DATA MINING

Assignment – 1

PYTHON REPORT

Submitted By

Lavanya Srinivasan – 1002040671

Prem Atul Jethwa – 1001861810

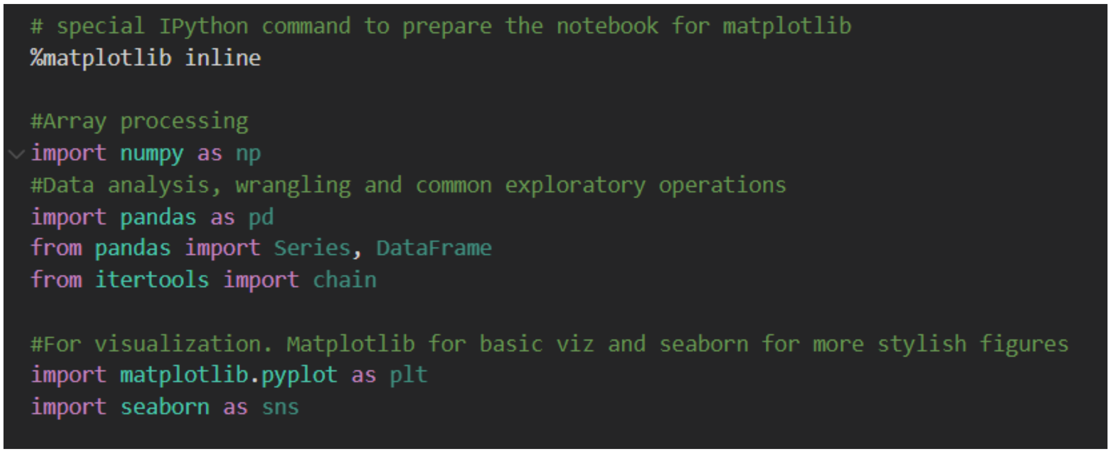
Shubham Sharma – 1001964524

Introduction:

In this Report we provide Exploratory Data Analysis for the given Vehicle Sales Dataset. We have accessed the data's using ranking, filtering, grouping and aggregation operations. We have provided visualization for requested data patters using matplotlib and seaborn and identified few interesting data patters.

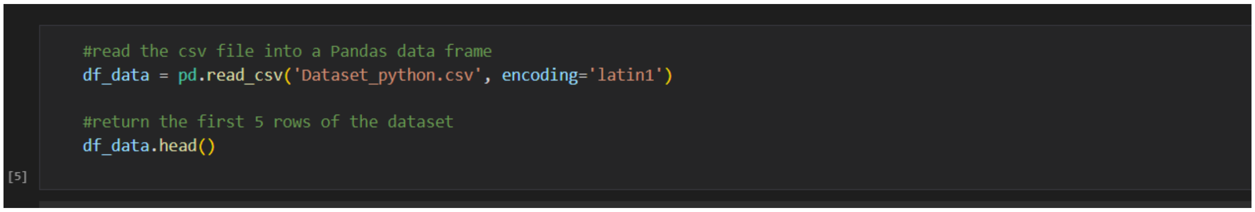
Creating and Accessing Data frame:

Required Packages:

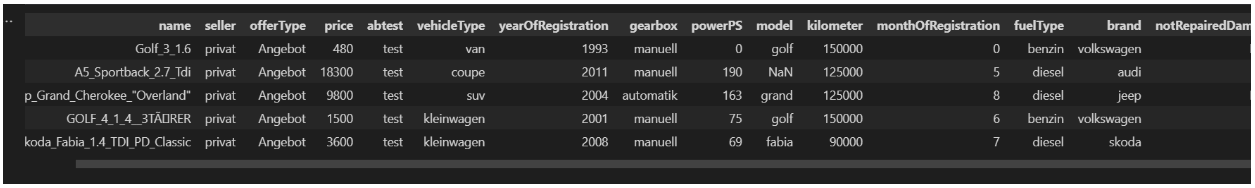


```
# special IPython command to prepare the notebook for matplotlib
%matplotlib inline

#Array processing
import numpy as np
#Data analysis, wrangling and common exploratory operations
import pandas as pd
from pandas import Series, DataFrame
from itertools import chain

#For visualization. Matplotlib for basic viz and seaborn for more stylish figures
import matplotlib.pyplot as plt
import seaborn as sns
```

Printing first 5 rows from the dataset:



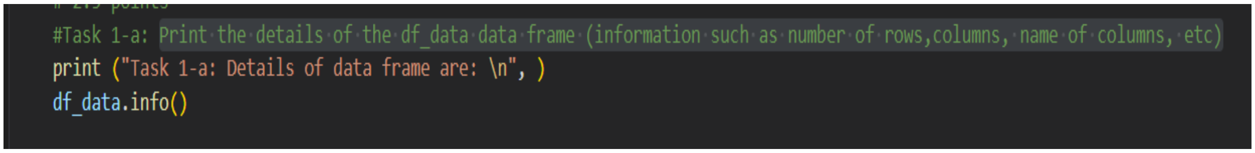
```
[5]
    #read the csv file into a Pandas data frame
    df_data = pd.read_csv('Dataset_python.csv', encoding='latin1')

    #return the first 5 rows of the dataset
    df_data.head()
```



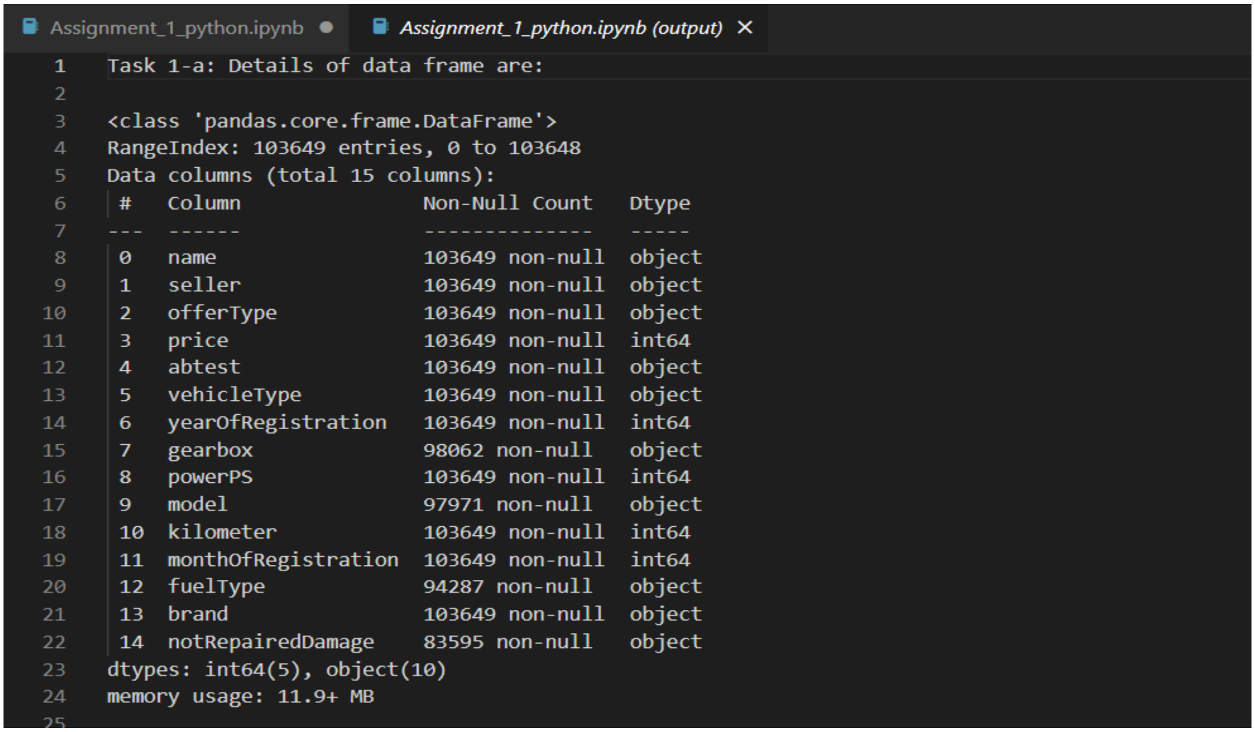
name	seller	offerType	price	abtest	vehicleType	yearOfRegistration	gearbox	powerPS	model	kilometer	monthOfRegistration	fuelType	brand	notRepairedDam
Golf_3_1.6	privat	Angebot	480	test	van	1993	manuell	0	golf	150000	0	benzin	volkswagen	
A5_Sportback_2.7_Tdi	privat	Angebot	18300	test	coupe	2011	manuell	190	NaN	125000	5	diesel	audi	
p_Grand_Cherokee_"Overland"	privat	Angebot	9800	test	suv	2004	automatik	163	grand	125000	8	diesel	jeep	
GOLF_4_1_4__3TÃ□RER	privat	Angebot	1500	test	kleinwagen	2001	manuell	75	golf	150000	6	benzin	volkswagen	
koda_Fabia_1.4_TDI_PD_Classic	privat	Angebot	3600	test	kleinwagen	2008	manuell	69	fabia	90000	7	diesel	skoda	

Task 1: Statistical Data Analysis:

1a. Print the details of the df_data data frame (information such as number of rows,columns, name of columns, etc)

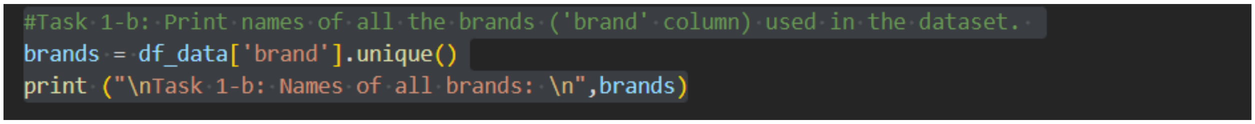


```
# 2.5 points
#Task 1-a: Print the details of the df_data data frame (information such as number of rows,columns, name of columns, etc)
print ("Task 1-a: Details of data frame are: \n", )
df_data.info()
```

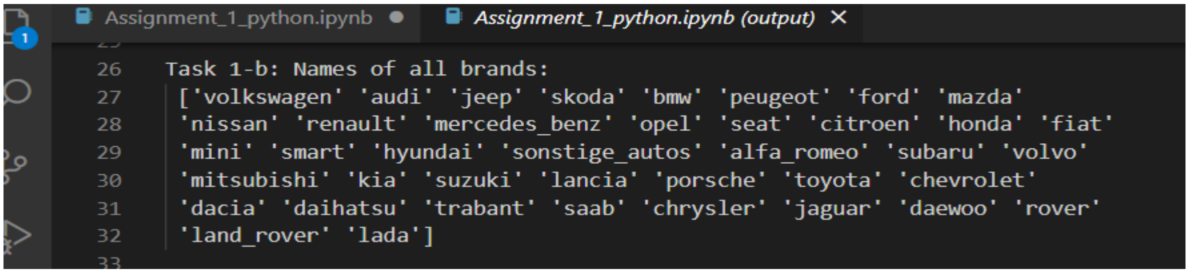


```
Task 1-a: Details of data frame are:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 103649 entries, 0 to 103648
Data columns (total 15 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   name                 103649 non-null  object
 1   seller               103649 non-null  object
 2   offerType            103649 non-null  object
 3   price                103649 non-null  int64
 4   abtest               103649 non-null  object
 5   vehicleType          103649 non-null  object
 6   yearOfRegistration   103649 non-null  int64
 7   gearbox              98062 non-null   object
 8   powerPS              103649 non-null  int64
 9   model                97971 non-null   object
 10  kilometer            103649 non-null  int64
 11  monthOfRegistration  103649 non-null  int64
 12  fuelType             94287 non-null   object
 13  brand                103649 non-null  object
 14  notRepairedDamage    83595 non-null   object
dtypes: int64(5), object(10)
memory usage: 11.9+ MB
```

1b. Print names of all brands:

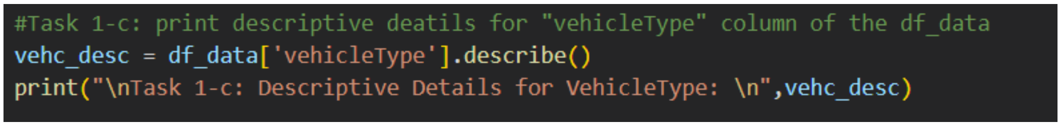


```
#Task 1-b: Print names of all the brands ('brand' column) used in the dataset.
brands = df_data['brand'].unique()
print ("\nTask 1-b: Names of all brands: \n",brands)
```

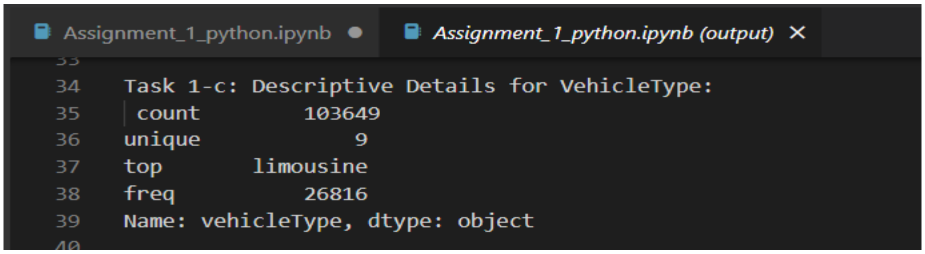


```
Task 1-b: Names of all brands:
 ['volkswagen' 'audi' 'jeep' 'skoda' 'bmw' 'peugeot' 'ford' 'mazda'
 'nissan' 'renault' 'mercedes_benz' 'opel' 'seat' 'citroen' 'honda' 'fiat'
 'mini' 'smart' 'hyundai' 'sonstige_autos' 'alfa_romeo' 'subaru' 'volvo'
 'mitsubishi' 'kia' 'suzuki' 'lancia' 'porsche' 'toyota' 'chevrolet'
 'dacia' 'daihatsu' 'trabant' 'saab' 'chrysler' 'jaguar' 'daewoo' 'rover'
 'land_rover' 'lada']
```

1c. Print descriptive details for vehicle Type from df_data:

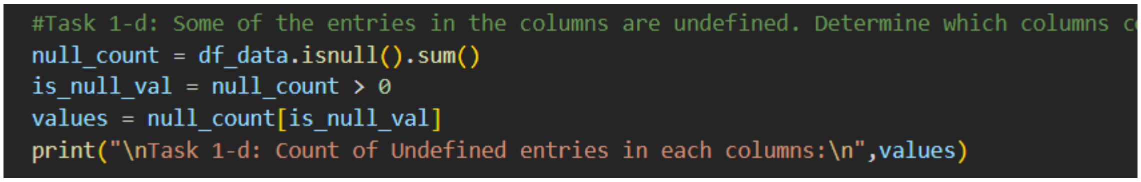


```
#Task 1-c: print descriptive deatils for "vehicleType" column of the df_data
vehc_desc = df_data['vehicleType'].describe()
print("\nTask 1-c: Descriptive Details for VehicleType: \n",vehc_desc)
```

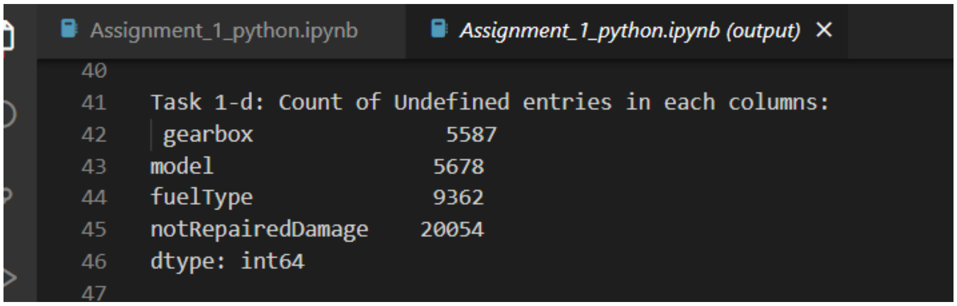


```
Task 1-c: Descriptive Details for VehicleType:
 count        103649
unique            9
top       limousine
freq          26816
Name: vehicleType, dtype: object
```

1d. Determining unidentified entries and printing the same.



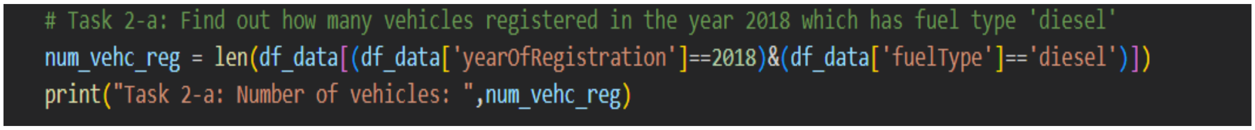
```
#Task 1-d: Some of the entries in the columns are undefined. Determine which columns c
null_count = df_data.isnull().sum()
is_null_val = null_count > 0
values = null_count[is_null_val]
print("\nTask 1-d: Count of Undefined entries in each columns:\n",values)
```



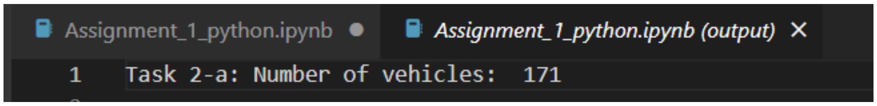
```

Task 1-d: Count of Undefined entries in each columns:
 gearbox              5587
model                5678
fuelType             9362
notRepairedDamage    20054
dtype: int64

```

Task 2: Aggregation, Filtering and Ranking:

2a. Printing how many vehicles registered in 2018 and with fuel type diesel.

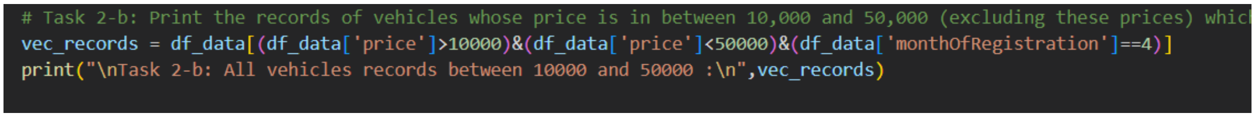


```
# Task 2-a: Find out how many vehicles registered in the year 2018 which has fuel type 'diesel'
num_vehc_reg = len(df_data[(df_data['yearOfRegistration']==2018)&(df_data['fuelType']=='diesel')])
print("Task 2-a: Number of vehicles: ",num_vehc_reg)
```

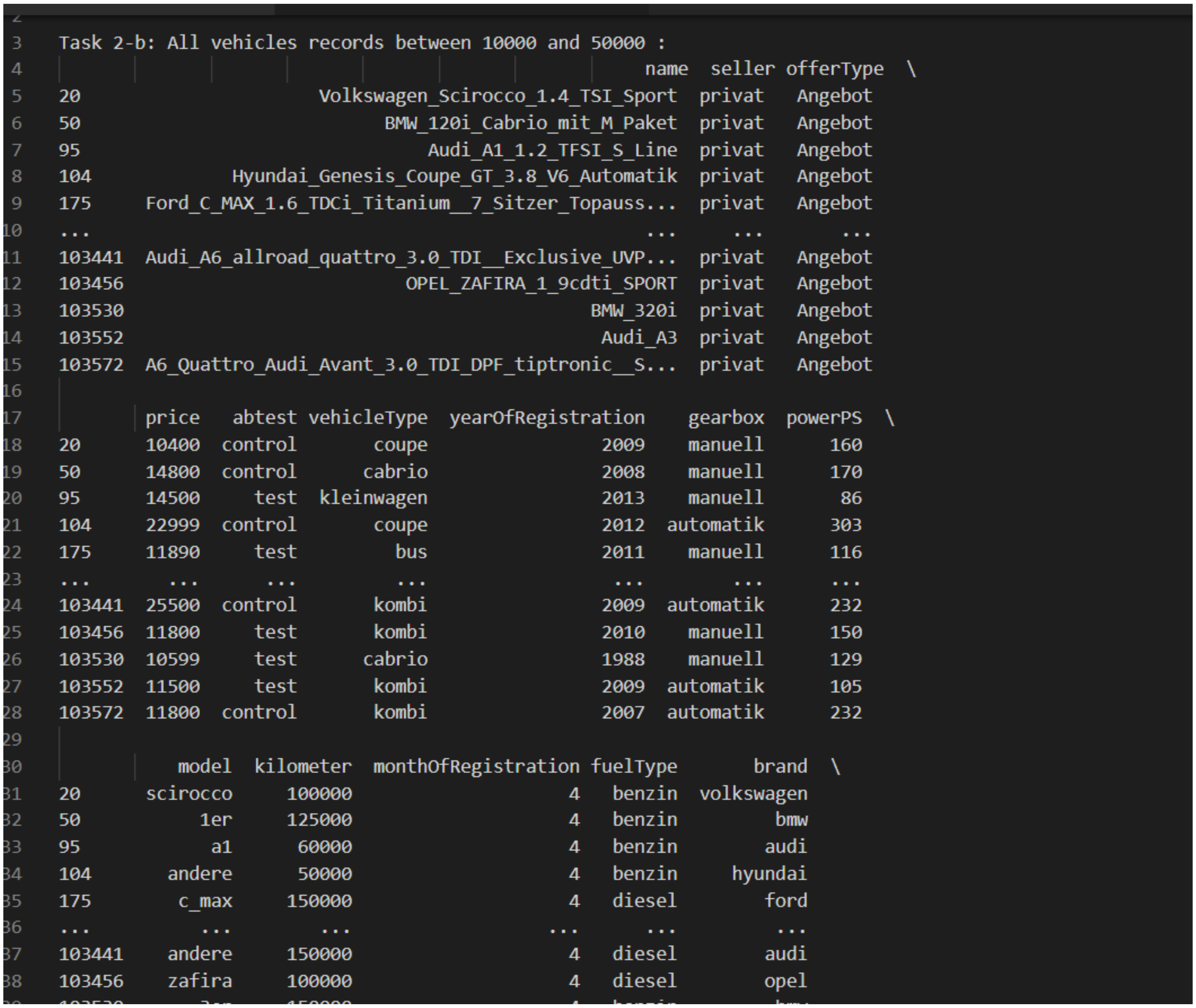


```
Task 2-a: Number of vehicles:  171
```

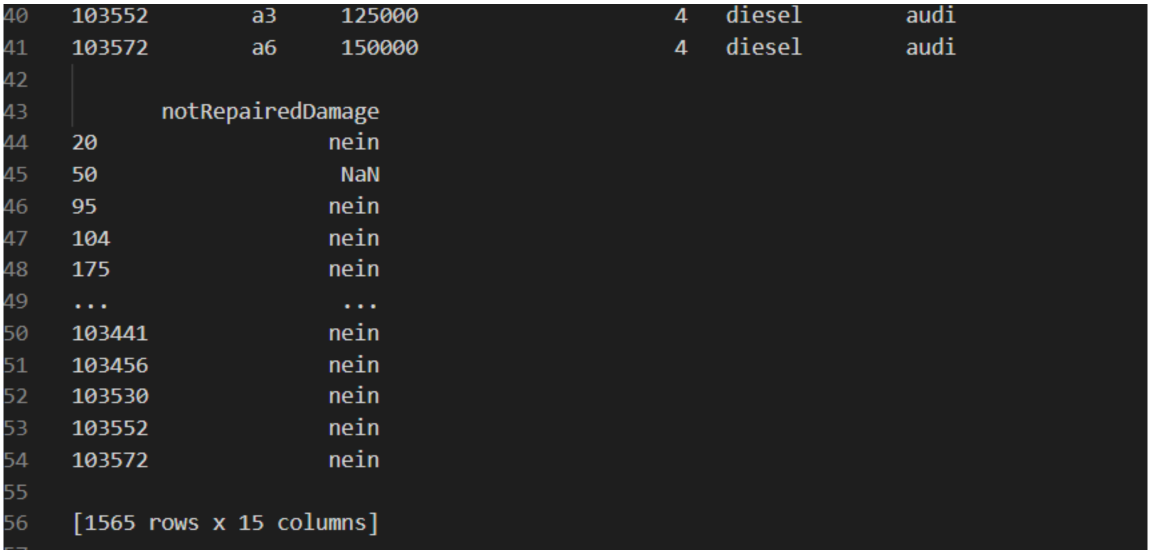
2b. Showing records of vehicles with price between 10000 and 50000.



```
# Task 2-b: Print the records of vehicles whose price is in between 10,000 and 50,000 (excluding these prices) whic
vec_records = df_data[(df_data['price']>10000)&(df_data['price']<50000)&(df_data['monthOfRegistration']==4)]
print("\nTask 2-b: All vehicles records between 10000 and 50000 :\n",vec_records)
```

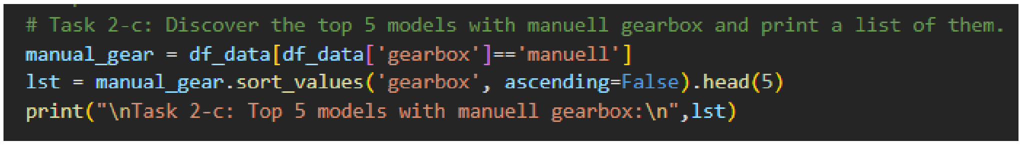


```
Task 2-b: All vehicles records between 10000 and 50000 :
                                                     name  seller offerType  \
20                        Volkswagen_Scirocco_1.4_TSI_Sport  privat   Angebot
50                              BMW_120i_Cabrio_mit_M_Paket  privat   Angebot
95                                  Audi_A1_1.2_TFSI_S_Line  privat   Angebot
104                Hyundai_Genesis_Coupe_GT_3.8_V6_Automatik  privat   Angebot
175     Ford_C_MAX_1.6_TDCi_Titanium__7_Sitzer_Topauss...  privat   Angebot
...                                                   ...     ...       ...
103441  Audi_A6_allroad_quattro_3.0_TDI__Exclusive_UVP...  privat   Angebot
103456                            OPEL_ZAFIRA_1_9cdti_SPORT  privat   Angebot
103530                                             BMW_320i  privat   Angebot
103552                                              Audi_A3  privat   Angebot
103572  A6_Quattro_Audi_Avant_3.0_TDI_DPF_tiptronic__S...  privat   Angebot

        price   abtest vehicleType  yearOfRegistration    gearbox  powerPS  \
20      10400  control       coupe                2009    manuell      160
50      14800  control      cabrio                2008    manuell      170
95      14500     test  kleinwagen                2013    manuell       86
104     22999  control       coupe                2012  automatik      303
175     11890     test         bus                2011    manuell      116
...       ...      ...         ...                 ...        ...      ...
103441  25500  control       kombi                2009  automatik      232
103456  11800     test       kombi                2010    manuell      150
103530  10599     test      cabrio                1988    manuell      129
103552  11500     test       kombi                2009  automatik      105
103572  11800  control       kombi                2007  automatik      232

           model  kilometer  monthOfRegistration fuelType       brand  \
20      scirocco     100000                    4   benzin  volkswagen
50           1er     125000                    4   benzin         bmw
95            a1      60000                    4   benzin        audi
104       andere      50000                    4   benzin     hyundai
175        c_max     150000                    4   diesel        ford
...          ...        ...                  ...      ...         ...
103441    andere     150000                    4   diesel        audi
103456    zafira     100000                    4   diesel        opel
103530       3er     150000                    4   benzin         bmw
```

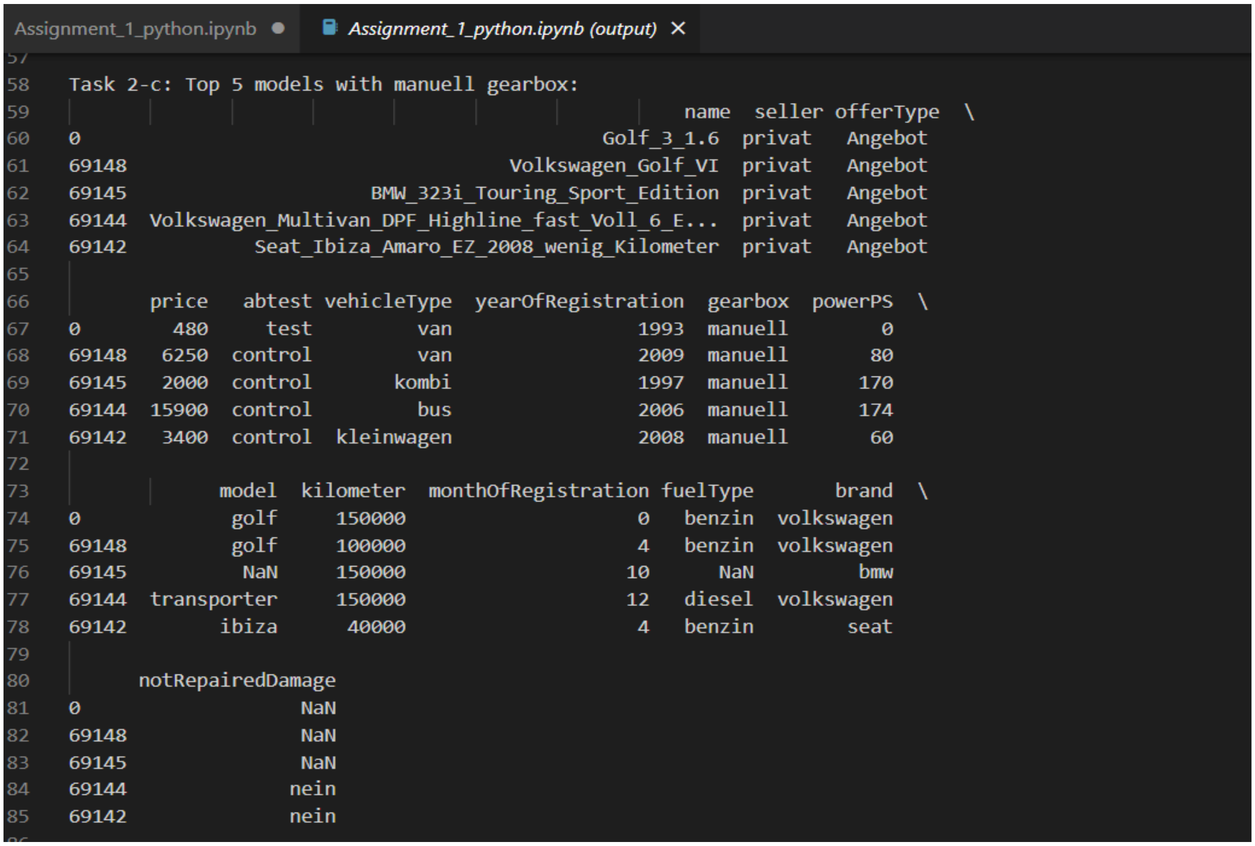


```
103552        a3     125000                    4   diesel        audi
103572        a6     150000                    4   diesel        audi

       notRepairedDamage
20                  nein
50                   NaN
95                  nein
104                 nein
175                 nein
...                  ...
103441              nein
103456              nein
103530              nein
103552              nein
103572              nein

[1565 rows x 15 columns]
```

2c. Top 5 models with ‘manuell’ gearbox from the dataset.



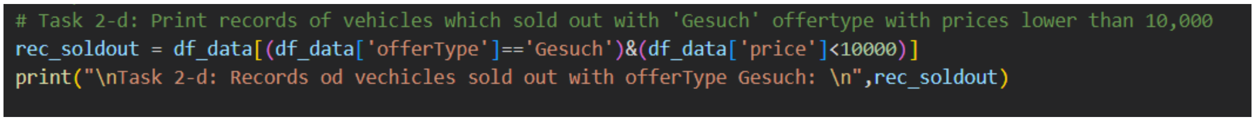
```python
# Task 2-c: Discover the top 5 models with manuell gearbox and print a list of them.
manual_gear = df_data[df_data['gearbox']=='manuell']
lst = manual_gear.sort_values('gearbox', ascending=False).head(5)
print("\nTask 2-c: Top 5 models with manuell gearbox:\n",lst)
```



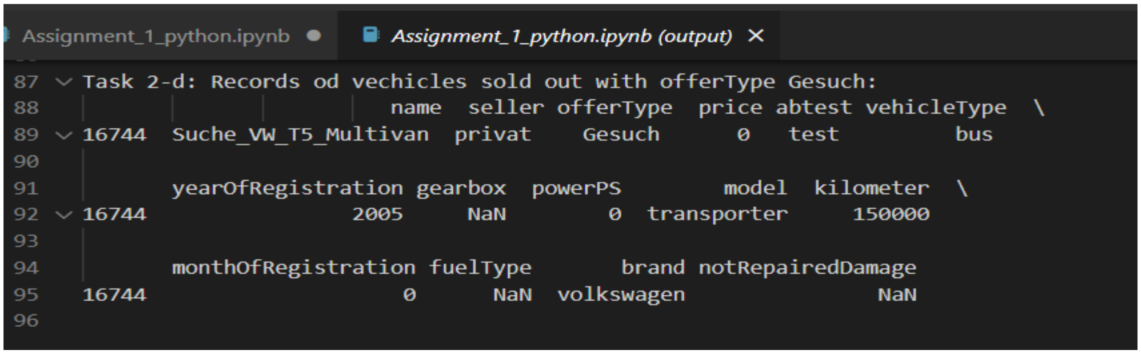
Assignment_1_python.ipynb ●
Assignment_1_python.ipynb (output) ×

```
Task 2-c: Top 5 models with manuell gearbox:
                                                    name  seller offerType  \
0                                             Golf_3_1.6  privat   Angebot
69148                                 Volkswagen_Golf_VI  privat   Angebot
69145                     BMW_323i_Touring_Sport_Edition  privat   Angebot
69144  Volkswagen_Multivan_DPF_Highline_fast_Voll_6_E...  privat   Angebot
69142           Seat_Ibiza_Amaro_EZ_2008_wenig_Kilometer  privat   Angebot

       price   abtest vehicleType  yearOfRegistration  gearbox  powerPS  \
0        480     test         van                1993  manuell        0
69148   6250  control         van                2009  manuell       80
69145   2000  control       kombi                1997  manuell      170
69144  15900  control         bus                2006  manuell      174
69142   3400  control  kleinwagen                2008  manuell       60

             model  kilometer  monthOfRegistration fuelType       brand  \
0             golf     150000                    0   benzin  volkswagen
69148         golf     100000                    4   benzin  volkswagen
69145          NaN     150000                   10      NaN         bmw
69144  transporter     150000                   12   diesel  volkswagen
69142        ibiza      40000                    4   benzin        seat

      notRepairedDamage
0                   NaN
69148               NaN
69145               NaN
69144              nein
69142              nein
```

2d. Showing vehicles sold with ‘Gesuch’ offer type with price lower than 10000.



```python
# Task 2-d: Print records of vehicles which sold out with 'Gesuch' offertype with prices lower than 10,000
rec_soldout = df_data[(df_data['offerType']=='Gesuch')&(df_data['price']<10000)]
print("\nTask 2-d: Records od vechicles sold out with offerType Gesuch: \n",rec_soldout)
```

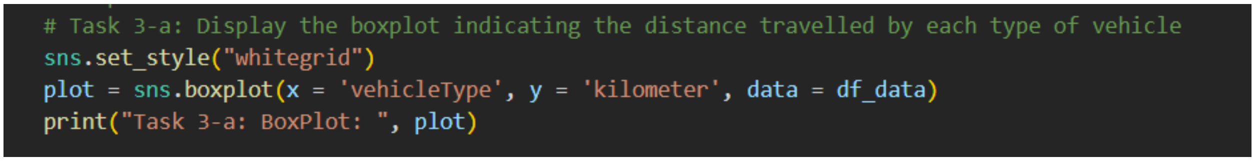


Assignment_1_python.ipynb ●
Assignment_1_python.ipynb (output) ×

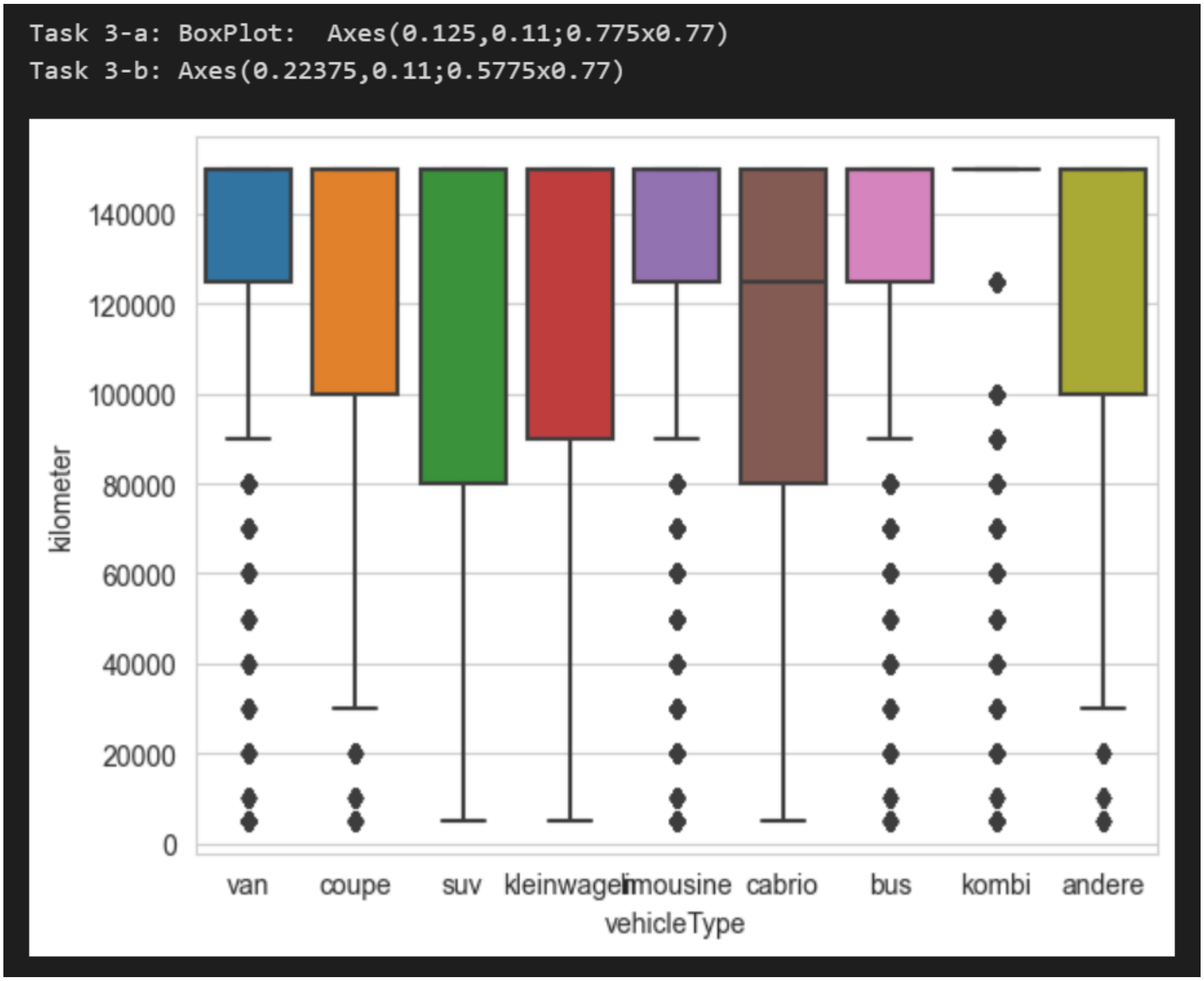
```
Task 2-d: Records od vechicles sold out with offerType Gesuch:
                       name  seller offerType  price abtest vehicleType  \
16744  Suche_VW_T5_Multivan  privat    Gesuch      0   test         bus

       yearOfRegistration gearbox  powerPS        model  kilometer  \
16744                2005     NaN        0  transporter     150000

       monthOfRegistration fuelType       brand notRepairedDamage
16744                    0      NaN  volkswagen               NaN
```

Task 3: Visualization

3a. Box plot indicating the distance travelled by each vehicle.

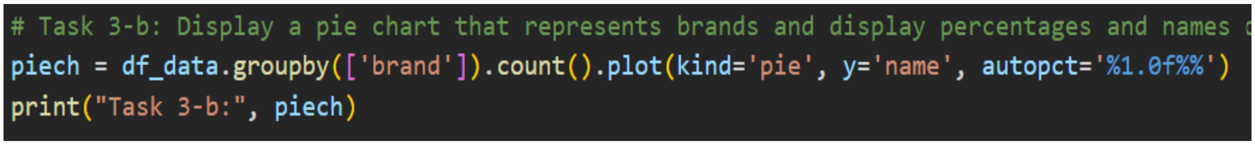


```
# Task 3-a: Display the boxplot indicating the distance travelled by each type of vehicle
sns.set_style("whitegrid")
plot = sns.boxplot(x = 'vehicleType', y = 'kilometer', data = df_data)
print("Task 3-a: BoxPlot: ", plot)
```

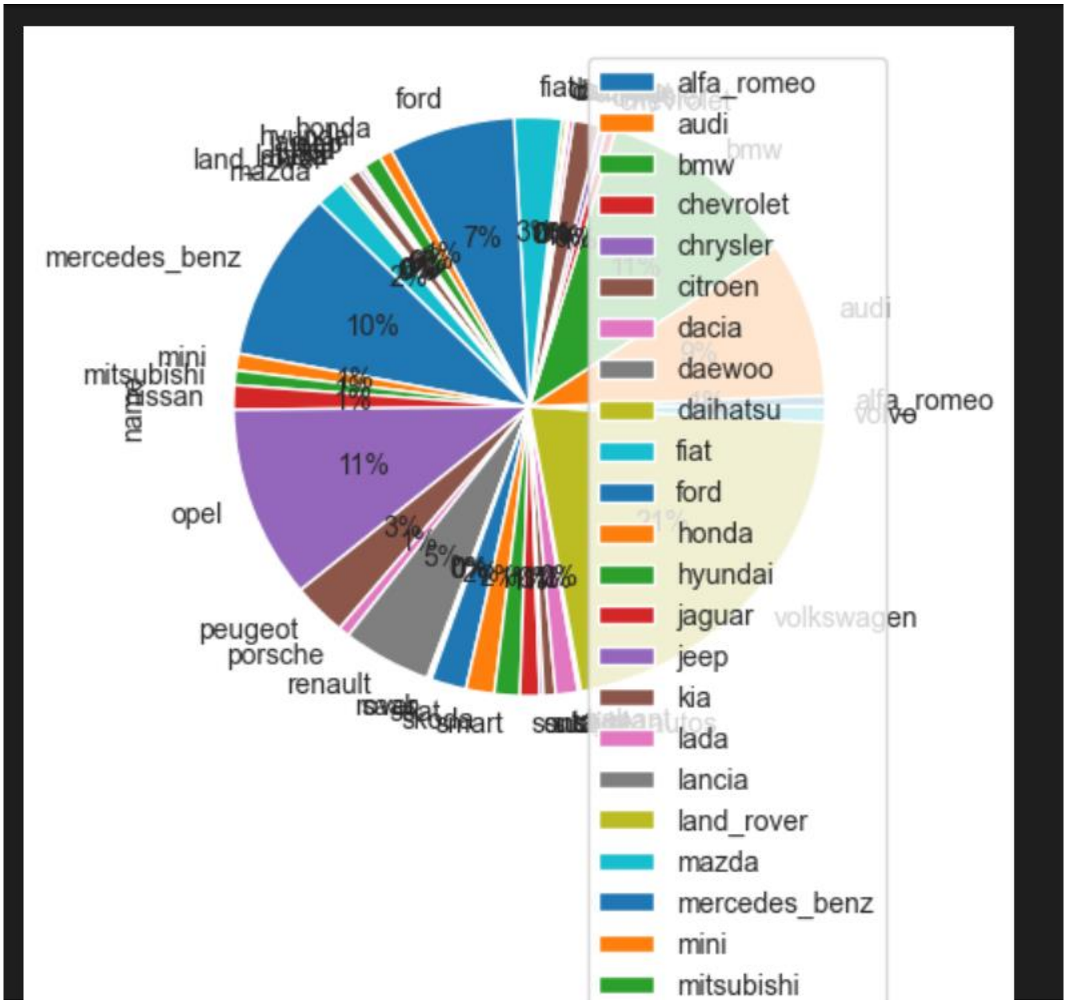


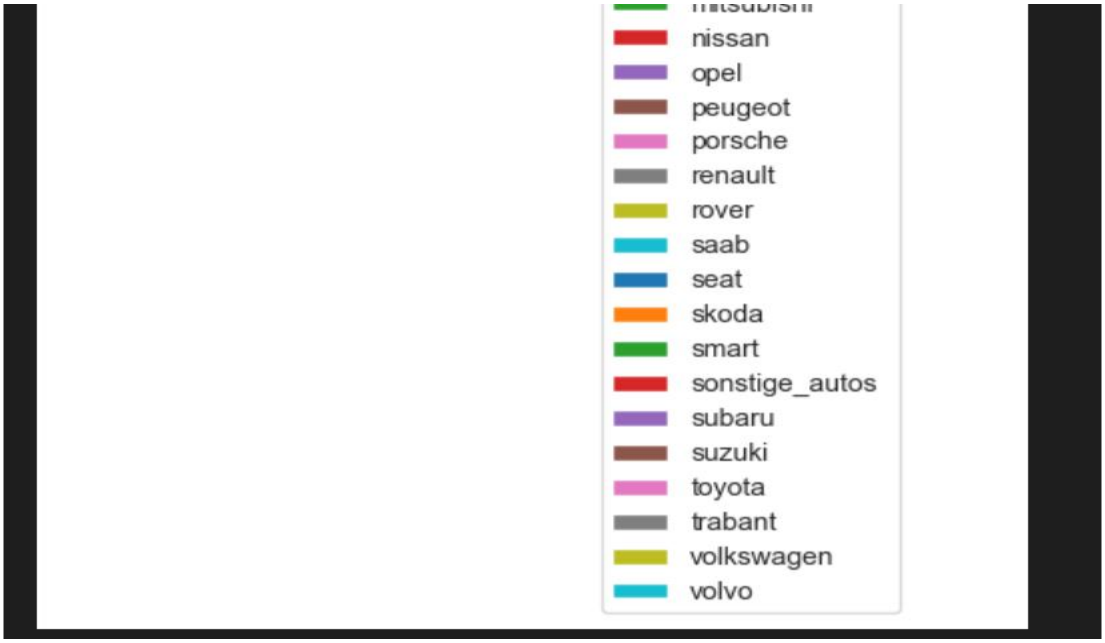
```
Task 3-a: BoxPlot:  Axes(0.125,0.11;0.775x0.77)
Task 3-b: Axes(0.22375,0.11;0.5775x0.77)
```

3b. Pie chart representing the brands with their percentage.



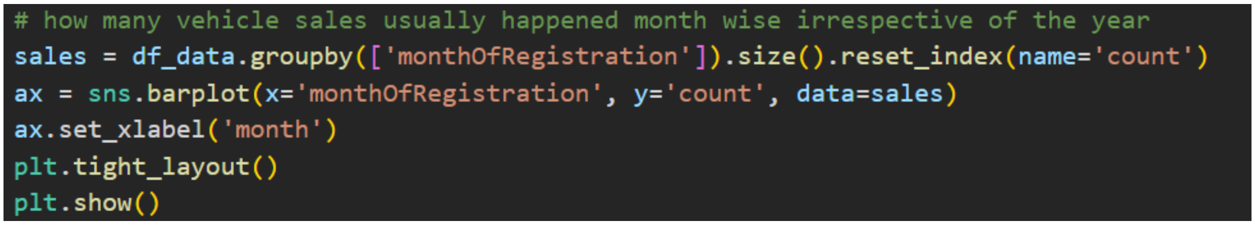
```
# Task 3-b: Display a pie chart that represents brands and display percentages and names
piech = df_data.groupby(['brand']).count().plot(kind='pie', y='name', autopct='%1.0f%%')
print("Task 3-b:", piech)
```


ford
fiat
chevrolet
bmw
honda
hyundai
land_rover
mazda
mercedes_benz
audi
mini
mitsubishi
nissan
name
alfa_romeo
volvo
opel
peugeot
porsche
renault
volkswagen
smart
7%
3%
11%
2%
10%
9%
1%
1%
11%
21%
3%
1%
5%
alfa_romeo
audi
bmw
chevrolet
chrysler
citroen
dacia
daewoo
daihatsu
fiat
ford
honda
hyundai
jaguar
jeep
kia
lada
lancia
land_rover
mazda
mercedes_benz
mini
mitsubishi

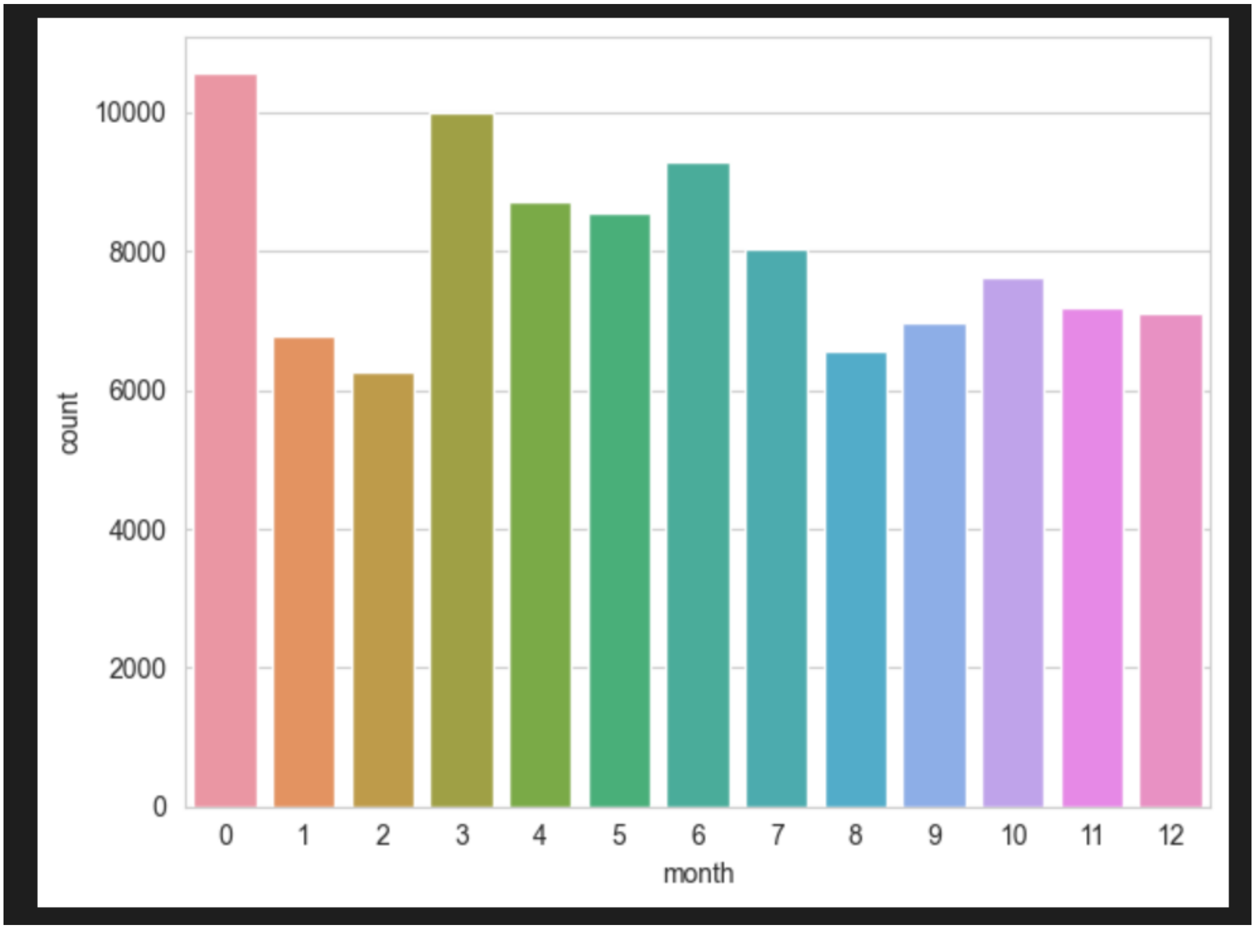

mitsubishi
nissan
opel
peugeot
porsche
renault
rover
saab
seat
skoda
smart
sonstige_autos
subaru
suzuki
toyota
trabant
volkswagen
volvo

Task 4: Insights from the Data

4a. Vehicle sales report month wise irrespective of year.

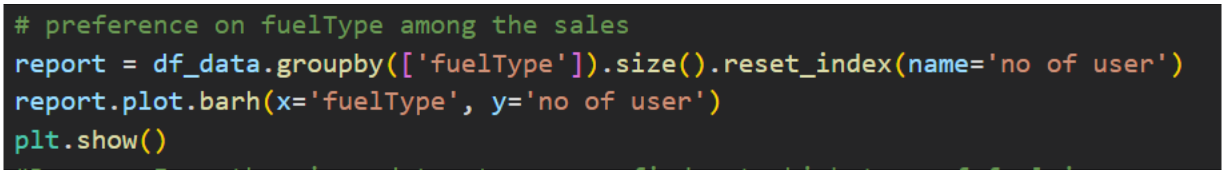


```
# how many vehicle sales usually happened month wise irrespective of the year
sales = df_data.groupby(['monthOfRegistration']).size().reset_index(name='count')
ax = sns.barplot(x='monthOfRegistration', y='count', data=sales)
ax.set_xlabel('month')
plt.tight_layout()
plt.show()
```

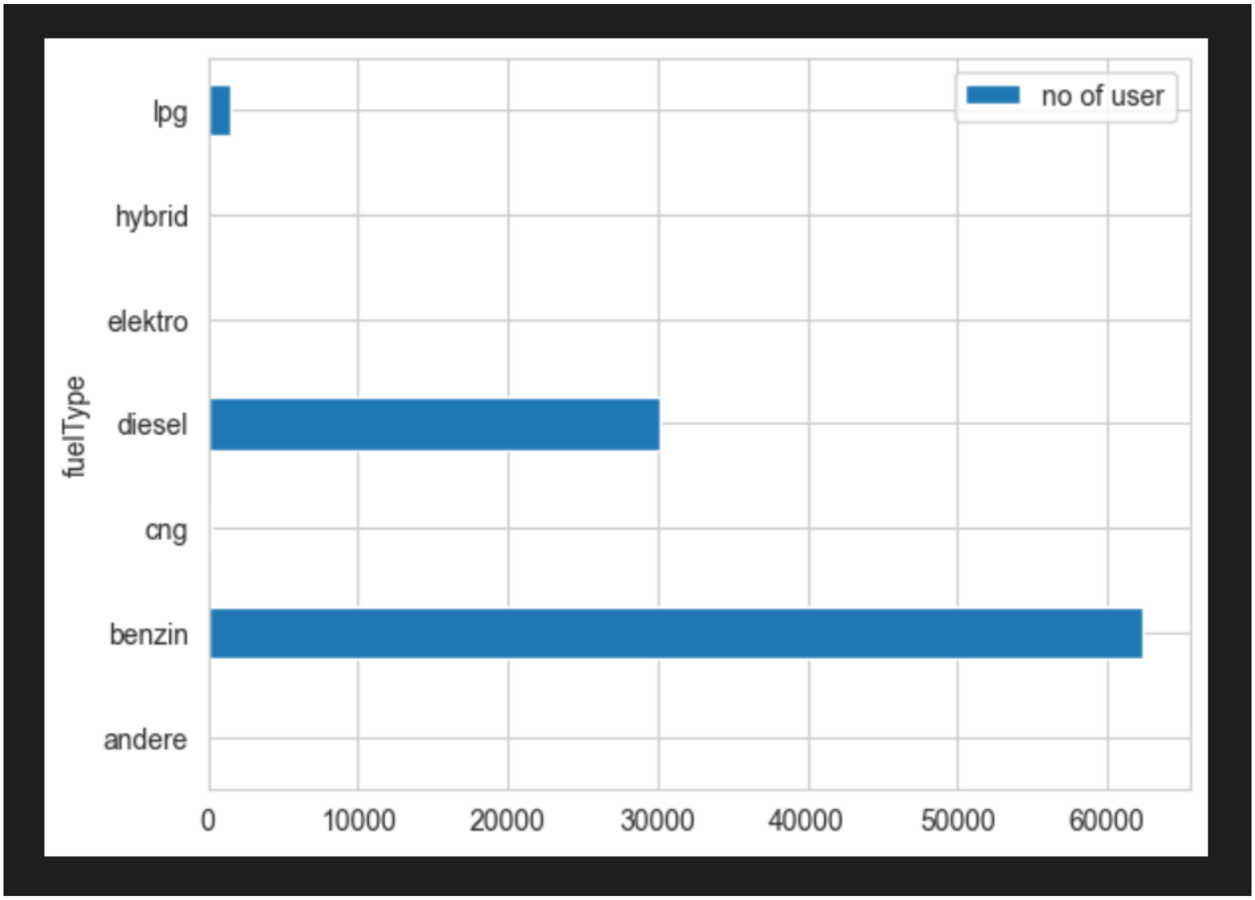


From the given dataset, one interesting fact is we can find out which month sales adds a great value to the revenue for each year. So, we plotted a graph and from the we can see in the month of 0,3,6 the sales percentage is more in each year and less in the month of 2.

4b. Preference on type of fuel among vehicles sales.



```
# preference on fuelType among the sales
report = df_data.groupby(['fuelType']).size().reset_index(name='no of user')
report.plot.barh(x='fuelType', y='no of user')
plt.show()
```



From the given dataset, we can find out which type of fuel is more popular among vehicle users. So, the graph shows 'benzin' is more popular among the vehicle and 'hybrid, elektro' are not famous among them.

Team Contributions:

Lavanya Srinivasan – 1002040671 – Python

Prem Atual Jethwa – 1001861810 – R

Shubham Sharma – 1001964524 - WEKA

References:

https://www.geeksforgeeks.org/box-plot-visualization-with-pandas-and-seaborn/

https://www.geeksforgeeks.org/how-to-create-pie-chart-from-pandas-dataframe/

https://stackoverflow.com/questions/36226083/how-to-find-which-columns-contain-any-nan-value-in-pandas-dataframe

https://stackoverflow.com/questions/47462690/how-to-get-top-5-values-from-pandas-dataframe