vehicle_images_dataset

The unit of observation is a vehicle image, therefore each row pertains to a vehicle image. Each subsection below pertains to a variable describing the observations, with the heading being the variable name. The reported numbers in each variable subsection in the format n(m) represent n total observations for the variable and m number of missing values for the variable.

```
! git clone https://github.com/jillianhaig/Project3_DS4002 # so we can access data loaded from shared github
```

```
Cloning into 'Project3_DS4002'...
remote: Enumerating objects: 132, done.
remote: Counting objects: 100% (132/132), done.
remote: Compressing objects: 100% (124/124), done.
remote: Total 132 (delta 59), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (132/132), 223.69 KiB | 3.39 MiB/s, done.
Resolving deltas: 100% (59/59), done.
```

```
from google.colab import drive
from google.colab import files
import zipfile
import os
import pandas as pd

from google.colab import auth
auth.authenticate_user()

# Mount Google Drive to access the dataset
drive.mount('/content/drive')

# Path to the zip file on Google Drive
zip_file_path = '/content/drive/My Drive/DS4002 Projects/Project 3/vehicleimages.zip'

# Directory where you want to extract the files
extract_to_path = '/content/vehicleimages'

# Unzip the dataset
with zipfile.ZipFile(zip_file_path, 'r') as zip_ref:
    zip_ref.extractall(extract_to_path)

print(f"Dataset unzipped to: {extract_to_path}")

image_paths = []
labels = []

vehicle_types = os.listdir(extract_to_path)

# For each subdirectory, get image paths
for vehicle_type in vehicle_types:
    vehicle_folder = os.path.join(extract_to_path, vehicle_type)

    if os.path.isdir(vehicle_folder):
        for img_file in os.listdir(vehicle_folder):
            if img_file.lower().endswith(('.jpg', '.jpeg', '.png', '.bmp')):
                image_path = os.path.join(vehicle_folder, img_file)
                image_paths.append(image_path)
                labels.append(vehicle_type)

df = pd.DataFrame({'image_path': image_paths, 'label': labels})
```

```
Mounted at /content/drive
Dataset unzipped to: /content/vehicleimages
```

image_path

This variable takes on string values and represents the path to access the image content when mounted to google colab.

```
print(df["image_path"].count(), "(", "0",")") # we already know all file paths are valid and there are no NAs because of our code working pr
```

```
4356 ( 0 )
```

```
df['image_path'].describe()
```

	image_path
count	4356
unique	4356
top	/content/vehicleimages/sedan/PHOTO_744.jpg
freq	1

label

This variable takes on values representing the vehicle type of the image being represented.

```
print(df["label"].count(), "(", sum(df["label"].isna()),")")
df["label"].unique()
```

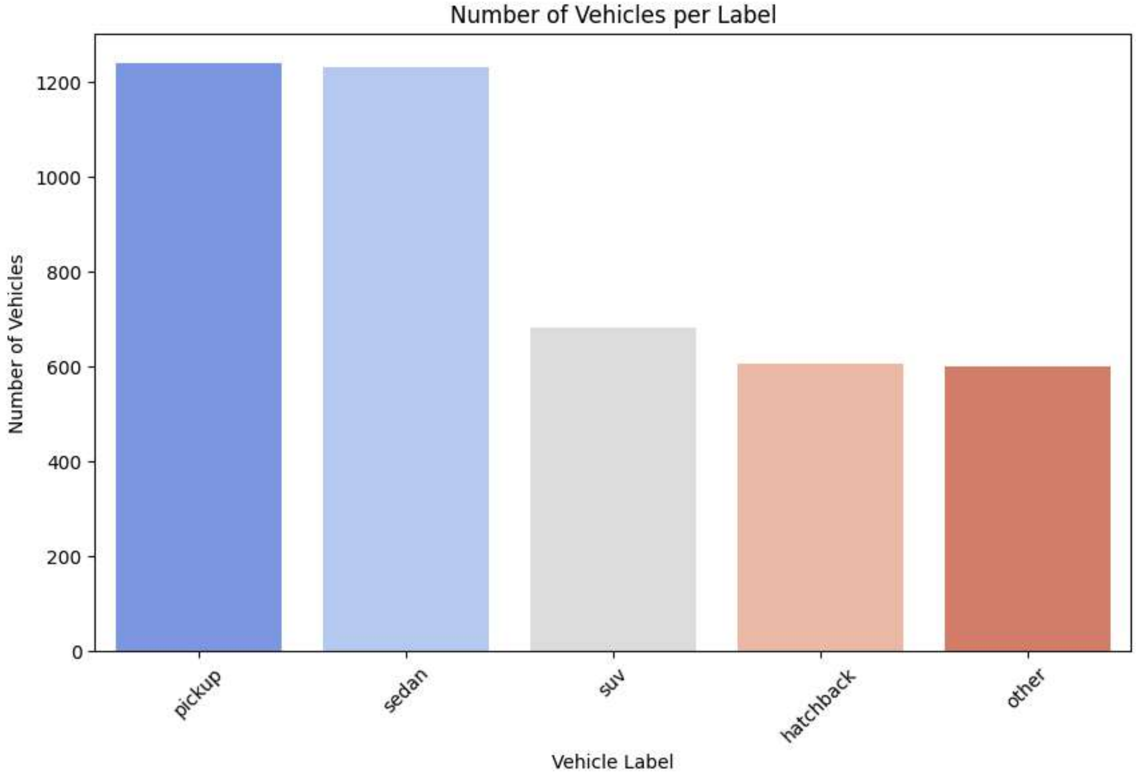
```
4356 ( 0 )
array(['sedan', 'hatchback', 'other', 'pickup', 'suv'], dtype=object)
```

```
df["label"].describe()
```

	label
count	4356
unique	5
top	pickup
freq	1240

```
import seaborn as sns
import matplotlib.pyplot as plt

# counts vehicles per label
label_counts = df['label'].value_counts()

# creates bar plot
plt.figure(figsize=(10, 6))
sns.barplot(x=label_counts.index, y=label_counts.values, palette='coolwarm')

plt.xlabel('Vehicle Label')
plt.ylabel('Number of Vehicles')
plt.title('Number of Vehicles per Label')

# shows plot
plt.xticks(rotation=45)
plt.show()
```

```
<ipython-input-12-f75f063f0d90>:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend

  sns.barplot(x=label_counts.index, y=label_counts.values, palette='coolwarm')
```


Number of Vehicles per Label
Number of Vehicles
pickup
sedan
suv
hatchback
other
Vehicle Label