Extracting Data

Extracting cars data from oto.com. The final table has 3 columns (Tipe Body, Tipe Brand, Brand)

In [152]:
```
#Import packages needed
import requests
import pandas as pd
from bs4 import BeautifulSoup
```

Extracting data for each car type

1. MPV Type

In [153]:
```
# Send a GET request to the web page for MPV type cars
url_mpv = 'https://www.oto.com/car1/mobil-mpv'
response = requests.get(url_mpv)
# Parse the HTML content
soup_mpv = BeautifulSoup(response.content, 'html.parser')

brand_mpv = soup_mpv.find_all('a', class_='vh-name')

# Extract and print the brand names for MPV type
mpv_data = [{'MPV', name.text.strip()} for name in brand_mpv]

# Create a DataFrame
df_mpv = pd.DataFrame(mpv_data, columns=['Tipe Body', 'Tipe Brand'])
```

In [154]:
```
# Display the DataFrame
df_mpv.head()
```

Out[154]:

	Tipe Body	Tipe Brand
0	MPV	Daihatsu Sigra
1	MPV	Mitsubishi Xpander
2	MPV	Toyota Calya
3	MPV	Toyota Avanza
4	MPV	Toyota Kijang Innova

2. SUV Type

In [155]:
```
# Send a GET request to the SUV page
url_suv = 'https://www.oto.com/car1/mobil-suv'
response_suv = requests.get(url_suv)

# Parse the HTML content
soup_suv = BeautifulSoup(response_suv.content, 'html.parser')

# Extract brand names for SUV type
brand_suv = soup_suv.find_all('a', class_='vh-name')

# Extract and print the brand names for SUV type
suv_data = [{'SUV', name.text.strip()} for name in brand_suv]

# Create a DataFrame
df_suv = pd.DataFrame(suv_data, columns=['Tipe Body', 'Tipe Brand'])
```

In [156]:
```
df_suv.tail()
```

Out[156]:

	Tipe Body	Tipe Brand
32	SUV	MG VS HEV
33	SUV	GWM Tank 500
34	SUV	GWM Haval H6
35	SUV	Chery Tiggo 5x
36	SUV	Wuling Almaz RS

3. Crossover Type

In [157]:
```
# Send a GET request to the Crossover page
url_crossover = 'https://www.oto.com/car1/mobil-crossover'
response_crossover = requests.get(url_crossover)

# Parse the HTML content
soup_crossover = BeautifulSoup(response_crossover.content, 'html.parser')

# Extract brand names for Crossover type
brand_crossover = soup_crossover.find_all('a', class_='vh-name')

# Extract and print the brand names for Crossover type
crossover_data = [{'Crossover', name.text.strip()} for name in brand_crossover]

# Create a DataFrame
df_crossover = pd.DataFrame(crossover_data, columns=['Tipe Body', 'Tipe Brand'])
```

In [158]:
```
df_crossover.head()
```

Out[158]:

	Tipe Body	Tipe Brand
0	Crossover	Honda WR-V
1	Crossover	Daihatsu Terios
2	Crossover	Hyundai Creta
3	Crossover	Honda HRV
4	Crossover	Chery Omoda 5

4. Hatchback Type

In [159]:
```
# Send a GET request to the Hatchback page
url_hatchback = 'https://www.oto.com/car1/mobil-hatchback'
response_hatchback= requests.get(url_hatchback)

# Parse the HTML content
soup_hatchback = BeautifulSoup(response_hatchback.content, 'html.parser')

# Extract brand names for Hatchback type
brand_hatchback = soup_hatchback.find_all('a', class_='vh-name')

# Extract and print the brand names for Hatchback type
hatchback_data = [{'Hatchback', name.text.strip()} for name in brand_hatchback]

# Create a DataFrame
df_hatchback = pd.DataFrame(hatchback_data, columns=['Tipe Body', 'Tipe Brand'])
```

In [160]:
```
df_hatchback.head()
```

Out[160]:

	Tipe Body	Tipe Brand
0	Hatchback	Honda Brio
1	Hatchback	Daihatsu Ayla
2	Hatchback	Toyota Agya
3	Hatchback	Renault KWID
4	Hatchback	Wuling Binguo EV

5. Sedan Type

In [161]:
```
# Send a GET request to the Sedan page
url_sedan = 'https://www.oto.com/car1/mobil-sedans'
response_sedan= requests.get(url_sedan)

# Parse the HTML content
soup_sedan = BeautifulSoup(response_sedan.content, 'html.parser')

# Extract brand names for Sedan type
brand_sedan = soup_sedan.find_all('a', class_='vh-name')

# Extract and print the brand names for Sedan type
sedan_data = [{'Sedan', name.text.strip()} for name in brand_sedan]

# Create a DataFrame
df_sedan = pd.DataFrame(sedan_data, columns=['Tipe Body', 'Tipe Brand'])
```

In [162]:
```
df_sedan.head()
```

Out[162]:

	Tipe Body	Tipe Brand
0	Sedan	MG 5 GT
1	Sedan	BYD Seal
2	Sedan	Honda Civic RS
3	Sedan	Tesla Model S
4	Sedan	Toyota Vios

6. Pickup Truck Type

In [163]:
```
# Send a GET request to the Pickup Truck page
url_pickup = 'https://www.oto.com/car1/mobil-pickup-trucks'
response_pickup= requests.get(url_pickup)

# Parse the HTML content
soup_pickup = BeautifulSoup(response_pickup.content, 'html.parser')

# Extract brand names for Pickup Truck type
brand_pickup = soup_pickup.find_all('a', class_='vh-name')

# Extract and print the brand names for Pickup Truck type
pickup_data = [{'Pickup Truck', name.text.strip()} for name in brand_pickup]

# Create a DataFrame
df_pickup = pd.DataFrame(pickup_data, columns=['Tipe Body', 'Tipe Brand'])
```

In [164]:
```
df_pickup.head()
```

Out[164]:

	Tipe Body	Tipe Brand
0	Pickup Truck	Daihatsu Gran Max PU
1	Pickup Truck	Suzuki Carry
2	Pickup Truck	Mitsubishi L300
3	Pickup Truck	Toyota Hilux
4	Pickup Truck	Isuzu Traga

7. Minivan Type

In [165]:
```
# Send a GET request to the Minivan page
url_minivans = 'https://www.oto.com/car1/mobil-minivans'
response_minivans= requests.get(url_minivans)

# Parse the HTML content
soup_minivans = BeautifulSoup(response_minivans.content, 'html.parser')

# Extract brand names for Minivan type
brand_minivans = soup_minivans.find_all('a', class_='vh-name')

# Extract and print the brand names for Minivan type
minivans_data = [{'Minivan', name.text.strip()} for name in brand_minivans]

# Create a DataFrame
df_minivans = pd.DataFrame(minivans_data, columns=['Tipe Body', 'Tipe Brand'])
```

In [166]:
```
df_minivans.head()
```

Out[166]:

	Tipe Body	Tipe Brand
0	Minivan	Toyota Hiace
1	Minivan	Suzuki APV Arena
2	Minivan	DFSK Gelora Electric
3	Minivan	Mercedes Benz V-Class
4	Minivan	Suzuki APV Arena

8. Coupe Type

In [167]:
```
# Send a GET request to the Coupe page
url_coupe = 'https://www.oto.com/car1/mobil-coupe'
response_coupe= requests.get(url_coupe)

# Parse the HTML content
soup_coupe = BeautifulSoup(response_coupe.content, 'html.parser')

# Extract brand names for Coupe type
brand_coupe = soup_coupe.find_all('a', class_='vh-name')

# Extract and print the brand names for Coupe type
coupe_data = [{'Coupe', name.text.strip()} for name in brand_coupe]

# Create a DataFrame
df_coupe = pd.DataFrame(coupe_data, columns=['Tipe Body', 'Tipe Brand'])
```

In [168]:
```
df_coupe.head()
```

Out[168]:

	Tipe Body	Tipe Brand
0	Coupe	Porsche 911
1	Coupe	Porsche Taycan
2	Coupe	Lamborghini Aventador
3	Coupe	Mclaren 765LT
4	Coupe	Mclaren 720S Spider

9. Van Type

In [169]:
```
# Send a GET request to the Van page
url_van = 'https://www.oto.com/car1/mobil-van'
response_van= requests.get(url_van)

# Parse the HTML content
soup_van = BeautifulSoup(response_van.content, 'html.parser')

# Extract brand names for Van type
brand_van = soup_van.find_all('a', class_='vh-name')

# Extract and print the brand names for Van type
van_data = [{'Van', name.text.strip()} for name in brand_van]

# Create a DataFrame
df_van = pd.DataFrame(van_data, columns=['Tipe Body', 'Tipe Brand'])
```

In [170]:
```
df_van.head()
```

Out[170]:

	Tipe Body	Tipe Brand
0	Van	Isuzu Traga
1	Van	Daihatsu Gran Max MB
2	Van	Daihatsu Luxio
3	Van	DFSK Gelora
4	Van	Mercedes Benz Sprinter

10. Wagon Type

In [171]:
```
# Send a GET request to the Wagon page
url_wagon = 'https://www.oto.com/car1/mobil-wagon'
response_wagon= requests.get(url_wagon)

# Parse the HTML content
soup_wagon = BeautifulSoup(response_wagon.content, 'html.parser')

# Extract brand names for Wagon type
brand_wagon = soup_wagon.find_all('a', class_='vh-name')

# Extract and print the brand names for Wagon type
wagon_data = [{'Wagon', name.text.strip()} for name in brand_wagon]

# Create a DataFrame
df_wagon = pd.DataFrame(wagon_data, columns=['Tipe Body', 'Tipe Brand'])
```

In [172]:
```
df_wagon
```

Out[172]:

	Tipe Body	Tipe Brand
0	Wagon	Porsche Taycan
1	Wagon	BMW M3 Touring
2	Wagon	Subaru Outback
3	Wagon	Mazda 6 Estate
4	Wagon	BMW 5 Series Touring
5	Wagon	Subaru WRX Wagon
6	Wagon	Audi RS 4 Avant
7	Wagon	Subaru Outback
8	Wagon	Mazda 6 Estate
9	Wagon	Subaru WRX Wagon
10	Wagon	Subaru Outback
11	Wagon	BMW M3 Touring

Combining Dataframes

combining all cars data

In [173]:
```
df_cars=pd.concat([df_mpv, df_suv, df_crossover, df_hatchback, df_sedan, df_pickup, df_minivans, df_coupe, df_van, df_wagon], axis=0,  ignore_index=True)
df_cars
```

Out[173]:

	Tipe Body	Tipe Brand
0	MPV	Daihatsu Sigra
1	MPV	Mitsubishi Xpander
2	MPV	Toyota Calya
3	MPV	Toyota Avanza
4	MPV	Toyota Kijang Innova
...	...	...
264	Wagon	Subaru Outback
265	Wagon	Mazda 6 Estate
266	Wagon	Subaru WRX Wagon
267	Wagon	Subaru Outback
268	Wagon	BMW M3 Touring

269 rows × 2 columns

In [174]:
```
df_cars.drop_duplicates(inplace=True, ignore_index=True)
df_cars.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 175 entries, 0 to 174
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Tipe Body   175 non-null    object
 1   Tipe Brand  175 non-null    object
dtypes: object(2)
memory usage: 2.9+ KB
```

In [175]:
```
#Adding Brand Column
df_cars['Brand'] = df_cars['Tipe Brand'].str.split().str[0]
df_cars
```

Out[175]:

	Tipe Body	Tipe Brand	Brand
0	MPV	Daihatsu Sigra	Daihatsu
1	MPV	Mitsubishi Xpander	Mitsubishi
2	MPV	Toyota Calya	Toyota
3	MPV	Toyota Avanza	Toyota
4	MPV	Toyota Kijang Innova	Toyota
...	...	...	...
170	Wagon	Subaru Outback	Subaru
171	Wagon	Mazda 6 Estate	Mazda
172	Wagon	BMW 5 Series Touring	BMW
173	Wagon	Subaru WRX Wagon	Subaru
174	Wagon	Audi RS 4 Avant	Audi

175 rows × 3 columns