

Assignment-2

SAMSRITHA RAAVI

16351883

Data Set

Data: <https://app.box.com/s/jm6pw202asu4xd3uypwtry2rqk691y1i>

	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine
1	Hyundai Creta 1.6 CRDI S	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC
2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	13 km/kg	1199 CC
3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC
4	Audi A4 New 2.0 TDI M&I	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC
6	Nissan Micra Diesel XV	Jaipur	2013	86999	Diesel	Manual	First	23.08 kmpl	1461 CC
7	Toyota Innova Crysta 2.8	Mumbai	2016	36000	Diesel	Automatic	First	11.36 kmpl	2755 CC
8	Volkswagen Vento Diesel	Pune	2013	64430	Diesel	Manual	First	20.54 kmpl	1598 CC
9	Tata Indica Vista QuadraJet	Chennai	2012	65932	Diesel	Manual	Second	22.3 kmpl	1248 CC
10	Maruti Ciaz Zeta	Kochi	2018	25692	Petrol	Manual	First	21.56 kmpl	1462 CC
11	Honda City 1.5 V AT Sunr	Kolkata	2012	60000	Petrol	Automatic	First	16.8 kmpl	1497 CC
12	Maruti Swift VDI BSIV	Jaipur	2015	64424	Diesel	Manual	First	25.2 kmpl	1248 CC
13	Land Rover Range Rover	Delhi	2014	72000	Diesel	Automatic	First	12.7 kmpl	2179 CC
14	Land Rover Freelander 2 T	Pune	2012	85000	Diesel	Automatic	Second	0.0 kmpl	2179 CC
15	Mitsubishi Pajero Sport 4	Delhi	2014	110000	Diesel	Manual	First	13.5 kmpl	2477 CC
16	Honda Amaze S i-Dtech	Kochi	2016	58950	Diesel	Manual	First	25.8 kmpl	1498 CC
17	Maruti Swift DDiS VDI	Jaipur	2017	25000	Diesel	Manual	First	28.4 kmpl	1248 CC
18	Renault Duster BSFS Dies	Kochi	2014	77469	Diesel	Manual	First	20.45 kmpl	1461 CC
19	Mercedes-Benz New C-Class	Bangalore	2014	78500	Diesel	Automatic	First	14.84 kmpl	2143 CC
20	BMW 3 Series 320d	Kochi	2014	32988	Diesel	Automatic	First	22.49 kmpl	1995 CC
21	Maruti S-Cross DDiS 200	Bangalore	2015	55392	Diesel	Manual	Second	23.65 kmpl	1248 CC

Data Clean:

- a) Look for the missing values in all the columns and either impute them (replace with mean, median, or mode) or drop them. Justify your action for this task.

Jupyter interface showing the initial data loading process. The code cell contains:

```
In [9]: import pandas as pd
In [10]: s=pd.read_csv(r"C:\Users\admin\Downloads\train.csv")
In [11]: s
```

The output displays the first few rows of the dataset:

Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price	Price
0	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.0	NaN	12.50
1	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	13 km/kg	1199 CC	88.7 bhp	5.0	8.61 Lakh	4.50
2	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7.0	NaN	6.00
3	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5.0	NaN	17.74
4	Nissan Micra Diesel XV	Japur	2013	96999	Diesel	Manual	First	23.08 kmpl	1461 CC	63.1 bhp	5.0	NaN	3.50
...
8642	Maruti Swift VDI	Delhi	2014	27365	Diesel	Manual	First	28.4 kmpl	1249 CC	74 bhp	5.0	7.88 Lakh	4.75
8643	Hyundai Xcent 1.1 CRDi S	Japur	2015	100000	Diesel	Manual	First	24.4 kmpl	1120 CC	71 bhp	5.0	NaN	4.00
...

Read The Dataset

Jupyter interface showing the data inspection and cleaning process. The code cell contains:

```
In [13]: #Null values
n_val = s.isnull().sum()
print("Null Values:\n", n_val)

Null Values:
Unnamed: 0      0
Name            0
Location        0
Year            0
Kilometers_Driven 0
Fuel_Type       0
Transmission    0
Owner_Type      0
Mileage         2
Engine          36
Power           36
Seats           38
New_Price       5032
Price           0
dtype: int64

In [14]: #removing
s['Mileage'] = s['Mileage'].str.extract('(d+\.d+)')
s['Mileage'] = s['Mileage'].astype(float)
s['Mileage'].fillna(s['Mileage'].median(), inplace=True)
s
```

B) Remove the units from some of the attributes and only keep the numerical values (for example remove kmpl from “Mileage”, CC from “Engine”, bhp from “Power”, and lakh from “New_price”).

```
In [14]: # Question 8
s['Mileage'] = s['Mileage'].str.extract('(\\d+\\.\\d+)')
s['Mileage'] = s['Mileage'].astype(float)
s['Mileage'].fillna(s['Mileage'].median(), inplace=True)
s
```

Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price	Price
0	Hyundai Creta 1.8 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67	1582 CC	126.2 bhp	5.0	NaN	12.50
1	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.19	1199 CC	88.7 bhp	5.0	8.61 Lakh	4.50
2	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77	1248 CC	88.76 bhp	7.0	NaN	6.00
3	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.20	1968 CC	140.8 bhp	5.0	NaN	17.74
4	Nissan Micra Diesel XV	Jaipur	2013	96999	Diesel	Manual	First	23.08	1461 CC	63.1 bhp	5.0	NaN	3.50

```
In [15]: s['Engine'] = s['Engine'].str.extract('(\\d+)')
s['Engine'] = s['Engine'].astype(float)
s['Power'] = s['Power'].str.extract('(\\d+\\.\\d+)')
s['Power'] = s['Power'].astype(float)
```

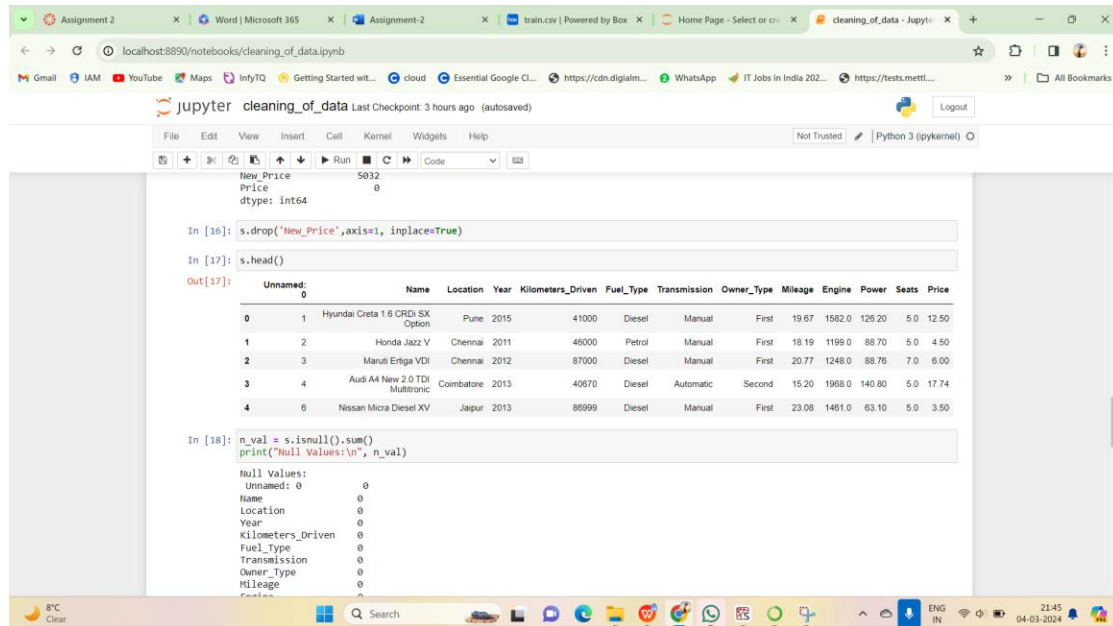
```
10 Power          5811 non-null object
11 Seats          5809 non-null float64
12 New_Price      815 non-null object
13 Price          5847 non-null float64
dtypes: float64(2), int64(3), object(9)
memory usage: 639.6+ KB
None

In [13]: #Null values
n_val = s.isnull().sum()
print("Null Values:\n", n_val)

Null Values:
Unnamed: 0      0
Name            0
Location        0
Year            0
Kilometers_Driven 0
Fuel_Type       0
Transmission     0
Owner_Type       0
Mileage          2
Engine          36
Power           36
Seats           38
New_Price       5832
Price           0
dtype: int64

In [14]: #removing
s['Mileage'] = s['Mileage'].str.extract('(\\d+\\.\\d+)')
s['Mileage'] = s['Mileage'].astype(float)
s['Mileage'].fillna(s['Mileage'].median(), inplace=True)
s
```

After Removing the null values



```
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (pykernel)
```

```
New_Price      5032  
Price           0  
dtype: int64
```

```
In [16]: s.drop('New_Price',axis=1, inplace=True)
```

```
In [17]: s.head()
```

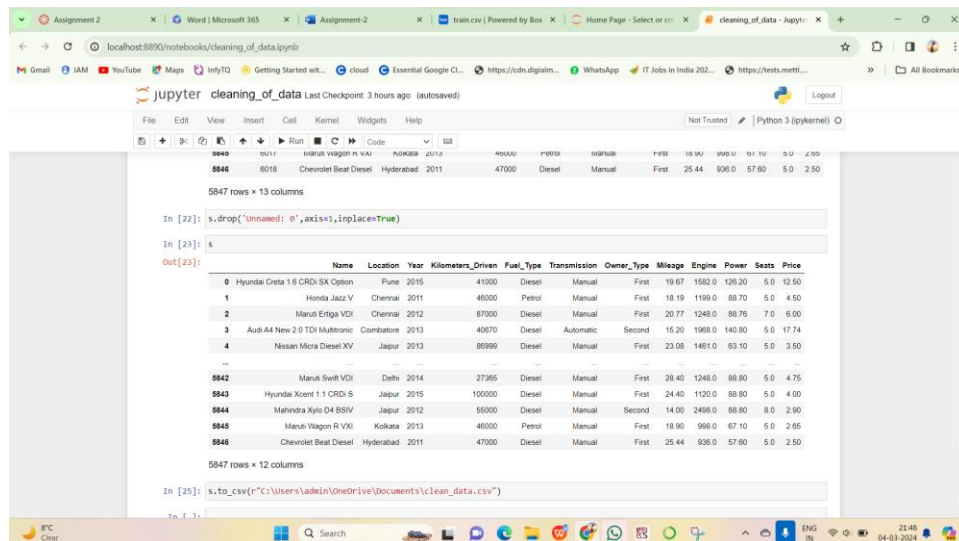
```
Out[17]:
```

Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price	
0	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67	1582.0	126.20	5.0	12.50
1	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.19	1199.0	88.70	5.0	4.50
2	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77	1248.0	88.76	7.0	6.00
3	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.20	1968.0	140.80	5.0	17.74
4	6	Nissan Micra Diesel XV	Jaipur	2013	86999	Diesel	Manual	First	23.08	1461.0	63.10	5.0	3.50

```
In [18]: n_val = s.isnull().sum()  
print("Null Values:\n", n_val)
```

```
Null Values:  
Unnamed: 0      0  
Name            0  
Location        0  
Year            0  
Kilometers_Driven 0  
Fuel_Type       0  
Transmission     0  
Owner_Type      0  
Mileage         0  
Engine          0  
Power           0  
Seats          0  
Price          0
```

Preprocessed data to clean data:



```
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (pykernel)
```

```
5040 0011 Maruti vagon R VDI Noida 2012 40000 Petrol Manual First 19.10 998.0 57.10 5.0 2.90  
5046 6018 Chevrolet Beat Diesel Hyderabad 2011 47000 Diesel Manual First 25.44 936.0 57.60 5.0 2.50
```

```
5847 rows x 13 columns
```

```
In [22]: s.drop('Unnamed: 0',axis=1,inplace=True)
```

```
In [23]: s
```

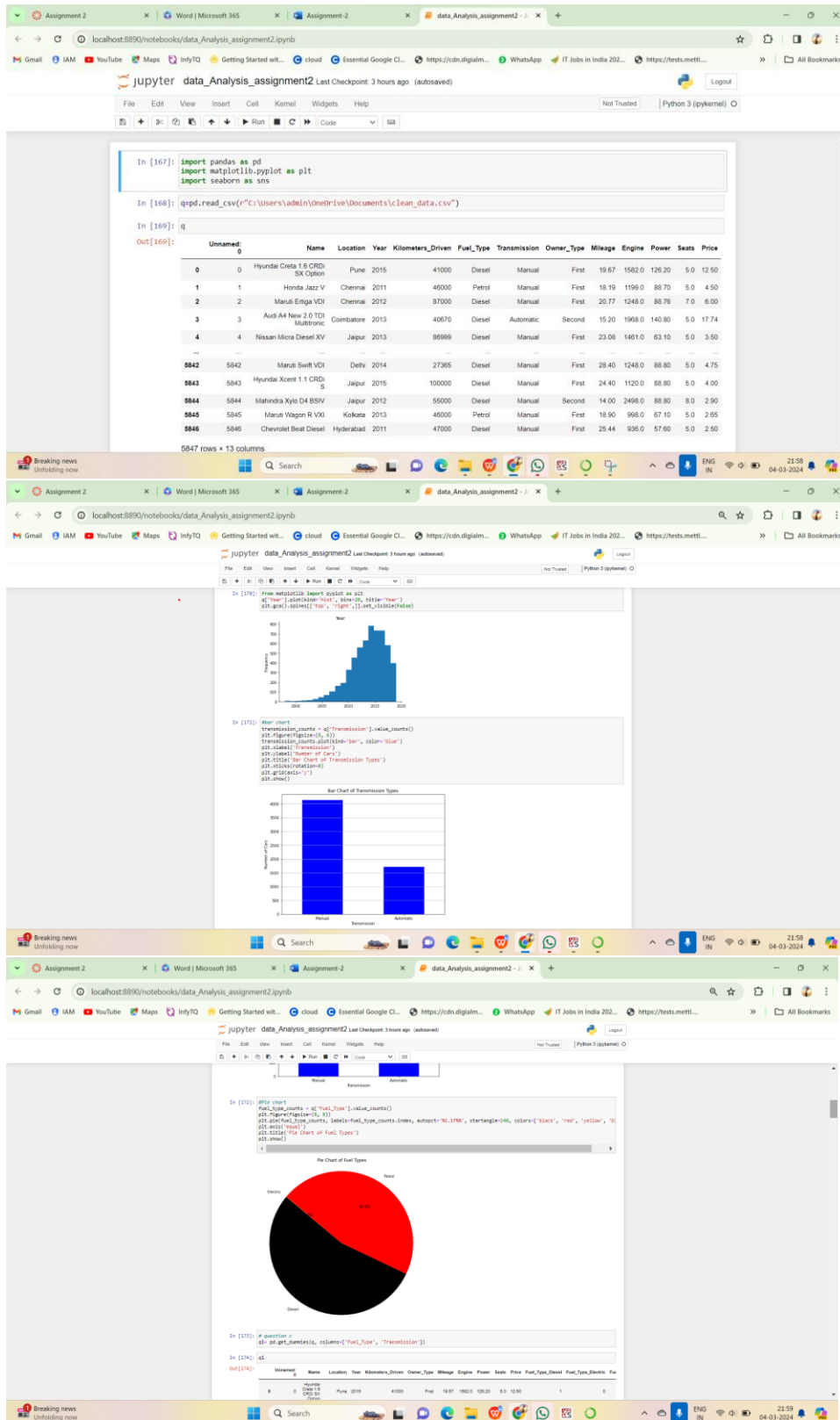
```
Out[23]:
```

Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price
0	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67	1582.0	126.20	5.0	12.50
1	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.19	1199.0	88.70	5.0	4.50
2	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77	1248.0	88.76	7.0	6.00
3	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.20	1968.0	140.80	5.0	17.74
4	Nissan Micra Diesel XV	Jaipur	2013	86999	Diesel	Manual	First	23.08	1461.0	63.10	5.0	3.50
...
5842	Maruti Swift VDI	Dehi	2014	27365	Diesel	Manual	First	28.40	1248.0	88.80	5.0	4.75
5843	Hyundai Xcent 1.1 CRDi S	Jaipur	2015	100000	Diesel	Manual	First	24.40	1120.0	88.80	5.0	4.00
5844	Maruti Kya D4 BSIV	Jaipur	2012	56000	Diesel	Manual	Second	14.00	2468.0	88.80	8.0	2.90
5845	Maruti Vagon R VDI	Kolkata	2013	46000	Petrol	Manual	First	18.90	968.0	67.10	5.0	2.65
5846	Chevrolet Beat Diesel	Hyderabad	2011	47000	Diesel	Manual	First	25.44	936.0	57.60	5.0	2.50

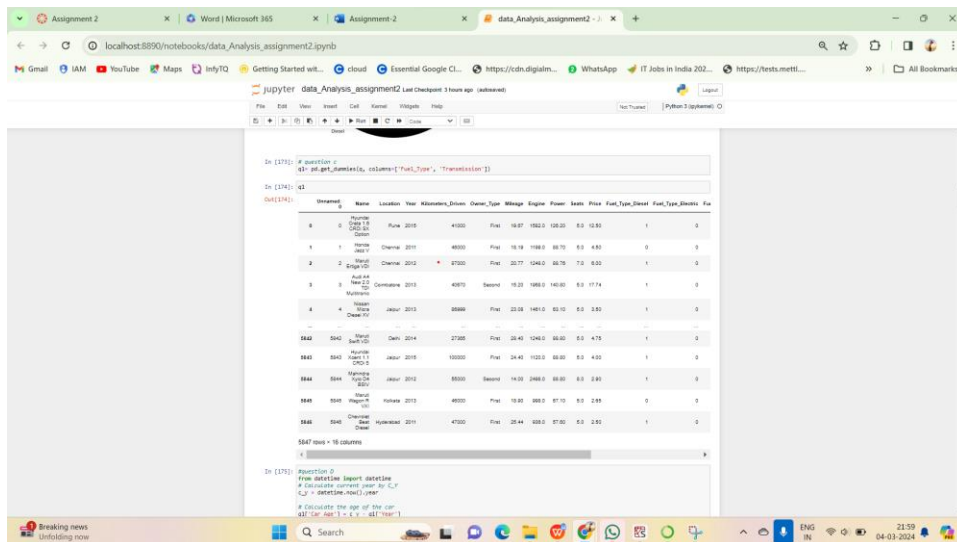
```
5847 rows x 12 columns
```

```
In [25]: s.to_csv("c:\\Users\\admin\\OneDrive\\Documents\\clean_data.csv")
```

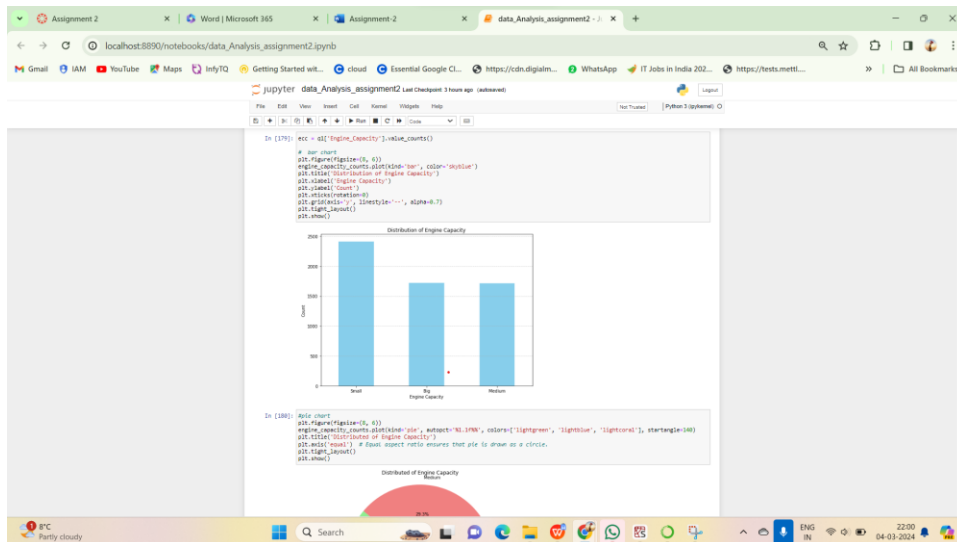
DATA ANALYSIS

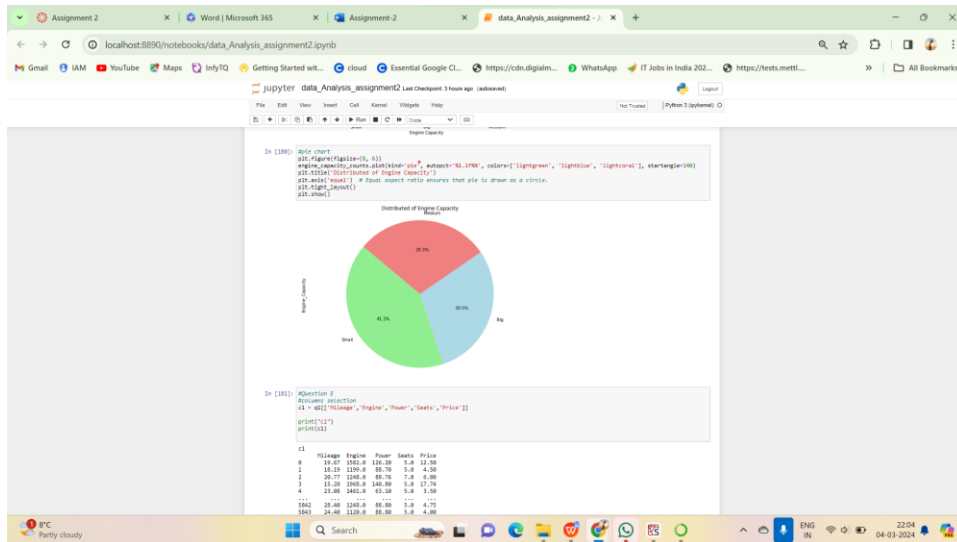


C) Change the categorical variables (“Fuel_Type” and “Transmission”) into numerical one hot encoded value.

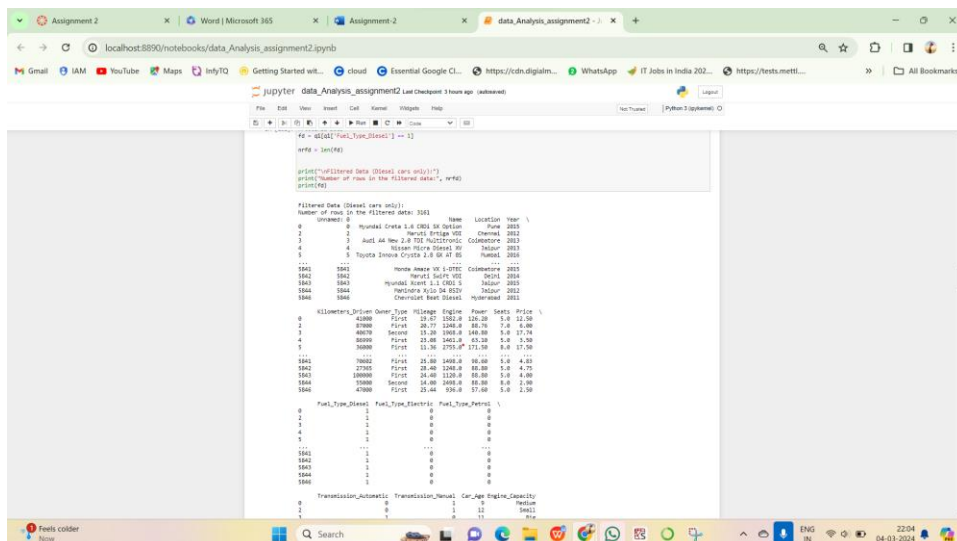


- d) Create one more feature and add this column to the dataset (you can use mutate function in R for this). For example, you can calculate the current age of the car by subtracting “Year” value from the current year.
- Added new column “Car_Age” and “Engine_Capacity”.





E) Perform select, filter, rename, mutate, arrange and summarize with group by operations (or their equivalent operations in python) on this dataset.



The screenshot shows a Jupyter Notebook interface with the following code and output:

```

In [181]: from pandas import read_csv
          df = read_csv('data_Analysis_assignment2.csv')
          print(df.columns)
          print(df)

```

The output displays the columns of the DataFrame and a preview of the data:

```

Renamed Columns:      Name      Locality      Year
0      Hummer H2      US Option      2002
1      Honda Jazz      Chennai      2011
2      Honda Civic      Chennai      2012
3      Audi A4 New 40 TFSI quattro      Germany      2013
4      Nissan Xterra SE      India      2013

0      2002      Hummer H2      US Option      2002
0800      2002      Hummer H2      US Option      2002
0800      2002      Hummer H2      US Option      2002
0800      2002      Hummer H2      US Option      2002
0800      2002      Hummer H2      US Option      2002

Kilometers      Drive      Owner_Type      Mileage_per_Liter      Engine      Power      Seats
0      40000      First      0      18.00      140.00      5.0
1      40000      First      18.35      120.00      5.0
2      80000      First      20.37      120.00      5.0
3      40070      Second      15.00      100.00      4.0
4      80000      First      15.40      140.00      5.0

0800      27500      First      20.40      100.00      5.0
0800      100000      First      20.40      110.00      5.0
0800      100000      Second      20.40      100.00      5.0
0800      40000      First      18.00      100.00      5.0
0800      40000      First      20.40      110.00      5.0

Price      Fuel_Type      Fuel_Type_Electric      Fuel_Type_Petrol
0      11.50      1      0
1      4.50      0      0
2      6.00      1      0
3      12.74      1      0
4      1.50      1      0

0800      4.75      1      0
0800      6.00      1      0
0800      2.00      0      0
0800      2.00      0      0
0800      2.00      0      0

```


[illegible]

Assignment 2 | Word | Microsoft 365 | Assignment 2 | data_Analysis_assignment2 - Jupyter

localhost:8890/notebooks/data_Analysis_assignment2.ipynb

Gmail IAM YouTube Maps InfoTQ Getting Started with... cloud Essential Google CL... https://cdn.digitall... WhatsApp IT Jobs in India 202... https://tests.mettl...

Jupyter data_Analysis_assignment2 Last checkpoint 3 hours ago (auto-saved)

File Edit View Insert Cell Kernel Help Test Tracker Python 3 (system)

1387 1388

```
[1387]: df.groupby('Location').agg({'Price': 'mean', 'Year': 'max'}).reset_index()
print(df)
```

Grouped Summary by Location:

Location	Price	Year
Mumbai	8.367048	2019
Bangalore	11.482750	2018
Chennai	7.092048	2019
Coimbatore	11.208048	2019
Delhi	9.809048	2019
Hyderabad	9.891923	2019
Jaipur	9.167048	2019
Kolkata	11.198109	2019
Ahmedabad	5.779048	2019
Pune	9.302048	2019
Pune	8.491088	2019

```
[1388]: df = df.groupby(['Location', 'Year']).size().reset_index(name='Count')
print(df)
```

Location Year Count

Location	Year	Count
Mumbai	2005	1
Mumbai	2006	1
Mumbai	2007	3
Mumbai	2008	8
Pune	2015	54
Pune	2016	87
Pune	2017	42
Pune	2018	52
Pune	2019	1

[1389]: df.groupby('Location').size().reset_index(name='Number_of_Cars')
print(df)

Location Number_of_Cars

Location	Number_of_Cars
Mumbai	218
Bangalore	252
Chennai	476
Coimbatore	631
Delhi	548
Hyderabad	734
Jaipur	489
Kolkata	688
Pune	762
Pune	388

Assignment 2 | Word | Microsoft 365 | Assignment 2 | data_Analysis_assignment2 - Jupyter

localhost:8890/notebooks/data_Analysis_assignment2.ipynb

Gmail IAM YouTube Maps InfoTQ Getting Started with... cloud Essential Google CL... https://cdn.digitall... WhatsApp IT Jobs in India 202... https://tests.mettl...

Jupyter data_Analysis_assignment2 Last checkpoint 3 hours ago (auto-saved)

File Edit View Insert Cell Kernel Help Test Tracker Python 3 (system)

1390 1391

```
[1390]: df.groupby('Location').size().reset_index(name='Number_of_Cars')
print(df)
```

Location Number_of_Cars

Location	Number_of_Cars
Mumbai	218
Bangalore	252
Chennai	476
Coimbatore	631
Delhi	548
Hyderabad	734
Jaipur	489
Kolkata	688
Pune	762
Pune	388

```
[1391]: plt.figure(figsize=(10, 8))
plt.pie(df['Number_of_Cars'].values.tolist(), labels=df['Location'].values.tolist(), autopct='%1.1f%%', startangle=140)
plt.title('Number of Cars by Location')
plt.show()
```

Number of Cars by Location

Number of Cars by Location

Location	Number_of_Cars	Percentage
Mumbai	218	18.5%
Bangalore	252	21.0%
Chennai	476	40.4%
Coimbatore	631	54.1%
Delhi	548	46.2%
Hyderabad	734	61.3%
Jaipur	489	41.5%
Kolkata	688	58.8%
Pune	762	63.5%
Pune	388	10.0%

```
[1392]: df.groupby('Car_Age').size().reset_index(name='Car_Count')
print(df)
print('Grouped data by car age:')
print(df)
```

Grouped data by car age:

