Filter, Drop Nulls, Dedupe

Use `data_08_v1.csv` and `data_18_v1.csv` . You should've created these data files in the previous section: *Cleaning Column Labels*.

In [11]:
```
import pandas as pd
import numpy as np

# load datasets

df_08 = pd.read_csv('data_08_v1.csv')
df_18 = pd.read_csv('data_18_v1.csv')
```

In [12]:
```
df_08.head()
```

Out[12]:

	model	displ	cyl	trans	drive	fuel	cert_region	veh_class	air_pollution_score	city_mpg
0	ACURA MDX	3.7	(6 cyl)	Auto-S5	4WD	Gasoline	CA	SUV	7	15
1	ACURA MDX	3.7	(6 cyl)	Auto-S5	4WD	Gasoline	FA	SUV	6	15
2	ACURA RDX	2.3	(4 cyl)	Auto-S5	4WD	Gasoline	CA	SUV	7	17
3	ACURA RDX	2.3	(4 cyl)	Auto-S5	4WD	Gasoline	FA	SUV	6	17
4	ACURA RL	3.5	(6 cyl)	Auto-S5	4WD	Gasoline	CA	midsize car	7	16

In [13]:
```
# view dimensions of dataset
df_08.shape
```

Out[13]:
```
(2404, 14)
```

In [14]:
```
# view dimensions of dataset
df_18.shape
```

Out[14]:
```
(1611, 14)
```

Filter by Certification Region

In [15]:
```
# filter datasets for rows following California standards
df_08 = df_08.query('cert_region =="CA"')
df_18 = df_18.query('cert_region =="CA"')
```

In [16]:
```
# confirm only certification region is California
df_08['cert_region'].unique()
```

Out[16]:
```
array(['CA'], dtype=object)
```

In [17]:
```
# confirm only certification region is California
df_18['cert_region'].unique()
```

Out[17]:
```
array(['CA'], dtype=object)
```

In [18]:
```
# drop certification region columns form both datasets

df_08.drop('cert_region', axis=1, inplace=True)
df_18.drop('cert_region', axis=1, inplace=True)
```

In [19]:
```
df_08.shape
```

Out[19]:
```
(1084, 13)
```

In [20]:
```
df_18.shape
```

Out[20]:
```
(798, 13)
```

Drop Rows with Missing Values

In [23]:
```
# view missing value count for each feature in 2008
df_08.isnull().sum()
```

Out[23]:
```
model                    0
displ                    0
cyl                     75
trans                   75
drive                   37
fuel                     0
veh_class                0
air_pollution_score      0
city_mpg                75
hwy_mpg                 75
cmb_mpg                 75
greenhouse_gas_score    75
smartway                 0
dtype: int64
```

In [24]:
```
# view missing value count for each feature in 2018
df_18.isnull().sum()
```

Out[24]:
```
model                   0
displ                   1
cyl                     1
trans                   0
drive                   0
fuel                    0
veh_class               0
air_pollution_score     0
city_mpg                0
hwy_mpg                 0
cmb_mpg                 0
greenhouse_gas_score    0
smartway                0
dtype: int64
```

In [28]:
```
# drop rows with any null values in both datasets

df_08.dropna(inplace=True)
df_18.dropna(inplace=True)
```

In [29]:
```
# checks if any of columns in 2008 have null values - should print False
df_08.isnull().sum().any()
```

Out[29]:
```
False
```

In [30]:
```
# checks if any of columns in 2018 have null values - should print False
df_18.isnull().sum().any()
```

Out[30]:
```
False
```

Dedupe Data

In [32]:
```
# print number of duplicates in 2008 and 2018 datasets
df_08.duplicated().sum()
```

Out[32]:
```
23
```

In [33]:
```
df_18.duplicated().sum()
```

Out[33]:
```
3
```

In [34]:
```
# drop duplicates in both datasets
df_08.drop_duplicates(inplace=True)
df_18.drop_duplicates(inplace=True)
```

In [36]:
```
# print number of duplicates again to confirm dedupe - should both be 0

df_08.duplicated().sum()
df_18.duplicated().sum()
```

Out[36]:
```
0
```

In [37]:
```
# save progress for the next section
df_08.to_csv('data_08_v2.csv', index=False)
df_18.to_csv('data_18_v2.csv', index=False)
```

In [ ]:
```

```