Car Dimensions: A Look Through Time

Have cars of the same model grown larger over the years, or does the change in size depend on the specific model?

To explore this question, we have a dataset in the file `car-dimensions.csv`, which presents the `length`, `width`, and `height` (all in millimeters) for 53 car models. This dataset includes measurements from the earliest generation of each model that we could find, as well as data from the latest model.

By analyzing this information, we can uncover trends and patterns in the evolution of car dimensions over time.

Overview of the dataframe

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('car-dimensions.csv')
df
```

✓ 0.3s Python

	brand_model	length	width	height	year	generation
0	Audi A4	4520	1733	1415	1994	first
1	Audi A4	4762	1847	1427	2015	latest
2	BMW 3 Series	4355	1610	1380	1975	first
3	BMW 3 Series	4829	1827	1442	2018	latest
4	BMW 7 Series	4860	1800	1430	1986	first
...	...	...	...	...	...	...
101	Toyota Sienna	5180	1990	1780	2020	latest
102	Volkswagen Golf	3705	1610	1395	1974	first
103	Volkswagen Golf	4633	1789	1498	2020	latest
104	Volkswagen Type 2	4280	1720	1940	1950	first
105	Volkswagen Type 2	5304	1904	2477	2015	latest

106 rows × 6 columns

First, we had to create separate data frames for both first and latest generation models, I also calculated for their volumes as well

```
# Get all the cars from the first generation
first_generation = df[df['generation'] == 'first']
# Volume of the cars
first_generation['early_gen_volume'] = (first_generation['length'] * first_generation['width'] * first_generation['height'])

first_generation
```

```
# Get all the cars from the latest generation
latest_generation = df[df['generation'] == 'latest']
# Volume of the cars
latest_generation['latest_gen_volume'] = (latest_generation['length'] * latest_generation['width'] * latest_generation['height'])
latest_generation
```

✓ 0.0s Python

Since we don’t need the year, length, width, and height anymore. We created another data frame that we can now use for the data visualization.

```
volume_compare = first_generation.merge(latest_generation, on='brand_model', how='left')
# Select only desired columns
columns_to_keep = ['brand_model', 'early_gen_volume', 'latest_gen_volume']
volume_compare = volume_compare[columns_to_keep]

volume_compare
```

✓ 0.0s Python

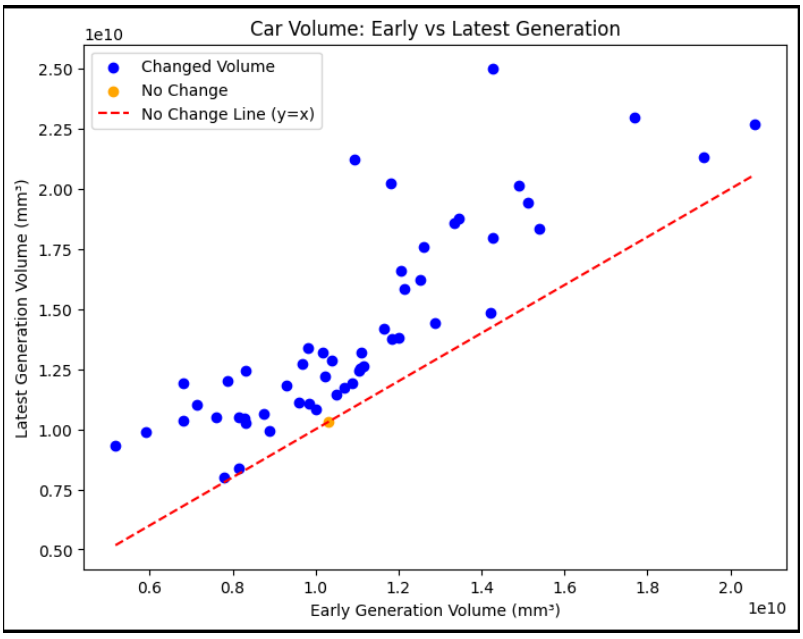
Time for the data visualization and inference of the data. I used a scatter plot in order to compare the Early and Latest Generation volumes.

```
# Identify unchanged points
unchanged = volume_compare[volume_compare['early_gen_volume'] == volume_compare['latest_gen_volume']]
changed = volume_compare[volume_compare['early_gen_volume'] != volume_compare['latest_gen_volume']]

plt.figure(figsize=(8, 6))
plt.scatter(changed['early_gen_volume'], changed['latest_gen_volume'], color='blue', label='Changed Volume')
plt.scatter(unchanged['early_gen_volume'], unchanged['latest_gen_volume'], color='orange', label='No Change')
plt.plot([volume_compare['early_gen_volume'].min(), volume_compare['early_gen_volume'].max()],
         [volume_compare['early_gen_volume'].min(), volume_compare['early_gen_volume'].max()],
         color='red', linestyle='--', label='No Change Line (y=x)')
plt.xlabel('Early Generation Volume (mm³)')
plt.ylabel('Latest Generation Volume (mm³)')
plt.title('Car Volume: Early vs Latest Generation')
plt.legend()
plt.show()
```

✓ 0.1s Python

Scatter plot



Inference:

The purpose of the red line is to serve as a reference for the two variables. It shows that the models above it means that these cars increased their volume in the latest generation of the model, while their distance from the line represent how big of a change happened. We can see that almost every brand model has increased their volume when the new generation arrives. There's only 1 brand model that did not change.