Used_car

January 27, 2026

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
df = pd.read_csv('vehicles.csv')
```

1 Display information about the data

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

[3]:
```
           id                                                url  \
0  7222695916  https://prescott.craigslist.org/cto/d/prescott…
1  7218891961  https://fayar.craigslist.org/ctd/d/bentonville…
2  7221797935  https://keys.craigslist.org/cto/d/summerland-k…
3  7222270760  https://worcester.craigslist.org/cto/d/west-br…
4  7210384030  https://greensboro.craigslist.org/cto/d/trinit…

                  region                        region_url  price  year  \
0               prescott    https://prescott.craigslist.org   6000   NaN
1           fayetteville       https://fayar.craigslist.org  11900   NaN
2           florida keys        https://keys.craigslist.org  21000   NaN
3  worcester / central MA   https://worcester.craigslist.org   1500   NaN
4             greensboro  https://greensboro.craigslist.org   4900   NaN

  manufacturer model condition cylinders  … size  type paint_color  \
0          NaN   NaN       NaN       NaN  …  NaN   NaN         NaN
1          NaN   NaN       NaN       NaN  …  NaN   NaN         NaN
2          NaN   NaN       NaN       NaN  …  NaN   NaN         NaN
3          NaN   NaN       NaN       NaN  …  NaN   NaN         NaN
4          NaN   NaN       NaN       NaN  …  NaN   NaN         NaN

  image_url description county state lat long posting_date
0       NaN         NaN    NaN    az NaN  NaN          NaN
1       NaN         NaN    NaN    ar NaN  NaN          NaN
2       NaN         NaN    NaN    fl NaN  NaN          NaN
3       NaN         NaN    NaN    ma NaN  NaN          NaN
```

```
4       NaN         NaN    NaN    nc NaN  NaN          NaN

[5 rows x 26 columns]
```

[4]:
```
df.isna().sum()
```

[4]:
```
id                  0
url                 0
region              0
region_url          0
price               0
year             1205
manufacturer    17646
model            5277
condition      174104
cylinders      177678
fuel             3013
odometer         4400
title_status     8242
transmission     2556
VIN            161042
drive          130567
size           306361
type            92858
paint_color    130203
image_url          68
description        70
county         426880
state               0
lat              6549
long             6549
posting_date       68
dtype: int64
```

[5]:
```
df.describe()
```

[5]:
```
                 id         price           year      odometer  county  \
count  4.268800e+05  4.268800e+05  425675.000000  4.224800e+05     0.0
mean   7.311487e+09  7.519903e+04    2011.235191  9.804333e+04     NaN
std    4.473170e+06  1.218228e+07       9.452120  2.138815e+05     NaN
min    7.207408e+09  0.000000e+00    1900.000000  0.000000e+00     NaN
25%    7.308143e+09  5.900000e+03    2008.000000  3.770400e+04     NaN
50%    7.312621e+09  1.395000e+04    2013.000000  8.554800e+04     NaN
75%    7.315254e+09  2.648575e+04    2017.000000  1.335425e+05     NaN
max    7.317101e+09  3.736929e+09    2022.000000  1.000000e+07     NaN

                lat           long
```

```
count  420331.000000  420331.000000
mean       38.493940     -94.748599
std         5.841533      18.365462
min       -84.122245    -159.827728
25%        34.601900    -111.939847
50%        39.150100     -88.432600
75%        42.398900     -80.832039
max        82.390818     173.885502
```

[6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 426880 entries, 0 to 426879
Data columns (total 26 columns):
 #   Column        Non-Null Count   Dtype  
---  ------        --------------   -----  
 0   id            426880 non-null  int64  
 1   url           426880 non-null  object 
 2   region        426880 non-null  object 
 3   region_url    426880 non-null  object 
 4   price         426880 non-null  int64  
 5   year          425675 non-null  float64
 6   manufacturer  409234 non-null  object 
 7   model         421603 non-null  object 
 8   condition     252776 non-null  object 
 9   cylinders     249202 non-null  object 
 10  fuel          423867 non-null  object 
 11  odometer      422480 non-null  float64
 12  title_status  418638 non-null  object 
 13  transmission  424324 non-null  object 
 14  VIN           265838 non-null  object 
 15  drive         296313 non-null  object 
 16  size          120519 non-null  object 
 17  type          334022 non-null  object 
 18  paint_color   296677 non-null  object 
 19  image_url     426812 non-null  object 
 20  description   426810 non-null  object 
 21  county        0 non-null       float64
 22  state         426880 non-null  object 
 23  lat           420331 non-null  float64
 24  long          420331 non-null  float64
 25  posting_date  426812 non-null  object 
dtypes: float64(5), int64(2), object(19)
memory usage: 84.7+ MB
```

[7]:
```
df.columns
```

[7]:
```
Index(['id', 'url', 'region', 'region_url', 'price', 'year', 'manufacturer',
       'model', 'condition', 'cylinders', 'fuel', 'odometer', 'title_status',
       'transmission', 'VIN', 'drive', 'size', 'type', 'paint_color',
       'image_url', 'description', 'county', 'state', 'lat', 'long',
       'posting_date'],
      dtype='object')
```

[9]:
```
df.dtypes
```

[9]:
```
id                int64
url              object
region           object
region_url       object
price             int64
year            float64
manufacturer     object
model            object
condition        object
cylinders        object
fuel             object
odometer        float64
title_status     object
transmission     object
VIN              object
drive            object
size             object
type             object
paint_color      object
image_url        object
description      object
county          float64
state            object
lat             float64
long            float64
posting_date     object
dtype: object
```

2 Data cleaning

[8]:
```
df=df.drop(["id","url","image_url","region_url","lat","long","county"],axis = 1)
```

[10]:
```
df_object = df.select_dtypes(include = 'object').head(3)
df_num = df.select_dtypes(exclude = 'object').head(3)
```

[11]:
```
df.select_dtypes(exclude = 'object').isna().sum()
```

[11]:
```
price          0
year        1205
odometer    4400
dtype: int64
```

[12]:
```
df.select_dtypes(include = 'object').isna().sum()
```

[12]:
```
region              0
manufacturer    17646
model            5277
condition      174104
cylinders      177678
fuel             3013
title_status     8242
transmission     2556
VIN            161042
drive          130567
size           306361
type            92858
paint_color    130203
description        70
state               0
posting_date       68
dtype: int64
```

[13]:
```
df.dropna(subset = ["model"],inplace = True)
df.dropna(subset = ["fuel"],inplace = True)
df.dropna(subset = ["transmission"],inplace = True)
df.dropna(subset = ["title_status"],inplace = True)
df.dropna(subset = ["posting_date"],inplace = True)
```

[ ]:
```

```

[14]:
```
df.select_dtypes(include = 'object').isna().sum()
```

[14]:
```
region              0
manufacturer    16764
model               0
condition      164850
cylinders      171086
fuel                0
title_status        0
transmission        0
VIN            157280
drive          122837
size           291571
type            89085
```

```
paint_color     119557
description          2
state                0
posting_date         0
dtype: int64
```

[15]:
```
df.select_dtypes(exclude = 'object').isna().sum()
```

[15]:
```
price          0
year         737
odometer    3508
dtype: int64
```

[16]:
```
df.dropna(subset = ["year"],inplace = True)
df.dropna(subset = ["odometer"],inplace = True)
```

[17]:
```
df.select_dtypes(exclude = 'object').isna().sum()
```

[17]:
```
price       0
year        0
odometer    0
dtype: int64
```

[18]:
```
df["price"].index
```

[18]:
```
Index([    27,     28,     29,     30,     31,     32,     33,     34,     35,
           36,
       …
       426870, 426871, 426872, 426873, 426874, 426875, 426876, 426877, 426878,
       426879],
      dtype='int64', length=405594)
```

[19]:
```
(405594 * (7.5/100))
```

[19]:
```
30419.55
```

[20]:
```
df[df["price"] ==0].head(5)
```

[20]:

	region	price	year	manufacturer	model	condition	cylinders	fuel	odometer	title_status	transmission	VIN
46	auburn	0	2011.0	jeep	compass	excellent	NaN	gas	99615.0	clean	automatic	NaN
126	auburn	0	2018.0	chevrolet	express cargo van	like new	6 cylinders	gas	68472.0	clean	automatic	1GCWGAFP8J1309579
127	auburn	0	2019.0	chevrolet	express cargo van	like new						
128	auburn	0	2018.0	chevrolet	express cargo van	like new						
191	birmingham	0	2015.0	nissan	sentra	excellent						

```
127  6 cylinders  gas   69125.0        clean    automatic  1GCWGAFP4K1214373
128  6 cylinders  gas   66555.0        clean    automatic  1GCWGAFPXJ1337903
191  4 cylinders  gas   99505.0        clean    automatic  3N1AB7AP8FY348505

    drive       size   type paint_color  \
46    NaN  full-size    SUV         NaN
126   rwd  full-size    van       white
127   rwd  full-size    van       white
128   rwd  full-size    van       white
191   fwd        NaN  sedan      silver

                                          description state  \
46   Call or text now 800-213-0336 Open 9:00-6:00pm…    al
126  2018 Chevrolet Express Cargo Van RWD 2500 135"…    al
127  2019 Chevrolet Express Cargo Van RWD 2500 135"…    al
128  2018 Chevrolet Express Cargo Van RWD 2500 135"…    al
191  2015 Nissan Sentra  by Benton Nissan of Oxford…    al

                posting_date
46   2021-04-30T16:35:11-0500
126  2021-04-12T11:20:35-0500
127  2021-04-12T11:20:00-0500
128  2021-04-12T11:19:58-0500
191  2021-05-04T11:00:42-0500
```

[21]:
```
top_30419 = df_num['price'].nlargest(30419)
bottom_100 = df_num['price'].nsmallest(30419)
```

[22]:
```
x=df[(df["price"] <= 3736928711) & (df["price"] >= 1111111)].index
y=df[df["price"] ==0].index
```

[23]:
```
df=df.drop(x,axis = 0)
```

[24]:
```
df=df.drop(y,axis = 0)
```

[25]:
```
df.describe()
```

[25]:

	price	year	odometer
count	375717.000000	375717.000000	3.757170e+05
mean	18935.740640	2011.096118	9.849603e+04
std	16144.694868	9.385032	1.976752e+05
min	1.000000	1900.000000	0.000000e+00
25%	7200.000000	2008.000000	3.812300e+04
50%	15499.000000	2013.000000	8.703700e+04
75%	27882.000000	2017.000000	1.356360e+05
max	1000000.000000	2022.000000	1.000000e+07

[26]:
```
print(df['price'].mean())
print(df['price'].median())
```

```
18935.740639896518
15499.0
```

[27]:
```
df.isna().sum()
```

[27]:
```
region              0
price               0
year                0
manufacturer    14714
model               0
condition      142243
cylinders      153971
fuel                0
odometer            0
title_status        0
transmission        0
VIN            147590
drive          112905
size           267055
type            82507
paint_color    106661
description         2
state               0
posting_date        0
dtype: int64
```

3 manufacturer

[28]:
```
df["manufacturer"].unique()
```

[28]:
```
array(['gmc', 'chevrolet', 'toyota', 'ford', 'jeep', 'nissan', 'ram',
       'mazda', 'cadillac', 'honda', 'dodge', 'lexus', 'jaguar', 'buick',
       'chrysler', 'volvo', 'audi', 'infiniti', 'lincoln', 'alfa-romeo',
       'subaru', nan, 'acura', 'hyundai', 'mercedes-benz', 'bmw',
       'mitsubishi', 'volkswagen', 'porsche', 'kia', 'ferrari', 'mini',
       'pontiac', 'fiat', 'rover', 'tesla', 'saturn', 'mercury',
       'harley-davidson', 'datsun', 'aston-martin', 'land rover'],
      dtype=object)
```

[29]:
```
df["manufacturer"].isna().sum()
```

[29]:
```
np.int64(14714)
```

[30]:
```
df["manufacturer"]=df["manufacturer"].fillna("Unknown")
```

[31]:
```
df["manufacturer"].isna().sum()
```

[31]:
```
np.int64(0)
```

[32]:
```
df["manufacturer"].unique()
```

[32]:
```
array(['gmc', 'chevrolet', 'toyota', 'ford', 'jeep', 'nissan', 'ram',
       'mazda', 'cadillac', 'honda', 'dodge', 'lexus', 'jaguar', 'buick',
       'chrysler', 'volvo', 'audi', 'infiniti', 'lincoln', 'alfa-romeo',
       'subaru', 'Unknown', 'acura', 'hyundai', 'mercedes-benz', 'bmw',
       'mitsubishi', 'volkswagen', 'porsche', 'kia', 'ferrari', 'mini',
       'pontiac', 'fiat', 'rover', 'tesla', 'saturn', 'mercury',
       'harley-davidson', 'datsun', 'aston-martin', 'land rover'],
      dtype=object)
```

4 condition

[33]:
```
df["condition"]=df["condition"].fillna("UN")
df["condition"].isna().sum()
```

[33]:
```
np.int64(0)
```

[34]:
```
df["condition"].unique()
```

[34]:
```
array(['good', 'excellent', 'fair', 'like new', 'UN', 'new', 'salvage'],
      dtype=object)
```

[35]:
```
prices = list(df["price"])
condition = list(df["condition"])
```

[36]:
```
len(prices) == len(condition)
```

[36]:
```
True
```

[37]:
```
df["price"].describe()
```

[37]:
```
count     375717.000000
mean       18935.740640
std        16144.694868
min            1.000000
25%         7200.000000
50%        15499.000000
75%        27882.000000
max      1000000.000000
Name: price, dtype: float64
```

[38]:
```
df["condition"].info()
```

```
<class 'pandas.core.series.Series'>
Index: 375717 entries, 27 to 426879
Series name: condition
Non-Null Count   Dtype
--------------   -----
375717 non-null  object
dtypes: object(1)
memory usage: 5.7+ MB
```

[39]:
```
condition[230239]
```

[39]:
```
'good'
```

[40]:
```
o=0
for i in condition:

    if i == 'UN':
        if prices[o] <= 1000000.0:
            condition[o] = condition[o].replace('UN',"new")
        elif  prices[o] <=27882.00:
            condition[o] = condition[o].replace('UN',"like new")
        elif  prices[o] <= 18935.0:
            condition[o] = condition[o].replace('UN',"excellent")
        elif  prices[o] <=13995.0:
            condition[o] = condition[o].replace('UN',"good")
        elif  prices[o] <=5995.0:
            condition[o] = condition[o].replace('UN',"fair")
        else :
            condition[o] = condition[o].replace("UN","salvage")
    o+=1
```

[41]:
```
test_2=pd.Series(condition)
```

[42]:
```
test_2.unique()
```

[42]:
```
array(['good', 'excellent', 'fair', 'like new', 'new', 'salvage'],
      dtype=object)
```

[43]:
```
df=df.drop(["condition"],axis=1)
```

[44]:
```
df.isna().sum()
```

[44]:
```
region              0
price               0
year                0
manufacturer        0
model               0
cylinders      153971
```

```
fuel                0
odometer            0
title_status        0
transmission        0
VIN            147590
drive          112905
size           267055
type            82507
paint_color    106661
description         2
state               0
posting_date        0
dtype: int64
```

[45]:
```
index =list(df.index)
```

[46]:
```
test_3=pd.DataFrame(condition,index=index,columns=["condition"])
```

[47]:
```
df=pd.concat([df,test_3],axis=1)
```

[48]:
```
df.isna().sum()
```

[48]:
```
region              0
price               0
year                0
manufacturer        0
model               0
cylinders      153971
fuel                0
odometer            0
title_status        0
transmission        0
VIN            147590
drive          112905
size           267055
type            82507
paint_color    106661
description         2
state               0
posting_date        0
condition           0
dtype: int64
```

5 cylinders

[49]:
```
df["cylinders"]=df["cylinders"].fillna("UN")
df["cylinders"].isna().sum()
```

[49]:
```
np.int64(0)
```

[50]:
```
df["cylinders"].unique()
```

[50]:
```
array(['8 cylinders', '6 cylinders', 'UN', '4 cylinders', '5 cylinders',
       'other', '3 cylinders', '10 cylinders', '12 cylinders'],
      dtype=object)
```

[51]:
```
cylinders = list(df["cylinders"])
```

[52]:
```
len(prices) == len(cylinders)
```

[52]:
```
True
```

[53]:
```
o=0
for i in cylinders:

    if i == 'UN':
        if prices[o] <= 1000000.0:
            cylinders[o] = cylinders[o].replace('UN',"12 cylinders")
        elif  prices[o] <=27882.00:
            cylinders[o] = cylinders[o].replace('UN',"10 cylinders")
        elif  prices[o] <= 18935.0:
            cylinders[o] = cylinders[o].replace('UN',"8 cylinders")
        elif  prices[o] <=13995.0:
            cylinders[o] = cylinders[o].replace('UN',"6 cylinders")
        elif  prices[o] <=5995.0:
            cylinders[o] = cylinders[o].replace('UN',"4 cylinders")
        else :
            cylinders[o] = cylinders[o].replace("UN","3 cylinders")
    o+=1
```

[54]:
```
test_4=pd.Series(cylinders)
```

[55]:
```
test_4.unique()
```

[55]:
```
array(['8 cylinders', '6 cylinders', '12 cylinders', '4 cylinders',
       '5 cylinders', 'other', '3 cylinders', '10 cylinders'],
      dtype=object)
```

[56]:
```
df=df.drop(["cylinders"],axis=1)
```

[57]:
```
index =list(df.index)
```

[58]:
```
test_5=pd.DataFrame(cylinders,index=index,columns=["cylinders"])
```

[59]:
```
df=pd.concat([df,test_5],axis=1)
```

[60]:
```
df.isna().sum()
```

[60]:
```
region              0
price               0
year                0
manufacturer        0
model               0
fuel                0
odometer            0
title_status        0
transmission        0
VIN            147590
drive          112905
size           267055
type            82507
paint_color    106661
description         2
state               0
posting_date        0
condition           0
cylinders           0
dtype: int64
```

[61]:
```
df.sample(5)
```

[61]:

	region	price	year	manufacturer	model	fuel	odometer	title_status	transmission	VIN	drive	size	type	cylinders
180751	annapolis	20995	2014.0	honda	pilot	gas	87300.0	clean	automatic	5FNYF4H93EB009411	4wd	NaN	SUV	6 cylinders
140506	chicago	17977	2014.0	ford	e350sd	other	97126.0	clean	other	NaN	NaN	NaN	van	8 cylinders
17038	yuma	39995	2017.0	ram	1500 lifted big horn crew cab 5.7 liter hemi	gas	48125.0	clean	automatic	3C6RR7LT4HG784712	4wd	NaN	pickup	8 cylinders
318312	eugene	9999	1999.0	toyota	tacoma	gas	249920.0							
101667	lakeland	29999	2017.0	rover	discovery sport	gas	31867.0							

```
318312        clean    automatic  4TANM92N9XZ436709   rwd        NaN   truck
101667        clean        other  SALCR2BG0HH636660   NaN  full-size     SUV

       paint_color                                        description state  \
180751      custom  2014 Honda Pilot Touring Sport Utility 4D - $2…    md
140506      yellow  (STOCK#: JYC-A50795)   VEHICLE DESCRIPTION: AC…    il
17038        white  2017 Ram 1500 LIFTED BIG HORN CREW CAB 5.7 LIT…    az
318312       white  1999 Toyota Tacoma Reg Cab PreRunner Auto -- $…    or
101667       white  CREDIT AMNESTY!  SE HABLA ESPANOL !!OPEN SUNDA…    fl

                    posting_date  condition    cylinders
180751  2021-04-26T12:03:22-0400        new  6 cylinders
140506  2021-05-03T14:55:52-0500        new  8 cylinders
17038   2021-04-16T17:07:32-0700        new  8 cylinders
318312  2021-04-19T09:32:48-0700        new  4 cylinders
101667  2021-04-10T10:47:14-0400  excellent  4 cylinders
```

6 drive

[62]:
```
df["drive"].unique()
```

[62]:
```
array([nan, 'rwd', '4wd', 'fwd'], dtype=object)
```

[63]:
```
df["drive"]=df["drive"].fillna("UN")
df["drive"].isna().sum()
```

[63]:
```
np.int64(0)
```

[64]:
```
df["drive"].unique()
```

[64]:
```
array(['UN', 'rwd', '4wd', 'fwd'], dtype=object)
```

[65]:
```
drive = list(df["drive"])
```

[66]:
```
len(prices) == len(drive)
```

[66]:
```
True
```

[67]:
```
o=0
for i in drive:

    if i == 'UN':
        if prices[o] <= 1000000.0:
            drive[o] = drive[o].replace('UN',"4wd")
        elif  prices[o] <= 18935.0:
            drive[o] = drive[o].replace('UN',"rwd")
        else :
```

```
            drive[o] = drive[o].replace("UN","fwd")
    o+=1
```

[ ]:

[68]:
```
test_6=pd.Series(drive)
```

[69]:
```
test_6.unique()
```

[69]:
```
array(['4wd', 'rwd', 'fwd'], dtype=object)
```

[70]:
```
df=df.drop(["drive"],axis=1)
```

[71]:
```
index =list(df.index)
```

[72]:
```
test_7=pd.DataFrame(drive,index=index,columns=["drive"])
```

[73]:
```
df=pd.concat([df,test_7],axis=1)
```

[74]:
```
df.isna().sum()
```

[74]:
```
region              0
price               0
year                0
manufacturer        0
model               0
fuel                0
odometer            0
title_status        0
transmission        0
VIN            147590
size           267055
type            82507
paint_color    106661
description         2
state               0
posting_date        0
condition           0
cylinders           0
drive               0
dtype: int64
```

7 type

[75]:
```
df["type"].unique()
```

[75]:
```
array(['pickup', 'truck', 'other', nan, 'coupe', 'SUV', 'hatchback',
       'mini-van', 'sedan', 'offroad', 'bus', 'convertible', 'wagon',
       'van'], dtype=object)
```

[76]:
```
df["manufacturer"].unique()
```

[76]:
```
array(['gmc', 'chevrolet', 'toyota', 'ford', 'jeep', 'nissan', 'ram',
       'mazda', 'cadillac', 'honda', 'dodge', 'lexus', 'jaguar', 'buick',
       'chrysler', 'volvo', 'audi', 'infiniti', 'lincoln', 'alfa-romeo',
       'subaru', 'Unknown', 'acura', 'hyundai', 'mercedes-benz', 'bmw',
       'mitsubishi', 'volkswagen', 'porsche', 'kia', 'ferrari', 'mini',
       'pontiac', 'fiat', 'rover', 'tesla', 'saturn', 'mercury',
       'harley-davidson', 'datsun', 'aston-martin', 'land rover'],
      dtype=object)
```

[77]:
```
manufacturer_type_map = {
    'jeep': 'SUV',
    'land rover': 'SUV',
    'rover': 'SUV',
    'gmc': 'pickup',
    'ram': 'pickup',
    'harley-davidson': 'other',

    'ferrari': 'coupe',
    'aston-martin': 'coupe',
    'porsche': 'coupe',
    'alfa-romeo': 'coupe',


    'mini': 'hatchback',
    'fiat': 'hatchback',
    'subaru': 'wagon',


    'mercedes-benz': 'sedan',
    'bmw': 'sedan',
    'audi': 'sedan',
    'lexus': 'sedan',
    'cadillac': 'sedan',
    'lincoln': 'sedan',
    'jaguar': 'sedan',
    'infiniti': 'sedan',
    'acura': 'sedan',
    'volvo': 'sedan',
    'buick': 'sedan',
    'chrysler': 'sedan',
```

```
    'toyota': 'sedan',
    'honda': 'sedan',
    'nissan': 'sedan',
    'ford': 'sedan',
    'chevrolet': 'sedan',
    'dodge': 'sedan',
    'hyundai': 'sedan',
    'kia': 'sedan',
    'mazda': 'sedan',
    'volkswagen': 'sedan',
    'mitsubishi': 'sedan',
    'saturn': 'sedan',
    'mercury': 'sedan',
    'pontiac': 'sedan',
    'datsun': 'sedan',
    'tesla': 'sedan',

    'Unknown': 'other'
}
```

[78]:
```
df['type'] = df['type'].fillna(df['manufacturer'].map(manufacturer_type_map))
```

[79]:
```
df['type'].isna().sum()
```

[79]:
```
np.int64(0)
```

[80]:
```
df["type"].unique()
```

[80]:
```
array(['pickup', 'truck', 'other', 'SUV', 'coupe', 'hatchback',
       'mini-van', 'sedan', 'offroad', 'bus', 'convertible', 'wagon',
       'van'], dtype=object)
```

[81]:
```
df.isna().sum()
```

[81]:
```
region              0
price               0
year                0
manufacturer        0
model               0
fuel                0
odometer            0
title_status        0
transmission        0
VIN            147590
size           267055
type                0
```

```
paint_color     106661
description          2
state                0
posting_date         0
condition            0
cylinders            0
drive                0
dtype: int64
```

8 size

[82]:
```
df["size"].unique()
```

[82]:
```
array([nan, 'full-size', 'mid-size', 'compact', 'sub-compact'],
      dtype=object)
```

[83]:
```
type_size_map = {
    'sedan': 'mid-size',
    'coupe': 'compact',
    'hatchback': 'sub-compact',
    'SUV': 'full-size',
    'pickup': 'full-size',
    'truck': 'full-size',
    'van': 'full-size',
    'mini-van': 'mid-size',
    'wagon': 'mid-size',
    'bus': 'full-size',
    'offroad': 'full-size',
    'convertible': 'compact',
    'other' : 'UNKNOWN'
}
```

[84]:
```
df['size'] = df['size'].fillna(df['type'].map(type_size_map))
```

[85]:
```
df.isna().sum()
```

[85]:
```
region              0
price               0
year                0
manufacturer        0
model               0
fuel                0
odometer            0
title_status        0
transmission        0
VIN            147590
size                0
```

```
type                0
paint_color     106661
description          2
state                0
posting_date         0
condition            0
cylinders            0
drive                0
dtype: int64
```

9 paint_color

[86]:
```
df["paint_color"] = df["paint_color"].fillna("UNKNOWN")
```

[87]:
```
df.isna().sum()
```

[87]:
```
region              0
price               0
year                0
manufacturer        0
model               0
fuel                0
odometer            0
title_status        0
transmission        0
VIN            147590
size                0
type                0
paint_color         0
description         2
state               0
posting_date        0
condition           0
cylinders           0
drive               0
dtype: int64
```

10 description

[88]:
```
df.dropna(subset=["description"],inplace = True)
```

[89]:
```
df.isna().sum()
```

[89]:
```
region              0
price               0
year                0
```

```
manufacturer         0
model                0
fuel                 0
odometer             0
title_status         0
transmission         0
VIN             147589
size                 0
type                 0
paint_color          0
description          0
state                0
posting_date         0
condition            0
cylinders            0
drive                0
dtype: int64
```

11 VIN

[90]:
```
df["VIN"] = df["VIN"].fillna("UNKNOWN")
```

[91]:
```
df.isna().sum()
```

[91]:
```
region          0
price           0
year            0
manufacturer    0
model           0
fuel            0
odometer        0
title_status    0
transmission    0
VIN             0
size            0
type            0
paint_color     0
description     0
state           0
posting_date    0
condition       0
cylinders       0
drive           0
dtype: int64
```

[92]:
```
df.sample(5)
```

[92]:
```
                       region  price    year manufacturer    model fuel  \
9023       flagstaff / sedona  22284  2020.0    chevrolet  trax lt  gas
94981   ft myers / SW florida  21998  2018.0       toyota     rav4  gas
317083                 eugene    142  2008.0        honda   accord  gas
60686             SF bay area   2500  2008.0        mazda   mazda5  gas
175840                 monroe   4900  1995.0         ford  f150 xl  gas

        odometer title_status transmission               VIN       size  \
9023        10.0        clean    automatic  KL7CJLSB0LB009667    UNKNOWN
94981    31980.0        clean    automatic            UNKNOWN   mid-size
317083   76328.0        clean    automatic  1HGCP26888A087378   mid-size
60686   169000.0        clean    automatic  JM1CR29L380312191   mid-size
175840  191000.0        clean       manual            UNKNOWN  full-size

         type paint_color                                        description  \
9023    other       white  Why Buy From Flagstaff Chevrolet?Welcome to Fl…
94981     SUV         red  Most common questions about this vehicle:  Wan…
317083  sedan     UNKNOWN  2008 Honda Accord 4dr I4 Auto EX-L w/Navi PZEV…
60686   sedan     UNKNOWN  2008 Mazda 5   Smogged and car runs but some m…
175840  truck       white  It is white with blue interior and has stereo …

       state              posting_date condition     cylinders drive
9023      az  2021-04-15T14:10:54-0700      good  12 cylinders   4wd
94981     fl  2021-04-26T09:20:15-0400       new   4 cylinders   fwd
317083    or  2021-04-26T09:09:47-0700       new   4 cylinders   fwd
60686     ca  2021-05-04T11:40:14-0700      fair   4 cylinders   4wd
175840    la  2021-04-28T09:19:44-0500       new  12 cylinders   4wd
```

12 VIN

13 description

14 posting_date

[93]:
```
df['posting_date'] = pd.to_datetime(df['posting_date'], utc=True)

df['posting_month'] = df['posting_date'].dt.month
df['posting_day'] = df['posting_date'].dt.day
df['posting_weekday'] = df['posting_date'].dt.weekday
df['posting_day'] = df['posting_date'].dt.day_name()
```

[94]:
```
df
```

[94]:
```
         region  price    year manufacturer                    model    fuel  \
27       auburn  33590  2014.0          gmc  sierra 1500 crew cab slt     gas
28       auburn  22590  2010.0    chevrolet            silverado 1500     gas
```

```
29       auburn  39590  2020.0    chevrolet       silverado 1500 crew     gas
30       auburn  30990  2017.0       toyota      tundra double cab sr     gas
31       auburn  15000  2013.0         ford                 f-150 xlt     gas
…          …     …      …            …                        …      …
426875  wyoming  23590  2019.0       nissan         maxima s sedan 4d     gas
426876  wyoming  30590  2020.0        volvo  s60 t5 momentum sedan 4d     gas
426877  wyoming  34990  2020.0     cadillac          xt4 sport suv 4d  diesel
426878  wyoming  28990  2018.0        lexus           es 350 sedan 4d     gas
426879  wyoming  30590  2019.0          bmw  4 series 430i gran coupe     gas

        odometer title_status transmission                VIN  …  \
27       57923.0        clean        other  3GTP1VEC4EG551563  …
28       71229.0        clean        other  1GCSCSE06AZ123805  …
29       19160.0        clean        other  3GCPWCED5LG130317  …
30       41124.0        clean        other  5TFRM5F17HX120972  …
31      128000.0        clean    automatic            UNKNOWN  …
…         …           …          …                  …  …
426875   32226.0        clean        other  1N4AA6AV6KC367801  …
426876   12029.0        clean        other  7JR102FKXLG042696  …
426877    4174.0        clean        other  1GYFZFR46LF088296  …
426878   30112.0        clean        other  58ABK1GG4JU103853  …
426879   22716.0        clean        other  WBA4J1C58KBM14708  …

       paint_color                                        description state  \
27           white  Carvana is the safer way to buy a car During t…    al
28            blue  Carvana is the safer way to buy a car During t…    al
29             red  Carvana is the safer way to buy a car During t…    al
30             red  Carvana is the safer way to buy a car During t…    al
31           black  2013 F-150 XLT V6 4 Door. Good condition. Leve…    al
…            …                                                 …   …
426875     UNKNOWN  Carvana is the safer way to buy a car During t…    wy
426876         red  Carvana is the safer way to buy a car During t…    wy
426877       white  Carvana is the safer way to buy a car During t…    wy
426878      silver  Carvana is the safer way to buy a car During t…    wy
426879     UNKNOWN  Carvana is the safer way to buy a car During t…    wy

                    posting_date  condition     cylinders drive posting_month  \
27     2021-05-04 17:31:18+00:00       good   8 cylinders   4wd             5
28     2021-05-04 17:31:08+00:00       good   8 cylinders   4wd             5
29     2021-05-04 17:31:25+00:00       good   8 cylinders   4wd             5
30     2021-05-04 15:41:31+00:00       good   8 cylinders   4wd             5
31     2021-05-03 19:02:03+00:00  excellent   6 cylinders   rwd             5
…                          …          …           …   …           …
426875 2021-04-04 09:21:31+00:00       good   6 cylinders   fwd             4
426876 2021-04-04 09:21:29+00:00       good  12 cylinders   fwd             4
426877 2021-04-04 09:21:17+00:00       good  12 cylinders   4wd             4
426878 2021-04-04 09:21:11+00:00       good   6 cylinders   fwd             4
```

```
426879 2021-04-04 09:21:07+00:00       good  12 cylinders   rwd             4

       posting_day  posting_weekday
27          Tuesday                1
28          Tuesday                1
29          Tuesday                1
30          Tuesday                1
31           Monday                0
…              …                  …
426875      Sunday                 6
426876      Sunday                 6
426877      Sunday                 6
426878      Sunday                 6
426879      Sunday                 6

[375715 rows x 22 columns]
```

[95]:
```
df['has_phone'] = df['description'].str.contains(r'\d{3}-\d{3}-\d{4}',␣
↪regex=True).astype(int)
df['desc_word_count'] = df['description'].apply(lambda x: len(str(x).split()))
df['has_warranty'] = df['description'].str.contains('warranty', case=False).
↪astype(int)
df['is_one_owner'] = df['description'].str.contains('one owner|1st owner',␣
↪case=False).astype(int)
```

[96]:
```
df["has_phone"].unique()
```

[96]:
```
array([1, 0])
```

[97]:
```
df["desc_word_count"].unique()
```

[97]:
```
array([ 681,  692,  690, …, 2870, 2781, 2934])
```

[98]:
```
df["has_warranty"].unique()
```

[98]:
```
array([0, 1])
```

[99]:
```
df["is_one_owner"].unique()
```

[99]:
```
array([0, 1])
```

[100]:
```
df['vin_country_code'] = df['VIN'].apply(lambda x: x[0] if x != "UNKNOWN" else␣
↪"UNKNOWN")
```

[101]:
```
country_map = {'1': 'USA', '4': 'USA', '5': 'USA', 'J': 'Japan', 'K': 'South␣
↪Korea', 'W': 'Germany'}
```

[102]:
```
df['origin_country'] = df['vin_country_code'].map(country_map).fillna('Other/
↪Unknown')
```

[103]:
```
df['origin_country']
```

[103]:
```
27          Other/Unknown
28                    USA
29          Other/Unknown
30                    USA
31          Other/Unknown
                ...
426875                USA
426876      Other/Unknown
426877                USA
426878                USA
426879            Germany
Name: origin_country, Length: 375715, dtype: object
```

15 Columns Cleaning

[104]:
```
df.columns
```

[104]:
```
Index(['region', 'price', 'year', 'manufacturer', 'model', 'fuel', 'odometer',
       'title_status', 'transmission', 'VIN', 'size', 'type', 'paint_color',
       'description', 'state', 'posting_date', 'condition', 'cylinders',
       'drive', 'posting_month', 'posting_day', 'posting_weekday', 'has_phone',
       'desc_word_count', 'has_warranty', 'is_one_owner', 'vin_country_code',
       'origin_country'],
      dtype='object')
```

Region

[107]:
```
df["region"].sample(43)
```

[107]:
```
271224              hudson valley
19683                 little rock
137472    spokane / coeur d'alene
411237                  wenatchee
51731                  sacramento
291171                 cincinnati
163331                 sioux city
189186               south coast
207453                   muskegon
41455               orange county
316546                     eugene
15640                      tucson
360489                  nashville
```

```
117604            tampa bay area
141244                   chicago
377931                   lubbock
91094                   delaware
282919             new hampshire
353405                cookeville
401150                bellingham
18202                fayetteville
210358           upper peninsula
398004                   roanoke
299690      dayton / springfield
321762                  portland
265665                    albany
421843                   madison
149598              st louis, MO
50119               reno / tahoe
383343           tyler / east TX
66399                    stockton
102351                     ocala
183934         cumberland valley
165961           kansas city, MO
369619       dallas / fort worth
378510        mcallen / edinburg
344599                charleston
34868                los angeles
117300            tampa bay area
76317                    denver
375653                   houston
69491             visalia-tulare
285245             new hampshire
Name: region, dtype: object
```

[131]:
```
df["region"]=df["region"].replace([",","/","'","-","  "]," ",regex=True)
```

[132]:
```
df["region"].sample(200)
```

[132]:
```
180578                 maine
150499            fort wayne
112640         south florida
46167                redding
6510         anchorage mat su
                ...
290424            cincinnati
248654             las vegas
351186     sioux falls SE SD
417378             green bay
126736    savannah hinesville
```

```
Name: region, Length: 200, dtype: object
```

15.0.1 Price

[133]:
```
df["price"].dtypes
```

[133]:
```
dtype('int64')
```

15.0.2 Year

[135]:
```
df["year"].dtypes
```

[135]:
```
dtype('float64')
```

[136]:
```
df["year"] = df["year"].astype(int)
```

[137]:
```
df["year"].dtypes
```

[137]:
```
dtype('int64')
```

[138]:
```
df["year"].unique
```

[138]:
```
<bound method Series.unique of 27        2014
28        2010
29        2020
30        2017
31        2013
          …
426875    2019
426876    2020
426877    2020
426878    2018
426879    2019
Name: year, Length: 375715, dtype: int64>
```

15.0.3 Manufacturer

[142]:
```
df["manufacturer"].unique()
```

[142]:
```
array(['gmc', 'chevrolet', 'toyota', 'ford', 'jeep', 'nissan', 'ram',
       'mazda', 'cadillac', 'honda', 'dodge', 'lexus', 'jaguar', 'buick',
       'chrysler', 'volvo', 'audi', 'infiniti', 'lincoln', 'alfa romeo',
       'subaru', 'Unknown', 'acura', 'hyundai', 'mercedes benz', 'bmw',
       'mitsubishi', 'volkswagen', 'porsche', 'kia', 'ferrari', 'mini',
       'pontiac', 'fiat', 'rover', 'tesla', 'saturn', 'mercury',
       'harley davidson', 'datsun', 'aston martin', 'land rover'],
      dtype=object)
```

[143]:
```
df["manufacturer"] = df["manufacturer"].replace(["-"]," ",regex=True)
```

[150]:
```
df["manufacturer"].unique()
```

[150]:
```
array(['gmc', 'chevrolet', 'toyota', 'ford', 'jeep', 'nissan', 'ram',
       'mazda', 'cadillac', 'honda', 'dodge', 'lexus', 'jaguar', 'buick',
       'chrysler', 'volvo', 'audi', 'infiniti', 'lincoln', 'alfa romeo',
       'subaru', 'Unknown', 'acura', 'hyundai', 'mercedes benz', 'bmw',
       'mitsubishi', 'volkswagen', 'porsche', 'kia', 'ferrari', 'mini',
       'pontiac', 'fiat', 'rover', 'tesla', 'saturn', 'mercury',
       'harley davidson', 'datsun', 'aston martin', 'land rover'],
      dtype=object)
```

[155]:
```
df["manufacturer"].dtypes
```

[155]:
```
dtype('O')
```

15.0.4 Model

[145]:
```
df["model"].unique()
```

[145]:
```
array(['sierra 1500 crew cab slt', 'silverado 1500',
       'silverado 1500 crew', …, 'gand wagoneer', '96 Suburban',
       'Paige Glenbrook Touring'], dtype=object)
```

[146]:
```
df["model"].sample(50)
```

[146]:
```
238702     f150 super cab xl pickup 4d
335452                              tl
259853          1500 crew cab big horn
119967                        regal t
209141                       escalade
90960                              x5
251134                     accord ex
262335            silverado 1500 crew
211627                  expedition el
263773                    elantra gls
333820        Genesis G80 3.8 Sedan 4D
223680                           f-350
310147                          forte
417770                          elite
385904                      mini-cab
150032                      corvette
144848                       durango
54300                        compass
61537                        liteace
9287                 300 300s sedan 4d
342834              International 4300
```

```
241637                       juke nismo
403546         lifted f350 power stroke
311712                           tacoma
86696                         xterra 4x4
140531       canyon extended cab pickup
329903           romeo stelvio ti sport
286518                  f350 king ranch
207387                      continental
150077                               3z
337648            silverado 1500 double
117982                      toytota rav4
288500       wrangler unlimited all new
179123                             1500
339374                           safari
347608                    envoy slt 4x4
366555             leaf sv hatchback 4d
273792                            rogue
128945                           lx 570
214744                          element
225501                           dakota
54284                            rx 330
201731                          sportage
315666    sierra 3500 slt 4dr crew cab
194727                            ml350
131772                  outback premium
269321        4runner sr5 sport utility
240723                          m-class
275773                    cc sport plus
262466                  f-250 xlt lariat
Name: model, dtype: object
```

[147]:
```
df=df.drop(["model"],axis = 1)
```

[149]:
```
df.columns
```

[149]:
```
Index(['region', 'price', 'year', 'manufacturer', 'fuel', 'odometer',
       'title_status', 'transmission', 'VIN', 'size', 'type', 'paint_color',
       'description', 'state', 'posting_date', 'condition', 'cylinders',
       'drive', 'posting_month', 'posting_day', 'posting_weekday', 'has_phone',
       'desc_word_count', 'has_warranty', 'is_one_owner', 'vin_country_code',
       'origin_country'],
      dtype='object')
```

15.0.5 Fuel

[157]:
```
df["fuel"].unique()
```

[157]:
```
array(['gas', 'other', 'diesel', 'hybrid', 'electric'], dtype=object)
```

15.0.6 Odometer

[162]:
```
df["odometer"].dtypes
```

[162]:
```
dtype('float64')
```

[163]:
```
df["odometer"]=df["odometer"].round()
```

[165]:
```
df["odometer"] = df["odometer"].astype(int)
```

[166]:
```
df["odometer"].dtypes
```

[166]:
```
dtype('int64')
```

15.0.7 Title_status

[169]:
```
df["title_status"].unique()
```

[169]:
```
array(['clean', 'rebuilt', 'lien', 'salvage', 'missing', 'parts only'],
      dtype=object)
```

15.0.8 Transmission

[171]:
```
df["transmission"].unique()
```

[171]:
```
array(['other', 'automatic', 'manual'], dtype=object)
```

15.0.9 VIN

[174]:
```
df["VIN"].unique()
```

[174]:
```
array(['3GTP1VEC4EG551563', '1GCSCSE06AZ123805', '3GCPWCED5LG130317', …,
       '2HGES15535H620534', '1FDWF37P64EA24868', 'SAJGX2749VCOO8376'],
      dtype=object)
```

[175]:
```
df=df.drop(["VIN"],axis=1)
```

[176]:
```
df.columns
```

[176]:
```
Index(['region', 'price', 'year', 'manufacturer', 'fuel', 'odometer',
       'title_status', 'transmission', 'size', 'type', 'paint_color',
       'description', 'state', 'posting_date', 'condition', 'cylinders',
       'drive', 'posting_month', 'posting_day', 'posting_weekday', 'has_phone',
       'desc_word_count', 'has_warranty', 'is_one_owner', 'vin_country_code',
       'origin_country'],
      dtype='object')
```

15.0.10 Size

[178]:
```
df["size"].unique()
```

[178]:
```
array(['full-size', 'UNKNOWN', 'compact', 'sub-compact', 'mid-size'],
      dtype=object)
```

[179]:
```
df["size"]=df["size"].replace("-"," ",regex=True)
```

15.0.11 Type

[182]:
```
df["type"].unique()
```

[182]:
```
array(['pickup', 'truck', 'other', 'SUV', 'coupe', 'hatchback',
       'mini-van', 'sedan', 'offroad', 'bus', 'convertible', 'wagon',
       'van'], dtype=object)
```

[183]:
```
df["type"]=df["type"].replace("-"," ",regex=True)
```

15.0.12 Paint_color

[186]:
```
df["paint_color"].unique()
```

[186]:
```
array(['white', 'blue', 'red', 'black', 'silver', 'grey', 'UNKNOWN',
       'brown', 'yellow', 'orange', 'green', 'custom', 'purple'],
      dtype=object)
```

15.0.13 Description

[188]:
```
df=df.drop(["description"],axis=1)
```

[190]:
```
df.columns
```

[190]:
```
Index(['region', 'price', 'year', 'manufacturer', 'fuel', 'odometer',
       'title_status', 'transmission', 'size', 'type', 'paint_color', 'state',
       'posting_date', 'condition', 'cylinders', 'drive', 'posting_month',
       'posting_day', 'posting_weekday', 'has_phone', 'desc_word_count',
       'has_warranty', 'is_one_owner', 'vin_country_code', 'origin_country'],
      dtype='object')
```

15.0.14 State

[192]:
```
df["state"].unique()
```

[192]:
```
array(['al', 'ak', 'az', 'ar', 'ca', 'co', 'ct', 'dc', 'de', 'fl', 'ga',
       'hi', 'id', 'il', 'in', 'ia', 'ks', 'ky', 'la', 'me', 'md', 'ma',
       'mi', 'mn', 'ms', 'mo', 'mt', 'nc', 'ne', 'nv', 'nj', 'nm', 'ny',
       'nh', 'nd', 'oh', 'ok', 'or', 'pa', 'ri', 'sc', 'sd', 'tn', 'tx',
```

```
       'ut', 'vt', 'va', 'wa', 'wv', 'wi', 'wy'], dtype=object)
```

15.0.15 Posting_date

[194]:
```
df=df.drop(["posting_date"],axis =1)
```

[195]:
```
df.columns
```

[195]:
```
Index(['region', 'price', 'year', 'manufacturer', 'fuel', 'odometer',
       'title_status', 'transmission', 'size', 'type', 'paint_color', 'state',
       'condition', 'cylinders', 'drive', 'posting_month', 'posting_day',
       'posting_weekday', 'has_phone', 'desc_word_count', 'has_warranty',
       'is_one_owner', 'vin_country_code', 'origin_country'],
      dtype='object')
```

15.0.16 Condition

[197]:
```
df["condition"].unique()
```

[197]:
```
array(['good', 'excellent', 'fair', 'like new', 'new', 'salvage'],
      dtype=object)
```

15.0.17 Cylinders

[199]:
```
df["cylinders"].unique()
```

[199]:
```
array(['8 cylinders', '6 cylinders', '12 cylinders', '4 cylinders',
       '5 cylinders', 'other', '3 cylinders', '10 cylinders'],
      dtype=object)
```

15.0.18 Drive

[201]:
```
df["drive"].unique()
```

[201]:
```
array(['4wd', 'rwd', 'fwd'], dtype=object)
```

15.0.19 posting_month

[203]:
```
df["posting_month"].unique()
```

[203]:
```
array([5, 4], dtype=int32)
```

15.0.20 Posting_day

[204]:
```
df["posting_day"].unique()
```

[204]:
```
array(['Tuesday', 'Monday', 'Sunday', 'Saturday', 'Friday', 'Thursday',
       'Wednesday'], dtype=object)
```

15.0.21 Posting_weekday

[208]:
```
df["posting_weekday"].unique()
```

[208]:
```
array([1, 0, 6, 5, 4, 3, 2], dtype=int32)
```

15.0.22 Has_phone

[209]:
```
df["has_phone"].unique()
```

[209]:
```
array([1, 0])
```

[210]:
```
df["has_phone"].dtypes
```

[210]:
```
dtype('int64')
```

15.0.23 Desc_word_count

[213]:
```
df["desc_word_count"].unique()
```

[213]:
```
array([ 681,  692,  690, …, 2870, 2781, 2934])
```

[214]:
```
df["desc_word_count"].dtypes
```

[214]:
```
dtype('int64')
```

15.0.24 Has_warranty

[215]:
```
df["has_warranty"].unique()
```

[215]:
```
array([0, 1])
```

[216]:
```
df["has_warranty"].dtypes
```

[216]:
```
dtype('int64')
```

15.0.25 Is_one_owner

[220]:
```
df["is_one_owner"].unique()
```

[220]:
```
array([0, 1])
```

15.0.26 Vin_country_code

[222]:
```
df["vin_country_code"].unique()
```

[222]:
```
array(['3', '1', '5', 'UNKNOWN', 'J', 'Z', '2', 'S', 'K', 'Y', '7', 'W',
       'L', '4', 'N', '0', 'B', 'M', '6', 'T', 'H', 'C', 'D', 'V', 'P',
       '8', 'I', 'F', 'A', 'U', '9', 'R', 'G', 'X', 'E', 'O'],
```

```
      dtype=object)
```

[223]:
```
df.columns
```

[223]:
```
Index(['region', 'price', 'year', 'manufacturer', 'fuel', 'odometer',
       'title_status', 'transmission', 'size', 'type', 'paint_color', 'state',
       'condition', 'cylinders', 'drive', 'posting_month', 'posting_day',
       'posting_weekday', 'has_phone', 'desc_word_count', 'has_warranty',
       'is_one_owner', 'vin_country_code', 'origin_country'],
      dtype='object')
```

15.0.27 Origin_country

[226]:
```
df["origin_country"].unique()
```

[226]:
```
array(['Other/Unknown', 'USA', 'Japan', 'South Korea', 'Germany'],
      dtype=object)
```

[229]:
```
df["origin_country"]=df["origin_country"].replace(["/Unknown"],"",regex= True)
```

[231]:
```
df["origin_country"].unique()
```

[231]:
```
array(['Other', 'USA', 'Japan', 'South Korea', 'Germany'], dtype=object)
```

[ ]:
```

```

16 Reset index

[ ]:
```

```

[241]:
```
df = df.reset_index(drop=True)
```

[243]:
```
df.sample()
```

[243]:
```
                region  price  year manufacturer fuel  odometer title_status  \
6577  anchorage mat su   9500  1979    chevrolet  gas     58522        clean

     transmission      size   type  … drive posting_month posting_day  \
6577       manual  mid size  sedan  …   4wd             4     Tuesday

     posting_weekday has_phone  desc_word_count has_warranty  is_one_owner  \
6577               1         0               79            0             0

      vin_country_code  origin_country
6577           UNKNOWN           Other

[1 rows x 24 columns]
```

[ ]:

[ ]:

17 Export New Data

[244]:
```
df.to_csv("D:/projects/USED_CAR/vehicles.csv/New_Data.csv")
```

[ ]: