Manchester
Metropolitan
University

Department of Computing and Mathematics

ASSESSMENT COVER SHEET 2023/24

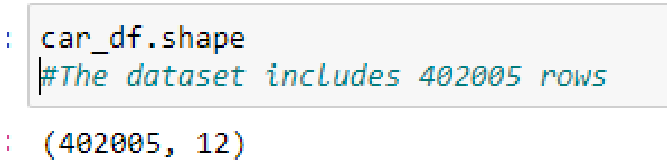
Student Name:	**POPOOLA, AYOMIDE PETER**
Student Id:	**23655243**
Unit Code and Title	**6G7V0026 Principles of Data Science**
Assessment Set By:	**L Gerber**
Assessment ID:	**1CWK100**
Assessment Title:	**Data Science Project**
Type:	**Individual**

1.0. DATA UNDERSTANDING AND EXPLORATION

1.1. Meaning and Type of Features (e.g., what the features convey/measure)

The car sales advert dataset used was provided by AutoTrader, one of the university industry partners which contains 402,005 rows and 12 columns. The features present in the data are collected data from the vehicles advertised which include brand, mileage, registration code, price, body type, fuel type, and vehicle condition. The dataset contains both categorical feature and numerical feature

1. Mileage: This indicates the distance (miles) a car has covered over a period of time (numerical feature)
2. Reg_code: This is a reg id given to all vehicles using the UK road, all cars before year 2000 were give a reg code in alphabet while the years over were in numbers (categorical and numerical)
3. Standard colour: This is the colour of the car (categorical feature)
4. Standard make: This is the company that produces the car (categorical feature)
5. Standard model: This is the varieties of cars produced by each of the standard car markers or manufacturer (categorical feature)
6. Vehicle condition: This is the condition of the car, whether they are used or brand new (categorical feature)
7. Year of registration: This is the year when the cars were registered to travel on the UK road (numerical feature)
8. Price: This is the purchasing amount of the cars (numerical feature)
9. Body type: This is the type of car shape, or the car purpose type (categorical feature)
10. Crossover car and van: Cross over are cars built on car platform, and crossover cars refer to versatile vehicles that blend features of both traditional sedans and sport utility vehicles (SUVs). A van is a type of motor vehicle typically characterized by its boxy shape, high roof, and spacious interior designed for transporting goods, people, or a combination of both. (categorical feature)
11. Fuel type: This is the fuel source used to power the vehicle (categorical feature)
12. Public reference: This is the unique value given to each row of advertised vehicle

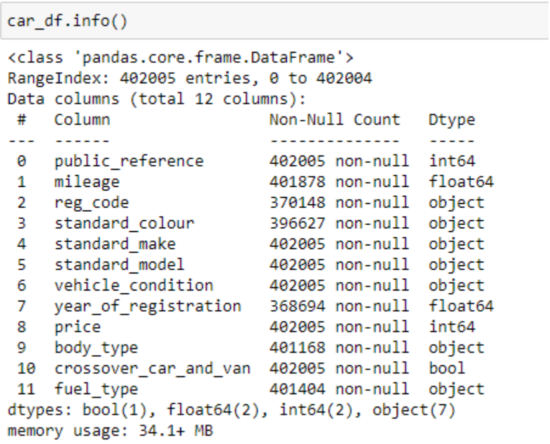


```
car_df.shape
#The dataset includes 402005 rows
```

```
(402005, 12)
```

Figure 1: The rows and columns of the car sales advert dataset

The dataset contains missing values. Figure 2 below shows the number of “non-null’’ values i.e. non missing values in each data columns. The year of registration has more missing numbers compare to others. The data type present are integer, float, object and Boolean

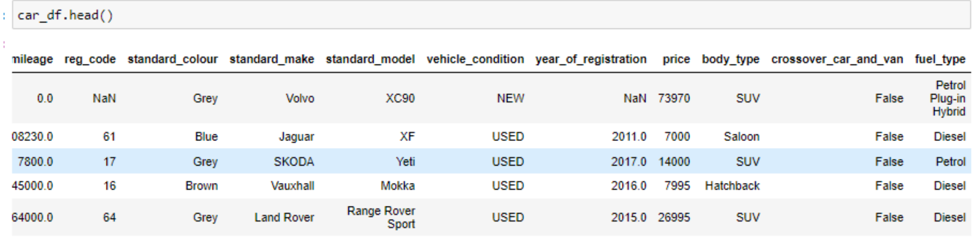


```
car_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 402005 entries, 0 to 402004
Data columns (total 12 columns):
 #   Column                 Non-Null Count   Dtype  
---  ------                 --------------   -----  
 0   public_reference       402005 non-null  int64  
 1   mileage                401878 non-null  float64
 2   reg_code               370148 non-null  object 
 3   standard_colour        396627 non-null  object 
 4   standard_make          402005 non-null  object 
 5   standard_model         402005 non-null  object 
 6   vehicle_condition      402005 non-null  object 
 7   year_of_registration   368694 non-null  float64
 8   price                  402005 non-null  int64  
 9   body_type              401168 non-null  object 
 10  crossover_car_and_van  402005 non-null  bool   
 11  fuel_type              401404 non-null  object 
dtypes: bool(1), float64(2), int64(2), object(7)
memory usage: 34.1+ MB
```

Figure 2: Car sales advert data description

Here is the overview of what the dataset entails:



```
car_df.head()
```

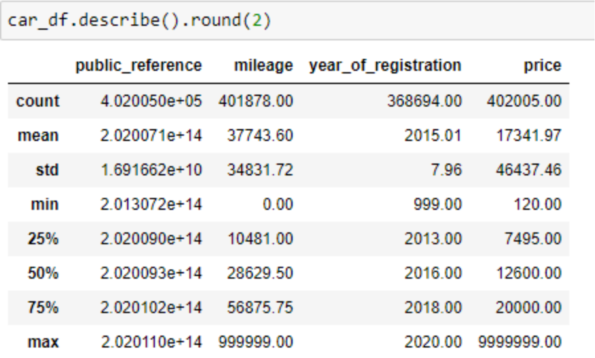
mileage	reg_code	standard_colour	standard_make	standard_model	vehicle_condition	year_of_registration	price	body_type	crossover_car_and_van	fuel_type
0.0	NaN	Grey	Volvo	XC90	NEW	NaN	73970	SUV	False	Petrol Plug-in Hybrid
08230.0	61	Blue	Jaguar	XF	USED	2011.0	7000	Saloon	False	Diesel
7800.0	17	Grey	SKODA	Yeti	USED	2017.0	14000	SUV	False	Petrol
45000.0	16	Brown	Vauxhall	Mokka	USED	2016.0	7995	Hatchback	False	Diesel
64000.0	64	Grey	Land Rover	Range Rover Sport	USED	2015.0	26995	SUV	False	Diesel

Figure 3: First 5 rows of the car sales advert dataset

1.2. Analysis of Distributions

In the figure below, the statistical distribution of the numerical feature is produced, the mean, median and standard deviation of the features are calculated, the maximum and the minimum value also help to understand the dataset range, the data percentile were also calculated, this gives an overview of the dataset.

The price of the cars has a mean value of 17,341 pounds with a standard deviation of 46,437. The price minimum value is 120 pounds. The 25%, 50%, and 75% percentiles are 7495, 12600, and 20000 respectively. The distribution for other numerical values such as mileage and year of registration were also calculated (see figure 3)

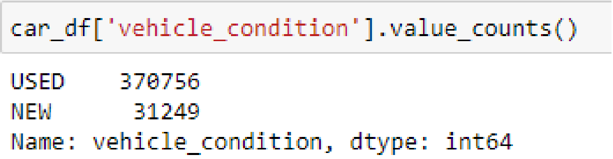


```
car_df.describe().round(2)
```

	public_reference	mileage	year_of_registration	price
count	4.020050e+05	401878.00	368694.00	402005.00
mean	2.020071e+14	37743.60	2015.01	17341.97
std	1.691662e+10	34831.72	7.96	46437.46
min	2.013072e+14	0.00	999.00	120.00
25%	2.020090e+14	10481.00	2013.00	7495.00
50%	2.020093e+14	28629.50	2016.00	12600.00
75%	2.020102e+14	56875.75	2018.00	20000.00
max	2.020110e+14	999999.00	2020.00	9999999.00

Figure 4: Distribution of car sales advert

The distribution of categorical features was also carried out. from figure 4, it can be seen that the vehicle condition of the cars was divided into new and used cars, the used cars are mostly advertised compare to the new vehicles. We can slightly understand that the seller deals or specialize in selling used vehicles more compare to the new vehicle

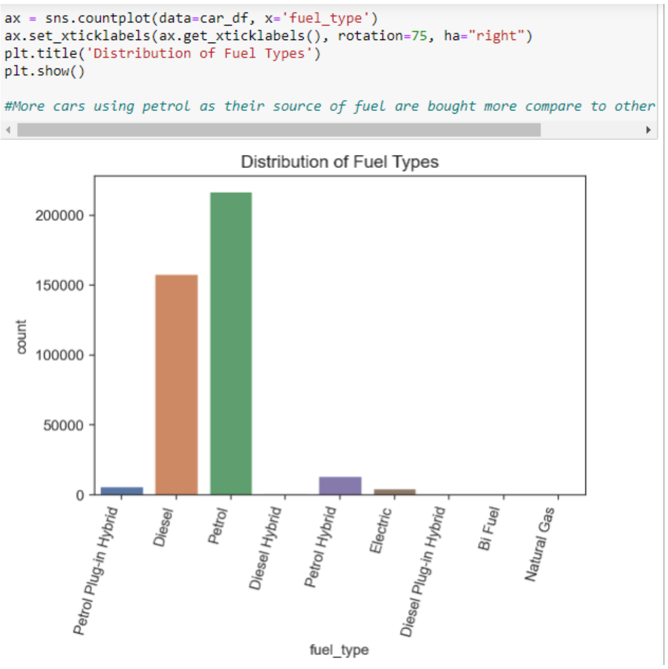


```
car_df['vehicle_condition'].value_counts()
```

```
USED    370756
NEW      31249
Name: vehicle_condition, dtype: int64
```

Figure 5: Distribution of categorical features

The vehicles most advertised are vehicles that uses petrol as their source of fuel which is above 200,000 in number followed by diesel vehicles. Most other vehicles using other source of fuel such as petrol plug-in hybrid, diesel hybrid, petrol hybrid, electric, bi-fuel, natural gas, diesel plug-in hybrid are less (figure 5).



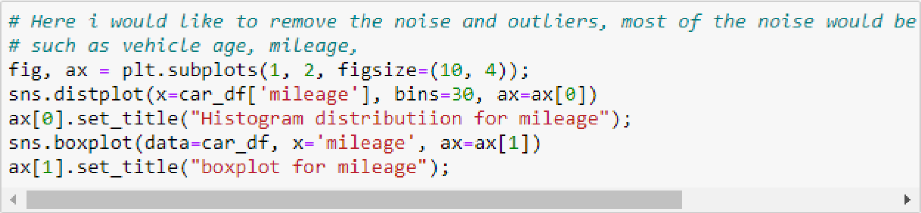
```
ax = sns.countplot(data=car_df, x='fuel_type')
ax.set_xticklabels(ax.get_xticklabels(), rotation=75, ha="right")
plt.title('Distribution of Fuel Types')
plt.show()

#More cars using petrol as their source of fuel are bought more compare to other
```

Figure 6: Distribution of car fuel type

Based on further analysis, the most advertised type of car the dealer has are vehicles whose body type are hatchback, followed by SUV with a count of 167,315 and 115,872 respectively.

Considering the distribution of mileage using boxplot and histogram, it is noted that the columns contain outliers. the box plot shows that the outliers started from about 120,000



```
# Here i would like to remove the noise and outliers, most of the noise would be
# such as vehicle age, mileage,
fig, ax = plt.subplots(1, 2, figsize=(10, 4));
sns.distplot(x=car_df['mileage'], bins=30, ax=ax[0])
ax[0].set_title("Histogram distributiion for mileage");
sns.boxplot(data=car_df, x='mileage', ax=ax[1])
ax[1].set_title("boxplot for mileage");
```

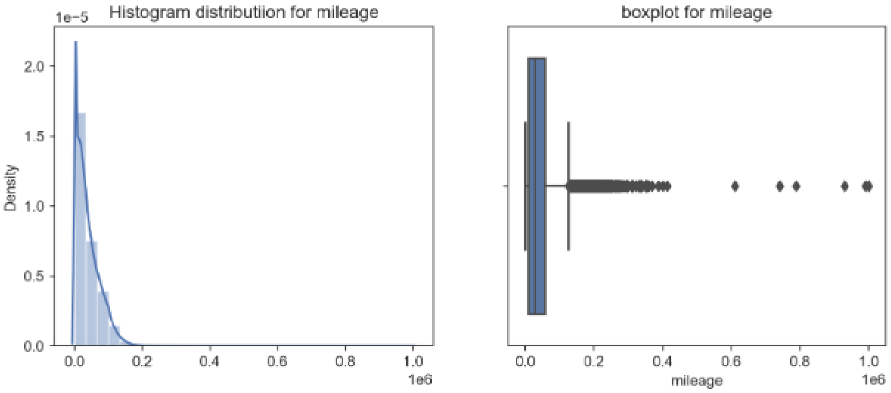
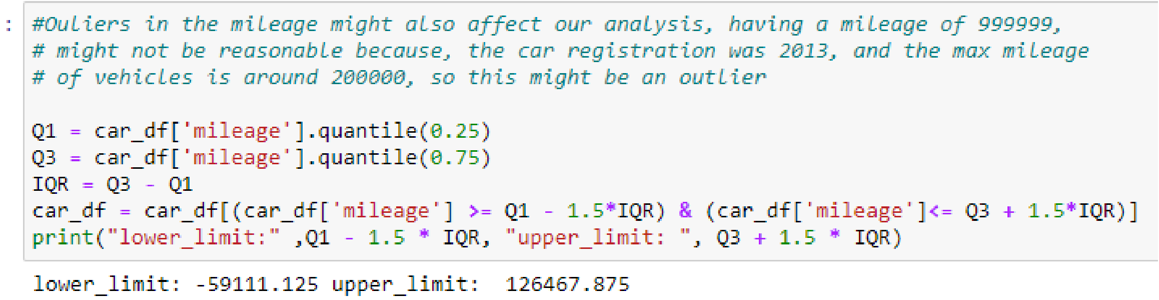


Figure 7: Distribution of mileage showing outliers

2.0. DATA PRE-PROCESSING

2.1. Data Cleaning (e.g., dealing with incorrect values, outliers)

At this stage, the missing values in each column will be treated based on suitable processes. Using the interquartile evaluation, the lower and the upper limit is set, therefore value below and above the limits are considered as outliers or noise and they are then removed



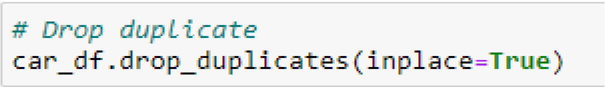
```
#Ouliers in the mileage might also affect our analysis, having a mileage of 999999,
# might not be reasonable because, the car registration was 2013, and the max mileage
# of vehicles is around 200000, so this might be an outlier

Q1 = car_df['mileage'].quantile(0.25)
Q3 = car_df['mileage'].quantile(0.75)
IQR = Q3 - Q1
car_df = car_df[(car_df['mileage'] >= Q1 - 1.5*IQR) & (car_df['mileage']<= Q3 + 1.5*IQR)]
print("lower_limit:" ,Q1 - 1.5 * IQR, "upper_limit: ", Q3 + 1.5 * IQR)
```

```
lower_limit: -59111.125 upper_limit:  126467.875
```

Figure 8: Removal of outliers in mileage column

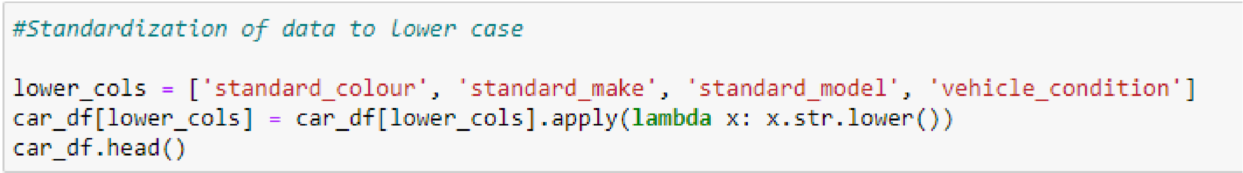
It was noticed that some values were repeated twice i.e. they were duplicated, they had multiple entry, so there is a need that one of the two rows is removed



```
# Drop duplicate
car_df.drop_duplicates(inplace=True)
```

Figure 9: Dropping of duplicated rows

In the dataset, the letter case is irregular, it is nice when all character has same letter case, the code below is used to convert all categorical features letter cases to lower case



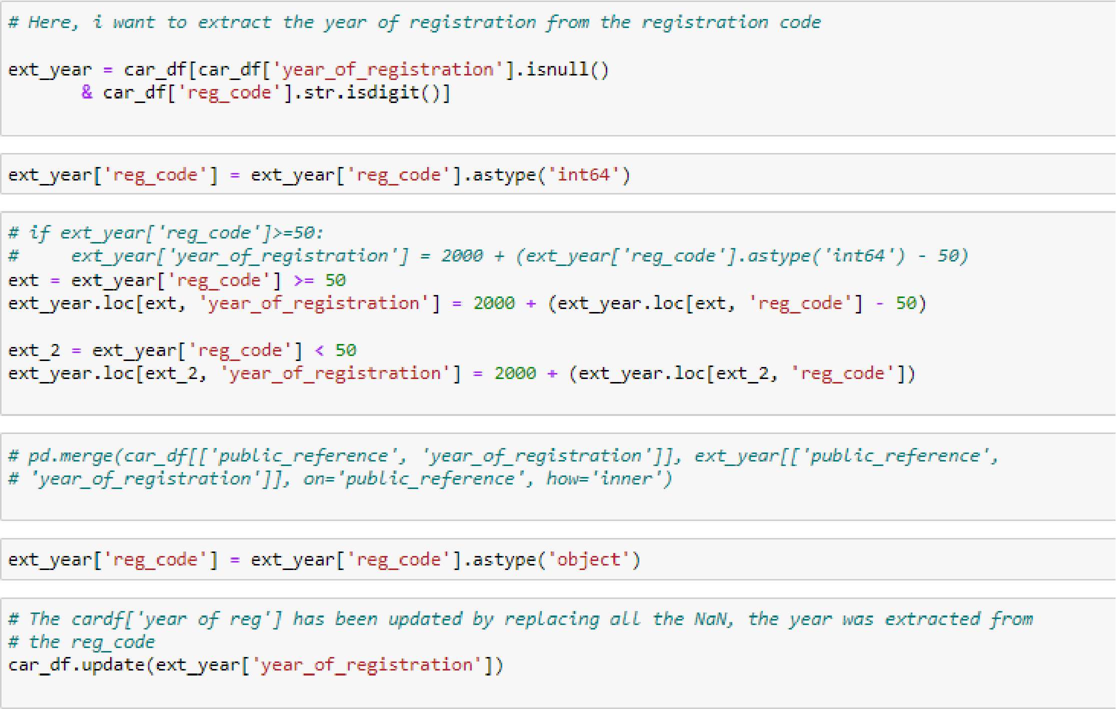
```
#Standardization of data to lower case

lower_cols = ['standard_colour', 'standard_make', 'standard_model', 'vehicle_condition']
car_df[lower_cols] = car_df[lower_cols].apply(lambda x: x.str.lower())
car_df.head()
```

Figure 10: Conversion of letter cases to lower case

2.2. Feature Engineering (e.g., deriving informative features)

Checking the number of missing values, it is seen that registration code and the year of registration have the highest number of missing values. Further check and readings revealed that the registration code was extracted from the year of registration, so therefore we can systematically derive the year of registration from the registration code and vice versa.



```
# Here, i want to extract the year of registration from the registration code

ext_year = car_df[car_df['year_of_registration'].isnull()
       & car_df['reg_code'].str.isdigit()]
```

```
ext_year['reg_code'] = ext_year['reg_code'].astype('int64')
```

```
# if ext_year['reg_code']>=50:
#     ext_year['year_of_registration'] = 2000 + (ext_year['reg_code'].astype('int64') - 50)
ext = ext_year['reg_code'] >= 50
ext_year.loc[ext, 'year_of_registration'] = 2000 + (ext_year.loc[ext, 'reg_code'] - 50)

ext_2 = ext_year['reg_code'] < 50
ext_year.loc[ext_2, 'year_of_registration'] = 2000 + (ext_year.loc[ext_2, 'reg_code'])
```

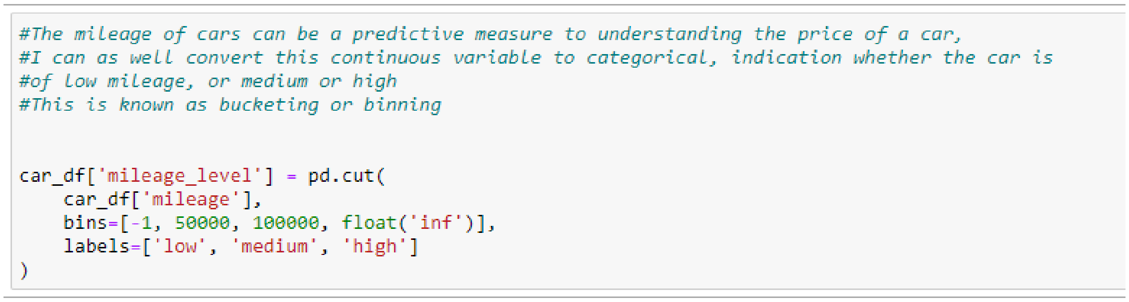
```
# pd.merge(car_df[['public_reference', 'year_of_registration']], ext_year[['public_reference',
# 'year_of_registration']], on='public_reference', how='inner')
```

```
ext_year['reg_code'] = ext_year['reg_code'].astype('object')
```

```
# The cardf['year of reg'] has been updated by replacing all the NaN, the year was extracted from
# the reg_code
car_df.update(ext_year['year_of_registration'])
```

Figure 11: Extraction of year of registration from the registration code

For better understanding of mileage, a new column named “Mileage Level” is created, it categorizes mileage into three levels: low (0-50,000), medium (50,001-100,000), and high (100,001 and above)

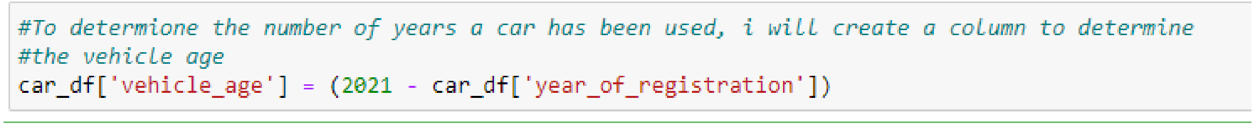


```
#The mileage of cars can be a predictive measure to understanding the price of a car,
#I can as well convert this continuous variable to categorical, indication whether the car is
#of low mileage, or medium or high
#This is known as bucketing or binning


car_df['mileage_level'] = pd.cut(
    car_df['mileage'],
    bins=[-1, 50000, 100000, float('inf')],
    labels=['low', 'medium', 'high']
)
```

Figure 12: Formation of mileage level column

Also, a new data column can be formed, which will give an informative expression, in this case, the vehicle age is derived from the year of registration.

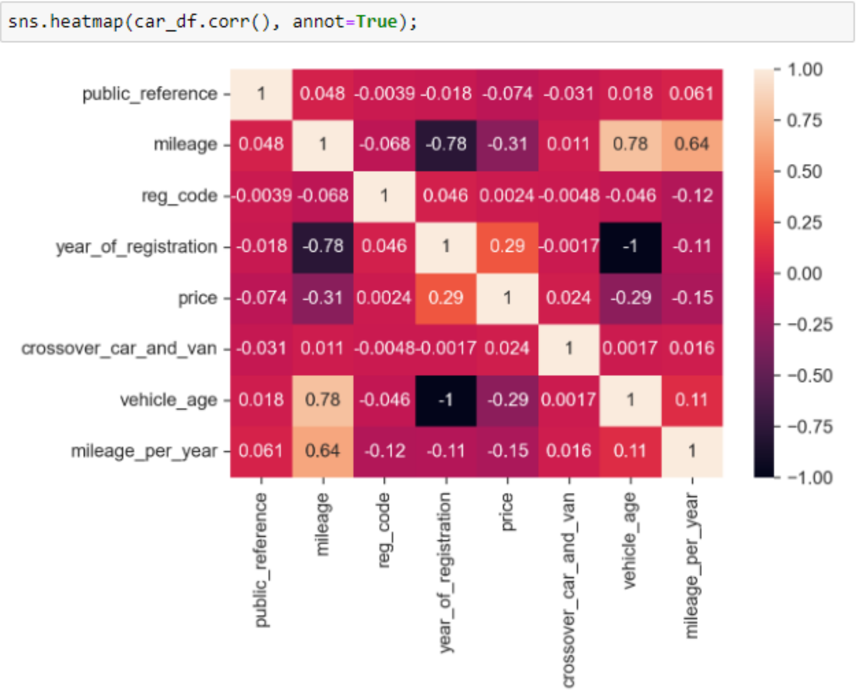


```
#To determione the number of years a car has been used, i will create a column to determine
#the vehicle age
car_df['vehicle_age'] = (2021 - car_df['year_of_registration'])
```

Figure 13: Creation of vehicle age column

2.3. Subsetting (e.g., feature selection and row sampling)

The correlation between the numerical features helps to understand the relationship between the features and aids feature selection. The year of registration and mileage had a stronger positive correlation, i.e. the older the car the higher the mileage. The mileage has a negative correlation with the car price, the higher the mileage the lower the price and most cars with higher mileage are used vehicles, from the analysis, new cars has their mileage less than 100. Crossover car and van, public reference, and reg code feature has a very low correlation when compared with other features

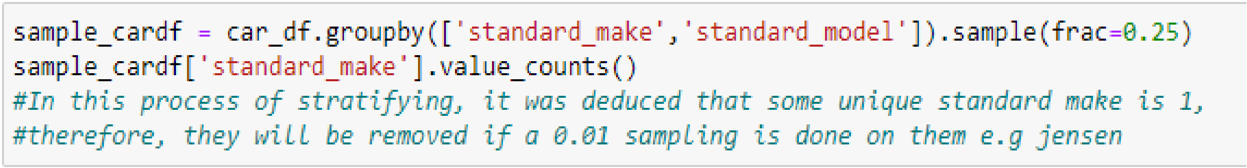


```
sns.heatmap(car_df.corr(), annot=True);
```

	public_reference	mileage	reg_code	year_of_registration	price	crossover_car_and_van	vehicle_age	mileage_per_year
public_reference	1	0.048	-0.0039	-0.018	-0.074	-0.031	0.018	0.061
mileage	0.048	1	-0.068	-0.78	-0.31	0.011	0.78	0.64
reg_code	-0.0039	-0.068	1	0.046	0.0024	-0.0048	-0.046	-0.12
year_of_registration	-0.018	-0.78	0.046	1	0.29	-0.0017	-1	-0.11
price	-0.074	-0.31	0.0024	0.29	1	0.024	-0.29	-0.15
crossover_car_and_van	-0.031	0.011	-0.0048	-0.0017	0.024	1	0.0017	0.016
vehicle_age	0.018	0.78	-0.046	-1	-0.29	0.0017	1	0.11
mileage_per_year	0.061	0.64	-0.12	-0.11	-0.15	0.016	0.11	1

Figure 14: Correlation matrix

To gain an insightful information about the data, it is preferred that a subset of the data is taken. The selection will be based on the vehicle manufacturer and models because each maker has different models with different prices, a subset in this format will give a general representation of the data.



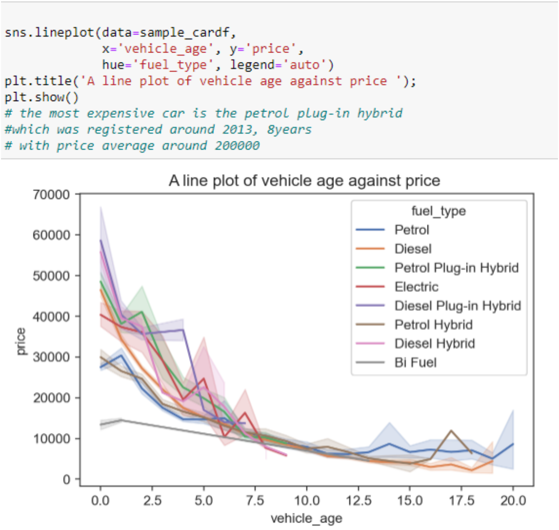
```
sample_cardf = car_df.groupby(['standard_make','standard_model']).sample(frac=0.25)
sample_cardf['standard_make'].value_counts()
#In this process of stratifying, it was deduced that some unique standard make is 1,
#therefore, they will be removed if a 0.01 sampling is done on them e.g jensen
```

Figure 15: Stratified sampling based on standard make and model

3.0. ANALYSIS OF ASSOCIATIONS AND GROUP DIFFERENCES

3.1. Quantitative-Quantitative

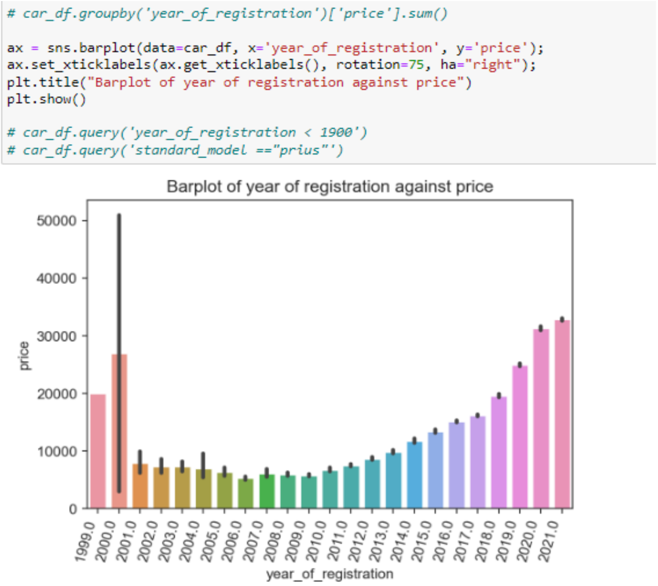
The line plot shows the price of the vehicle as the vehicle age increases, there is a clear conclusion that the price of the car decrease as the age of vehicle increases. This is also categorized according to their fuel type; diesel plug-in hybrid vehicles are more expensive even when they are new at 0 years.



```
sns.lineplot(data=sample_cardf,
             x='vehicle_age', y='price',
             hue='fuel_type', legend='auto')
plt.title('A line plot of vehicle age against price ');
plt.show()
# the most expensive car is the petrol plug-in hybrid
#which was registered around 2013, 8years
# with price average around 200000
```

Figure 16: A line plot of vehicle age against price, categorized by fuel type

The figure below shows the average price in each year, from 1999 to 2009, the average price reduces until 2009 and then started increasing in 2010 till 2021. It can be deduced that the car price also increases as the year goes by, factor causing this might be inflation. In the year 2006, the average car price is low.



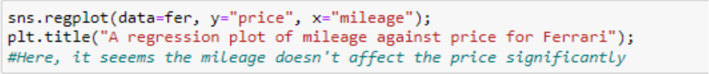
```
# car_df.groupby('year_of_registration')['price'].sum()

ax = sns.barplot(data=car_df, x='year_of_registration', y='price');
ax.set_xticklabels(ax.get_xticklabels(), rotation=75, ha="right");
plt.title("Barplot of year of registration against price")
plt.show()

# car_df.query('year_of_registration < 1900')
# car_df.query('standard_model =="prius"')
```

Figure 17: A bar plot of year of registration against price

Considering the ferrari car brand, being one of the brand known for its luxury, high mileage appears to have a lesser significance on the price, Ferrari cars seem to retain their value well, as the price does not significantly decrease with an increase in mileage



```
sns.regplot(data=fer, y="price", x="mileage");
plt.title("A regression plot of mileage against price for Ferrari");
#Here, it seeems the mileage doesn't affect the price significantly
```

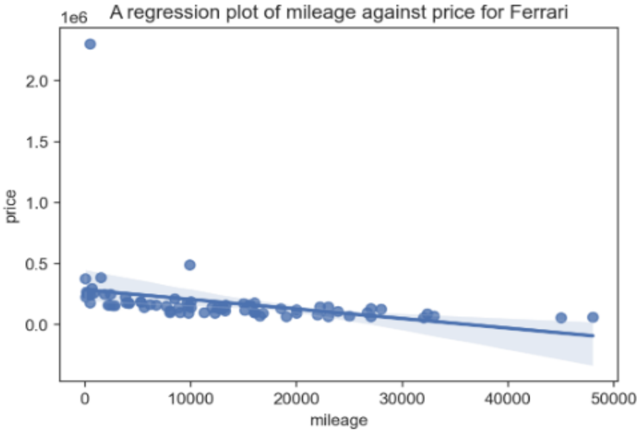
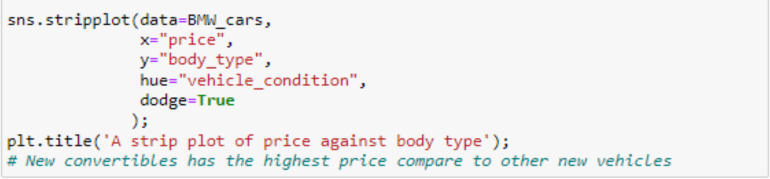


Figure 18: A regplot of mileage against price for Ferrari cars

3.2. Quantitative-Categorical

In the figure below representing data for BMW vehicles, used vehicles with MPV body types are reported to be more expensive, compared to other body types. Additionally, there are no new vehicles with a price less than 20,000 pounds, new hatchback vehicles are least expensive of all the new cars advertised



```
sns.stripplot(data=BMW_cars,
              x="price",
              y="body_type",
              hue="vehicle_condition",
              dodge=True
             );
plt.title('A strip plot of price against body type');
# New convertibles has the highest price compare to other new vehicles
```

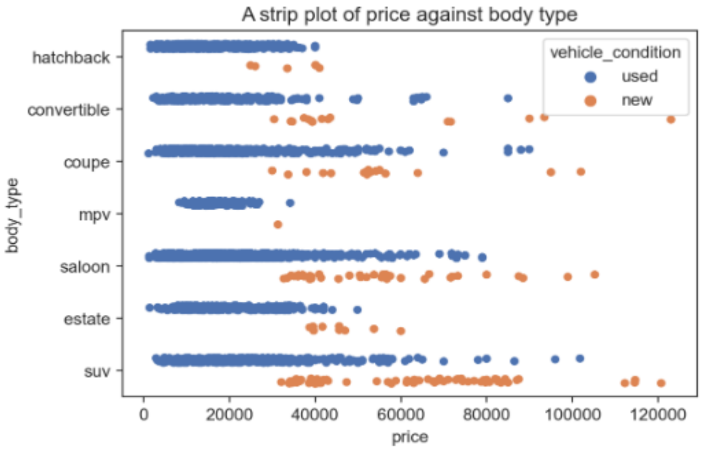
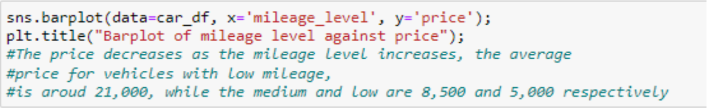


Figure 19: A stripplot of price against price for BMW vehicles

To understand how vehicles lose their price values when the mileage increases, the low mileage are (0 to 50000), medium (50,001 to 100000), and high (150,000 and above). The average price for cars with low mileage is approximately 21,000 pounds, for medium mileage it is around 8,000 pounds, and for high mileage, it is approximately 5,000 pounds. This indicates that once a car is used and its mileage exceeds 50,000, the vehicle is likely to experience a depreciation of over 50% in its market value.



```
sns.barplot(data=car_df, x='mileage_level', y='price');
plt.title("Barplot of mileage level against price");
#The price decreases as the mileage level increases, the average
#price for vehicles with low mileage,
#is aroud 21,000, while the medium and low are 8,500 and 5,000 respectively
```

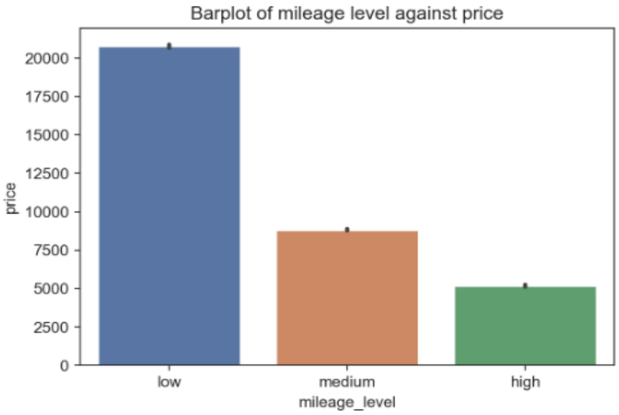
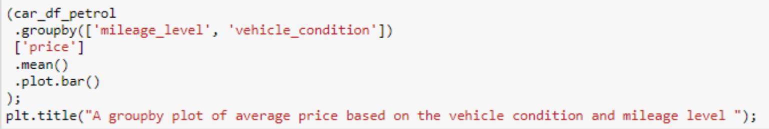


Figure 20: A bar plot of mileage level against price for all vehicles

Taking a subset of cars that uses petrol as their fuel source, the vehicles that have low mileage and are new are about 8,000 pounds more expensive than used petrol vehicles with low mileage, the used petrol vehicles with high mileages are very cheap with price about 3,000 pounds



```
(car_df_petrol
 .groupby(['mileage_level', 'vehicle_condition'])
 ['price']
 .mean()
 .plot.bar()
);
plt.title("A groupby plot of average price based on the vehicle condition and mileage level ");
```

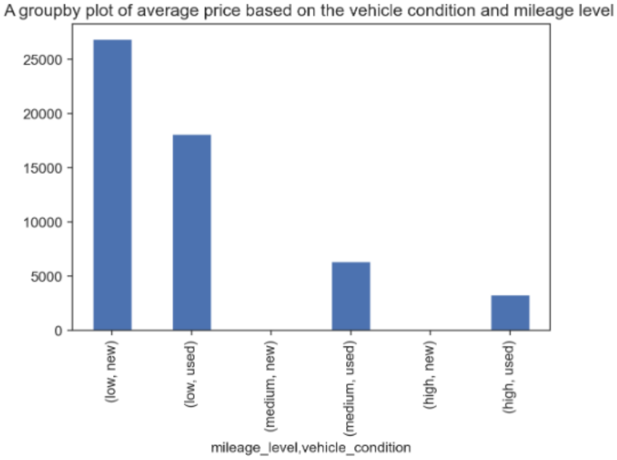
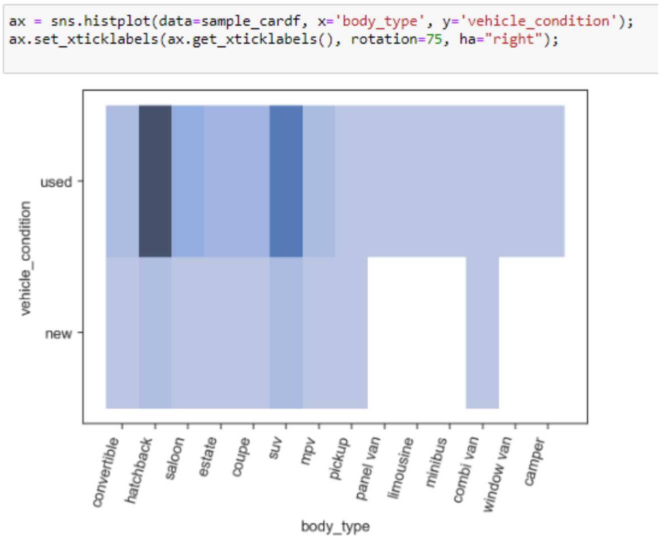


Figure 21: A barplot of mileage level and vehicle condition against price for petrol vehicles

3.3. Categorical-Categorical

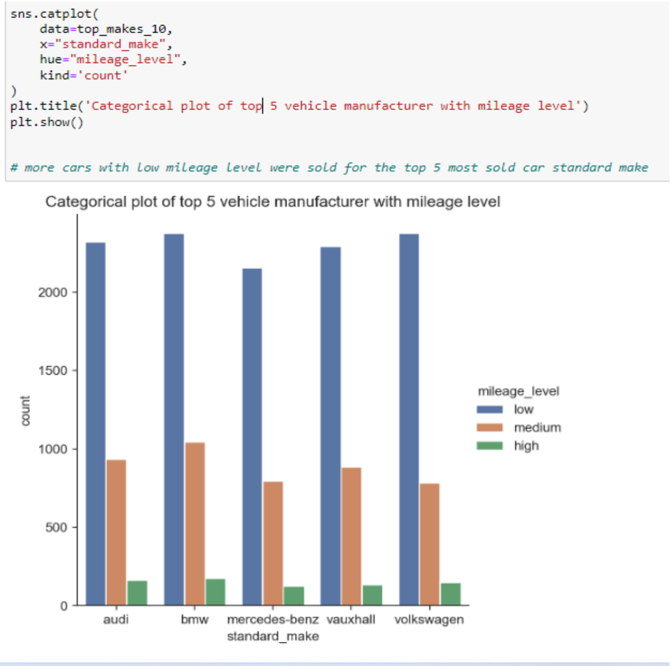
In the figure below, more used hatchback vehicles were advertised followed by used SUV vehicles. No new cars whose body type is panel van, limousine, minibus, window van and camper were advertised, only their used vehicle condition. The predominant vehicle body types being purchased are hatchbacks and SUVs



```
ax = sns.histplot(data=sample_cardf, x='body_type', y='vehicle_condition');
ax.set_xticklabels(ax.get_xticklabels(), rotation=75, ha="right");
```

Figure 22: A histplot of body type against vehicle condition

The top five most purchased vehicle manufacturer have almost the same number of purchases across the 3 mileage levels. The difference in their count are minimal, just a little difference. BMW vehicles are more likely the most advertised vehicles across the three-mileage level than all other top five vehicle makes.



```
sns.catplot(
    data=top_makes_10,
    x="standard_make",
    hue="mileage_level",
    kind='count'
)
plt.title('Categorical plot of top 5 vehicle manufacturer with mileage level')
plt.show()

# more cars with low mileage level were sold for the top 5 most sold car standard make
```

Figure 23: A categorical plot of top 5 vehicle manufacturer with their mileage level