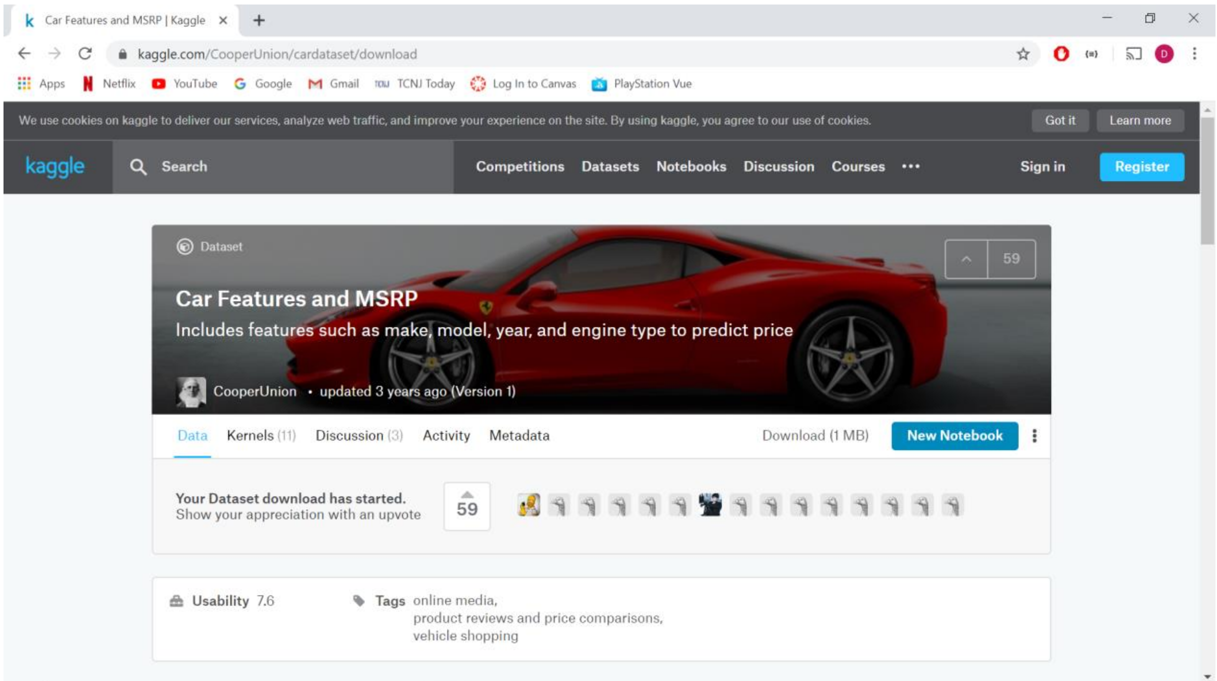
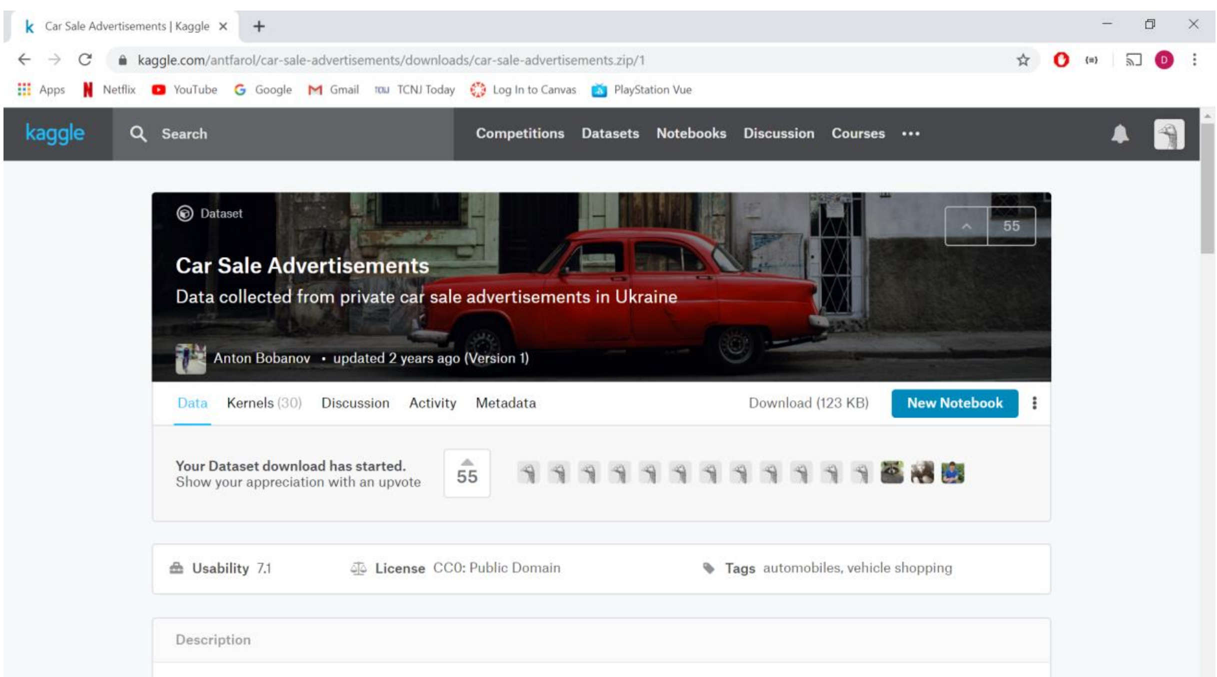
Danyelle Tolud

ETL Project Final Report

Extract

The datasets were found on Kaggle:

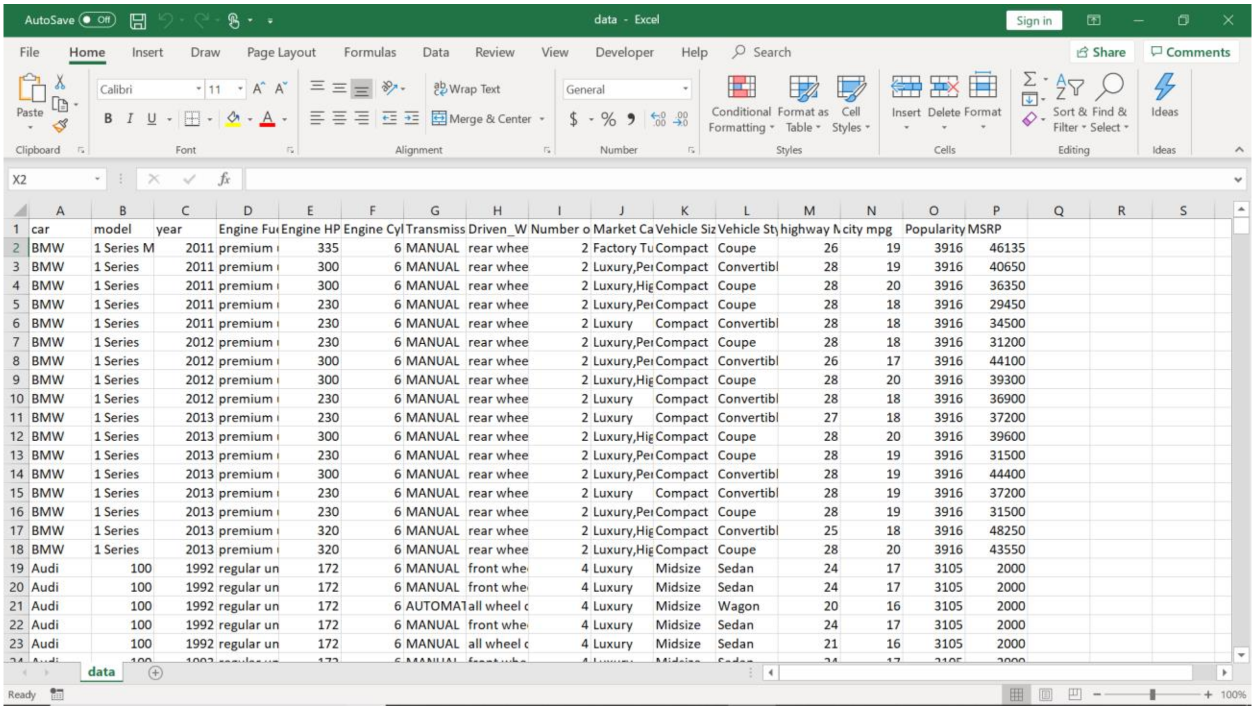




Downloaded the CSV Files:

(car data and popularity)

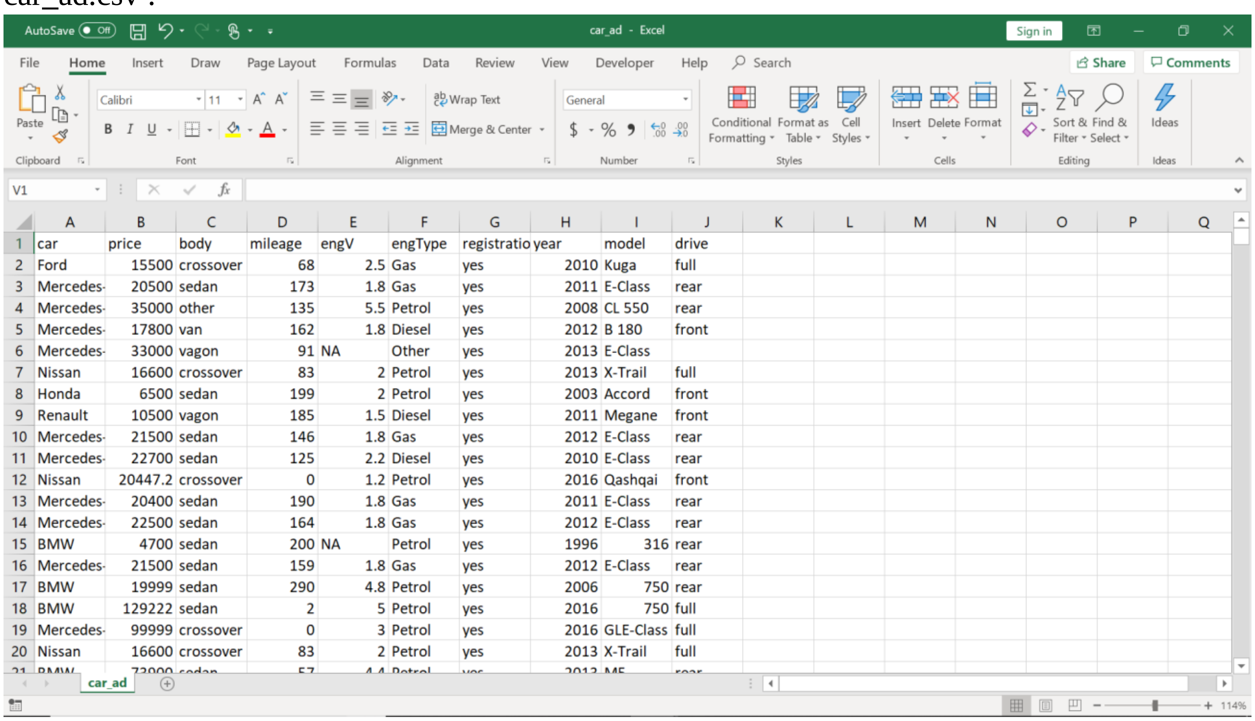
data.csv :



	car	model	year	Engine Fu	Engine HP	Engine Cyl	Transmiss	Driven_W	Number o	Market Ca	Vehicle Siz	Vehicle Sty	highway M	city mpg	Popularity	MSRP
2	BMW	1 Series M	2011	premium	335	6	MANUAL	rear whee	2	Factory Tu	Compact	Coupe	26	19	3916	46135
3	BMW	1 Series	2011	premium	300	6	MANUAL	rear whee	2	Luxury,Pe	Compact	Convertibl	28	19	3916	40650
4	BMW	1 Series	2011	premium	300	6	MANUAL	rear whee	2	Luxury,Hig	Compact	Coupe	28	20	3916	36350
5	BMW	1 Series	2011	premium	230	6	MANUAL	rear whee	2	Luxury,Pe	Compact	Coupe	28	18	3916	29450
6	BMW	1 Series	2011	premium	230	6	MANUAL	rear whee	2	Luxury	Compact	Convertibl	28	18	3916	34500
7	BMW	1 Series	2012	premium	230	6	MANUAL	rear whee	2	Luxury,Pe	Compact	Coupe	28	18	3916	31200
8	BMW	1 Series	2012	premium	300	6	MANUAL	rear whee	2	Luxury,Pe	Compact	Convertibl	26	17	3916	44100
9	BMW	1 Series	2012	premium	300	6	MANUAL	rear whee	2	Luxury,Hig	Compact	Coupe	28	20	3916	39300
10	BMW	1 Series	2012	premium	230	6	MANUAL	rear whee	2	Luxury	Compact	Convertibl	28	18	3916	36900
11	BMW	1 Series	2013	premium	230	6	MANUAL	rear whee	2	Luxury	Compact	Convertibl	27	18	3916	37200
12	BMW	1 Series	2013	premium	300	6	MANUAL	rear whee	2	Luxury,Hig	Compact	Coupe	28	20	3916	39600
13	BMW	1 Series	2013	premium	230	6	MANUAL	rear whee	2	Luxury,Pe	Compact	Coupe	28	19	3916	31500
14	BMW	1 Series	2013	premium	300	6	MANUAL	rear whee	2	Luxury,Pe	Compact	Convertibl	28	19	3916	44400
15	BMW	1 Series	2013	premium	230	6	MANUAL	rear whee	2	Luxury	Compact	Convertibl	28	19	3916	37200
16	BMW	1 Series	2013	premium	230	6	MANUAL	rear whee	2	Luxury,Pe	Compact	Coupe	28	19	3916	31500
17	BMW	1 Series	2013	premium	320	6	MANUAL	rear whee	2	Luxury,Hig	Compact	Convertibl	25	18	3916	48250
18	BMW	1 Series	2013	premium	320	6	MANUAL	rear whee	2	Luxury,Hig	Compact	Coupe	28	20	3916	43550
19	Audi	100	1992	regular un	172	6	MANUAL	front whe	4	Luxury	Midsize	Sedan	24	17	3105	2000
20	Audi	100	1992	regular un	172	6	MANUAL	front whe	4	Luxury	Midsize	Sedan	24	17	3105	2000
21	Audi	100	1992	regular un	172	6	AUTOMAT	all wheel c	4	Luxury	Midsize	Wagon	20	16	3105	2000
22	Audi	100	1992	regular un	172	6	MANUAL	front whe	4	Luxury	Midsize	Sedan	24	17	3105	2000
23	Audi	100	1992	regular un	172	6	MANUAL	all wheel c	4	Luxury	Midsize	Sedan	21	16	3105	2000

(car advertising and sales data)

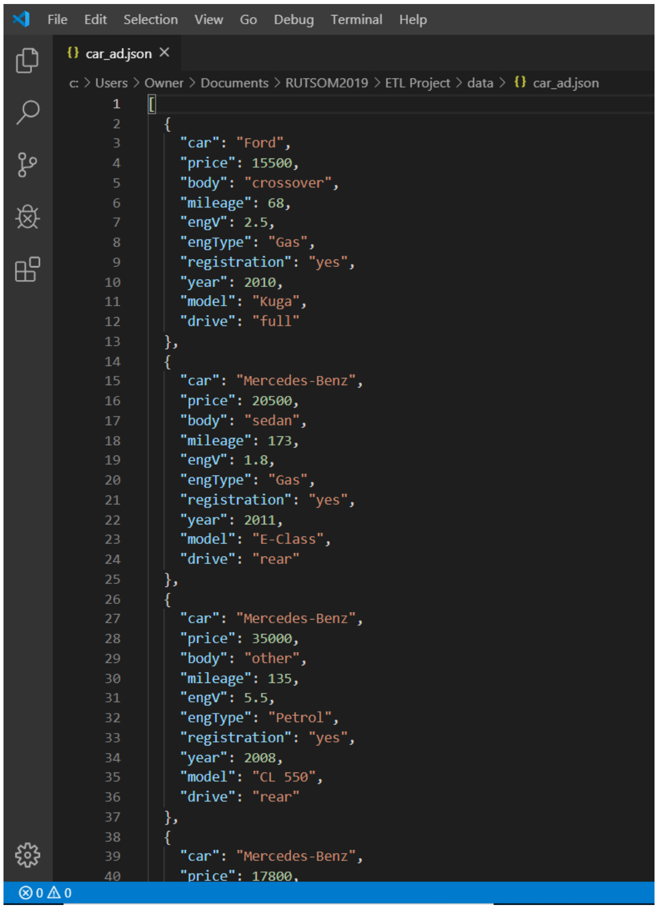
car_ad.csv :



	car	price	body	mileage	engV	engType	registratio	year	model	drive
2	Ford	15500	crossover	68	2.5	Gas	yes	2010	Kuga	full
3	Mercedes-	20500	sedan	173	1.8	Gas	yes	2011	E-Class	rear
4	Mercedes-	35000	other	135	5.5	Petrol	yes	2008	CL 550	rear
5	Mercedes-	17800	van	162	1.8	Diesel	yes	2012	B 180	front
6	Mercedes-	33000	vagon	91	NA	Other	yes	2013	E-Class	
7	Nissan	16600	crossover	83	2	Petrol	yes	2013	X-Trail	full
8	Honda	6500	sedan	199	2	Petrol	yes	2003	Accord	front
9	Renault	10500	vagon	185	1.5	Diesel	yes	2011	Megane	front
10	Mercedes-	21500	sedan	146	1.8	Gas	yes	2012	E-Class	rear
11	Mercedes-	22700	sedan	125	2.2	Diesel	yes	2010	E-Class	rear
12	Nissan	20447.2	crossover	0	1.2	Petrol	yes	2016	Qashqai	front
13	Mercedes-	20400	sedan	190	1.8	Gas	yes	2011	E-Class	rear
14	Mercedes-	22500	sedan	164	1.8	Gas	yes	2012	E-Class	rear
15	BMW	4700	sedan	200	NA	Petrol	yes	1996	316	rear
16	Mercedes-	21500	sedan	159	1.8	Gas	yes	2012	E-Class	rear
17	BMW	19999	sedan	290	4.8	Petrol	yes	2006	750	rear
18	BMW	129222	sedan	2	5	Petrol	yes	2016	750	full
19	Mercedes-	99999	crossover	0	3	Petrol	yes	2016	GLE-Class	full
20	Nissan	16600	crossover	83	2	Petrol	yes	2013	X-Trail	full

converted >>> CSV to JSON

car_ad.json :



Transform

For the ETL process, the data was extracted from the original data.csv file and a converted csv to json car_ad.json file. Then the data was cleaned, transformed, and joined into one dataframe that included data from both datasets, joined by car make, model, and year to include popularity and price from the two different datasets. This was done in order to analyze car popularity and car price together as they are not both included in either dataset but are in the combined dataframe.

jupyter data_etl Last Checkpoint: an hour ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run Code

In [1]:

```python
import pandas as pd
from sqlalchemy import create_engine
```

Store CSV into DataFrame

In [2]:

```python
csv_file = "../data/data.csv"
data_df = pd.read_csv(csv_file)
data_df.head()
```

Out[2]:

	Make	Model	Year	Engine Fuel Type	Engine HP	Engine Cylinders	Transmission Type	Driven_Wheels	Number of Doors	Market Category	Vehicle Size
0	BMW	1 Series M	2011	premium unleaded (required)	335.0	6.0	MANUAL	rear wheel drive	2.0	Factory Tuner,Luxury,High-Performance	Compact
1	BMW	1 Series	2011	premium unleaded (required)	300.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,Performance	Compact
2	BMW	1 Series	2011	premium unleaded (required)	300.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,High-Performance	Compact

jupyter data_etl Last Checkpoint: an hour ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run Code

Create new data with select columns

In [3]:

```python
new_data_df = data_df[['Make', 'Model', 'Year', 'Popularity' ]].copy()
new_data_df.head()
```

Out[3]:

	Make	Model	Year	Popularity
0	BMW	1 Series M	2011	3916
1	BMW	1 Series	2011	3916
2	BMW	1 Series	2011	3916
3	BMW	1 Series	2011	3916
4	BMW	1 Series	2011	3916

Store JSON data into a DataFrame

In [4]:

```python
json_file = "../data/car_ad.json"
car_ad_df = pd.read_json(json_file)
car_ad_df.head()
```

Out[4]:

	body	car	drive	engType	engV	mileage	model	price	registration	year
0	crossover	Ford	full	Gas	2.5	68	Kuga	15500.0	yes	2010
1	sedan	Mercedes-Benz	rear	Gas	1.8	173	E-Class	20500.0	yes	2011
2	other	Mercedes-Benz	rear	Petrol	5.5	135	CL 550	35000.0	yes	2008
3	van	Mercedes-Benz	front	Diesel	1.8	162	B 180	17800.0	yes	2012

jupyter data_etl Last Checkpoint: an hour ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run Code

In [5]:
```
new_car_ad_df = car_ad_df[["car", "model", "year", "price"]].copy()
new_car_ad_df.head()
```

Out[5]:

	car	model	year	price
0	Ford	Kuga	2010	15500.0
1	Mercedes-Benz	E-Class	2011	20500.0
2	Mercedes-Benz	CL 550	2008	35000.0
3	Mercedes-Benz	B 180	2012	17800.0
4	Mercedes-Benz	E-Class	2013	33000.0

Connect to local database

In [8]:
```
rds_connection_string = "postgres:password@localhost:5432/car_db"
engine = create_engine(f'postgresql://{rds_connection_string}')
```

Check for tables

In [9]:
```
engine.table_names()
```

jupyter data_etl Last Checkpoint: an hour ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run Code

Use pandas to load csv converted DataFrame into database

In [10]:
```
new_car_ad_df.to_sql(name='car', con=engine, if_exists='append', index=False)
```

Use pandas to load json converted DataFrame into database

In [11]:
```
new_car_ad_df.to_sql(name='model', con=engine, if_exists='append', index=False)
```

Confirm data has been added by querying the customer_name table

- NOTE: can also check using pgAdmin

In [12]:
```
pd.read_sql_query('select * from car', con=engine).head()
```

Out[12]:

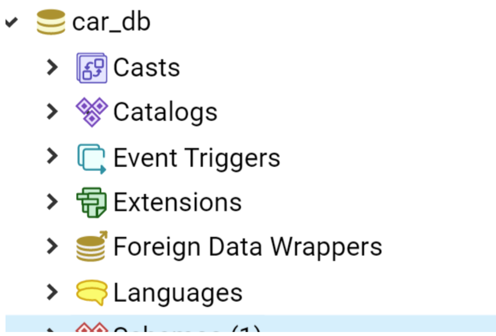
	car	model	year	price
0	Ford	Kuga	2010	15500.0
1	Mercedes-Benz	E-Class	2011	20500.0
2	Mercedes-Benz	CL 550	2008	35000.0
3	Mercedes-Benz	B 180	2012	17800.0
4	Mercedes-Benz	E-Class	2013	33000.0

Confirm data has been added by querying the customer_location table ¶

In [13]:
```
pd.read_sql_query('select * from model', con=engine).head()
```

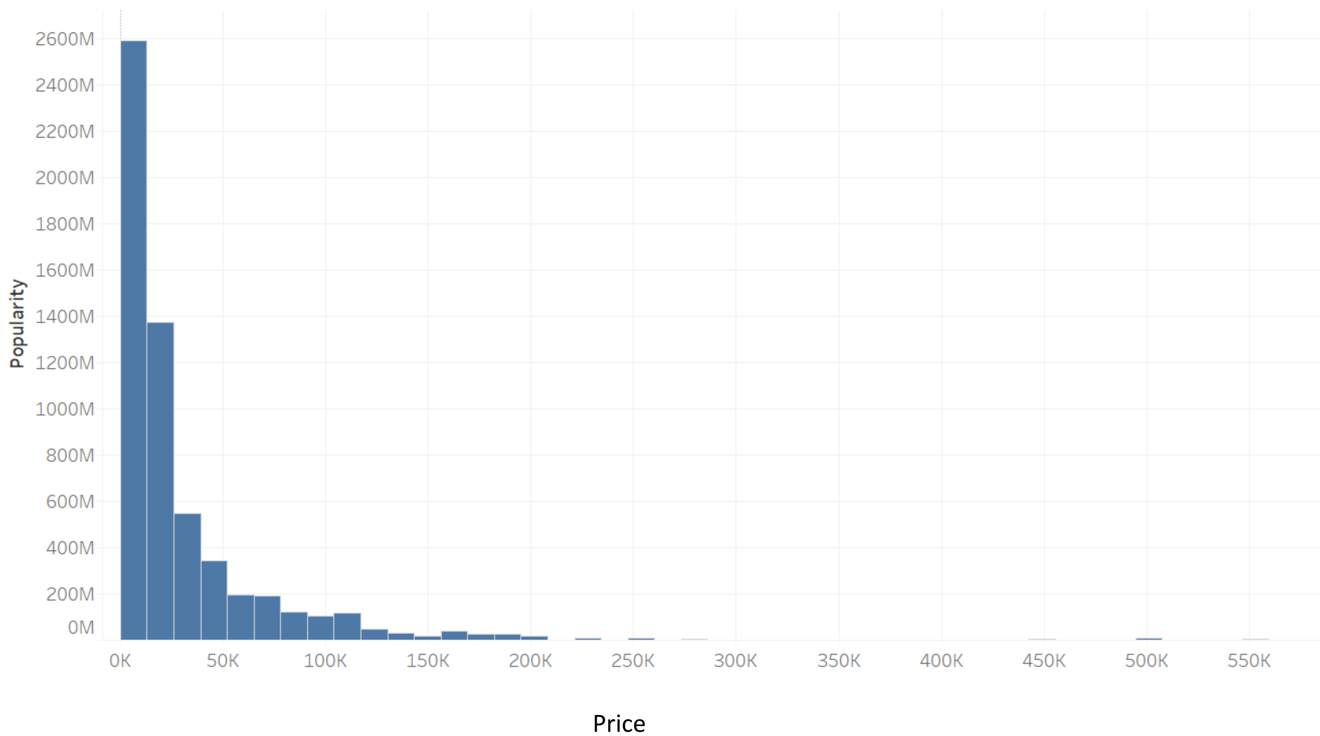
Out[13]:

	car	model	year	price
0	Ford	Kuga	2010	15500.0
1	Mercedes-Benz	E-Class	2011	20500.0



Load

Either dataset included some of the same data on cars, which made it possible to join the data together. From the cars.csv dataset we included car make, noted as simply as car, car model, year and popularity. From the car_ad.json dataset we included car make, model, and year as well, but also price. Because either set included popularity or price but not both, we cleaned the data and loaded them together to analyze both. We can now observe car popularity by price.



From the results, shown in the graph above, we can see a negative correlation—as price increases car popularity decreases. More expensive cars are less popular, and we can see that the most popular cars cost a lot less. This may not necessarily be because cheaper cars are more well-known, perform any better, or have any higher efficiency, etc. but maybe just because they are more affordable. A lot more people can afford cheaper cars, so this causes more expensive cars to be less "popular" and purchased less frequently. However, just because more expensive cars are not bought as often does not mean they have a poor reputation or need increased advertising. People may be well-aware of an expensive car and want to purchase it, more than any other car, but still choose to purchase a cheaper car because it is within their budget. Advertisers should be aware of this when marketing ads, so instead of increasing advertising, they could make it more efficient. Given that price is strongly correlated to popularity, advertisers need focus on promoting affordability if they wish to increase popularity. Future analysis could include looking at cars sales by different demographics. Because cost is indicative of popularity, we could see how income or location ties into that. Then in terms of marketing, companies directly target the appropriate demographic to have more efficient means of advertising.