car price prediction

Description of the data set:

Here we use data that studies informations about cars including the following parameters:

1. (Make)>> the manifacture company of the car
2. (Model)>> the model of the car
3. (Year)>> year of manifacture
4. (Engine Fuel Type)
5. (Engine HP) >> Horse Power
6. (Engine Cylinders)>> number of cylinders
7. (Transmission Type)>> Automatic/Manual
8. (Driven_Wheels)>>Front/all
    A. (Number of Doors)
    B. (Market Category)>>crossover/Luxury
    C. (Vehicle Size)
    D. (Vehicle Style)
    E. (highway MPG)
    F. (city mpg)
    G. (Popularity)
    H. (MSRP)>> Manufacturer's Suggested [Retail Price]>> Our Target

plan for data exploration:

1. cleaning data
    - removing unimportant data
    - dealing with missing (NaN) values if found.
2. feature engineering
    - visualizing the data and see the data distribution
    - deal with skewed distribution if found
3. Variable Selection
    - encoding for categorical variables
    - feature scalling for continuous variables

In [ ]:

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
%matplotlib inline
```

1. cleaning data

In [ ]:

```
df = pd.read_csv("data/data.csv")
df.head(3).T
```

Out[ ]:

	0	1	2

	0	1	2
Make	BMW	BMW	BMW
Model	1 Series M	1 Series	1 Series
Year	2011	2011	2011
Engine Fuel Type	premium unleaded (required)	premium unleaded (required)	premium unleaded (required)
Engine HP	335.0	300.0	300.0
Engine Cylinders	6.0	6.0	6.0
Transmission Type	MANUAL	MANUAL	MANUAL
Driven_Wheels	rear wheel drive	rear wheel drive	rear wheel drive
Number of Doors	2.0	2.0	2.0
Market Category	Factory Tuner,Luxury,High-Performance	Luxury,Performance	Luxury,High-Performance
Vehicle Size	Compact	Compact	Compact
Vehicle Style	Coupe	Convertible	Coupe
highway MPG	26	28	28
city mpg	19	19	20
Popularity	3916	3916	3916
MSRP	46135	40650	36350

In [ ]:

```
df.describe()
```

Out[ ]:

	Year	Engine HP	Engine Cylinders	Number of Doors	highway MPG	city mpg	Popul
count	11914.000000	11845.00000	11884.000000	11908.000000	11914.000000	11914.000000	11914.00
mean	2010.384338	249.38607	5.628829	3.436093	26.637485	19.733255	1554.91
std	7.579740	109.19187	1.780559	0.881315	8.863001	8.987798	1441.85
min	1990.000000	55.00000	0.000000	2.000000	12.000000	7.000000	2.00
25%	2007.000000	170.00000	4.000000	2.000000	22.000000	16.000000	549.00
50%	2015.000000	227.00000	6.000000	4.000000	26.000000	18.000000	1385.00
75%	2016.000000	300.00000	6.000000	4.000000	30.000000	22.000000	2009.00
max	2017.000000	1001.00000	16.000000	4.000000	354.000000	137.000000	5657.00

In [ ]:

```
df.columns = df.columns.str.lower().str.replace(' ', '_')
string_columns = df.dtypes[df.dtypes == 'object'].index
for col in string_columns:
 df[col] = df[col].str.lower().str.replace(' ', '_')
```

In [ ]:
```
df.head(3).T
```

Out[ ]:

	0	1	
make	bmw	bmw	bn
model	1_series_m	1_series	1_seri
year	2011	2011	20
engine_fuel_type	premium_unleaded_(required)	premium_unleaded_(required)	premium_unleaded_(require
engine_hp	335.0	300.0	30(
engine_cylinders	6.0	6.0	6
transmission_type	manual	manual	manu
driven_wheels	rear_wheel_drive	rear_wheel_drive	rear_wheel_dri
number_of_doors	2.0	2.0	2
market_category	factory_tuner,luxury,high-performance	luxury,performance	luxury,high-performan
vehicle_size	compact	compact	compa
vehicle_style	coupe	convertible	cou
highway_mpg	26	28	
city_mpg	19	19	
popularity	3916	3916	39
msrp	46135	40650	363

In [ ]:
```
df.isnull().sum()
```

Out[ ]:
```
make                    0
model                   0
year                    0
engine_fuel_type        3
engine_hp              69
engine_cylinders       30
transmission_type       0
driven_wheels           0
number_of_doors         6
market_category      3742
vehicle_size            0
vehicle_style           0
highway_mpg             0
city_mpg                0
popularity              0
msrp                    0
dtype: int64
```

In [ ]:
```
df = df.fillna(0)
```

In [ ]:
```
df.isnull().sum()
```

Out[ ]:
```
make                 0
model                0
```

```
year                 0
engine_fuel_type     0
engine_hp            0
engine_cylinders     0
transmission_type    0
driven_wheels        0
number_of_doors      0
market_category      0
vehicle_size         0
vehicle_style        0
highway_mpg          0
city_mpg             0
popularity           0
msrp                 0
dtype: int64
```

In [ ]:

```
df['model'].unique()
```

Out[ ]:

```
array(['1_series_m', '1_series', '100', '124_spider', '190-class',
       '2_series', '200', '200sx', '240sx', '240', '2',
       '3_series_gran_turismo', '3_series', '300-class', '3000gt', '300',
       '300m', '300zx', '323', '350-class', '350z', '360', '370z', '3',
       '4_series_gran_coupe', '4_series', '400-class', '420-class',
       '456m', '458_italia', '4c', '4runner', '5_series_gran_turismo',
       '5_series', '500-class', '500e', '500', '500l', '500x', '550',
       '560-class', '570s', '575m', '57', '599', '5',
       '6_series_gran_coupe', '6_series', '600-class', '6000',
       '612_scaglietti', '626', '62', '650s_coupe', '650s_spider', '6',
       '7_series', '718_cayman', '740', '760', '780', '8_series', '80',
       '850', '86', '9-2x', '9-3_griffin', '9-3', '9-4x', '9-5', '9-7x',
       '9000', '900', '90', '911', '928', '929', '940', '944', '960',
       '968', 'a3', 'a4_allroad', 'a4', 'a5', 'a6', 'a7', 'a8',
       'acadia_limited', 'acadia', 'accent', 'acclaim',
       'accord_crosstour', 'accord_hybrid', 'accord_plug-in_hybrid',
       'accord', 'achieva', 'activehybrid_5', 'activehybrid_7',
       'activehybrid_x6', 'aerio', 'aerostar', 'alero', 'allante',
       'allroad_quattro', 'allroad', 'alpina_b6_gran_coupe', 'alpina_b7',
       'alpina', 'altima_hybrid', 'altima', 'amanti', 'amg_gt', 'armada',
       'arnage', 'aspen', 'aspire', 'astro_cargo', 'astro', 'ats_coupe',
       'ats-v', 'ats', 'aurora', 'avalanche', 'avalon_hybrid', 'avalon',
       'avenger', 'aventador', 'aveo', 'aviator', 'axxess', 'azera',
       'aztek', 'azure_t', 'azure', 'b-class_electric_drive',
       'b-series_pickup', 'b-series_truck', 'b-series', 'b9_tribeca',
       'baja', 'beetle_convertible', 'beetle', 'beretta',
       'black_diamond_avalanche', 'blackwood', 'blazer', 'bolt_ev',
       'bonneville', 'borrego', 'boxster', 'bravada', 'breeze',
       'bronco_ii', 'bronco', 'brooklands', 'brougham', 'brz', 'c-class',
       'c-max_hybrid', 'c30', 'c36_amg', 'c43_amg', 'c70', 'c8',
       'cabriolet', 'cabrio', 'cadenza', 'caliber', 'california_t',
       'california', 'camaro', 'camry_hybrid', 'camry_solara', 'camry',
       'canyon', 'caprice', 'captiva_sport', 'caravan', 'carrera_gt',
       'cascada', 'catera', 'cavalier', 'cayenne', 'cayman_s', 'cayman',
       'cc', 'celebrity', 'celica', 'century', 'challenger', 'charger',
       'chevy_van', 'ciera', 'cirrus', 'city_express', 'civic_crx',
       'civic_del_sol', 'civic', 'c/k_1500_series', 'c/k_2500_series',
       'cl-class', 'cla-class', 'cl', 'classic', 'clk-class', 'cls-class',
       'cobalt', 'colorado', 'colt', 'concorde',
       'continental_flying_spur_speed', 'continental_flying_spur',
       'continental_gt_speed_convertible', 'continental_gt_speed',
       'continental_gt3-r', 'continental_gt', 'continental_gtc_speed',
       'continental_gtc', 'continental_supersports_convertible',
       'continental_supersports', 'continental', 'contour_svt', 'contour',
       'corniche', 'corolla_im', 'corolla', 'corrado', 'corsica',
       'corvette_stingray', 'corvette', 'coupe', 'cr-v', 'cr-z',
       'cressida', 'crossfire', 'crosstour', 'crosstrek',
       'crown_victoria', 'cruze_limited', 'cruze', 'ct_200h', 'ct6',
       'cts_coupe', 'cts-v_coupe', 'cts-v_wagon', 'cts-v', 'cts_wagon',
       'cts', 'cube', 'custom_cruiser', 'cutlass_calais', 'cutlass_ciera',
       'cutlass_supreme', 'cutlass', 'cx-3', 'cx-5', 'cx-7', 'cx-9',
       'dakota', 'dart', 'dawn', 'daytona', 'db7', 'db9_gt', 'db9', 'dbs',
       'defender', 'deville', 'diablo', 'diamante', 'discovery_series_ii',
       'discovery_sport', 'discovery', 'dts', 'durango', 'dynasty',
       'e-150', 'e-250', 'e-class', 'e-golf', 'e-series_van',
       'e-series_wagon', 'e55_amg', 'echo', 'eclipse_spyder', 'eclipse',
       'edge', 'eighty-eight_royale', 'eighty-eight', 'elantra_coupe',
       'elantra_gt', 'elantra_touring', 'elantra', 'eldorado', 'electra',
       'element', 'elise', 'enclave', 'encore', 'endeavor', 'entourage',
       'envision', 'envoy_xl', 'envoy_xuv', 'envoy', 'enzo', 'eos',
       'equator', 'equinox', 'equus', 'es_250', 'es_300h', 'es_300',
       'es_330', 'es_350', 'escalade_esv', 'escalade_ext',
       'escalade_hybrid', 'escalade', 'escape_hybrid', 'escape', 'escort',
       'esprit', 'estate_wagon', 'esteem', 'eurovan', 'evora_400',
       'evora', 'ex35', 'excel', 'exige', 'ex', 'expedition',
       'explorer_sport_trac', 'explorer_sport', 'explorer', 'expo',
       'express_cargo', 'express', 'f-150_heritage',
       'f-150_svt_lightning', 'f-150', 'f-250', 'f12_berlinetta', 'f430',
       'festiva', 'ff', 'fiesta', 'firebird', 'fit_ev', 'fit',
       'five_hundred', 'fj_cruiser', 'fleetwood', 'flex', 'flying_spur',
       'focus_rs', 'focus_st', 'focus', 'forenza', 'forester', 'forte',
       'fox', 'fr-s', 'freelander', 'freestar', 'freestyle', 'frontier',
       'fusion_hybrid', 'fusion', 'fx35', 'fx45', 'fx50', 'fx', 'g-class',
       'g_convertible', 'g_coupe', 'g_sedan', 'g20', 'g35',
       'g37_convertible', 'g37_coupe', 'g37_sedan', 'g37', 'g3', 'g5',
       'g6', 'g80', 'g8', 'galant', 'gallardo', 'genesis_coupe',
       'genesis', 'ghibli', 'ghost_series_ii', 'ghost', 'gl-class',
       'gla-class', 'glc-class', 'gle-class_coupe', 'gle-class', 'gli',
       'glk-class', 'gls-class', 'golf_alltrack', 'golf_gti', 'golf_r',
       'golf_sportwagen', 'golf', 'grand_am', 'grand_caravan',
       'grand_prix', 'grand_vitara', 'grand_voyager', 'gransport',
       'granturismo_convertible', 'granturismo', 'gs_200t', 'gs_300',
       'gs_350', 'gs_400', 'gs_430', 'gs_450h', 'gs_460', 'gs_f', 'gt-r',
       'gt', 'gti', 'gto', 'gx_460', 'gx_470', 'h3', 'h3t', 'hhr',
       'highlander_hybrid', 'highlander', 'horizon', 'hr-v', 'hs_250h',
       'huracan', 'i-miev', 'i30', 'i35', 'i3', 'ia', 'ilx_hybrid', 'ilx',
       'impala_limited', 'impala', 'imperial', 'impreza_wrx', 'impreza',
       'im', 'insight', 'integra', 'intrepid', 'intrigue', 'iq',
       'is_200t', 'is_250_c', 'is_250', 'is_300', 'is_350_c', 'is_350',
       'is_f', 'j30', 'jetta_gli', 'jetta_hybrid', 'jetta_sportwagen',
       'jetta', 'jimmy', 'journey', 'juke', 'justy', 'jx', 'k900',
       'kizashi', 'lacrosse', 'lancer_evolution', 'lancer_sportback',
       'lancer', 'land_cruiser', 'landaulet', 'laser', 'le_baron',
       'le_mans', 'leaf', 'legacy', 'legend', 'lesabre', 'levante', 'lfa',
       'lhs', 'loyale', 'lr2', 'lr3', 'lr4', 'ls_400', 'ls_430', 'ls_460',
       'ls_600h_l', 'ls', 'lss', 'ltd_crown_victoria', 'lucerne',
       'lumina_minivan', 'lumina', 'lx_450', 'lx_470', 'lx_570',
       'm-class', 'm2', 'm30', 'm35', 'm37', 'm3', 'm4_gts', 'm45', 'm4',
       'm56', 'm5', 'm6_gran_coupe', 'm6', 'macan', 'magnum',
       'malibu_classic', 'malibu_hybrid', 'malibu_limited', 'malibu_maxx',
       'malibu', 'mark_lt', 'mark_viii', 'mark_vii', 'matrix', 'maxima',
       'maybach', 'mazdaspeed_3', 'mazdaspeed_6', 'mazdaspeed_mx-5_miata',
       'mazdaspeed_protege', 'm', 'mdx', 'metris', 'metro',
       'mighty_max_pickup', 'millenia', 'mirage_g4', 'mirage', 'mkc',
       'mks', 'mkt', 'mkx', 'mkz_hybrid', 'mkz', 'ml55_amg', 'model_s',
       'monaco', 'montana_sv6', 'montana', 'monte_carlo', 'montero_sport',
       'montero', 'mp4-12c', 'mpv', 'mr2_spyder', 'mr2', 'mulsanne',
       'murano_crosscabriolet', 'murano', 'murcielago',
       'mustang_svt_cobra', 'mustang', 'mx-3', 'mx-5_miata', 'mx-6',
       'navajo', 'navigator', 'neon', 'new_beetle', 'new_yorker',
       'ninety-eight', 'nitro', 'nsx', 'nv200', 'nx_200t', 'nx_300h',
       'nx', 'odyssey', 'omni', 'optima_hybrid', 'optima', 'outback',
       'outlander_sport', 'outlander', 'pacifica', 'panamera',
       'park_avenue', 'park_ward', 'paseo', 'passat', 'passport',
       'pathfinder', 'phaeton', 'phantom_coupe', 'phantom_drophead_coupe',
       'phantom', 'pickup', 'pilot', 'precis', 'prelude', 'previa',
       'prius_c', 'prius_prime', 'prius_v', 'prius', 'prizm', 'probe',
       'protege5', 'protege', 'prowler', 'pt_cruiser', 'pulsar', 'q3',
       'q40', 'q45', 'q50', 'q5', 'q60_convertible', 'q60_coupe', 'q70',
       'q7', 'quattroporte', 'quest', 'qx4', 'qx50', 'qx56', 'qx60',
       'qx70', 'qx80', 'qx', 'r-class', 'r32', 'r8', 'rabbit', 'raider',
       'rainier', 'rally_wagon', 'ram_150', 'ram_250', 'ram_50_pickup',
       'ram_cargo', 'ram_pickup_1500', 'ram_van', 'ram_wagon',
       'ramcharger', 'range_rover_evoque', 'range_rover_sport',
       'range_rover', 'ranger', 'rapide_s', 'rapide', 'rav4_ev',
       'rav4_hybrid', 'rav4', 'rc_200t', 'rc_300', 'rc_350', 'rc_f',
       'rdx', 'reatta', 'regal', 'regency', 'rendezvous', 'reno',
       'reventon', 'ridgeline', 'rio', 'riviera', 'rl', 'rlx',
       'roadmaster', 'rogue_select', 'rogue', 'rondo', 'routan', 'rs_4',
       'rs_5', 'rs_6', 'rs_7', 'rsx', 'rx_300', 'rx_330', 'rx_350',
       'rx_400h', 'rx_450h', 'rx-7', 'rx-8', 's-10_blazer', 's-10',
       's-15_jimmy', 's-15', 's-class', 's2000', 's3', 's40', 's4', 's5',
       's60_cross_country', 's60', 's6', 's70', 's7', 's80', 's8', 's90',
       'safari_cargo', 'safari', 'samurai', 'santa_fe_sport', 'santa_fe',
       'savana_cargo', 'savana', 'sc_300', 'sc_400', 'sc_430', 'scoupe',
       'sebring', 'sedona', 'sentra', 'sephia', 'sequoia', 'seville',
       'shadow', 'shelby_gt350', 'shelby_gt500', 'sidekick', 'sienna',
       'sierra_1500_classic', 'sierra_1500_hybrid', 'sierra_1500',
       'sierra_1500hd', 'sierra_c3', 'sierra_classic_1500', 'sigma',
       'silhouette', 'silver_seraph', 'silverado_1500_classic',
       'silverado_1500_hybrid', 'silverado_1500', 'sixty_special',
       'skylark', 'sl-class', 'slc-class', 'slk-class', 'slr_mclaren',
       'sls_amg_gt_final_edition', 'sls_amg_gt', 'sls_amg', 'slx',
       'solstice', 'sonata_hybrid', 'sonata', 'sonic', 'sonoma',
       'sorento', 'soul_ev', 'soul', 'spark_ev', 'spark', 'spectra',
       'spirit', 'sportage', 'sportvan', 'spyder', 'sq5', 'srt_viper',
       'srx', 'ss', 'ssr', 'stanza', 'stealth', 'stratus', 'sts-v', 'sts',
       'suburban', 'sunbird', 'sundance', 'sunfire', 'superamerica',
       'supersports_convertible_isr', 'supra', 'svx', 'swift', 'sx4',
       'syclone', 't100', 'tacoma', 'tahoe_hybrid', 'tahoe_limited/z71',
       'tahoe', 'taurus_x', 'taurus', 'tc', 'tempo', 'tercel', 'terrain',
       'terraza', 'thunderbird', 'tiburon', 'tiguan', 'titan', 'tl',
       'tlx', 'toronado', 'torrent', 'touareg_2', 'touareg',
       'town_and_country', 'town_car', 'tracker', 'trailblazer_ext',
       'trailblazer', 'trans_sport', 'transit_connect', 'transit_wagon',
       'traverse', 'trax', 'tribeca', 'tribute_hybrid', 'tribute',
       'truck', 'tsx_sport_wagon', 'tsx', 'tt_rs', 'tt', 'tts', 'tucson',
       'tundra', 'typhoon', 'uplander', 'v12_vanquish', 'v12_vantage_s',
       'v12_vantage', 'v40', 'v50', 'v60_cross_country', 'v60', 'v70',
       'v8_vantage', 'v8', 'v90', 'vanagon', 'vandura', 'van', 'vanquish',
       'vanwagon', 'veloster', 'venture', 'venza', 'veracruz', 'verano',
       'verona', 'versa_note', 'versa', 'veyron_16.4', 'vibe', 'vigor',
       'viper', 'virage', 'vitara', 'voyager', 'windstar_cargo',
       'windstar', 'wraith', 'wrx', 'x-90', 'x1', 'x3', 'x4', 'x5_m',
       'x5', 'x6_m', 'x6', 'xa', 'xb', 'xc60', 'xc70', 'xc90', 'xc', 'xd',
       'xg300', 'xg350', 'xl-7', 'xl7', 'xlr-v', 'xlr', 'xt5', 'xterra',
       'xts', 'xt', 'xv_crosstrek', 'yaris_ia', 'yaris', 'yukon_denali',
       'yukon_hybrid', 'yukon_xl', 'yukon', 'z3', 'z4_m', 'z4', 'z8',
       'zdx', 'zephyr'], dtype=object)
```

In [ ]:

```
df=df.drop(columns='model')
df.head()
```

Out[ ]:

	make	year	engine_fuel_type	engine_hp	engine_cylinders	transmission_type	driven

	make	year	engine_fuel_type	engine_hp	engine_cylinders	transmission_type	driven
0	bmw	2011	premium_unleaded_(required)	335.0	6.0	manual	rear_wh
1	bmw	2011	premium_unleaded_(required)	300.0	6.0	manual	rear_wh
2	bmw	2011	premium_unleaded_(required)	300.0	6.0	manual	rear_wh
3	bmw	2011	premium_unleaded_(required)	230.0	6.0	manual	rear_wh
4	bmw	2011	premium_unleaded_(required)	230.0	6.0	manual	rear_wh

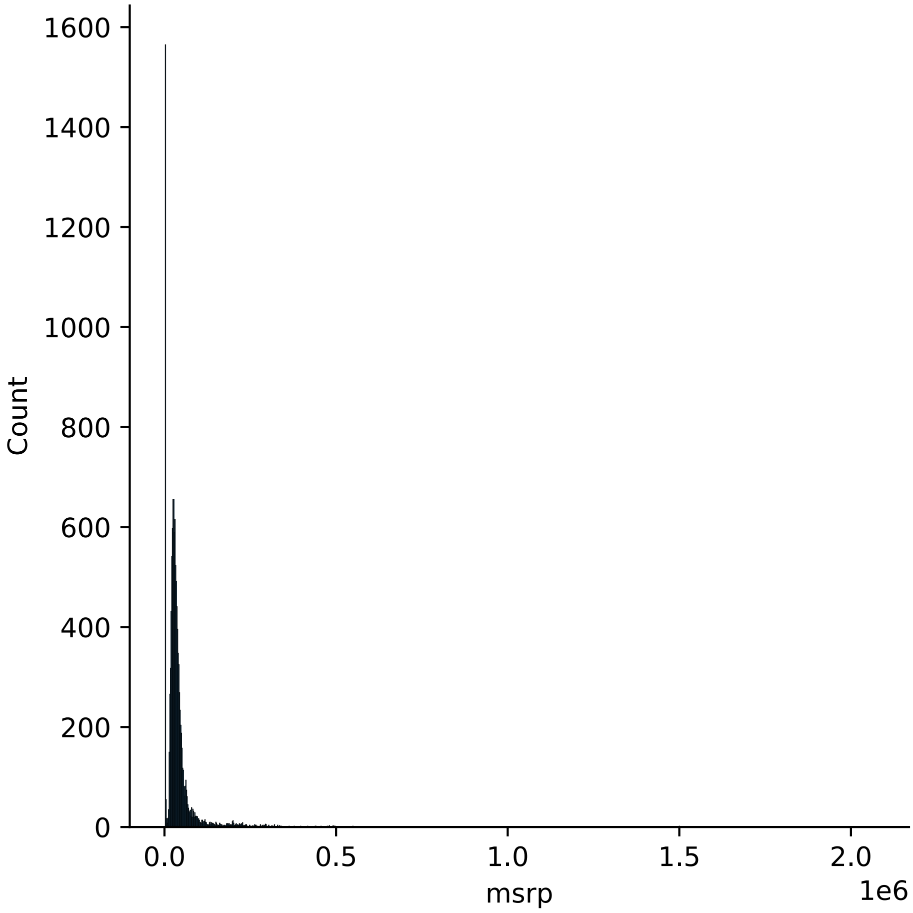
2. feature engineering

In [ ]:

```
sns.displot(df.msrp)
```

Out[ ]:

```
<seaborn.axisgrid.FacetGrid at 0x1a4658ffbb0>
```



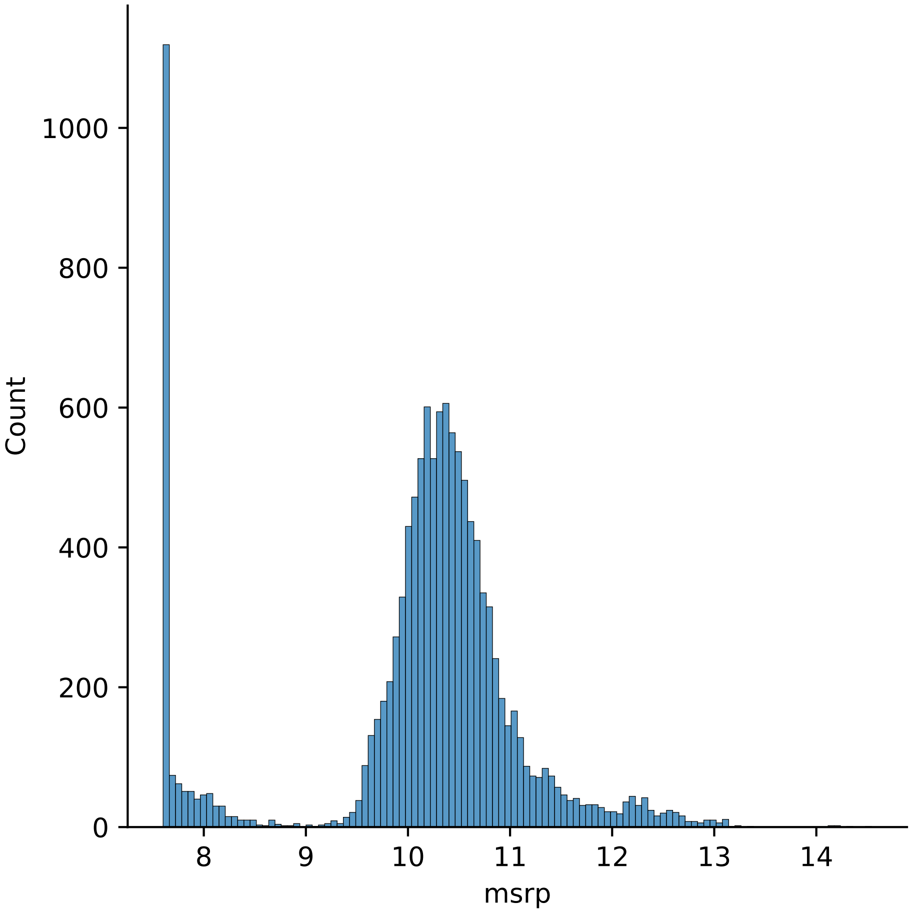
data is positive skewed

In [ ]:

```
sns.displot(np.log1p(df.msrp))
```

Out[ ]:

```
<seaborn.axisgrid.FacetGrid at 0x1a39c656af0>
```



3. Variable Selection

In [ ]:

```
#One-hot encoding for dummy variables
string_columns = df.dtypes[df.dtypes == 'object'].index.to_list()
df[string_columns].T
```

Out[ ]:

	0	1	
make	bmw	bmw	bn
engine_fuel_type	premium_unleaded_(required)	premium_unleaded_(required)	premium_unleaded_(require
transmission_type	manual	manual	manu
driven_wheels	rear_wheel_drive	rear_wheel_drive	rear_wheel_dri
market_category	factory_tuner,luxury,high-performance	luxury,performance	luxury,high-performan
vehicle_size	compact	compact	compa
vehicle_style	coupe	convertible	cou

7 rows × 11914 columns

In [ ]:

```
df = pd.get_dummies(df,columns=string_columns,drop_first=True)
df.describe()
```

Out[ ]:

	year	engine_hp	engine_cylinders	number_of_doors	highway_mpg	city_mpg
count	11914.000000	11914.000000	11914.000000	11914.000000	11914.000000	11914.000000

	year	engine_hp	engine_cylinders	number_of_doors	highway_mpg	city_mpg
mean	2010.384338	247.941749	5.614655	3.434363	26.637485	19.733255
std	7.579740	110.507669	1.800554	0.884460	8.863001	8.987798
min	1990.000000	0.000000	0.000000	0.000000	12.000000	7.000000
25%	2007.000000	170.000000	4.000000	2.000000	22.000000	16.000000
50%	2015.000000	225.000000	6.000000	4.000000	26.000000	18.000000
75%	2016.000000	300.000000	6.000000	4.000000	30.000000	22.000000
max	2017.000000	1001.000000	16.000000	4.000000	354.000000	137.000000

8 rows × 160 columns

Pair plot of features

Now we have a filtered dataset, we will generate visuals to better understand the target and feature-target relationships

3. Hypothesis Testing

A statistical hypothesis is an assumption about a population parameter. This assumption may or may not be true. Hypothesis testing refers to the formal procedures used by statisticians to accept or reject statistical hypotheses.

In [ ]:

```
%pylab inline
%config InlineBackend.figure_formats = ['svg']

import matplotlib.pyplot as plt
from scipy import stats
import math
```

```
Populating the interactive namespace from numpy and matplotlib
```

Null hypothesis.

The null hypothesis, denoted by Ho, is usually the hypothesis that sample observations result purely from chance.

Hypothesis Tests Statisticians follow a formal process to determine whether to reject a null hypothesis, based on sample data. This process, called hypothesis testing, consists of four steps:

```
* State the hypotheses: This involves stating the null and alternative
hypotheses. The hypotheses are stated in such a way that they are
mutually  exclusive. That is, if one is true, the other must be
false.
* Formulate an analysis plan: The analysis plan describes how to use
sample data to evaluate the null hypothesis. The evaluation often
focuses around a single test statistic.
* Analyze sample data: Find the value of the test statistic (mean
score, proportion, t statistic, z-score, etc.) described in the
analysis plan.
* Interpret results: Apply the decision rule described in the analysis
```

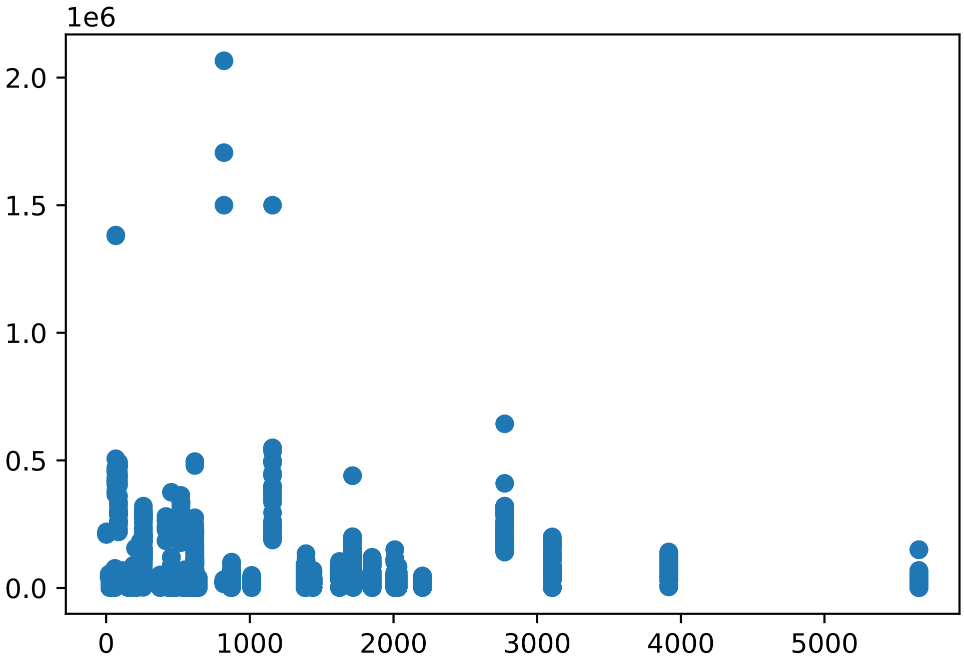
```
plan. If the value of the test statistic is unlikely, based on the
null     hypothesis, reject the null hypothesis.
```

In [ ]:

```
plt.plot(df.popularity,df.msrp,ls=" ",marker="o")
```

Out[ ]:

```
[<matplotlib.lines.Line2D at 0x1a467a22b50>]
```



In [ ]:

```
hle = np.array(df.popularity)
ylwd = np.array(df.msrp)
stats.ttest_ind(hle, ylwd, equal_var = False)
```

Out[ ]:

```
Ttest_indResult(statistic=-70.87148117260806, pvalue=0.0)
```

The p value is 0 which is smaller than 0.05. Therefore we reject the null hypothesis at 5% significance level, meaning that the popularity increase is not the same as in msrp increase.

Next Steps

trying to get more features using the year column with other features