28e648>
```


In [39]:
```
sns.catplot(x='writing score', kind='box', data=student)
```

Out[39]:
```
<seaborn.axisgrid.FacetGrid at 0x1f4b342db28>
```