Data Science Academy Assignment

TIM BAHARI JAYA

Anggota:

Aryo Hastungkoro Harimurti Mukarta

Nardiena Althafia Pratama

Thareq Wibisono

Contents

Latar Belakang

Abad ke-21 adalah era dimana informasi merupakan salah satu asset penting dalam kehidupan sehari-hari manusia. Tanpa adanya informasi yang luas, sekelompok organisasi akan kalah bersaing dengan kompetitornya, dan seseorang akan ketinggalan jaman karena tidak mengikuti perkembangan informasi dunia. Di era dimana informasi menjadi penting, dibutuhkan sebuah ilmu berbasis pemrogaman komputer untuk memahami banyak informasi tersebut agar dapat memperoleh kesimpulan yang objektif dalam mengambil sebuah keputusan. Lahirlah sebuah ilmu bernama *Data Science* yaitu ilmu yang menggabungkan pemikiran akar antara komputer/IT, matematika dan statistika, dan ilmu bisnis. Seorang *data scientist* merupakan seseorang yang mahir dalam analisis statistikal yang secara bersamaan mampu menggunakan ilmu komputernya untuk menghasilkan sebuah informasi berupa trend serta menganalisis trend tersebut secara empirik.

Akibat dari lonjakan informasi ini, banyak perusahaan yang akhirnya menyadari betapa pentingnya sebuah informasi dalam perkembangan perusahaan. Hal tersebutlah yang secara tidak langsung menghasilkan lonjakan permintaan bagi para *data scientist* dengan gaji yang menggiurkan sehingga membuat banyak orang berlomba-lomba untuk menjadi bagian dari maestro informasi tersebut. Berbagai kursus *data science* pun hadir di kalangan masyarakat sebagai media pengenalan terhadap esensi dari istilah tersebut. Salah satu ajang Pendidikan *data science* yang populer di kalangan anak muda adalah COMPFEST.

COMPFEST merupakan ajang Pendidikan *data science* yang mempunyai peran untuk memperdalam bidang *data science* bagi masyarakat. Dengan hadirnya COMPFEST ini, peserta diharapkan mempunyai fondasi yang kuat dalam meniti karir yang berhubungan dengan *data science* baik itu dalam ranah R&D maupun dalam industri.

Pada ajang COMPFEST kali ini terdapat seleksi masuk bagi para calon peserta. Metode seleksi yang digunakan COMPFEST adalah pemberian tugas berupa sebuah dataframe dalam format file csv yang ditujukan bagi para peserta agar dapat menjawab pertanyaan-pertanyaan di dalam tugas tersebut. Dataframe yang diberikan oleh COMPFEST pada tahun ini berupa data pemakaian mobil bekas di India yang terdiri dari 6019 baris x 12 kolom dimana tiap baris merepresentasikan merek individu yang independen dari baris lainnya, sedangkan kolom merepresentasikan parameter-parameter jual merek individu tersebut.

Dalam tahap ini peserta akan diuji kemampuan *data* science mereka dalam menghadapi tantangan seperti terdapatnya *missing value* dan *outliers* yang dapat mempengaruhi hasil

analisis serta mengatur data sedemikian rupa agar data dapat diolah secara statistik. Selain menjawab pertanyaan yang tertera pada tugas, peserta juga dihimbau untuk melakukan analisis tambahan mengenai data yang sudah diolah tersebut agar terdapat hubungan berdasarkan tinjauan literatur antara parameter satu dengan lainnya.

Jawaban Soal

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

Untuk menjawab pertanyaan tersebut kami menggunakan Bahasa pemrograman

```
filename="C:/Users/Thareq/Documents/Thareq Library/Compfest/used_car_data.csv"
df=pd.read_csv(filename)
```

Figur 1.1: *Fungsi pd.Series untuk mengimport file csv untuk diolah*

python dengan memasukan paket pandas. Paket pandas sangat berguna untuk membaca file csv dan mengubahnya menjadi bentuk *DataFrame*. *DataFrame* tersebut kami simpan dalam variable df seperti yang terlihat pada figur 1.1. Hasil dari pembacaan file csv tersebut terpampang pada figur 1.2 dibawah ini.

	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats
0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5.0
1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.0
2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5.0

Figur 1.2: *DataFrame hasil dari import*

Dalam kasus soal ini, kami hanya fokus untuk memecahkan berapa jumlah merek unik dalam kolom “Name”. Gambaran umum akan Bahasa program python yang kami gunakan dapat dilihat pada figur 1.3 dimana kode kami terdiri dari 8 baris.

```
List_make=list(df['Name'])
make=[]
for i in range(0, len(List_make)):
    row_split=List_make[i].split()
    First_makeTerm=row_split[0]
    make.append(First_makeTerm)
series_ver=pd.Series(make)
series_ver.value_counts()
```

Figur 1.3*: Gambaran umum untuk kode pertanyaan nomor 1*

Prinsip umum dari kode pada figur 1.3 adalah mendapatkan *value* unik yang nantinya akan dihitung berapa jumlah tiap *value* unik tersebut dengan menggunakan fungsi pandas value_counts(). Alur dari kode tersebut dibagi menjadi 3 proses yaitu

pengubahan *DataFrame* menjadi list, pemisahan kalimat untuk tiap *value* list, pengambilan kata pertama dari tiap baris, pengubahan list menjadi series, dan perhitungan *value* unik.

Pertama-tama kami membuat list untuk kolom "Name" dan dimasukan dalam variabel List_make. Setelah itu kami membuat list kosong dan disimpan dalam variabel make yang nantinya sebagai tempat menampung merek yang sudah di potong hingga tersisa kata pertama dari keseluruhan kalimat. Proses pemotongan kalimat untuk variabel List_make dilakukan dengan menggunakan iterasi for dimana iterasi tersebut dilakukan sepanjang variabel list List_make dengan menggunakan fungsi range(0, len(List_make)). Setiap iterasi akan dilakukan proses pemotongan dengan menggunakan fungsi split() yang nantinya disimpan dalam variabel row_split. Proses dilanjutkan dengan mangambil kata pertama dari hasil pemotongan setiap iterasi sehingga hanya tersisa merek mobil dan disimpan dalam variabel First_makeTerm yang diikuti dengan penambahan ke variabel make dengan menggunakan funsi append(). Proses tersebut diulang hingga list terakhir dalam variabel List_make. Kemudian kami mengubah bentuk data dari variabel make menjadi series dengan menggunakan fungsi paket Pandas pd.Series yang dilanjutkan dengan perhitungan *value* unik dengan menggunakan fungsi value_counts(). Hasil dari perhitungan value_counts() dapat dilihat pada figur 1.4 dibawah ini.

```
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
SMART               1
LAMBORGHINI         1
BENTLEY             1
AMBASSADOR          1
dtype: int64
```

Figur 1.4: *Hasil dari fungsi value.counts() pada kolom "Name" yang sudah di split. Terdapat 31 merek mobil bekas yang terdapat di India*

2. *Kota apa yang memiliki mobil bekas paling banyak?*

```
df['Location'].value_counts()
```

Figur 2.1: *Fungsi value.counts() yang diperuntukan untuk kolom "Location"*

Soal ini dapat dengan mudah kami temukan dengan menggunakan fungsi value.counts() pada kolom "Location" dan dilakukan hanya dengan satu baris kode

seperti yang terlihat pada figur 2.1. Hasil dari perhitungan fungsi value.counts() dapat dilihat pada figur 2.2 dibawah. Dapat dilihat bahwa kota Mumbai memiliki mobil bekas paling banyak dibandingkan kota lainnya.

```
Mumbai          790
Hyderabad       742
Kochi           651
Coimbatore      636
Pune            622
Delhi           554
Kolkata         535
Chennai         494
Jaipur          413
Bangalore       358
Ahmedabad       224
Name: Location, dtype: int64
```

Figur 2.2: *Hasil dari fungsi value.counts() pada kolom "Location"*

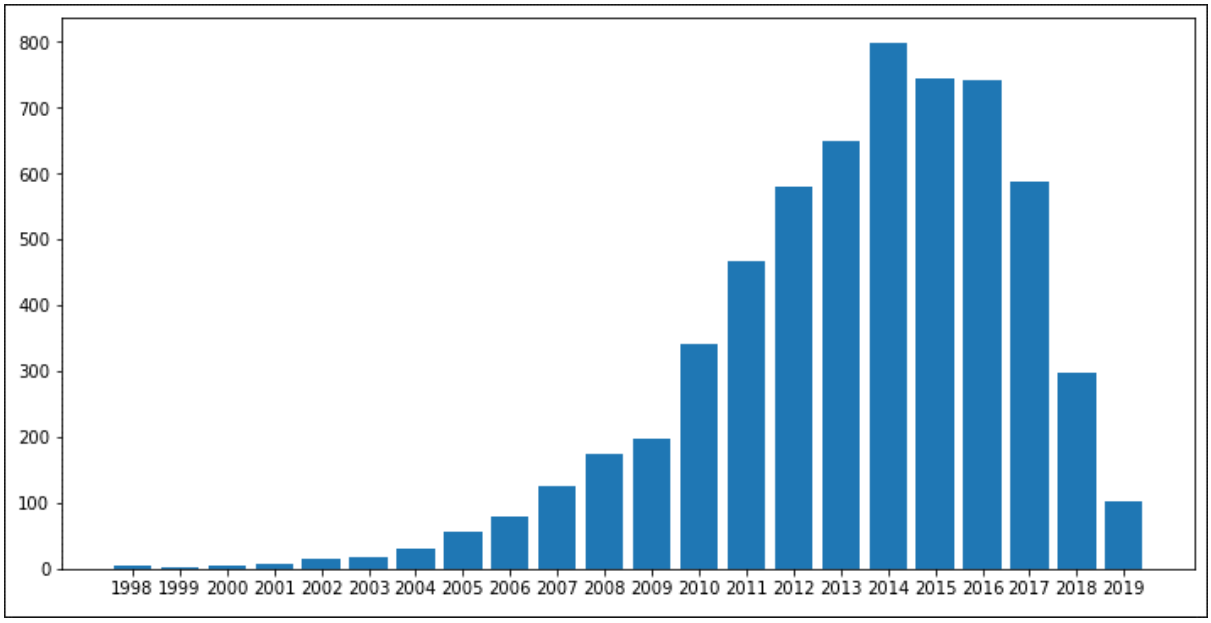
3. *Bagaimana distribusi tahun edisi mobil-mobil bekas tersebut?*

Kami membuat fungsi yang bertujuan untuk menghitung frekuensi data untuk tiap tahun. Fungsi dapat dilihat pada figur di bawah.

```
def distribusiTahun(data):
    tahun = {}
    fig= plt.figure(figsize=(12,6))
    for mobil in data:
        tahunTemp = mobil
        if tahunTemp not in tahun:
            tahun[tahunTemp] = 1
        else:
            tahun[tahunTemp] += 1
    tahunSorted = sorted(tahun)
    tahunJumlah = []
    for i in range (0,len(tahunSorted)):
        tahunJumlah.append(tahun[tahunSorted[i]])
    plt.bar(range(len(tahunJumlah)), list(tahunJumlah), align='center')
    plt.xticks(range(len(tahunSorted)), list(tahunSorted))
    plt.show()
    print('Data Jumlah Mobil per Merk')
    for i in range (0,len(tahunSorted)):
        print(str(tahunSorted[i]) + ": " + str(tahunJumlah[i]))
```

Figur 3.1*: Fungsi untuk melihat distribusi tahun*

Lalu kita membuat plot untuk mempermudah visualisasi persebaran data. Hasil plot dari fungsi diatas adalah figur berikut.



***Figur 3.2**: Histogram distribusi tahun*

Dengan frekuensi mobil terjual pertahun dapat dilihat di figur berikut.

```
Data Jumlah Mobil per Merk
1998: 4
1999: 2
2000: 4
2001: 8
2002: 15
2003: 17
2004: 31
2005: 57
2006: 78
2007: 125
2008: 174
2009: 198
2010: 342
2011: 466
2012: 580
2013: 649
2014: 797
2015: 744
2016: 741
2017: 587
2018: 298
2019: 102
```

***Figur 3.3**: Data frekuensi penjualan mobil per tahun*

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

Dalam kasus soal ini, kami fokus pada pengerjaan kolom Kilometers_Driven. Gambaran umum akan kode *python* yang kami gunakan pada soal ini dapat dilihat pada figur 4.1 dibawah ini.

```
list_NameColumn=list(df['Kilometers_Driven'])
count_temp = 0
for i in range(0, len(list_NameColumn)):
    if list_NameColumn[i]<100000:
        count_temp += 1
print('Jumlah mobil dengan jarak tempuh kurang dari 100.000 km ada ' + str(count_temp), 'mobil')
```

```
Jumlah mobil dengan jarak tempuh kurang dari 100.000 km ada 5470 mobil
```

Figur 4.1: *Gambaran umum untuk kode pertanyaan nomor 4*

Prinsip umum dari kode pada figur 3.1 terbagi menjadi 3 bagian yaitu mengubah bentuk *DataFrame* menjadi bentuk list sehingga dapat diiterasikan, kemudian menghitung jumlah *value* kilometers_Driven dibawah 100.000 km dengan menggunakan *statement* kondisi if, setelah itu proses tersebut diiterasikan untuk semua *value* di list, dan terakhir di print untuk menampilkan hasilnya.

Hal pertama yang kami lakukan adalah mengubah bentuk series di *DataFrame* menjadi list dengan menggunakan fungsi list() dan disimpan pada variabel list_NameColumn. Kemudian kami membuat variabel count_temp dengan *value* int 0 yang nantinya akan digunakan untuk menghitung jumlah mobil dengan jarak tempuh kurang dari 100.000 km. Untuk menghitung permasalahan tersebut kami menggunakan *statement* kondisi if dimana jika *value* di list kurang dari 100.000 km maka variabel count_temp akan bertambah 1. Proses ini diiterasikan dengan menggunakan iterasi for dengan range sepanjang list list_NameColumn. Hasil dari count_temp selanjutnya akan di tampilkan dengan menggunakan fungsi print() seperti yang ditunjukan pada figur 4.1

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

Untuk menjawab pertanyaan ini, kami mencari distribusi data pada kolom *Kilometers_Driven* dengan menggunakan *method* "*describe()*".

```
73 # Question 5
74 def stats(col):
75     print(round(col.describe()))
76
```

Figur 5.1*: Fungsi untuk melihat deskripsi box plot untuk kolom Kilometers_Driven*

Hasilnya sebagai berikut:

```
In [46]: stats(kiloCol)
count       6019.0
mean       58738.0
std        91269.0
min          171.0
25%        34000.0
50%        53000.0
75%        73000.0
max      6500000.0
Name: Kilometers_Driven, dtype: float64
```

***Figur 5.2**: Deskripsi box plot untuk kolom Kilometers_Driven*

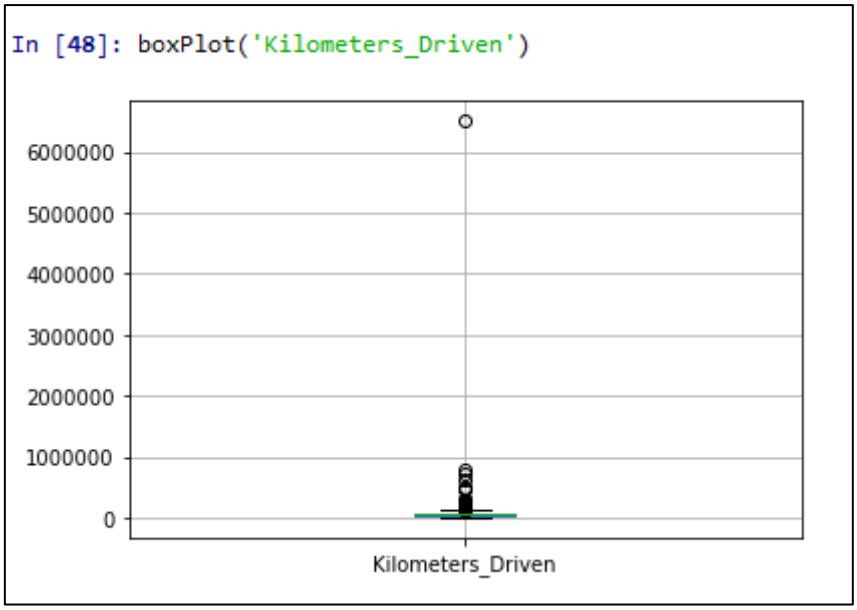
Dari hasil deskripsi box plot, dapat diinterpretasikan bahwa total jarak pemakaian bisa dikategorikan sebagai rendah jika berada di bawah kuartil pertama (25%), yaitu **34000 kilometer**, dan bisa dikategorikan sebagai tinggi jika berada di atas kuartil ketiga (75%), yaitu **73000 kilometer**. Kami menggunakan kuartil pertama dan kuartil ketiga sebagai batas untuk kategori 'rendah' dan 'tinggi' karena mereka merupakan batas pada rentang interkuartil. Rentang interkuartil mengukur 50% nilai tengah dari seluruh data, jadi dapat digunakan sebagai ukuran penyebaran yang andal.

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

Untuk mengetahui outlier pada kolom tersebut, kami menggunakan metode yang sama seperti pada soal sebelumnya – box plot – tetapi sekarang dengan memvisualisasikannya. Ini kode yang kami gunakan untuk mencapai ini:

```
78 def boxPlot(colName):
79     read_data.boxplot(column=colName)
80     plt.savefig(colName+'.png')
```

***Figur 6.1**: Fungsi box plot untuk kolom Kilometers_Driven*



***Figur 6.2**: Gambar box plot untuk kolom Kilometers_Driven*

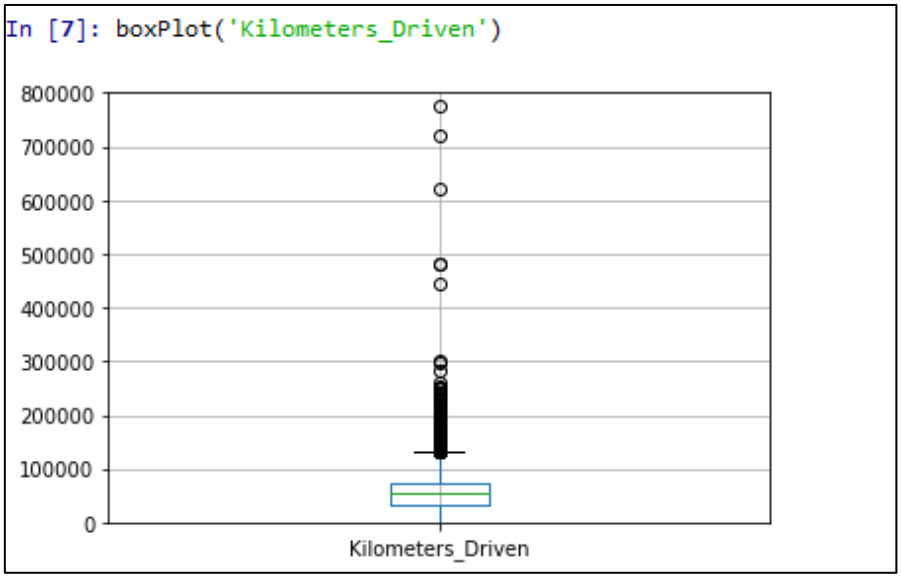
Pada figur boxplot di atas, bulatan-bulatan merupakan outlier-outlier. Akan tetapi, boxplotnya sendiri tidak terlihat terlalu jelas karena adanya satu outlier yang nilainya sangat jauh dari yang lain, yaitu dengan **nilai sekitar 6500000**.

Agar dapat melihat outlier-outlier selain itu, kami memperbesar figur boxplot tersebut dengan menggunakan "plt.ylim(miny, maxy)" pada kode kami.

```
78 def boxPlot(colName):
79     read_data.boxplot(column=colName)
80     plt.ylim(0, 800000)
81     plt.savefig(colName+'.png')
```

***Figur 6.3**: Fungsi box plot untuk kolom Kilometers_Driven (diperbesar)*

Setelah menjalankan kode tersebut, hasilnya sebagai berikut.



***Figur 6.4**: Gambar box plot untuk kolom Kilometers_Driven yang sudah diperbesar*

Pada figur ini, terlihat lebih jelas outlier-outlier yang lain. Semua outlier tampaknya berada **di atas nilai sekitar 120000**.

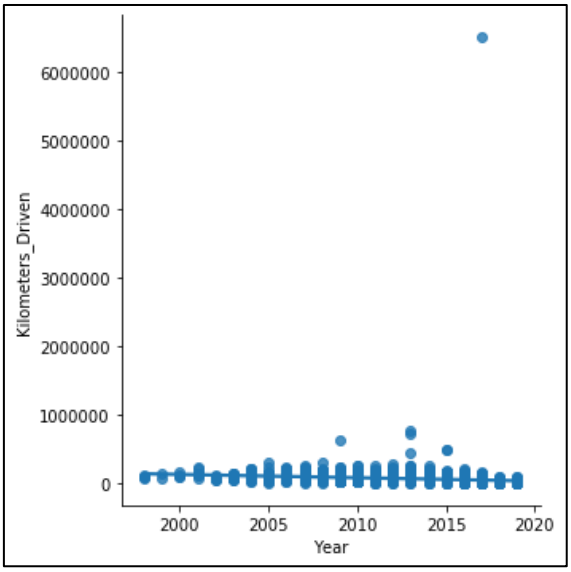
7. *Apakah tahun pembuatan mobil berpengaruh terhadap total jarak pemakaian? Sertakan argumen yang mendukung jawaban.*

Untuk mengetahui apakah tahun pembuatan mobil dan total jarak pemakaian berkorelasi, kami membuat visualisasi regresi linear menggunakan method *lmplot* dari seaborn.

```
84 def linearReg(colXName, colYName):
85     sns.lmplot(x=colXName, y=colYName, data=read_data)
86     ax = plt.gca()
```

***Figur 7.1**: Fungsi untuk membuat grafik regresi linear untuk Year vs Kilometers_Driven*

Hasil kode tersebut sebagai berikut.



***Figur 7.2**: Grafik regresi linear untuk Year vs Kilometers_Driven*

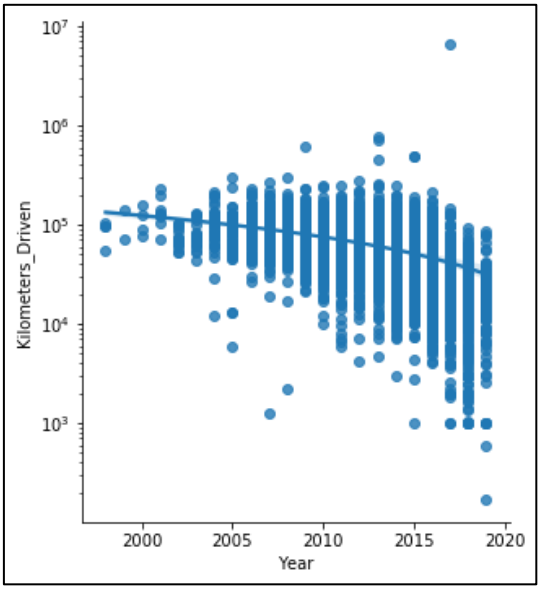
Dari figur di atas, kami tidak dapat melihat korelasi antara dua variabel tersebut karena adanya outlier yang sangat jauh nilainya dari data yang lain. Outlier tersebut juga tidak masuk akal karena tahun pembuatannya berada di sekitar tahun 2018, akan tetapi total jarak pemakaiannya melebihi enam kali lipat total jarak pemakaian semua mobil yang tahun pembuatannya sebelum tahun 2018. Kemungkinan besar data tahun pembuatan untuk outlier tersebut merupakan *typo* dan seharusnya 600.000, bukan 6.000.000, karena 600.000 jauh lebih masuk akal sebagai total jarak pemakaian mobil yang dibuat pada tahun 2018.

Agar dapat melihat ada tidaknya korelasi dengan lebih jelas, kami memutuskan untuk menghilangkan outlier tersebut. Kami menggunakan method *set_yscale('log')* untuk memperbesar figur boxplot.

```
84 def linearReg(colXName, colYName):
85     sns.lmplot(x=colXName, y=colYName, data=read_data)
86     ax = plt.gca()
87     ax.set_yscale('log')
```

***Figur 7.3**: Fungsi untuk membuat grafik regresi linear untuk Year vs Kilometers_Driven (diperbesar)*

Hasil kode tersebut sebagai berikut.



***Figur 7.4**: Grafik regresi linear untuk Year vs Kilometers_Driven yang sudah diperbesar*

Dari figur yang sudah diperbesar ini, terlihat bahwa ada korelasi negative, walau tidak terlalu signifikan, antara tahun pembuatan mobil dan total jarak pemakaiannya. Ini memang sesuai ekspektasi kita karena tentunya semakin baru tahun pembuatan mobil, maka semakin kecil rentang waktu dari waktu pembuatan dengan waktu sekarang, sehingga lebih kecil pula total jarak pemakaiannya.

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

```
df['Owner_Type'].value_counts()
```

```
First             4929
Second             968
Third              113
Fourth & Above       9
Name: Owner_Type, dtype: int64
```

***Figur 8.1:** Gambaran umum untuk kode pertanyaan nomor 8*

Kode yang digunakan untuk menjawab soal ini mirip dengan nomor 2 dimana kami menggunakan value_counts() untuk menghitung *value* unik dari kolom "Owner_Type" seperti yang tertera pada figur 8.1 diatas. Jika dilihat dari figur tersebut, dapat diperhatikan bahwa untuk kepemilikan ketiga ada 113 dan untuk kepemilikan keempat atau lebih ada 9. Proses penjumlahan dapat dilakukan dengan pertama-tama membuat variabel Own_typList yang mengandung list berupa *value* pada figur 5.1 dan diperlihatkan pada figur 8.2 dibawah. Kemudian kami ambil *value* nomor 2 dan 3 untuk dijumlahkan dan ditampilkan hasilnya seperti yang terlihat pada figur 5.2 yaitu berjumlah 122.

```
Own_typList=df['Owner_Type'].value_counts().tolist()
Own_typList[2]+Own_typList[3]
```

```
122
```

Figur 8.2: *Kode penjumlahan serta jawaban dari penjumlahan tersebut*

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

Untuk menjawab pertanyaan ini, kami membuat fungsi dimana setiap tipe bahan bakar dimasukkan sebagai keys dalam dictionary. Value dari setiap tipe bahan bakar adalah jumlah mobil yang menggunakan tipe bahan bakar tersebut.

Kemudian kami membuat dictionary lagi, dengan tipe bahan bakar sebagai keys. Value dari setiap tipe bahan bakar dalam dictionary ini adalah hasil penjumlahan nilai mileage untuk setiap mobil dengan tipe bahan bakar tersebut.

Total mileage untuk setiap tipe bahan bakar (value dari dictionary kedua) dibagi dengan jumlah mobil dengan tipe bahan bakar tersebut (value dari dictionary pertama) untuk mendapat nilai rata-rata.

```python
def mostEconomicalFuel(fuelCol, mileageCol):
    fuelDict = {}
    mileageDict = {}
    mileageAvg = {}
    for fuel in fuelCol:
        if fuel not in fuelDict:
            fuelDict[fuel] = 1
        else:
            fuelDict[fuel] += 1
    for fuel in fuelDict:
        tempData = read_data.loc[read_data.Fuel_Type == fuel, 'Mileage']
        for i in tempData:
            if i is not None:
                if fuel not in mileageDict:
                    mileageDict[fuel] = float(str(i).split()[0])
                else:
                    mileageDict[fuel] += float(str(i).split()[0])
        mileageAvg[fuel] = mileageDict[fuel]/fuelDict[fuel]

    print(fuelDict)
    print(mileageDict, "\n")
    print('Average mileage for each fuel type:\n')
    for i in mileageAvg:
        print(i, ":", mileageAvg[i])
```

***Figur 9.1**: Fungsi yang memberi informasi mengenai rata-rata mileage untuk setiap tipe bahan bakar*

Hasil fungsinya sebagai berikut.

```
In [17]: mostEconomicalFuel(fuelCol, mileageCol)
{'CNG': 56, 'Diesel': 3205, 'Petrol': 2746, 'LPG': 10, 'Electric': 2}
{'CNG': 1423.4099999999999, 'Diesel': 59678.65000000024, 'Petrol': 47822.14999999983, 'LPG': 193.85, 'Electric': nan}

Average mileage for each fuel type:

CNG : 25.41803571428571
Diesel : 18.62048361934485
Petrol : 17.415203932993382
LPG : 19.384999999999998
Electric : nan
```

***Figur 9.2**: Hasil fungsi mostEconomicalFuel() yang memberi informasi mengenai rata-rata mileage untuk setiap tipe bahan bakar*

Dari hasil tersebut, dapat terlihat bahwa tipe bahan bakar yang paling hemat adalah ‘Petrol’ dengan nilai rata-rata mileage: 17.415 kmpl.

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

Kami akan menguji hubungan antar setiap variabel ‘Name’, ‘Location’, ‘Year’, ‘Kilometers_Driven’, ‘Fuel_Type’, ‘Transmission’, ‘Owner_Type’, ‘Mileage’, ‘Engine’, ‘Power’, dan ‘Seats’, terhadap variabel ‘Price’. Kami menggunakan metode regresi linear berganda.

[3]

```
file = r'used_car_data.csv'
mobilData = pd.read_csv(file)
mobilData
```

	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
...	...	...	...	...	...	...	...	...	...	...	...	...
6014	Maruti Swift VDI	Delhi	2014	27365	Diesel	Manual	First	28.4 kmpl	1248 CC	74 bhp	5.0	4.75
6015	Hyundai Xcent 1.1 CRDi S	Jaipur	2015	100000	Diesel	Manual	First	24.4 kmpl	1120 CC	71 bhp	5.0	4.00
6016	Mahindra Xylo D4 BSIV	Jaipur	2012	55000	Diesel	Manual	Second	14.0 kmpl	2498 CC	112 bhp	8.0	2.90
6017	Maruti Wagon R VXI	Kolkata	2013	46000	Petrol	Manual	First	18.9 kmpl	998 CC	67.1 bhp	5.0	2.65
6018	Chevrolet Beat Diesel	Hyderabad	2011	47000	Diesel	Manual	First	25.44 kmpl	936 CC	57.6 bhp	5.0	2.50

6019 rows × 12 columns

***Figur 10.1**: Data mobil bekas di India*

Jika dilihat dari *DataFrame* diatas, beberapa data yang terdapat di dataset tersebut masih berupa data kualitatif dengan tipe data '*string*'. Untuk mengatasi hal tersebut, kami mengubah data-data tersebut menjadi data kategorik agar dapat dianalisis menggunakan model regresi linear.

```
def quantify(data):
    dataKualitatif = []
    for i in data:
        dataKualitatif.append(str(i).upper())
    kategori = []
    for i in dataKualitatif:
        if i not in kategori:
            kategori.append(i)
    dataKategorik = []
    for i in dataKualitatif:
        for j in kategori:
            if i == j:
                dataKategorik.append(kategori.index(j))
    return dataKategorik
```

***Figur 10.2**: Fungsi untuk mengubah data kualitatif menjadi data kategorik*

Hal yang pertama kami lakukan adalah membuat fungsi yang dapat dilihat pada figur diatas. Prinsip utama dari figur tersebut adalah dengan membandingkan sebuah data kualitatif dengan posisinya. *dataKualitatif* merupakan *list* yang berisi semua *value* dari suatu variabel. *kategori* merupakan *list* yang berisi semua *value* unik dari suatu variabel. *dataKategorik* merupakan *list* yang berisi *value* dari *dataKualitatif* yang telah dilabeli sesuai posisi pada *list kategori*.

Variabel yang menggunakan fungsi *quantify(x)* adalah 'Name', 'Location', 'Fuel_Type', 'Transmission', dan 'Owner_Type'.

```
def unitCut(data):
    cut = []
    for i in data:
        if str(i).split()[0] == 'null':
            cut.append(0)
        else:
            cut.append(float(str(i).split()[0]))
    return cut
```

***Figur 10.3**: Fungsi untuk menghapus satuan*

Selanjutnya, kami akan memotong *value* yang memiliki satuan didalamnya. Hal ini diperlukan untuk memudahkan kita dalam pembuatan model. Prinsip umum dari figur diatas adalah dengan memasukan kata pertama (yang berupa data *float*) dari sebuah *value ke* dalam *list* baru. *cut* merupakan list baru yang berisi *value* yang sudah merupakan data *float*. Jika terdapat data yang berisi ‘null’, kita ubah menjadi 0.

Variabel yang menggunakan fungsi *unitCut(x)* adalah ‘Mileage’, ‘Engine’, dan ‘Power’.

```
for i in range(0,len(mobilData.Fuel_Type)):
    if mobilData.Fuel_Type.index[i] == 'Electric':
        mobilData.Mileage.index[i] = 0

mobilData = mobilData.dropna()
```

***Figur 10.4**: Skrip untuk membersihkan data*

Setelah itu, kami akan membersihkan baris yang memiliki *missing value*. Kami tahu bahwa mobil listrik tidak menggunakan bahan bakar, maka setiap mobil dengan ‘Fuel_Type’ berupa *Electric*, ‘Mileage’ dari baris tersebut adalah 0. Setelah itu, kami menghapus semua baris yang memiliki *missing value* dengan *dropna( )* seperti yang terlihat pada figur 10.4.

```
variables = [quantify(mobilData.Name), quantify(mobilData.Location), list(mobilData.Year),
             list(mobilData.Kilometers_Driven), quantify(mobilData.Fuel_Type),
             quantify(mobilData.Transmission), quantify(mobilData.Owner_Type),
             unitCut(mobilData.Mileage), unitCut(mobilData.Engine), unitCut(mobilData.Power),
             list(mobilData.Seats), list(mobilData.Price)]
```

***Figur 10.5**: List berisi data yang sudah di kuantifikasi*

Setelah itu, kita membuat *array* yang berisi semua data yang telah kita olah dengan kedua fungsi yang baru kita buat. Lalu, kita akan membuat file *csv* baru yang berisi dataframe yang sudah kita olah.

```
export = pd.DataFrame(variables, index = list(mobilData.columns.values)).transpose()
export.to_csv (r'mobilBekas.csv', index = False, header=True)

print (export)
```

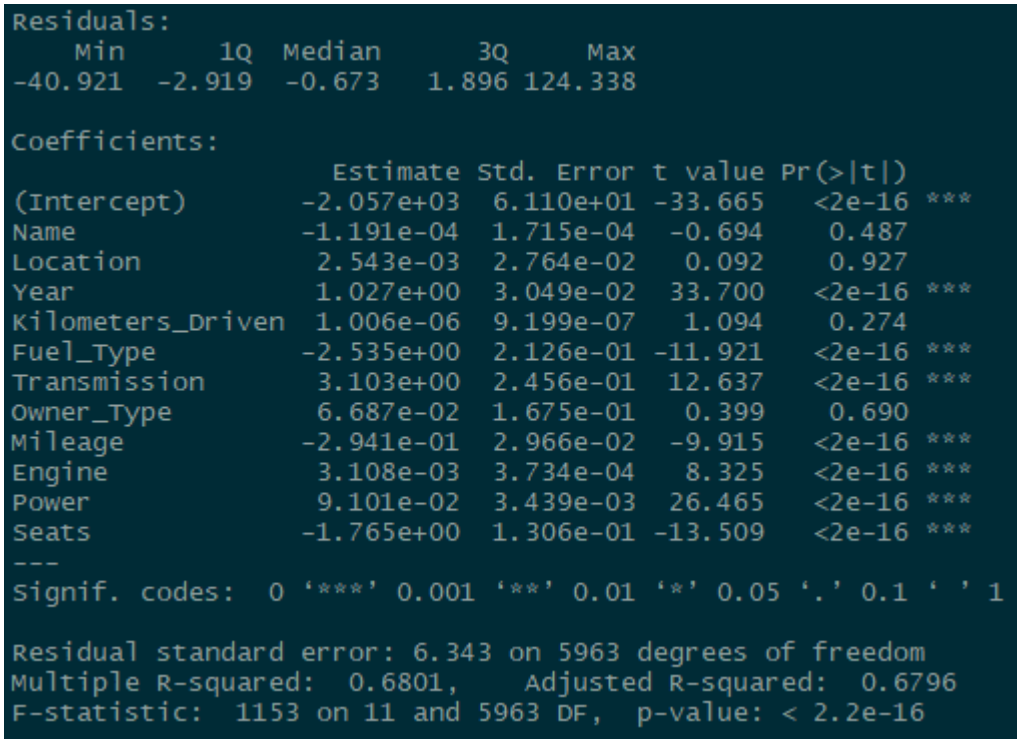
```
Name  Location    Year  Kilometers_Driven  Fuel_Type  Transmission  \
0        0.0     0.0  2010.0            72000.0        0.0           0.0
1        1.0     1.0  2015.0            41000.0        1.0           0.0
2        2.0     2.0  2011.0            46000.0        2.0           0.0
3        3.0     2.0  2012.0            87000.0        1.0           0.0
4        4.0     3.0  2013.0            40670.0        1.0           1.0
...      ...     ...     ...                ...        ...           ...
5970    52.0     8.0  2014.0            27365.0        1.0           0.0
5971    47.0     5.0  2015.0           100000.0        1.0           0.0
5972  1830.0     5.0  2012.0            55000.0        1.0           0.0
5973   200.0     7.0  2013.0            46000.0        2.0           0.0
5974   684.0     4.0  2011.0            47000.0        1.0           0.0

      Owner_Type  Mileage  Engine   Power  Seats  Price
0            0.0    26.60   998.0   58.16    5.0   1.75
1            0.0    19.67  1582.0  126.20    5.0  12.50
2            0.0    18.20  1199.0   88.70    5.0   4.50
3            0.0    20.77  1248.0   88.76    7.0   6.00
4            1.0    15.20  1968.0  140.80    5.0  17.74
...          ...      ...     ...     ...    ...    ...
5970         0.0    28.40  1248.0   74.00    5.0   4.75
5971         0.0    24.40  1120.0   71.00    5.0   4.00
5972         1.0    14.00  2498.0  112.00    8.0   2.90
```

***Figur 10.6**: Mengekspor dataframe akhir dengan format CSV*

Kita membuat model linear berupa:

$$Price = \beta_0 + \beta_1 Name + \beta_2 Location + \beta_3 Year + \beta_4 Kilometers\_Driven + \beta_5 Fuel\_Type + \beta_6 Transmission + \beta_7 Owner\_Type + \beta_8 Mileage + \beta_9 Engine + \beta_{10} Power + \beta_{11} Seats$$

Dimana β_0 merupakan titik potong model terhadap sumbu Y dan $\beta_1, \dots, \beta_{11}$ merupakan koefisien dari tiap variabel selain 'Price'. Lalu, kami mengolah *data frame* yang baru kita buat dengan menggunakan Bahasa pemrograman R seperti yang ditampilkan pada figur 10.7



```
Residuals:
    Min      1Q  Median      3Q     Max
-40.921  -2.919  -0.673   1.896 124.338

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       -2.057e+03  6.110e+01 -33.665   <2e-16 ***
Name              -1.191e-04  1.715e-04  -0.694    0.487
Location           2.543e-03  2.764e-02   0.092    0.927
Year               1.027e+00  3.049e-02  33.700   <2e-16 ***
Kilometers_Driven  1.006e-06  9.199e-07   1.094    0.274
Fuel_Type         -2.535e+00  2.126e-01 -11.921   <2e-16 ***
Transmission       3.103e+00  2.456e-01  12.637   <2e-16 ***
Owner_Type         6.687e-02  1.675e-01   0.399    0.690
Mileage           -2.941e-01  2.966e-02  -9.915   <2e-16 ***
Engine             3.108e-03  3.734e-04   8.325   <2e-16 ***
Power              9.101e-02  3.439e-03  26.465   <2e-16 ***
Seats             -1.765e+00  1.306e-01 -13.509   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.343 on 5963 degrees of freedom
Multiple R-squared:  0.6801,    Adjusted R-squared:  0.6796
F-statistic:  1153 on 11 and 5963 DF,  p-value: < 2.2e-16
```

***Figur 10.7**: Summary dari model linear awal*

Dengan R, kami mendapatkan *summary* dari model yang kita buat. Maka, model kami menjadi:

$$Price = -2057 - 0.0001Name + 0.0025Location + 1.027Year + 0.000001Kilometers\_Driven - 2.535Fuel\_Type + 3.103Transmission + 0.067Owner\_Type - 0.29Mileage + 0.0031Engine + 0.091Power - 1.77Seats$$

Berdasarkan figur 10.7, kita dapat melihat bahwa tidak semua variabel yang ada adalah variabel yang signifikan terhadap model. Kami menggunakan metode *Stepwise (Both Direction)* untuk menghilangkan variabel yang tidak signifikan terhadap model.

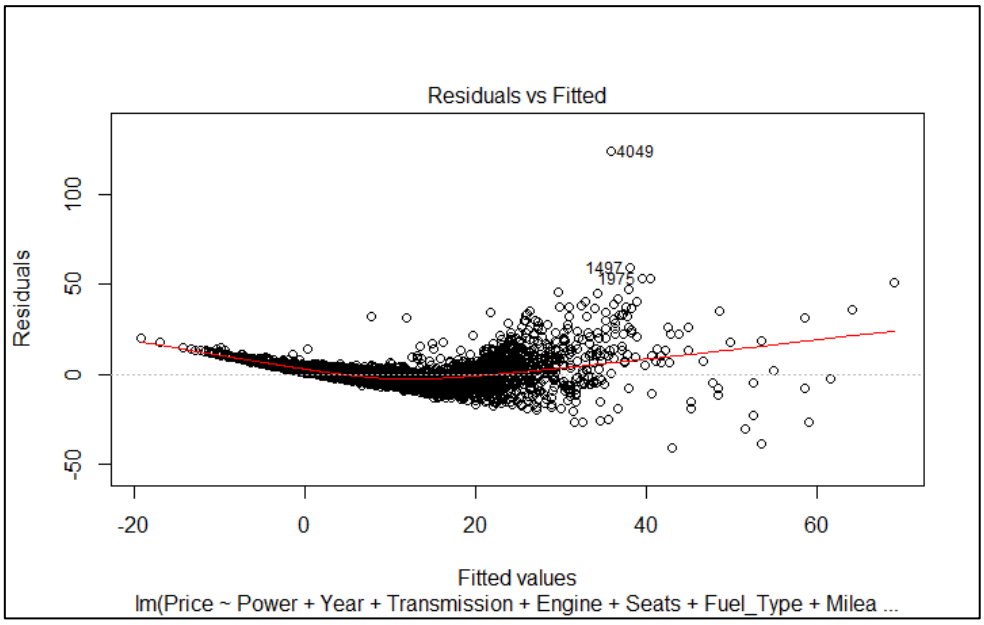
```
Residuals:
    Min      1Q  Median      3Q     Max
-40.988  -2.924  -0.668   1.901 124.213

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.038e+03  5.656e+01 -36.028   <2e-16 ***
Power        9.105e-02  3.433e-03  26.521   <2e-16 ***
Year         1.018e+00  2.826e-02  36.028   <2e-16 ***
Transmission 3.097e+00  2.451e-01  12.636   <2e-16 ***
Engine       3.102e-03  3.719e-04   8.341   <2e-16 ***
Seats       -1.757e+00  1.304e-01 -13.473   <2e-16 ***
Fuel_Type   -2.561e+00  2.115e-01 -12.109   <2e-16 ***
Mileage     -2.934e-01  2.963e-02  -9.903   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.342 on 5967 degrees of freedom
Multiple R-squared:   0.68,     Adjusted R-squared:  0.6797
F-statistic:  1812 on 7 and 5967 DF,  p-value: < 2.2e-16
```

***Figur 10.8**: Summary dari model final*

Figur 10.8 diatas menampilkan variabel-variabel yang signifikan terhadap model. Maka, model kita menjadi

$$Price = -2038 + 0.091Power + 1.018Year + 3.1Transmission + 0.003Engine - 1.76Seats - 2.56Fuel\_Type - 0.29Mileage$$

Perbandingan nilai *residual* dengan nilai *fitted* digambarkan pada figur 10.9 dibawah



***Figur 10.9**: Plot Residual vs Fitted*

Maka, dari model diatas, kita dapat mengambil kesimpulan bahwa faktor-faktor yang mempengaruhi harga mobil bekas di India adalah tenaga maksimum dari mesin

('Power'), tahun edisi mobil ('Year'), tipe transmisi ('Transmission'), kapasitas mesin ('Engine'), jumlah kursi ('Seats'), tipe bahan bakar ('Fuel_Type'), dan tingkat konsumsi bahan bakar ('Mileage').

Hasil Analisis Tambahan

Dalam menjawab pertanyaan-pertanyaan diatas, kami menemukan beberapa hal yang dapat dianalisis lebih lanjut teruatama dalam korelasi antara parameter-parameter mobil bekas terhadap pengaruhnya ke harga mobil tersebut. Beberapa parameter yang kami anggap memiliki korelasi terhadap harga dari mobil bekas adalah "Engine" dan "Seats" yang akan dibahas pada paragraf di bawah.

Korelasi antara "Engine" dan "Price" dari mobil dalam basis manufaktur terdapat pada penggunaan proses manufaktur *drilling* dan *boring*. Kolom "Engine" pada kasus ini mengacu kepada ukuran silinder pembakaran piston didalam mesin dalam satuan volume (cc). Semakin besar ukuran silinder, semakin besar pula cc yang tertera dalam spesifikasi mesin. Hal ini tentunya berpengaruh terhadap daya yang dibutuhkan mesin *drilling* dan *boring* untuk memakan material kerja dimana kedua parameter, daya dan ukuran makan mesin, tersebut memiliki hubungan berbading lurus.

Ketika melihat korelasi antara jumlah kursi dalam mobil atau "Seats" dan "Price" dari data yang kami analisis, kami menemukan bahwa variabel "Seats" juga memengaruhi "Price" dari mobil yang akan dijual. Jumlah kursi dalam mobil tentunya memiliki efek pada ukuran kesuluruhan mobil itu sendiri – semakin banyak kursi dalam mobil, semakin besar volume mobil tersebut karena perlunya ruang yang lebih besar untuk memuat kursi. Mobil yang semakin besar tentunya akan memiliki harga yang lebih tinggi. Oleh karena itu, jumlah kursi dalam mobil dan harga mobil tersebut memiliki korelasi positif.

Kesimpulan

Pertanyaan nomor 1 hingga nomor 9 menitik beratkan cara kita mendapatkan informasi tertentu dari sebuah variabel di sebuah data frame. Sementara pertanyaan nomor 10 membantu kita mencari variabel yang signifikan terhadap variabel 'Price'. Informasi yang kita dapat dari menjawab pertanyaan-pertanyaan yang diberikan dapat membantu kita dalam mengambil berbagai keputusan, seperti hal-hal apa sajakah yang sebaiknya kita tonjolkan saat memasarkan mobil bekas.

Dalam membangun model linear yang bagus, kita sering menemukan berbagai kendala seperti *missing value*, dan data kualitatif. Kita harus menghilangkan data-data yang memiliki *missing value* dan mengkuantifikasi data-data kualitatif agar data-data tersebut diproses dengan baik.

Model akhir yang kita dapat adalah sebagai berikut.

$$Price = -2038 + 0.091Power + 1.018Year + 3.1Transmission + 0.003Engine - 1.76Seats - 2.56Fuel\_Type - 0.29Mileage$$

Dari model tersebut, kita dapat mengetahui pengali pada tiap variabel, sehingga kita tau signifikansi dari perubahan nilai dari tiap-tiap variabel terhadap variabel 'Price'.

Daftar Pusaka (APA style)

Kalpakjian, S., & Schmid, S. R. (2010). *Manufacturing engineering and technology*. Pearson Education.

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2013). *Introduction to Linear Regression Analysis*. Wiley.

Mendenhall, W., & Sincich, T. (2011). *A second course in statistics: regression analysis*. Pearson.

Kanginan, M., & Kartiwa, A. (2010). *Aktif Belajar Matematika: untuk kelas XI*. Pusat Perbukuan Kementerian Pendidikan Nasional .