ANALISIS DATASET MOBIL BEKAS DI INDIA

DATA SCIENCE ACADEMY COMPFEST 12

Oleh:

TIM - PK2MABA

Anggota :

Mohammad Fachryza Purwanto Putra

Julia Ferlin

Fawwaz Roja Mahardhika

TIM - PK2MABA

2020

Latar belakang

Data Science merupakan gabungan irisan dari berbagai ilmu pengetahuan mulai dari matematika, statistika serta ilmu komputer untuk tujuan melakukan analisa data dari suatu himpunan data mulai dari skala kecil sampai besar dengan mengaplikasikan algoritma tertentu untuk tujuan menggali data atau mendapatkan pola data serta dapat melakukan prediksi data untuk membantu dalam melakukan pengambilan keputusan dan dapat digunakan untuk membuat sistem cerdas yang dapat belajar dengan sendirinya *(machine learning).* Teknologi *big data* pada era revolusi industri 4.0 terus mengalami perkembangan, sehingga mengharuskan manusia untuk dapat mengolah sebuah data secara sedemikian rupa agar dapat bermanfaat untuk manusia dalam menjalankan proses kehidupan sehari-sehari nya.

Kaggle adalah salah satu situs dan *platform* pembelajaran yang terkenal di dunia *Data Science* yang terdiri dari lebih dari 6000 *dataset* yang dapat diunduh dalam format CSV. *Dataset* ini diberikan oleh suatu perusahaan, dengan suatu deskripsi masalah tertentu. Dengan terdapatnya kompetisi membuat model terbaik untuk menganalisa dan memprediksi suatu dataset, dapat membantu para ilmuwan data pemula untuk mengembangkan ilmu *Data Science* mereka.

Berikut dibawah ini merupakan hasil analisa, visualisasi data serta pembahasan dari salah satu *dataset* yang terdapat pada Kaggle yaitu dataset mengenai mobil bekas yang terdapat di India yang meliputi 6019 baris dan 12 kolom. Beberapa kolom tersebut ialah, *Name, Location, Year, Kilometers Driven, Fuel Type, Transmission, Owner Type, Mileage, Engine, Power, Seats, dan Price.* Hasil analisa dan pembahasan dari dataset ini digunakan untuk mengikuti proses seleksi *Data Science Academy* yang diselenggarakan oleh COMPFEST 12.

Jawaban Soal

1. Merek mobil apa saja yang tersedia dan ada berapa banyak mobil untuk tiap merek tersebut?

   Jawab :

   Pertama kita harus membaca data CSV (user_car_data.csv) terlebih dahulu.

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

df = pd.read_csv('~/python/Compfest/Seleksi/used_car_data.csv')
df.head()
```

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns

        df = pd.read_csv('~/python/Compfest/Seleksi/used_car_data.csv')
        df.head()
```

Out[1]:

	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price
0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5.0	1.75
1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.0	12.50
2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5.0	4.50
3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7.0	6.00
4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5.0	17.74

Setelah itu, kita akan mengisi data kosong terlebih dahulu menggunakan fungsi fillna(), kemudian dilanjutkan dengan menggunakan fungsi describe() untuk menunjukkan rangkuman statisitik seperti yang terlihat pada gambar output di bawah ini.

```
df2 = df.fillna(0)
df2.describe(include='all')
```

```
In [51]: df2 = df.fillna(0)
         df2.describe(include='all')
```

Out[51]:

	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price
count	6019	6019	6019.000000	6.019000e+03	6019	6019	6019	6019	6019	6019	6019.000000	6019.000000
unique	1876	11	NaN	NaN	5	2	4	443	147	373	NaN	NaN
top	Mahindra XUV500 W8 2WD	Mumbai	NaN	NaN	Diesel	Manual	First	18.9 kmpl	1197 CC	74 bhp	NaN	NaN
freq	49	790	NaN	NaN	3205	4299	4929	172	606	235	NaN	NaN
mean	NaN	NaN	2013.358199	5.873838e+04	NaN	NaN	NaN	NaN	NaN	NaN	5.241901	9.479468
std	NaN	NaN	3.269742	9.126884e+04	NaN	NaN	NaN	NaN	NaN	NaN	0.918025	11.187917
min	NaN	NaN	1998.000000	1.710000e+02	NaN	NaN	NaN	NaN	NaN	NaN	0.000000	0.440000
25%	NaN	NaN	2011.000000	3.400000e+04	NaN	NaN	NaN	NaN	NaN	NaN	5.000000	3.500000
50%	NaN	NaN	2014.000000	5.300000e+04	NaN	NaN	NaN	NaN	NaN	NaN	5.000000	5.640000
75%	NaN	NaN	2016.000000	7.300000e+04	NaN	NaN	NaN	NaN	NaN	NaN	5.000000	9.950000
max	NaN	NaN	2019.000000	6.500000e+06	NaN	NaN	NaN	NaN	NaN	NaN	10.000000	160.000000

Dilanjutkan dengan pendefinisian fungsi extract_merk menggunakan kata kunci def dengan parameter fungsi berupa text. Fungsi ini digunakan untuk memisahkan kata pertama pada value yang dimiliki oleh column Name untuk menjadi column baru yaitu column Merk.

```
def extract_merk(text):
  merk = text.split(' ', 1)[0]
  merk = merk.lower()
  merk = merk.capitalize()
  return merk

df2['Merk'] = df2['Name'].apply(lambda x: extract_merk(x))
df2.head()
```

```
In [52]: def extract_merk(text):
           merk = text.split(' ', 1)[0]
           merk = merk.lower()
           merk = merk.capitalize()
           return merk

         df2['Merk'] = df2['Name'].apply(lambda x: extract_merk(x))
         df2.head()
```

Out[52]:

	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price	Merk
0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5.0	1.75	Maruti
1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.0	12.50	Hyundai
2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5.0	4.50	Honda
3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7.0	6.00	Maruti
4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5.0	17.74	Audi

Selanjutnya kita akan mencetak keluaran menggunakan fungsi print(), keluaran pertama yaitu untuk menampilkan unique value terkait ”Merk mobil yang tersedia:” pada column Merk. Kemudian keluaran kedua yaitu untuk menampilkan jumlah total unique value pada column Merk atau ”Jumlah merk mobil yang tersedia:”.

```
#Merek mobil apa saja yang tersedia
print("Merk mobil yang tersedia:\n",df2.Merk.unique())

#Jumlah seluruh merek mobil
print("\nJumlah merk mobil yang tersedia:\n",df2.Merk.nunique())
```

```
In [54]: #Merek mobil apa saja yang tersedia
         print("Merk mobil yang tersedia:\n",df2.Merk.unique())

         #Jumlah seluruh merek mobil
         print("\nJumlah merk mobil yang tersedia:\n",df2.Merk.nunique())

Merk mobil yang tersedia:
 ['Maruti' 'Hyundai' 'Honda' 'Audi' 'Nissan' 'Toyota' 'Volkswagen' 'Tata'
 'Land' 'Mitsubishi' 'Renault' 'Mercedes-benz' 'Bmw' 'Mahindra' 'Ford'
 'Porsche' 'Datsun' 'Jaguar' 'Volvo' 'Chevrolet' 'Skoda' 'Mini' 'Fiat'
 'Jeep' 'Smart' 'Ambassador' 'Isuzu' 'Force' 'Bentley' 'Lamborghini']

Jumlah merk mobil yang tersedia:
 30
```

Merk mobil yang tersedia: ['Maruti' 'Hyundai' 'Honda' 'Audi' 'Nissan' 'Toyota' 'Volkswagen' 'Tata' 'Land' 'Mitsubishi' 'Renault' 'Mercedes-benz' 'Bmw' 'Mahindra' 'Ford' 'Porsche' 'Datsun' 'Jaguar' 'Volvo' 'Chevrolet' 'Skoda' 'Mini' 'Fiat' 'Jeep' 'Smart' 'Ambassador' 'Isuzu' 'Force' 'Bentley' 'Lamborghini']

Jumlah merk mobil yang tersedia: 30

Terakhir merupakan keluaran berupa total jumlah mobil untuk setiap merk. Pada keluaran ini menggunakan fungsi value_counts() pada column Merk untuk menghitung total jumlah di setiap merk.

```
#banyak mobil untuk tiap merek
print("\nJumlah mobil di tiap merk:\n", df2['Merk'].value_counts())
```

```
In [55]: #banyak mobil untuk tiap merek
         print("\nJumlah mobil di tiap merk:\n", df2['Merk'].value_counts())
```

```
Jumlah mobil di tiap merk:
 Maruti           1211
Hyundai           1107
Honda              608
Toyota             411
Mercedes-benz      318
Volkswagen         315
Ford               300
Mahindra           272
Bmw                267
Audi               236
Tata               186
Skoda              173
Renault            145
Chevrolet          121
Nissan              91
Land                60
Jaguar              40
Fiat                28
Mitsubishi          27
```

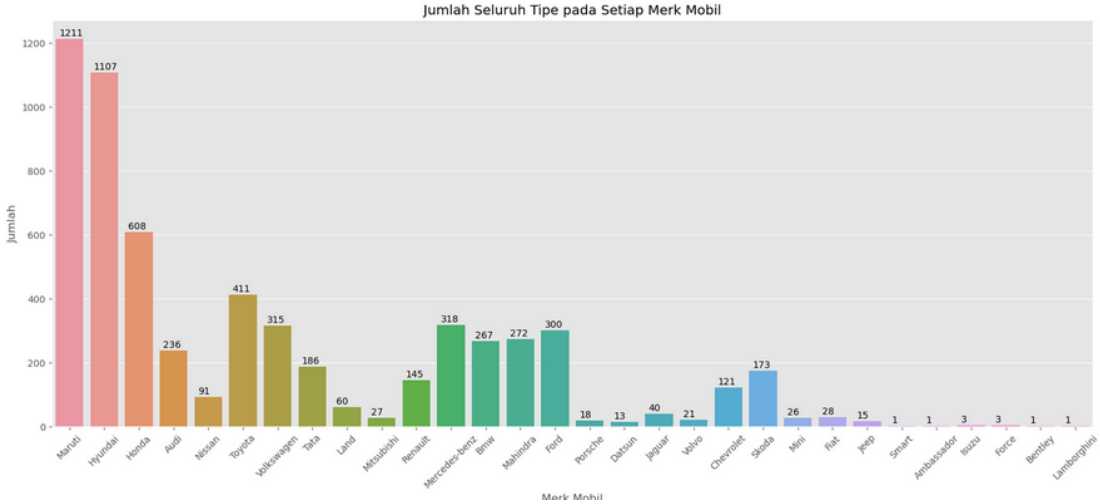
Jumlah mobil di tiap merk:

Maruti : 1211; Hyundai : 1107; Honda : 608; Toyota : 411; Mercedes-benz : 318; Volkswagen : 315; Ford : 300; Mahindra : 272; Bmw : 267; Audi : 236; Tata : 186; Skoda : 173; Renault : 145; Chevrolet : 121; Nissan : 91; Land : 60; Jaguar : 40; Fiat : 28; Mitsubishi : 27; Mini : 26; Volvo : 21; Porsche : 18; Jeep : 15; Datsun : 13; Isuzu : 3; Force : 3; Bentley : 1; Ambassador : 1; Lamborghini : 1; Smart : 1

Berikut merupakan tampilan visualisasi data menggunakan library Seaborn untuk menampilkan total jumlah mobil pada setiap merk.

```
plt.figure(figsize=(20,8))
ax = sns.countplot(x="Merk", data=df2)
plt.title('Jumlah Seluruh Tipe pada Setiap Merk Mobil')
plt.xlabel('Merk Mobil')
plt.ylabel('Jumlah')
plt.xticks(rotation=45)

for p in ax.patches:
    ax.annotate('{}'.format(p.get_height()), (p.get_x()+0.1,
p.get_height()+10))
```



2. Kota apa yang memiliki mobil bekas paling banyak?

   Dilakukan pendefinisian fungsi used_car_label untuk memberikan label Yes dan No pada value di column Owner_Type. Label ini akan menjadi column baru yang bernama used_car_label

```
def used_car_label(text):
  label = 'Yes'
  if text.lower() == 'first':
    label = 'No'
  return label

df2['used_car_label'] = df2['Owner_Type'].apply(lambda x:
used_car_label(x))
df2.head()
```

Out[97]:

	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price	Merk	used_car_label
0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5.0	1.75	Maruti	No
1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.0	12.50	Hyundai	No
2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5.0	4.50	Honda	No
3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7.0	6.00	Maruti	No
4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5.0	17.74	Audi	Yes

Berikut merupakan data keluaran mengenai jumlah mobil bekas di berbagai kota di India.

```
print('Data of used car in each city:')
df2.groupby(['Location']).used_car_label.value_counts().unstack(fill_v
alue=0).loc[:,'Yes']
```

```
Data of used car in each city:

Out[117]: Location
          Ahmedabad      38
          Bangalore     127
          Chennai       159
          Coimbatore     65
          Delhi          97
          Hyderabad      84
          Jaipur        113
          Kochi          37
          Kolkata        32
          Mumbai        137
          Pune          201
          Name: Yes, dtype: int64
```

Sedangkan untuk kota yang memiliki jumlah mobil bekas terbanyak di India yaitu **Kota Pune** dengan 201 mobil.

```
print('City with max car:')
dataset =
df2.groupby(['Location']).used_car_label.value_counts().unstack(fill_v
alue=0).loc[:,'Yes']
```

```
dataset = pd.DataFrame(dataset)
dataset.loc[dataset['Yes'].idxmax()]
```

```
City with max car:

Out[19]: Yes     201
         Name: Pune, dtype: int64
```

3. Bagaimana distribusi tahun edisi mobil-mobil bekas tersebut?

   Berikut merupakan tampilan visualisasi data menggunakan histogram yang menampilkan distribusi frekuensi mobil mobil bekas berdasarkan edisi tahun. Ditampilkan bahwa frekuensi jumlah mobil bekas kepemilikan kedua mencapai peak nya di antara tahun 2010-2011. Sedangkan untuk kepemilikian ketiga mencapai peak di sekitaran tahun 2006. Kemudian untuk kepemilikan keempat dan keatas mencapai peak di pertengahan tahun 2007.

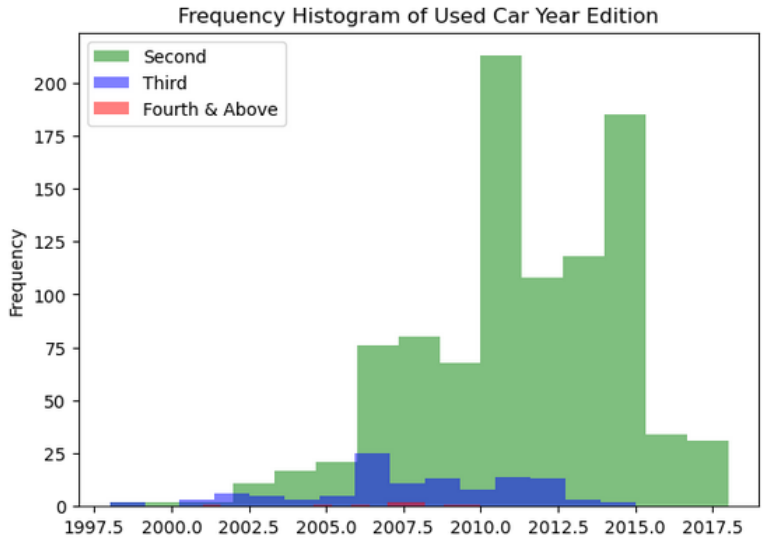
```
import matplotlib.pyplot as plt
%matplotlib inline
plt.rcParams.update({'figure.figsize':(7,5), 'figure.dpi':100})

x1 = df2.loc[df2.Owner_Type=='Second', 'Year']
x2 = df2.loc[df2.Owner_Type=='Third', 'Year']
x3 = df2.loc[df2.Owner_Type=='Fourth & Above', 'Year']

kwargs = dict(alpha=0.5, bins=15)

plt.hist(x1, **kwargs, color='g', label='Second')
plt.hist(x2, **kwargs, color='b', label='Third')
plt.hist(x3, **kwargs, color='r', label='Fourth & Above')

plt.gca().set(title='Frequency Histogram of Used Car Year Edition',
ylabel='Frequency')
plt.legend();
```

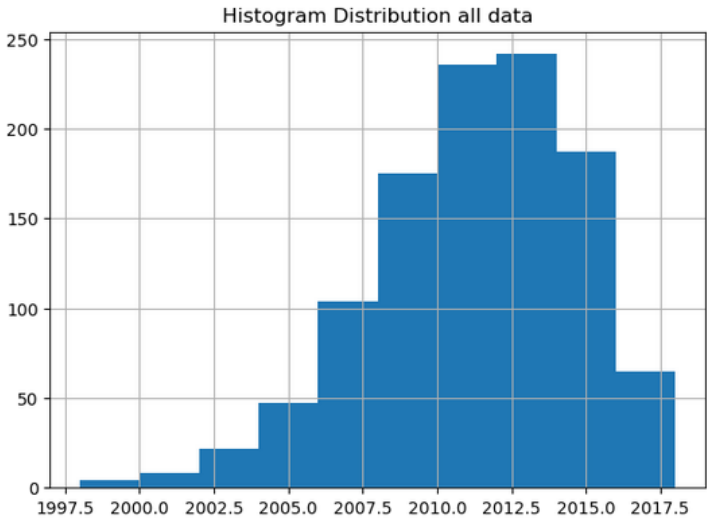


Berikut merupakan tambahan mengenai tampilan visualisasi data yang menampilkan keseluruhan data distribusi mobil mulai dari kepemilikan pertama, kedua, ketiga maupun keempat dan lebih.

```
year = df2[['Year','used_car_label']].copy()
year = year[year.used_car_label != 'No']

year.hist(column='Year')
plt.title('Histogram Distribution all data')
```

```
Out[23]: Text(0.5, 1.0, 'Histogram Distribution all data')
```



4. Berapa banyak mobil yang memiliki total jarak pemakaian di bawah 100.000 kilometer?

   Berikut merupakan jumlah banyak mobil dengan jarak total pemaikaian dibawah 100.000 kilometer yaitu **5470 mobil**.

```
jarak_mobil = df[(df["Kilometers_Driven"] < 100000)]
jarak_mobil["Kilometers_Driven"].count()
```

```
In [26]: jarak_mobil = df[(df["Kilometers_Driven"] < 100000)]
         jarak_mobil["Kilometers_Driven"].count()

Out[26]: 5470
```

5. Pada batas berapa kilometer total jarak pemakaian bisa dikategorikan sebagai rendah atau tinggi? Sertakan argumen yang mendukung jawaban.

   Untuk mengetahui jarak pemakaian tinggi atau rendah kita perlu mencari rata2 jarak pemakaian dari data yang kita miliki (mean). Karena data yang kita miliki merupakan data sample, kita perlu melakukan generalisasi agar data dapat menggambarkan populasi. Kita menggunakan confidence interval untuk dapat mendapatkan gambaran rata - rata dari data populasi dengan confidence level 95%.

Hasil :

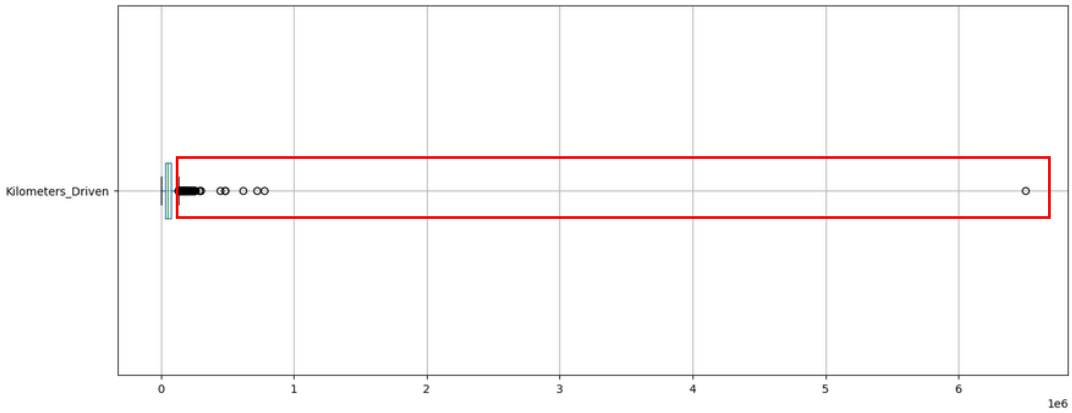
```
Mean: 53488.81055526904
Sample size: 5817
Standar Deviation of sample: 27007.073738688898
Upper Limit Value : 54182.84986577422
Lower Limit Value : 52794.77124476386
```

Dari hasil tersebut kita mendapatkan upper limit dan lower limit dari interval rata - rata data populasi. Dimana, jika kilometer total jarak lebih besar dari upper limit maka dapat dikategorikan sebagai tinggi dan jila lebih rendah dari lower limit dapat dikategorikan sebagai rendah.

6. Apakah terdapat outlier pada kolom Kilometers_Driven? Sertakan argumen yang mendukung jawaban.

   Untuk menampilkan outlier pada column Kilometers_Driven kita dapat melakukan visualisasi data menggunakan boxplot yang dapat menggambarkan bentuk distribusi data. Data outlier merupakan data yang berada diluar di antara Q1 (lower bound) dan Q3 (upper bound).

```
fig = plt.figure(figsize =(15, 6))
data = pd.DataFrame(df2['Kilometers_Driven'])
data.boxplot(vert=False)
plt.show()
```



Berdasarkan visualisasi gambar diatas, ditemukan bahwa **column Kilometers_Driven memiliki data outlier.** Karena berdasarkan boxplot yang dilihat di gambar diatas terdapat nilai yang lebih besar dari Upper Limit (Q3), nilai tersebut terindikasi sebagai outlier dari data.

Untuk membersihkan data kita dari data outlier, maka kita dapat menggunakan metode IQR dan quartile. Berikut merupakan hasil data setelah dilakukan pembersihan terhadap data outlier.

```
#clean outlier
import numpy as np
Q1 = np.quantile(df2['Kilometers_Driven'],0.25)
Q3 = np.quantile(df2['Kilometers_Driven'],0.75)
IQR = Q3 - Q1
lower_bound = Q1 -(1.5 * IQR)
upper_bound = Q3 +(1.5 * IQR)
clean_data = df2[(df2.Kilometers_Driven >= lower_bound ) &
(df2.Kilometers_Driven <= upper_bound)]
clean_data.head()
```

Out[33]:

	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price	Merk	used_car_label
0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5.0	1.75	Maruti	No
1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.0	12.50	Hyundai	No
2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5.0	4.50	Honda	No
3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7.0	6.00	Maruti	No
4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5.0	17.74	Audi	Yes

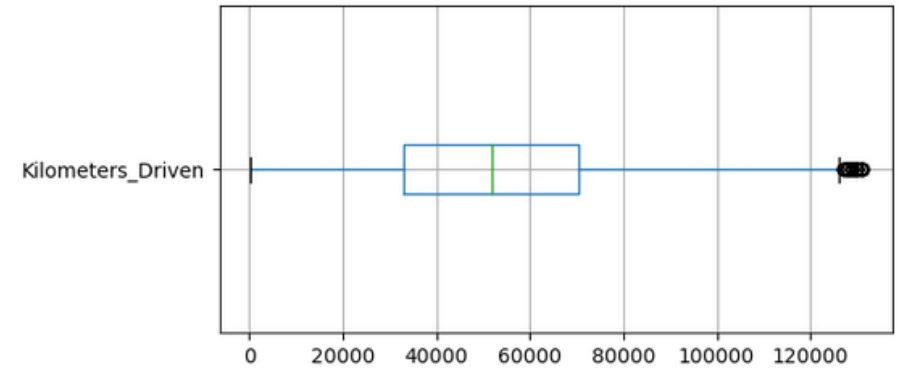
Dengan menggunakan fungsi info pada dataframe clean_data, maka akan ditampilkan keseluruhan informasi yang terletak pada dataframe clean_data. Mulai dari index, column, jumlah value, dan tipe data pada setiap column.

```
clean_data.info()
```

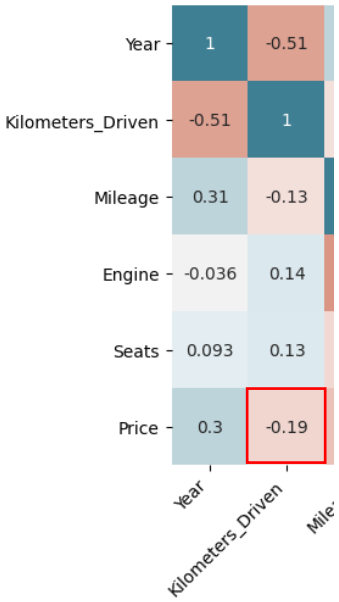
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5817 entries, 0 to 6018
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Name               5817 non-null   object
 1   Location           5817 non-null   object
 2   Year               5817 non-null   int64
 3   Kilometers_Driven  5817 non-null   int64
 4   Fuel_Type          5817 non-null   object
 5   Transmission       5817 non-null   object
 6   Owner_Type         5817 non-null   object
 7   Mileage            5817 non-null   object
 8   Engine             5817 non-null   object
 9   Power              5817 non-null   object
 10  Seats              5817 non-null   float64
 11  Price              5817 non-null   float64
 12  Merk               5817 non-null   object
 13  used_car_label     5817 non-null   object
dtypes: float64(2), int64(2), object(10)
memory usage: 681.7+ KB
```

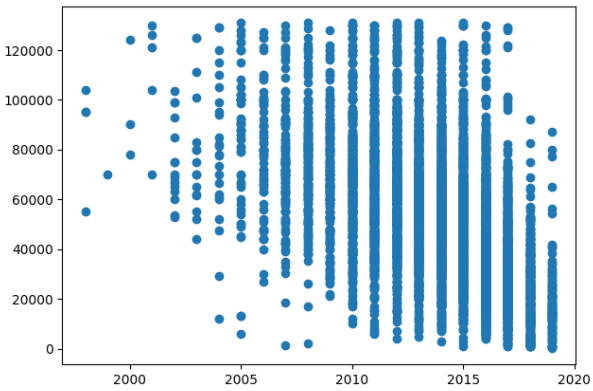
Berikut merupakan hasil tampilan visualisasi data dengan menggunakan boxplot setelah dilakukannya proses pembersihan terhadap data outlier.

```
fig = plt.figure(figsize =(6, 3))
data = pd.DataFrame(clean_data['Kilometers_Driven'])
data.boxplot(vert=False)
plt.show()
```



7. Apakah tahun pembuatan mobil berpengaruh terhadap total jarak pemakaian? Sertakan argumen yang mendukung jawaban.





Dari distribusi data pada gambar diatas tidak terlihat pengaruh yang kuat antara tahun pembuatan dan jarak pemakaian. Didukung juga dengan heatmap yang menunjukkan nilai korelasi antara tahun dan kilometers driven hanya sebesar 19% (Korelasi negatif) dimana bisa kita simpulkan bahwa tahun tidak berpengaruh terhadap total jarak pemakaian.

8. Berapa banyak mobil yang merupakan kepemilikan ketiga atau lebih?

   Pertama kita harus memisahkan data terlebih dahulu berdasarkan column Owner_Type dan Location. Serta dengan melakukan agregasi count pada column Location. Maka setelah itu akan ditampilkan column baru bernama Location yang selanjutnya akan diberikan nama Jumlah. Kemudian akan dijalankan fungsi pivot_table untuk menjadikan Location sebagai index. Hasilnya dapat dilihat pada gambar dibawah.

```
df_bekas = df.groupby(["Owner_Type","Location"]).agg({"Location":
"count"})
df_bekas = df_bekas.rename(columns ={"Location": "Jumlah"})
df_reset = df_bekas.reset_index()
bekas = df_reset.pivot_table(index="Location", columns="Owner_Type",
values="Jumlah", fill_value=0)
bekas.head()
```

Out[99]:

Owner_Type	First	Fourth & Above	Second	Third
Location				
Ahmedabad	186	0	38	0
Bangalore	231	1	114	12
Chennai	335	2	121	36
Coimbatore	571	1	63	1
Delhi	457	0	94	3

   Dikarenakan data yang dicari yaitu data banyak mobil yang merupakan kepemilikan ketiga atau lebih, maka akan dilakukan drop pada column First dan Second. Selanjutnya akan dilakukan penjumlahan pada jumlah mobil bekas kepemilikan ketiga atau lebih pada setiap kota dan penjumlahan total pada jumlah mobil bekas kepemilikan ketiga dari setiap kota serta penjumlahan total pada jumlah mobil bekas kepemilikan keempat dan lebih.

```
bekas3 = bekas.drop(['First','Second'], axis = 1, inplace = False)
bekas3.loc['Total per Owner_Type']= bekas3.sum()

col_list2 = list(bekas3)
bekas3['Total'] = bekas3[col_list2].sum(axis=1)
bekas3
```

Out[100]:

Owner_Type Location	Fourth & Above	Third	Total
Ahmedabad	0	0	0
Bangalore	1	12	13
Chennai	2	36	38
Coimbatore	1	1	2
Delhi	0	3	3
Hyderabad	0	0	0
Jaipur	1	15	16
Kochi	0	3	3
Kolkata	0	0	0
Mumbai	2	14	16
Pune	2	29	31
Total per Owner_Type	9	113	122

Berikut merupakan tampilan Total per Owner Type yang memiliki **total 122 mobil**, dengan hasil penjumlahan dari kepemilikan ketiga sejumlah **113 mobil** dan kepemilikan keempat dan lebih sejumlah **9 mobil.**

```
bekas3.loc['Total per Owner_Type',:]
```

```
In [101]: bekas3.loc['Total per Owner_Type',:]

Out[101]: Owner_Type
          Fourth & Above      9
          Third             113
          Total             122
          Name: Total per Owner_Type, dtype: int64
```

9. Tipe bahan bakar apa yang memiliki mileage (konsumsi bahan bakar) paling hemat?

   Terlebih dahulu kita akan menggunakan fungsi fillna() untuk mengisi data kosong dengan 0 (nol). Setelah itu akan dilakukan pemisahan value yang terdapa pada column Mileage, value baru tersebut akan dimasukkan kedalam 2 column baru yaitu Mileage_f dan Satuan. Hasilnya dapat dilihat pada gambar dibawah.

```
mileage = df[['Fuel_Type','Mileage']].fillna(0)

baru = mileage["Mileage"].str.split(" ", n = 1, expand = True)
mileage["Mileage_f"]= baru[0]
mileage["Satuan"]= baru[1]
mileage.head()
```

Out[119]:

	Fuel_Type	Mileage	Mileage_f	Satuan
0	CNG	26.6 km/kg	26.6	km/kg
1	Diesel	19.67 kmpl	19.67	kmpl
2	Petrol	18.2 kmpl	18.2	kmpl
3	Diesel	20.77 kmpl	20.77	kmpl
4	Diesel	15.2 kmpl	15.2	kmpl

Setelah itu akan dibuat dataframe baru yang bernama new_mileage yang berisi column Fuel_Type dan Mileage_f dari dataframe Mileage. Kita akan isi data kosong terlebih dahulu dengan fillna(). Selanjutnya akan dilakukan perubahan tipe data column Mileage_f menjadi tipe data float yang sebelumnya bertipe data str.

```
new_mileage = mileage[['Fuel_Type','Mileage_f']]
new_mileage = new_mileage.fillna(0)
new_mileage['Mileage_f'] = new_mileage['Mileage_f'].astype(float)
```

Selanjutnya dilakukan pemisahan data berdasarkan Fuel_Type menggunakan groupby() dengan agregasi mean pada column Mileage_f. Kemudian akan dilakukan sorting value untuk menentukan tipe bahan bakar mana yang terhemat ke tipe bahan bakar yang terboros.

```
new_mileage = new_mileage.groupby(["Fuel_Type"]).agg({"Mileage_f":
"mean"})
new_mileage = new_mileage.sort_values("Mileage_f", ascending = False)
new_mileage
```

Out[121]:

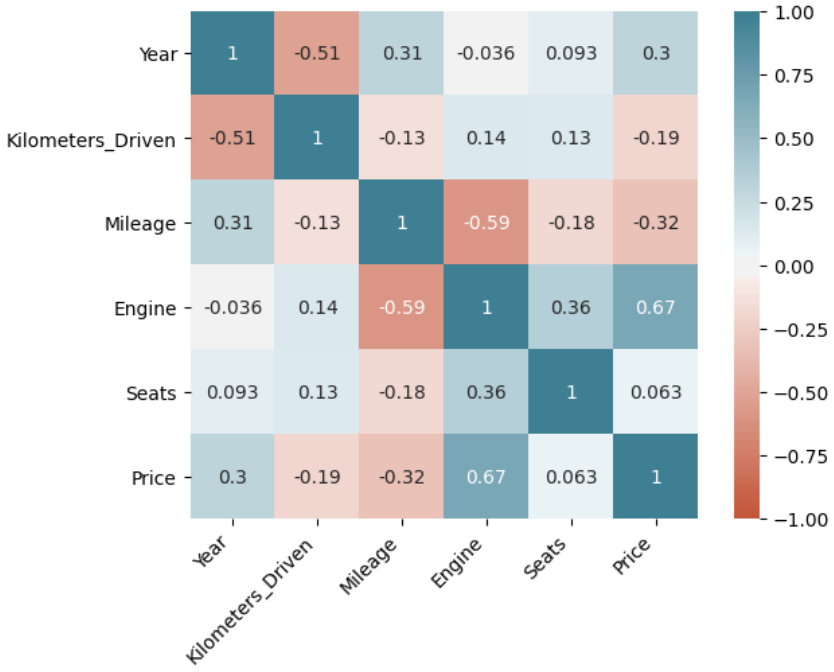
	Mileage_f
Fuel_Type	
CNG	25.418036
LPG	19.385000
Diesel	18.620484
Petrol	17.415204
Electric	0.000000

Maka, ditemukan bahwa tipe bahan bakar diurutkan dari yang terhemat adalah **CNG, LPG, Diesel dan Petrol.** Sedangkan untuk bahan bakar elektric tidak termasuk dalam urutan bahan bakar minyak atau gas yang hemat maupun boros dikarenakan tidak adanya data

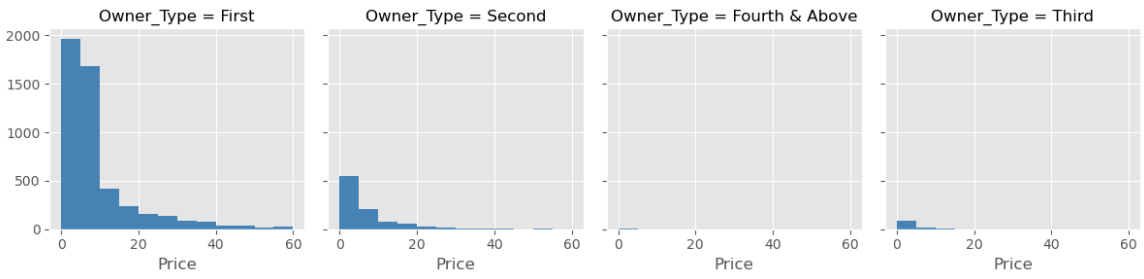
Mileage berdasarkan dataset diatas sehingga diisikan data berupa 0 (nol). Selain itu bahan bakar electric merupakan bahan bakar yang ramah lingkungan.

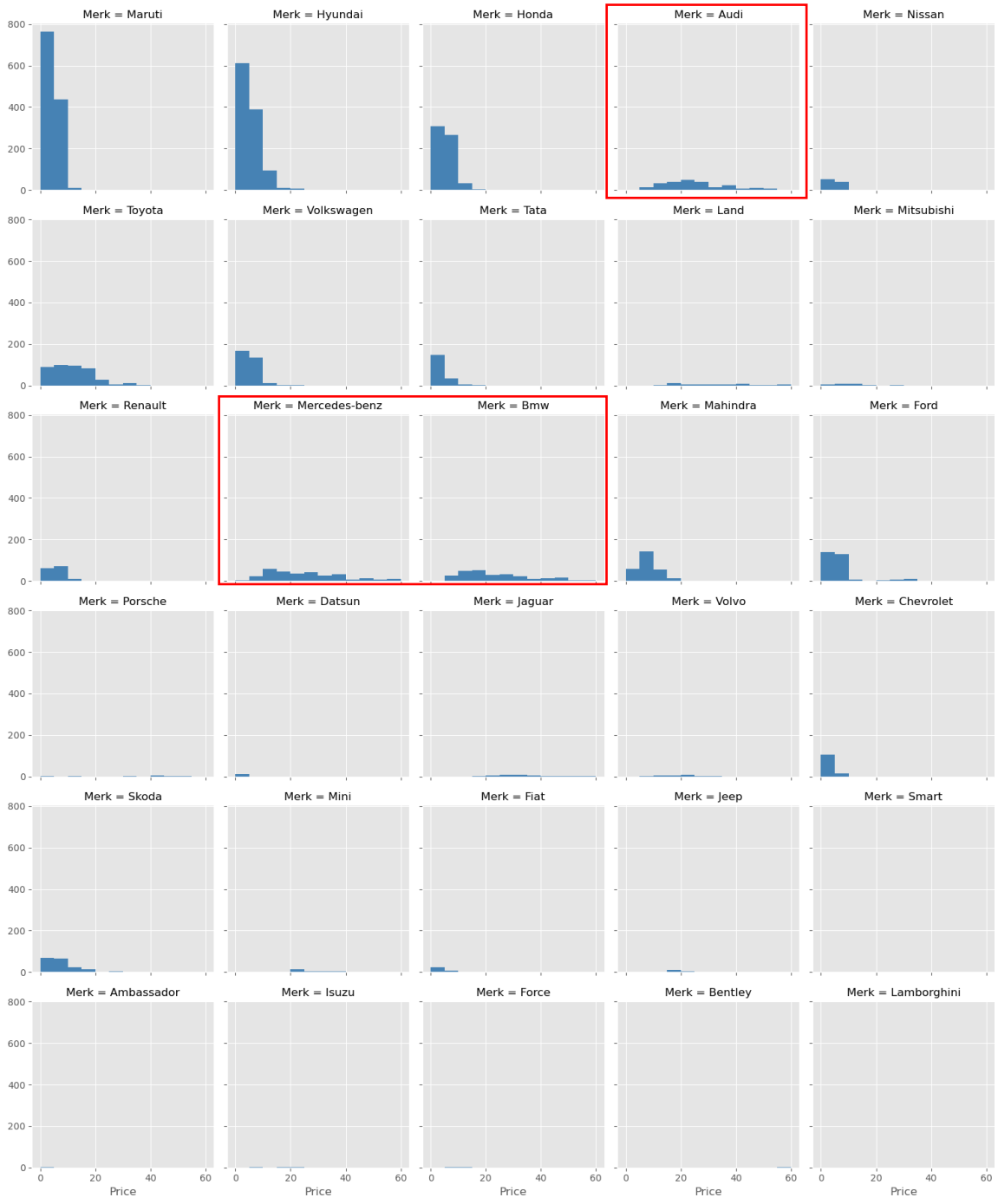
10. Apa saja faktor-faktor yang mempengaruhi harga mobil bekas di India? Sertakan argumen yang mendukung jawaban.

    Berdasarkan dengan numeric variable yang terdapat pada data, nilai korelasi yang paling tinggi yang berpengaruh terhadap harga mobil adalah Engine dengan korelasi positive. Hal ini menunjukkan bahwa Engine berpengaruh terhadap harga mobil namun tidak terlalu signifikan hanya 60%.



Untuk mengetahui variable lain kita melihat dari distribusi data berdasarkan kategori (Merk dan Owner_Type)





Persebaran data berdasarakan owner_type menunjukkan bahwa setiap jenis owner_type memiliki persebaran data yang sama. Hal ini mengindikasikan bahwa harga mobil tidak terpengaruh jika dilihat dari kepemilikan.

Sedangkan jika dilihat berdasarkan merk atau brand ada perbedaan persebaran data dari beberapa brand. Seperti audi, mercedes, dan bmw yang persebaran datanya cenderung tinggi di harga menengah tidak seperti Maruti atau Hyundai. Hal ini menunjukkan bahwa memang merk dari mobil berpengaruh terhadap harga mobil karena perbedaan persebaran

data pada tiap merk. Jadi dapat disimpulkan hal – hal yang mempengaruhi harga mobil adalah Engine dan Merk.

Hasil Analisis Tambahan

Merk	Price
Lamborghini	120.000000
Bentley	59.000000
Porsche	48.348333
Land	39.259500
Jaguar	37.632250
Mini	26.896923
Mercedes-benz	26.809874
Audi	25.537712
Bmw	25.243146
Volvo	18.802857

Dengan dilakukan nya pemisahan data menggunakan fungsi groupby berdasarkan column Merk dengan agregasi mean atau rata rata pada column Price. Maka berdasarkan visualisasi data tersebut, kita dapat mengetahui bahwa berikut adalah 10 merk mobil termahal di India. Dengan di urutan nomor 1 yaitu merk Lamborghini dengan harga rata rata yaitu 120 INR Lakhs.

Kesimpulan

Dari hasil analisa dataset mengenai mobil bekas yang terdapat di India, dapat dikatakan bahwa di India sendiri pun pasar mobil bekas dapat dikatakan memiliki basis pasar yang sangat besar. Banyak yang mempertimbangkan untuk membeli mobil bekas daripada harus membeli yang baru. Ketika kita membeli mobil baru dan kemudian menjualnya hanya beberapa hari saja tanpa ada default, harga mobil berkurang hingga 30%.

Dalam pembelian mobil bekas tersebut, hal yang paling diperhatikan dan berpengaruh ialah Engine karena berdasarkan dengan numeric variabel pada data, nilai korelasi paling tinggi yang paling berpengaruh pada pembelian mobil ialah Engine sebesar 60%. Sementara jarak tempuh (mileage) yang paling tidak berpengaruh.

DAFTAR REFERENSI

Galarnyk, M., 2018. *Toward Data Science.* [Online] Available at: https://towardsdatascience.com/how-to-use-and-create-a-z-table-standard-normal-table-240e21f36e53#:~:text=%2Dscore%20%3D%201.0)-,To%20use%20the%20z%2Dscore%20table%2C%20start%20on%20the%20left,8413%20which%20is%20the%20probability. [Accessed 16 July 2020].

Glen, S., 2020. *Statistics How To.* [Online] Available at: https://www.statisticshowto.com/probability-and-statistics/confidence-interval/ [Accessed 18 July 2020].

Z Tables, n.d. *Z Tables.* [Online] Available at: https://www.ztable.net/ [Accessed 2020 July 17].