```
#Important Libraries
import pandas as pd


df = pd.read_csv("/content/drive/MyDrive/Final_Sample (1).csv")
```

```
/usr/local/lib/python3.8/dist-packages/IPython/core/interactiveshell.py:3326: DtypeWarning: Columns (7) have mixed types.Specify dtype c
  exec(code_obj, self.user_global_ns, self.user_ns)
```

```
df
```

	Unnamed: 0	vin	back_legroom	body_type	city	city_fuel_economy	daysonmarket	dealer_zip	engine_cylinde
0	57071	1HGCV1F42LA123443	40.4 in	Sedan	Tipp City	30.0	15	45371	
1	56716	1GNERFKW4LJ280771	38.4 in	SUV / Crossover	Tampa	18.0	47	33619	
2	50056	5J8TB1H51AA006383	37.6 in	SUV / Crossover	Oak Lawn	17.0	4	60453	
3	50327	3GKALVEV0LL339320	39.7 in	SUV / Crossover	Marshfield	25.0	6	54449	
4	94738	3C4PDDGG9KT794869	36.1 in	SUV / Crossover	Newton	16.0	10	2458	
...	...	...	...	...	...	...	...	...	
199995	43655	3KPA24AD5LE341137	33.5 in	Sedan	San Antonio	33.0	8	78238	
199996	72169	2T3H1RFV1KW001728	37.8 in	SUV / Crossover	Phoenix	26.0	11	85014	
199997	17747	KL7CJKSB1LB333124	35.7 in	SUV / Crossover	Ozark	26.0	41	65721	
199998	44443	WAUFGAFC8EN111898	37.4 in	Sedan	Fargo	18.0	9	58103	
199999	83398	19XZE4F59ME003513	37.4 in	Sedan	Maumee	NaN	11	43537	I4 Hyt

200000 rows × 56 columns

Body Type Features:

First, number of unique values and number of vehicles for different body types:

```
print("Number of unique values = " , len(df["body_type"].unique()))
print("Number of cars for different body types: ")
df["body_type"].value_counts()
```

```
Number of unique values =  7
Number of cars for different body types:
SUV / Crossover    114892
Sedan               60398
Hatchback            7122
Minivan              6449
Coupe                5676
Wagon                3351
Convertible          2112
Name: body_type, dtype: int64
```

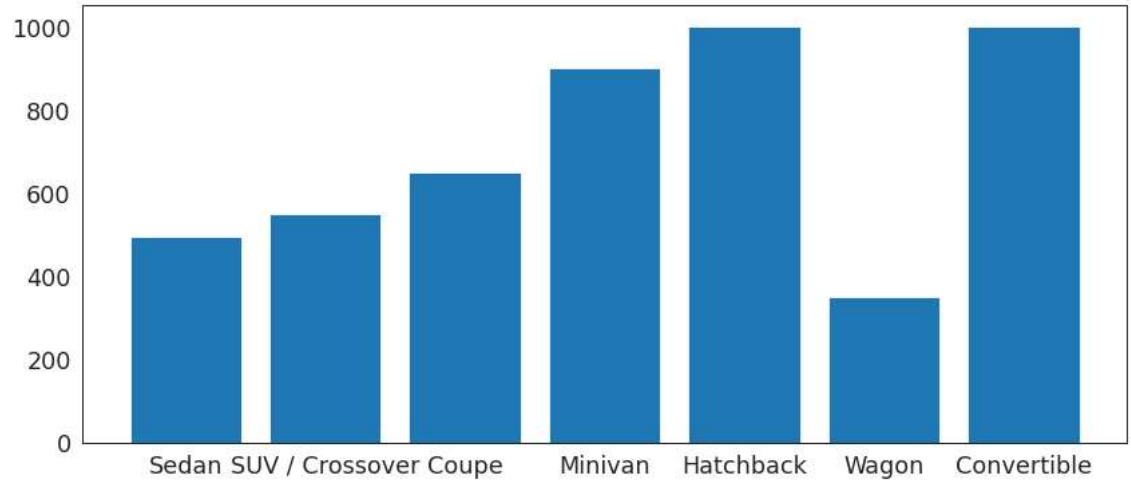
Second, Number of missing values:

```
print("Number of missing values = ", df["body_type"].isna().sum())
```

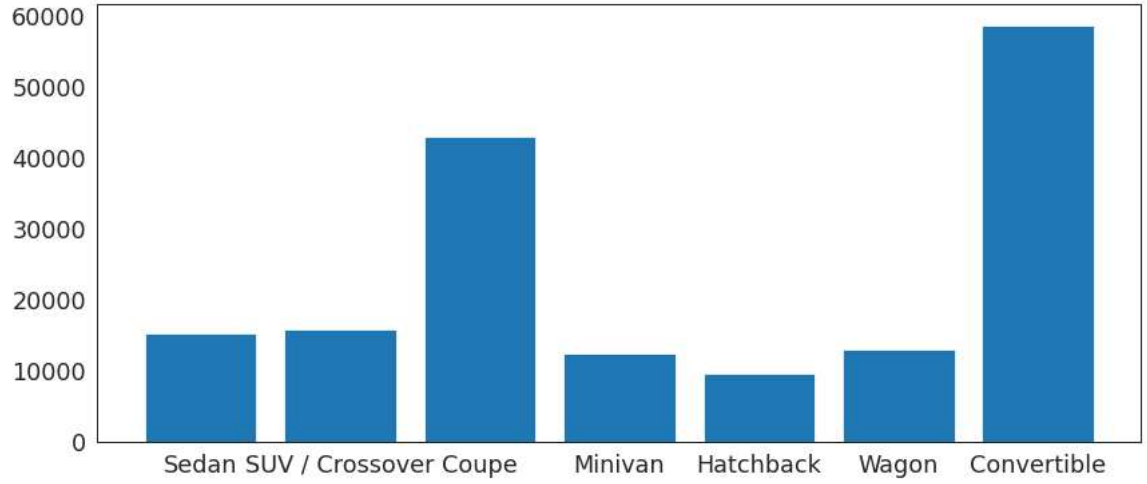
```
Number of missing values =  0
```

```
import matplotlib.pyplot as plt
lis = df["body_type"].unique()
means = {}
maxs = {}
mins = {}
stds= {}
for item in lis:
  df1 = df[df.body_type == item]
  m = df1["price"].mean()
  means[item] = m
  m = df1["price"].max()
  maxs[item] = m
  m = df1["price"].min()
  mins[item] = m
  m = df1["price"].std()
  stds[item] = m

print("Plotting the means of prices for different Body Types")
names = list(means.keys())
values = list(means.values())
plt.figure(figsize=(14,6))
plt.bar(range(len(means)), values, tick_label=names)
plt.show()

print("Plotting the maximum values of price for different Body Types")
names = list(maxs.keys())
values = list(maxs.values())
plt.figure(figsize=(14,6))
plt.bar(range(len(means)), values, tick_label=names)
plt.show()


print("Plotting the minimum values of price for different Body Types")
names = list(mins.keys())
values = list(mins.values())
plt.figure(figsize=(14,6))
plt.bar(range(len(means)), values, tick_label=names)
plt.show()


print("Plotting standard deviations values of price for different Body Types")
names = list(stds.keys())
values = list(stds.values())
plt.figure(figsize=(14,6))
plt.bar(range(len(stds)), values, tick_label=names)
plt.show()
```


```
Plotting the minimum values of price for different Body Types
```




```
Plotting standard deviations values of price for different Body Types
```



From the previous results, we can see that different body types have an effect on the price based on the means, maximum values, minimum values, standard deviations of different car body types. So, it is clearly important to keep this feature. And since, this feature contains categorical data it is important to convert it to numerical data. We decided to use one hot encoding for this feature.

Converting Body types from categorical to numerical data using one hot encoding.

```
new_cols = {}
for val in lis:
  new_cols[val] = []
for i in range(0, len(df)):
  target_col =df.at[i,"body_type"]
  for val in lis:
    if(target_col == val):
      new_cols[val].append(1)
    else:
      new_cols[val].append(0)



for val in lis:
  df[val] = new_cols[val]


df.info()
```

```
      7   dealer_zip            200000 non-null  object
      8   engine_cylinders      194920 non-null  object
      9   engine_displacement   191357 non-null  float64
      10  engine_type           194920 non-null  object
      11  exterior_color        199996 non-null  object
      12  fleet                 108332 non-null  object
      13  frame_damaged         108332 non-null  object
      14  franchise_dealer      200000 non-null  bool
      15  franchise_make        160488 non-null  object
      16  front_legroom         192540 non-null  object
      17  fuel_tank_volume      192540 non-null  object
      18  fuel_type             196297 non-null  object
      19  has_accidents         108332 non-null  object
      20  height                192540 non-null  object
      21  highway_fuel_economy  176380 non-null  float64
      22  horsepower            191357 non-null  float64
      23  interior_color        199990 non-null  object
      24  isCab                 108332 non-null  object
      25  is_new                200000 non-null  bool
      26  latitude              200000 non-null  float64
      27  length                192540 non-null  object
      28  listed_date           200000 non-null  object
      29  listing_color         200000 non-null  object
      30  listing_id            200000 non-null  int64
      31  longitude             200000 non-null  float64
      32  major_options         188278 non-null  object
      33  make_name             200000 non-null  object
      34  maximum_seating       192540 non-null  object
      35  mileage               191049 non-null  float64
      36  model_name            200000 non-null  object
      37  owner_count           103263 non-null  float64
      38  power                 170366 non-null  object
      39  price                 200000 non-null  float64
      40  salvage               108332 non-null  object
      41  savings_amount        200000 non-null  int64
      42  seller_rating         197294 non-null  float64
      43  sp_id                 199993 non-null  float64
      44  sp_name               200000 non-null  object
      45  theft_title           108332 non-null  object
      46  torque                167437 non-null  object
      47  transmission          195802 non-null  object
      48  transmission_display  195802 non-null  object
      49  trimId                195391 non-null  object
      50  trim_name             195368 non-null  object
      51  wheel_system          193140 non-null  object
      52  wheel_system_display  193140 non-null  object
      53  wheelbase             192540 non-null  object
      54  width                 192540 non-null  object
      55  year                  200000 non-null  int64
      56  Sedan                 200000 non-null  int64
      57  SUV / Crossover       200000 non-null  int64
      58  Coupe                 200000 non-null  int64
      59  Minivan               200000 non-null  int64
      60  Hatchback             200000 non-null  int64
      61  Wagon                 200000 non-null  int64
      62  Convertible           200000 non-null  int64
     dtypes: bool(2), float64(11), int64(12), object(38)
     memory usage: 93.5+ MB
```

Back Legroom Feature:

As we can see, the **back** legroom feature describes the space available for legs in the **back** seats. So, it is mainly a distance so, it should be a numeric value. However, it is a string object in the data because the unit is included. As a result, the first step in cleaning the data in this feature will be removing the units and converting the data from strings to floats

```
df["back_legroom"].unique()
```

```
array(['40.4 in', '38.4 in', '37.6 in', '39.7 in', '36.1 in', '35.6 in',
       '33.7 in', '41.1 in', '37.2 in', '36.6 in', '34.8 in', '33.1 in',
       '38.2 in', nan, '35.7 in', '41.4 in', '38.6 in', '40.3 in',
       '35.8 in', '35.1 in', '39.1 in', '39.6 in', '39.8 in', '40.1 in',
       '37 in', '35.3 in', '34.3 in', '38 in', '37.8 in', '38.3 in',
       '36 in', '39.5 in', '42 in', '37.4 in', '40.2 in', '38.9 in',
       '40.9 in', '29.9 in', '38.7 in', '36.5 in', '37.3 in', '37.9 in',
       '41.7 in', '34.7 in', '39 in', '36.2 in', '40.7 in', '33.2 in',
       '31.9 in', '40.6 in', '39.3 in', '34 in', '38.5 in', '31 in',
       '31.4 in', '--', '39.9 in', '33 in', '30.9 in', '39.4 in',
       '36.8 in', '35.2 in', '31.7 in', '32 in', '32.7 in', '31.2 in',
       '41.5 in', '35 in', '33.5 in', '38.1 in', '34.6 in', '35.4 in',
       '34.2 in', '41 in', '37.5 in', '36.7 in', '41.3 in', '40 in',
       '33.9 in', '32.9 in', '38.8 in', '32.6 in', '42.4 in', '33.3 in',
       '27.1 in', '33.4 in', '34.4 in', '34.5 in', '36.4 in', '39.2 in',
       '29.8 in', '35.9 in', '37.1 in', '36.3 in', '29.2 in', '34.1 in',
       '36.9 in', '35.5 in', '29 in', '34.9 in', '44.3 in', '37.7 in',
       '33.6 in', '42.6 in', '27.3 in', '41.9 in', '30.8 in', '33.8 in',
       '30.5 in', '32.8 in', '32.1 in', '46.8 in', '30.6 in', '28.2 in',
       '47.5 in', '32.5 in', '42.3 in', '31.1 in', '31.3 in', '31.8 in',
       '30.3 in', '27.2 in', '32.2 in', '43 in', '30.4 in', '41.6 in',
       '44.4 in', '29.5 in', '40.5 in', '41.8 in', '30.1 in', '43.2 in',
       '25.9 in', '27 in', '26.4 in', '27.6 in', '32.4 in', '32.3 in',
       '42.9 in', '30 in', '28.5 in', '31.5 in', '40.8 in', '43.4 in',
       '43.7 in', '26.8 in', '28.1 in', '28.9 in', '30.2 in', '3.5 in',
       '42.2 in', '23.8 in', '44.1 in', '13 in', '24.8 in', '41.2 in',
       '28.3 in', '24 in', '47.4 in', '28.7 in', '20.2 in', '28.8 in',
       '29.3 in', '14 in', '30.7 in', '27.4 in', '23.1 in', '42.1 in',
       '27.5 in', '31.6 in', '46.9 in', '43.3 in', '23.4 in', '28 in',
       '0 in', '29.6 in', '28.6 in', '27.7 in', '23.7 in', '23 in',
       '49 in', '26 in', '24.9 in'], dtype=object)
```

```
from pandas.core.api import notnull
for i in range(0, len(df)):
  temp = df.at[i,"back_legroom"]
  if(temp != '--' and temp is not None ):
    x= type("ojmlk")
    if(type(temp) == x):
      temp1 = temp.replace(" in", '')
      df['back_legroom'] = df['back_legroom'].replace([temp],float(temp1))

  elif (temp == "--"):
      df['back_legroom'] = df['back_legroom'].replace([temp],None)
```

```
pip install fitter
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: fitter in /usr/local/lib/python3.8/dist-packages (1.5.2)
Requirement already satisfied: scipy>=0.18 in /usr/local/lib/python3.8/dist-packages (from fitter) (1.7.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages (from fitter) (1.22.4)
Requirement already satisfied: click in /usr/local/lib/python3.8/dist-packages (from fitter) (8.1.3)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.8/dist-packages (from fitter) (3.5.3)
Requirement already satisfied: joblib in /usr/local/lib/python3.8/dist-packages (from fitter) (1.2.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.8/dist-packages (from fitter) (4.64.1)
Requirement already satisfied: pandas in /usr/local/lib/python3.8/dist-packages (from fitter) (1.3.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.8/dist-packages (from matplotlib->fitter) (23.0)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.8/dist-packages (from matplotlib->fitter) (2.8.2)
Requirement already satisfied: pyparsing>=2.2.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib->fitter) (3.0.9)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib->fitter) (1.4.4)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-packages (from matplotlib->fitter) (0.11.0)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.8/dist-packages (from matplotlib->fitter) (8.4.0)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.8/dist-packages (from matplotlib->fitter) (4.38.0)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.8/dist-packages (from pandas->fitter) (2022.7.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-packages (from python-dateutil>=2.7->matplotlib->fitter) (1.15.
```

Statistics of the Back legroom feature after converting it to float

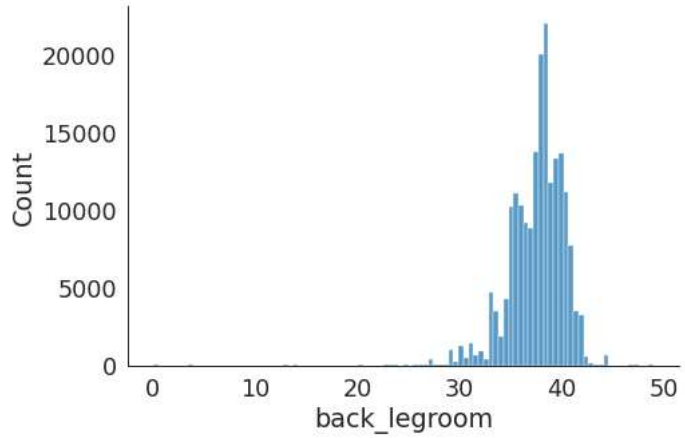
```
print("Number of missing values = ", df["back_legroom"].isna().sum())
```

```
Number of missing values =  7570
```

Statistics before induction of missing values:

```
import numpy as np
import pandas as pd
import seaborn as sns
from fitter import Fitter, get_common_distributions, get_distributions
from scipy import stats
sns.set_style('white')
sns.set_context("paper", font_scale = 2)
sns.displot(data=df, x="back_legroom", kind="hist", bins = 100, aspect = 1.5)
outlirs = df.describe()["back_legroom"]

print("The mean = ",outlirs["mean"])
print("The standard deviation = ", outlirs["std"])
print("The minimum value =", outlirs["min"])
print("The maximum value =", outlirs["max"])
print("The median = ", outlirs["50%"])
print("The first quartile = ", outlirs["25%"])
print("The third quartile = ", outlirs["75%"])
print("Distribution of the back legroom feature using histogram")
```

```
The mean =  37.621562646157045
The standard deviation =  2.528247331488303
The minimum value = 0.0
The maximum value = 49.0
The median =  38.0
The first quartile =  36.1
The third quartile =  39.5
Distribution of the back legroom feature using histogram
```




```
array = df["back_legroom"].to_numpy()
#print(array)
print("Distribution of the back legroom feature using Box plot")
sns.boxplot(df['back_legroom'])
```

```
Distribution of the back legroom feature using Box plot
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From \
```

We can see that the data contains outliers from the box plot.

Induction of missing values: We decided to assign the nulll values of the back legroom feature with the average value over the data set.

```
import math
average = df["back_legroom"].mean()
for i in range(0, len(df)):

  if(math.isnan(df.at[i, "back_legroom"])):

    df.loc[i,"back_legroom"] = average
```

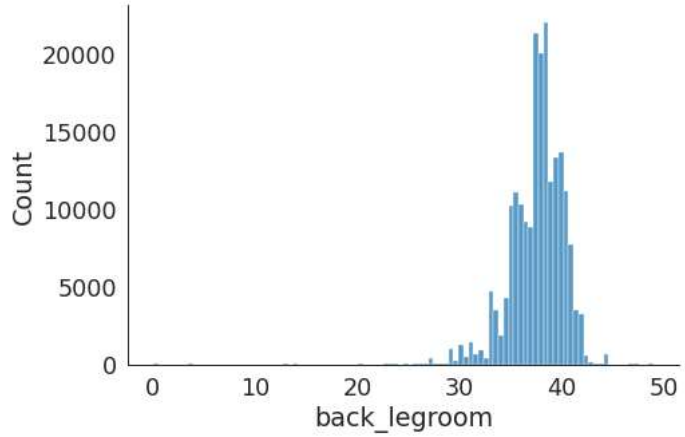
Statistics after induction of missing values:

```
print("Number of missing values = ", df["back_legroom"].isna().sum())
```

```
Number of missing values =  0
```

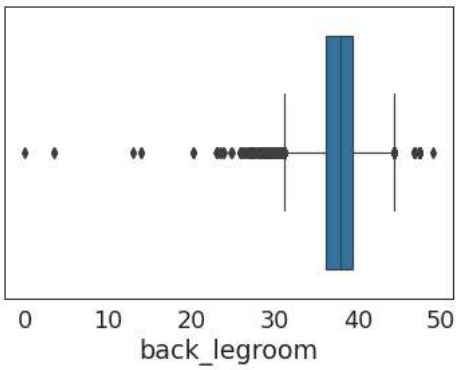
```
import numpy as np
import pandas as pd
import seaborn as sns
from fitter import Fitter, get_common_distributions, get_distributions
from scipy import stats
sns.set_style('white')
sns.set_context("paper", font_scale = 2)
sns.displot(data=df, x="back_legroom", kind="hist", bins = 100, aspect = 1.5)
outlirs = df.describe()["back_legroom"]
print("The mean = ",outlirs["mean"])
print("The standard deviation = ", outlirs["std"])
print("The minimum value =", outlirs["min"])
print("The maximum value =", outlirs["max"])
print("The median = ", outlirs["50%"])
print("The first quartile = ", outlirs["25%"])
print("The third quartile = ", outlirs["75%"])
print("Distribution of the back legroom feature using histogram")
```

```
The mean =  37.621562646157045
The standard deviation =  2.479938477268288
The minimum value = 0.0
The maximum value = 49.0
The median =  38.0
The first quartile =  36.1
The third quartile =  39.4
Distribution of the back legroom feature using histogram
```




```
array = df["back_legroom"].to_numpy()
#print(array)
print("Distribution of the back legroom feature using Box plot")
sns.boxplot(df['back_legroom'])
```

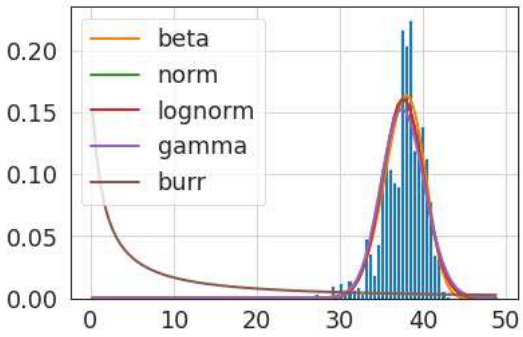
```
Distribution of the back legroom feature using Box plot
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From \
  warnings.warn(
<AxesSubplot:xlabel='back_legroom'>
```




```
BL = df["back_legroom"].values

BL = list(BL)
f = Fitter(BL,
           distributions=['gamma',
                          'lognorm',
                          "beta",
                          "burr",
                          "norm"], timeout = 240 )
f.fit()
f.summary()
```

```
Fitting 5 distributions: 100%|██████████| 5/5 [00:22<00:00,  4.55s/it]
```

	sumsquare_error	aic	bic	kl_div	ks_statistic	ks_pvalue
beta	0.020693	4443.207193	-3.216762e+06	inf	0.076949	0.0
norm	0.025655	6422.033169	-3.173797e+06	inf	0.095970	0.0
lognorm	0.025857	6658.315422	-3.172213e+06	inf	0.097502	0.0
gamma	0.029691	9500.140354	-3.144559e+06	inf	0.107404	0.0
burr	0.332004	976.927673	-2.661688e+06	inf	0.631717	0.0



From the above figure, we can see that the beta and normal distribution fits the data the most

```
import scipy.stats
r = scipy.stats.pearsonr(df["back_legroom"], df["price"])
print("Correlation Coeffecient = ", r[0])
```

```
Correlation Coeffecient =  0.17575136513911482
```

The final step in dealing with the data is to scale it. We will use Satndard z scaling to help us deal with outliers

```
des = df.describe()["back_legroom"]
m = des["mean"]
std = des["std"]
for i in range(0, len(df)):
```

```
  df.loc[i, "back_legroom"] = (df.loc[i, "back_legroom"] - m)/std
print(df["back_legroom"])
```

```
0         1.120365
1         0.313894
2        -0.008695
3         0.838100
4        -0.613549
              ...
199995   -1.661962
199996    0.071952
199997   -0.774843
199998   -0.089342
199999   -0.089342
Name: back_legroom, Length: 200000, dtype: float64
```

```
df.info()
```

```
 7   dealer_zip            200000 non-null  object
 8   engine_cylinders      194920 non-null  object
 9   engine_displacement   191357 non-null  float64
 10  engine_type           194920 non-null  object
 11  exterior_color        199996 non-null  object
 12  fleet                 108332 non-null  object
 13  frame_damaged         108332 non-null  object
 14  franchise_dealer      200000 non-null  bool
 15  franchise_make        160488 non-null  object
 16  front_legroom         192540 non-null  object
 17  fuel_tank_volume      192540 non-null  object
 18  fuel_type             196297 non-null  object
 19  has_accidents         108332 non-null  object
 20  height                192540 non-null  object
 21  highway_fuel_economy  176380 non-null  float64
 22  horsepower            191357 non-null  float64
 23  interior_color        199990 non-null  object
 24  isCab                 108332 non-null  object
 25  is_new                200000 non-null  bool
 26  latitude              200000 non-null  float64
 27  length                192540 non-null  object
 28  listed_date           200000 non-null  object
 29  listing_color         200000 non-null  object
 30  listing_id            200000 non-null  int64
 31  longitude             200000 non-null  float64
 32  major_options         188278 non-null  object
 33  make_name             200000 non-null  object
 34  maximum_seating       192540 non-null  object
 35  mileage               191049 non-null  float64
 36  model_name            200000 non-null  object
 37  owner_count           103263 non-null  float64
 38  power                 170366 non-null  object
 39  price                 200000 non-null  float64
 40  salvage               108332 non-null  object
 41  savings_amount        200000 non-null  int64
 42  seller_rating         197294 non-null  float64
 43  sp_id                 199993 non-null  float64
 44  sp_name               200000 non-null  object
 45  theft_title           108332 non-null  object
 46  torque                167437 non-null  object
 47  transmission          195802 non-null  object
 48  transmission_display  195802 non-null  object
 49  trimId                195391 non-null  object
 50  trim_name             195368 non-null  object
 51  wheel_system          193140 non-null  object
 52  wheel_system_display  193140 non-null  object
 53  wheelbase             192540 non-null  object
 54  width                 192540 non-null  object
 55  year                  200000 non-null  int64
 56  Sedan                 200000 non-null  int64
 57  SUV / Crossover       200000 non-null  int64
 58  Coupe                 200000 non-null  int64
 59  Minivan               200000 non-null  int64
 60  Hatchback             200000 non-null  int64
 61  Wagon                 200000 non-null  int64
 62  Convertible           200000 non-null  int64
dtypes: bool(2), float64(12), int64(12), object(37)
memory usage: 93.5+ MB
```

Front Legroom Feature:

As we can see, the front legroom feature describes the space available for legs in the Front seats. So, it is mainly a distance so, it should be a numeric value. However, it is a string object in the data because the unit is included. As a result, the first step in cleaning the data in this feature will be removing the units and converting the data from strings to floats

```
df["front_legroom"].unique()
```

```
array(['42.3 in', '41 in', '41.8 in', '40.9 in', '40.8 in', '45.5 in',
       '42.2 in', '41.6 in', '41.4 in', '46.1 in', '43.6 in', '41.5 in',
       nan, '40.3 in', '41.1 in', '41.3 in', '41.2 in', '43 in',
       '45.8 in', '44.1 in', '44.3 in', '42.5 in', '42.9 in', '44.5 in',
       '45.3 in', '42.4 in', '42.1 in', '43.9 in', '40.2 in', '39.9 in',
       '43.1 in', '42 in', '41.7 in', '40.5 in', '40.7 in', '44.2 in',
       '42.6 in', '--', '43.7 in', '43.3 in', '40.4 in', '42.8 in',
       '43.2 in', '43.5 in', '40.6 in', '41.9 in', '45 in', '39.6 in',
       '44 in', '43.8 in', '44.4 in', '39.1 in', '45.1 in', '37.7 in',
       '45.7 in', '39.8 in', '46.9 in', '39 in', '36.6 in', '45.4 in',
       '52.5 in', '67 in', '43.4 in', '40 in', '44.7 in', '42.7 in',
       '53.6 in', '46.3 in', '44.8 in', '39.5 in', '46.4 in', '40.1 in',
       '38.9 in', '38 in', '44.6 in', '39.4 in', '38.7 in', '39.3 in',
       '45.9 in', '46 in', '38.6 in', '38.4 in', '33 in'], dtype=object)
```

```
from pandas.core.api import notnull
for i in range(0, len(df)):
  temp = df.at[i,"front_legroom"]
  if(temp != '--' and temp is not None ):
    x= type("ojmlk")
    if(type(temp) == x):
      temp1 = temp.replace(" in", '')
      df['front_legroom'] = df['front_legroom'].replace([temp],float(temp1))

  elif (temp == "--"):
      df['front_legroom'] = df['front_legroom'].replace([temp],None)
```

Statistics of the Back legroom feature after converting it to float

```
print("Number of missing values = ", df["front_legroom"].isna().sum())
```

```
Number of missing values =  7521
```

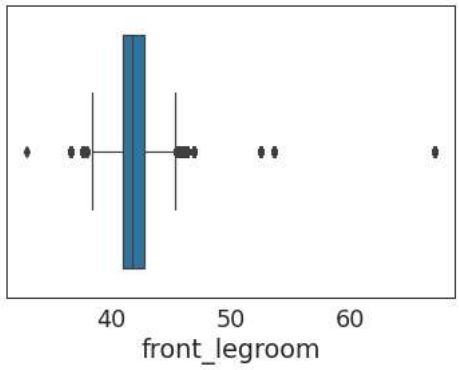
Statistics before induction of missing values:

```
import·numpy·as·np
import·pandas·as·pd
import·seaborn·as·sns
from·fitter·import·Fitter,·get_common_distributions,·get_distributions
from·scipy·import·stats
sns.set_style('white')
sns.set_context("paper",·font_scale·=·2)
sns.displot(data=df,·x="front_legroom",·kind="hist",·bins·=·100,·aspect·=·1.5)
outlirs·=·df.describe()["front_legroom"]
print("The·mean·=·",outlirs["mean"])
print("The·standard·deviation·=·",·outlirs["std"])
print("The·minimum·value·=",·outlirs["min"])
print("The·maximum·value·=",·outlirs["max"])
print("The·median·=·",·outlirs["50%"])
print("The·first·quartile·=·",·outlirs["25%"])
print("The·third·quartile·=·",·outlirs["75%"])
print("Distribution·of·the·front·legroom·feature·using·histogram")
```

```
The mean =  42.06399815044758
The standard deviation =  1.4409755400728699
The minimum value = 33.0
The maximum value = 67.0
The median =  41.8
The first quartile =  41.0
The third quartile =  42.8
Distribution of the front legroom feature using histogram
```

```
array = df["front_legroom"].to_numpy()
#print(array)
print("Distribution of the front legroom feature using Box plot")
sns.boxplot(df['front_legroom'])
```

```
Distribution of the front legroom feature using Box plot
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From v
  warnings.warn(
<AxesSubplot:xlabel='front_legroom'>
```



Induction of missing values

We will assign the null values to the average

```
import math
average = df["front_legroom"].mean()
for i in range(0, len(df)):

  if(math.isnan(df.at[i, "front_legroom"])):

    df.loc[i,"front_legroom"] = average
```

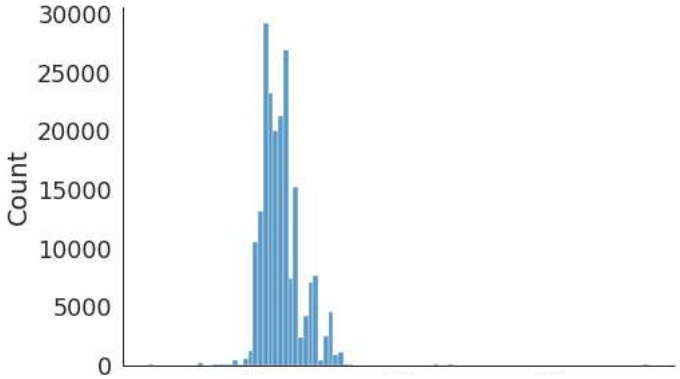
Statistics after the induction of missing values:

```
print("Number of missing values = ", df["front_legroom"].isna().sum())
```

```
Number of missing values =  0
```

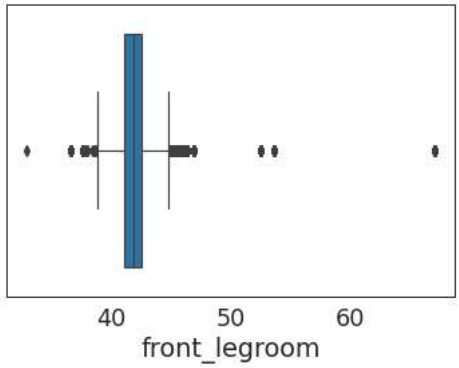
```
import·numpy·as·np
import·pandas·as·pd
import·seaborn·as·sns
from·fitter·import·Fitter,·get_common_distributions,·get_distributions
from·scipy·import·stats
sns.set_style('white')
sns.set_context("paper",·font_scale·=·2)
sns.displot(data=df,·x="front_legroom",·kind="hist",·bins·=·100,·aspect·=·1.5)
outlirs·=·df.describe()["front_legroom"]
print("The·mean·=·",outlirs["mean"])
print("The·standard·deviation·=·",·outlirs["std"])
print("The·minimum·value·=",·outlirs["min"])
print("The·maximum·value·=",·outlirs["max"])
print("The·median·=·",·outlirs["50%"])
print("The·first·quartile·=·",·outlirs["25%"])
print("The·third·quartile·=·",·outlirs["75%"])
print("Distribution·of·the·front·legroom·feature·using·histogram")
```

```
The mean =  42.06399815044758
The standard deviation =  1.4136218375330107
The minimum value = 33.0
The maximum value = 67.0
The median =  41.9
The first quartile =  41.1
The third quartile =  42.6
Distribution of the front legroom feature using histogram
```




```
array·=·df["front_legroom"].to_numpy()
#print(array)
print("Distribution·of·the·front·legroom·feature·using·Box·plot")
sns.boxplot(df['front_legroom'])
```

```
Distribution of the front legroom feature using Box plot
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From \
  warnings.warn(
<AxesSubplot:xlabel='front_legroom'>
```

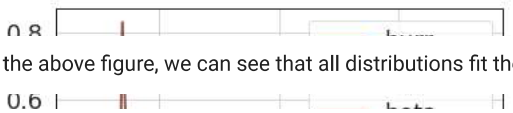


From the above plot, we can see that the data contain some outliers

```
BL·=·df["front_legroom"].values

BL·=·list(BL)
f·=·Fitter(BL,
··········distributions=['gamma',
·····················'lognorm',
·····················"beta",
·····················"burr",
·····················"norm"],·timeout·=·240·)
f.fit()
f.summary()
```

```
Fitting 5 distributions: 100%|██████████| 5/5 [00:24<00:00,  4.88s/it]
```

	sumsquare_error	aic	bic	kl_div	ks_statistic	ks_pvalue
burr	0.073301	2591.996098	-2.963802e+06	inf	0.072413	0.0
lognorm	0.099992	5015.417477	-2.901711e+06	inf	0.087710	0.0
beta	0.101885	inf	-2.897947e+06	inf	0.090287	0.0
norm	0.140841	8228.919011	-2.833215e+06	inf	0.120671	0.0
gamma	1.833372	inf	-2.319947e+06	inf	0.978154	0.0



From the above figure, we can see that all distributions fit the data except of the gamma distribution

```
import scipy.stats
r = scipy.stats.pearsonr(df["front_legroom"], df["price"])
print("Correlation Coeffecient = ", r[0])
```

```
Correlation Coeffecient =  0.0037592106046482076
```

The final step in dealing with the data is to scale it. We will use Satndard z scaling to help us deal with outliers.

```
des = df.describe()["front_legroom"]
m = des["mean"]
std = des["std"]
for i in range(0, len(df)):
  df.loc[i, "front_legroom"] = (df.loc[i, "front_legroom"] - m)/std
print(df["front_legroom"])
```

```
0         0.166948
1        -0.752675
2        -0.186753
3        -0.823416
4        -0.894156
            ...
199995    0.025468
199996   -0.752675
199997   -0.894156
199998   -0.540454
199999    0.166948
Name: front_legroom, Length: 200000, dtype: float64
```

Fuel Tank Volume Feature: As we can see, the Fuel tank volume feature describes the number of gallons in the fuel tank. So, it is mainly a distance so, it should be a numeric value. However, it is a string object in the data because the unit is included. As a result, the first step in cleaning the data in this feature will be removing the units and converting the data from strings to floats

```
df["fuel_tank_volume"].unique()
```

```
array(['14.8 gal', '19.4 gal', '18 gal', '15.6 gal', '21.1 gal',
       '18.5 gal', '15.8 gal', '16.5 gal', '17 gal', '20.5 gal',
       '19.5 gal', '15.9 gal', '12.8 gal', '16.4 gal', nan, '14 gal',
       '13.2 gal', '24.6 gal', '17.4 gal', '12.7 gal', '33.5 gal',
       '15.3 gal', '18.8 gal', '16.6 gal', '11.9 gal', '22 gal',
       '19.2 gal', '28 gal', '14.5 gal', '18.6 gal', '21.5 gal', '24 gal',
       '13.5 gal', '31 gal', '20 gal', '22.5 gal', '14.4 gal', '16 gal',
       '19 gal', '17.7 gal', '15.7 gal', '14.7 gal', '12.4 gal',
       '11.4 gal', '21.9 gal', '9.2 gal', '18.1 gal', '21 gal',
       '10.8 gal', '19.8 gal', '15.5 gal', '17.2 gal', '13.6 gal',
       '13 gal', '16.9 gal', '14.9 gal', '12 gal', '23.3 gal', '18.4 gal',
       '26 gal', '22.7 gal', '17.3 gal', '16.2 gal', '15.1 gal',
       '17.9 gal', '10.5 gal', '23 gal', '14.3 gal', '19.1 gal',
       '27.8 gal', '17.1 gal', '11.1 gal', '9.3 gal', '21.7 gal',
       '13.7 gal', '9.5 gal', '14.2 gal', '27.7 gal', '11.6 gal',
       '23.6 gal', '11.3 gal', '19.3 gal', '10.6 gal', '18.3 gal',
       '13.1 gal', '12.2 gal', '16.1 gal', '22.4 gal', '17.5 gal',
       '17.8 gal', '23.8 gal', '25 gal', '26.4 gal', '32 gal', '21.8 gal',
       '8.9 gal', '9 gal', '25.1 gal', '16.3 gal', '25.4 gal', '15 gal',
       '28.3 gal', '20.1 gal', '11.8 gal', '23.5 gal', '27.6 gal',
       '22.2 gal', '20.3 gal', '10 gal', '2.4 gal', '23.7 gal', '--',
       '20.6 gal', '21.6 gal', '7 gal', '18.2 gal', '21.4 gal',
       '16.8 gal', '22.8 gal', '27 gal', '8.7 gal', '20.4 gal', '44 gal',
       '15.4 gal', '37 gal', '1.9 gal', '27.3 gal', '15.2 gal',
       '24.3 gal', '12.1 gal', '24.5 gal', '7.7 gal', '29 gal', '11 gal',
       '16.7 gal', '14.6 gal', '12.6 gal', '17.6 gal', '23.9 gal',
```

```
            '22.3 gal', '23.2 gal', '12.3 gal', '24.1 gal', '2.3 gal',
            '21.2 gal', '39 gal', '7.8 gal', '30 gal', '8 gal', '31.5 gal',
            '42 gal', '8.5 gal', '22.6 gal', '20.2 gal', '10.3 gal',
            '25.6 gal'], dtype=object)
```

```
from pandas.core.api import notnull
for i in range(0, len(df)):
  temp = df.at[i,"fuel_tank_volume"]
  if(temp != '--' and temp is not None ):
    x= type("ojmlk")
    if(type(temp) == x):
      temp1 = temp.replace(" gal", '')
      df['fuel_tank_volume'] = df['fuel_tank_volume'].replace([temp],float(temp1))

  elif (temp == "--"):
      df['fuel_tank_volume'] = df['fuel_tank_volume'].replace([temp],None)
```

```
df.info()
```

```
      7   dealer_zip            200000 non-null  object
      8   engine_cylinders      194920 non-null  object
      9   engine_displacement   191357 non-null  float64
      10  engine_type           194920 non-null  object
      11  exterior_color        199996 non-null  object
      12  fleet                 108332 non-null  object
      13  frame_damaged         108332 non-null  object
      14  franchise_dealer      200000 non-null  bool
      15  franchise_make        160488 non-null  object
      16  front_legroom         200000 non-null  float64
      17  fuel_tank_volume      192536 non-null  float64
      18  fuel_type             196297 non-null  object
      19  has_accidents         108332 non-null  object
      20  height                192540 non-null  object
      21  highway_fuel_economy  176380 non-null  float64
      22  horsepower            191357 non-null  float64
      23  interior_color        199990 non-null  object
      24  isCab                 108332 non-null  object
      25  is_new                200000 non-null  bool
      26  latitude              200000 non-null  float64
      27  length                192540 non-null  object
      28  listed_date           200000 non-null  object
      29  listing_color         200000 non-null  object
      30  listing_id            200000 non-null  int64
      31  longitude             200000 non-null  float64
      32  major_options         188278 non-null  object
      33  make_name             200000 non-null  object
      34  maximum_seating       192540 non-null  object
      35  mileage               191049 non-null  float64
      36  model_name            200000 non-null  object
      37  owner_count           103263 non-null  float64
      38  power                 170366 non-null  object
      39  price                 200000 non-null  float64
      40  salvage               108332 non-null  object
      41  savings_amount        200000 non-null  int64
      42  seller_rating         197294 non-null  float64
      43  sp_id                 199993 non-null  float64
      44  sp_name               200000 non-null  object
      45  theft_title           108332 non-null  object
      46  torque                167437 non-null  object
      47  transmission          195802 non-null  object
      48  transmission_display  195802 non-null  object
      49  trimId                195391 non-null  object
      50  trim_name             195368 non-null  object
      51  wheel_system          193140 non-null  object
      52  wheel_system_display  193140 non-null  object
      53  wheelbase             192540 non-null  object
      54  width                 192540 non-null  object
      55  year                  200000 non-null  int64
      56  Sedan                 200000 non-null  int64
      57  SUV / Crossover       200000 non-null  int64
      58  Coupe                 200000 non-null  int64
      59  Minivan               200000 non-null  int64
      60  Hatchback             200000 non-null  int64
      61  Wagon                 200000 non-null  int64
      62  Convertible           200000 non-null  int64
     dtypes: bool(2), float64(14), int64(12), object(35)
     memory usage: 93.5+ MB
```

Statistics of the fuel tank volume feature after converting it to float

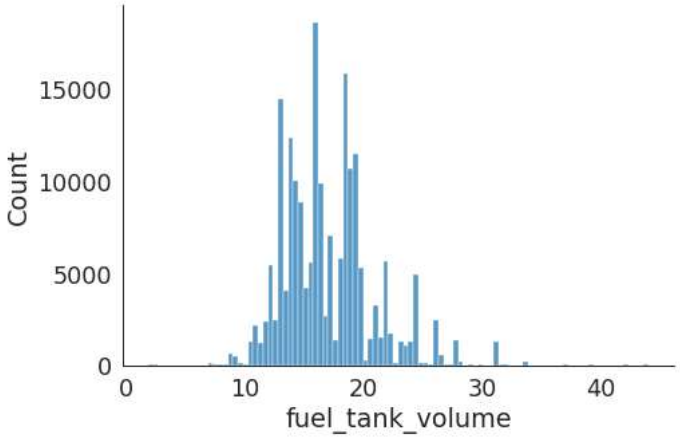
```
print("Number of missing values = ",df["fuel_tank_volume"].isna().sum())
```

```
Number of missing values =  7464
```

Statistics before induction of missing values

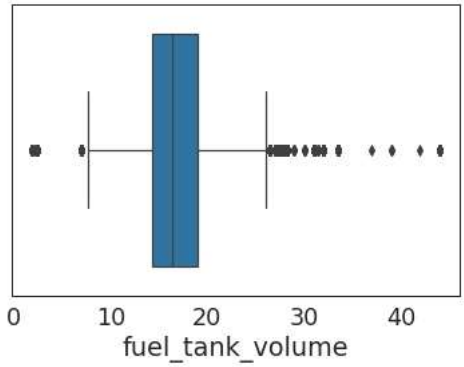
```
import numpy as np
import pandas as pd
import seaborn as sns
from fitter import Fitter, get_common_distributions, get_distributions
from scipy import stats
sns.set_style('white')
sns.set_context("paper", font_scale = 2)
sns.displot(data=df, x="fuel_tank_volume", kind="hist", bins = 100, aspect = 1.5)
outlirs = df.describe()["fuel_tank_volume"]
print("The mean = ",outlirs["mean"])
print("The standard deviation = ", outlirs["std"])
print("The minimum value =", outlirs["min"])
print("The maximum value =", outlirs["max"])
print("The median = ", outlirs["50%"])
print("The first quartile = ", outlirs["25%"])
print("The third quartile = ", outlirs["75%"])
print("Distribution of the fuel_tank_volume feature using histogram")
```

```
The mean =  17.054020546806832
The standard deviation =  3.7535953714758654
The minimum value = 1.9
The maximum value = 44.0
The median =  16.4
The first quartile =  14.3
The third quartile =  19.0
Distribution of the fuel_tank_volume feature using histogram
```




```
array = df["fuel_tank_volume"].to_numpy()
#print(array)
print("Distribution of the front legroom feature using Box plot")
sns.boxplot(df['fuel_tank_volume'])
```

```
Distribution of the front legroom feature using Box plot
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From v
  warnings.warn(
<AxesSubplot:xlabel='fuel_tank_volume'>
```



Induction of Missing values:

We will assign the null values to the average of the data set

```
import math
average = df["fuel_tank_volume"].mean()
for i in range(0, len(df)):

  if(math.isnan(df.at[i, "fuel_tank_volume"])):

    df.loc[i,"fuel_tank_volume"] = average
```

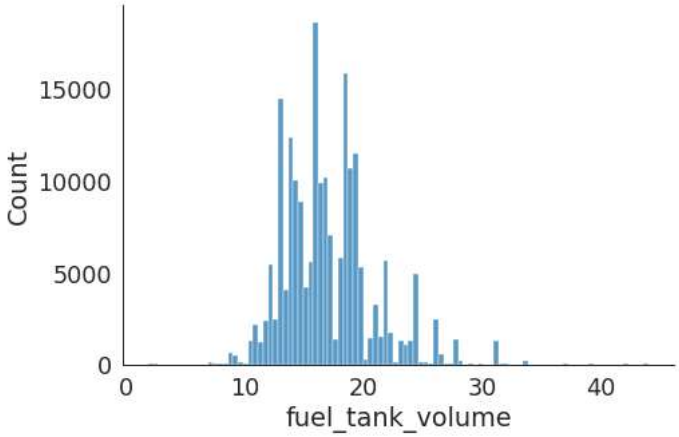
Statistcs after induction of missing values:

```
print("Number of missing values = ", df["fuel_tank_volume"].isna().sum())
```

```
Number of missing values =  0
```

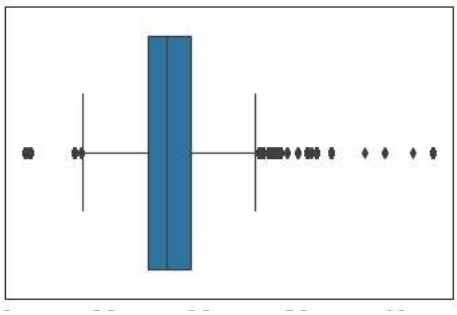
```
import numpy as np
import pandas as pd
import seaborn as sns
from fitter import Fitter, get_common_distributions, get_distributions
from scipy import stats
sns.set_style('white')
sns.set_context("paper", font_scale = 2)
sns.displot(data=df, x="fuel_tank_volume", kind="hist", bins = 100, aspect = 1.5)
outlirs = df.describe()["fuel_tank_volume"]
print("The mean = ",outlirs["mean"])
print("The standard deviation = ", outlirs["std"])
print("The minimum value =", outlirs["min"])
print("The maximum value =", outlirs["max"])
print("The median = ", outlirs["50%"])
print("The first quartile = ", outlirs["25%"])
print("The third quartile = ", outlirs["75%"])
print("Distribution of the ffuel_tank_volume feature using histogram")
```

```
The mean =  17.05402054680683
The standard deviation =  3.682886945964384
The minimum value = 1.9
The maximum value = 44.0
The median =  16.5
The first quartile =  14.5
The third quartile =  19.0
Distribution of the ffuel_tank_volume feature using histogram
```




```
array = df["fuel_tank_volume"].to_numpy()
#print(array)
print("Distribution of the front legroom feature using Box plot")
sns.boxplot(df['fuel_tank_volume'])
```

```
Distribution of the front legroom feature using Box plot
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From \
  warnings.warn(
<AxesSubplot:xlabel='fuel_tank_volume'>
```

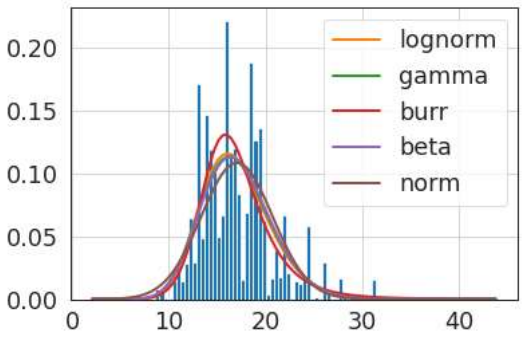


From the above figure, we can see that the data contains many outliers

```
BL = df["fuel_tank_volume"].values

BL = list(BL)
f = Fitter(BL,
           distributions=['gamma',
                          'lognorm',
                          "beta",
                          "burr",
                          "norm"], timeout = 240 )
f.fit()
f.summary()
```

```
Fitting 5 distributions: 100%|██████████| 5/5 [00:18<00:00,  3.66s/it]
```

	sumsquare_error	aic	bic	kl_div	ks_statistic	ks_pvalue
lognorm	0.068071	1597.925036	-2.978619e+06	inf	0.060335	0.0
gamma	0.068951	1628.750407	-2.976048e+06	inf	0.065489	0.0
burr	0.069404	1275.640881	-2.974729e+06	inf	0.070081	0.0
beta	0.069504	1649.549882	-2.974440e+06	inf	0.068144	0.0
norm	0.074418	1793.662049	-2.960802e+06	inf	0.083447	0.0



From this figure, we can see that any type of distributions fits this data

```
import scipy.stats
r = scipy.stats.pearsonr(df["fuel_tank_volume"], df["price"])
print("Correlation Coeffecient = ", r[0])
```

```
Correlation Coeffecient =  0.36505753459402823
```

The final step is to scale the data of this feature using standard z scaling

```
des = df.describe()["fuel_tank_volume"]
m = des["mean"]
std = des["std"]
for i in range(0, len(df)):
  df.loc[i, "fuel_tank_volume"] = (df.loc[i, "fuel_tank_volume"] - m)/std
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.8/dist-packages/pandas/core/indexes/base.py in get_loc(self, key, method, tolerance)
   3360             try:
-> 3361                 return self._engine.get_loc(casted_key)
   3362             except KeyError as err:

                                 4 frames
pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'front_lfuel_tank_volumeegroom'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.8/dist-packages/pandas/core/indexes/base.py in get_loc(self, key, method, tolerance)
   3361                 return self._engine.get_loc(casted_key)
   3362             except KeyError as err:
-> 3363                 raise KeyError(key) from err
   3364
   3365         if is_scalar(key) and isna(key) and not self.hasnans:

KeyError: 'front_lfuel_tank_volumeegroom'
```

SEARCH STACK OVERFLOW

```
print(df["fuel_tank_volume"])
```

```
0        -0.612025
1         0.636995
2         0.256858
3        -0.394805
4         1.098589
            ...
199995   -1.399451
199996   -0.693483
199997   -0.829246
199998    0.745605
199999   -1.752435
Name: fuel_tank_volume, Length: 200000, dtype: float64
```

Height Feature:

As we can see, the height feature describes the height of the car. So, it is mainly a distance so, it should be a numeric value. However, it is a string object in the data because the unit is included. As a result, the first step in cleaning the data in this feature will be removing the units and converting the data from strings to floats

```
df["height"].unique()
```

```
array(['57.1 in', '70.7 in', '65.1 in', '65.4 in', '69.5 in', '58.1 in',
       '54.2 in', '68.5 in', '67.8 in', '58.7 in', '71.5 in', '67.4 in',
       '56.9 in', '56.3 in', '65.2 in', nan, '56.5 in', '57.3 in',
       '69.3 in', '66.2 in', '57.8 in', '66.5 in', '77.7 in', '66.1 in',
       '58.9 in', '58.5 in', '69.9 in', '64.4 in', '66.3 in', '62.7 in',
       '66.7 in', '67.7 in', '77.2 in', '67 in', '65.7 in', '58 in',
       '71 in', '73.6 in', '75.9 in', '64.1 in', '54.6 in', '64.6 in',
       '64.8 in', '74.4 in', '57.4 in', '58.2 in', '57.9 in', '70.2 in',
       '66.4 in', '53.1 in', '68.6 in', '65.5 in', '58.4 in', '71.9 in',
       '69.6 in', '55.7 in', '70.6 in', '76.5 in', '59.2 in', '66.6 in',
       '69 in', '68.1 in', '68.7 in', '67.9 in', '67.2 in', '55.6 in',
       '72.7 in', '50.7 in', '63.6 in', '62.5 in', '55.4 in', '55.1 in',
       '67.3 in', '67.1 in', '64.9 in', '75.8 in', '57.5 in', '56.4 in',
       '61.6 in', '54 in', '76.8 in', '56.8 in', '56 in', '57.7 in',
       '66.9 in', '76.4 in', '69.4 in', '70.4 in', '57.2 in', '57 in',
       '56.7 in', '70 in', '66 in', '65.3 in', '70.1 in', '66.8 in',
       '64.2 in', '56.6 in', '59.8 in', '60 in', '75.7 in', '63.4 in',
       '71.7 in', '59.1 in', '68 in', '55.9 in', '63 in', '68.9 in',
       '68.4 in', '59.4 in', '60.7 in', '70.9 in', '68.3 in', '51.2 in',
       '63.5 in', '65 in', '71.3 in', '64.7 in', '71.6 in', '59.7 in',
       '65.6 in', '52.9 in', '50.9 in', '60.6 in', '69.8 in', '61 in',
       '54.9 in', '59 in', '62.4 in', '57.6 in', '58.8 in', '63.2 in',
       '63.9 in', '63.1 in', '64 in', '72.2 in', '67.5 in', '59.6 in',
       '49.9 in', '54.3 in', '51.5 in', '54.4 in', '60.1 in', '76.9 in',
       '48.8 in', '68.2 in', '58.3 in', '50.2 in', '70.8 in', '68.8 in',
       '76.6 in', '60.5 in', '54.5 in', '77.4 in', '75.5 in', '54.8 in',
       '74.2 in', '60.4 in', '65.9 in', '73.5 in', '76.3 in', '55.8 in',
```

```
            '53.7 in', '55.3 in', '72.4 in', '50.1 in', '69.1 in', '47.8 in',
            '73.7 in', '76.2 in', '56.2 in', '72.6 in', '59.3 in', '49 in',
            '55 in', '64.5 in', '62.2 in', '61.8 in', '78.7 in', '69.7 in',
            '73 in', '47.2 in', '59.5 in', '47.7 in', '77 in', '75.6 in',
            '55.5 in', '70.3 in', '65.8 in', '61.3 in', '72.5 in', '60.8 in',
            '72 in', '51.6 in', '43.9 in', '62.9 in', '54.7 in', '51.3 in',
            '50.8 in', '62.8 in', '52 in', '51.8 in', '72.8 in', '51 in',
            '53.5 in', '56.1 in', '71.8 in', '73.4 in', '61.5 in', '74 in',
            '51.1 in', '45.9 in', '78 in', '48.7 in', '76.1 in', '52.6 in',
            '--', '45.8 in', '75.2 in', '53.3 in', '79.2 in', '63.3 in',
            '78.1 in', '74.3 in', '47.5 in', '54.1 in', '62.6 in', '73.9 in',
            '55.2 in', '74.1 in', '49.1 in', '53.2 in', '74.6 in', '61.7 in',
            '67.6 in', '62 in', '51.4 in', '58.6 in', '53.9 in', '48.5 in',
            '46.4 in', '49.4 in', '48.9 in', '60.9 in', '78.3 in', '63.7 in',
            '50.6 in', '53.8 in', '44.7 in', '71.1 in', '61.9 in', '53 in',
            '47.3 in', '73.8 in', '79.7 in', '48.6 in', '51.9 in', '52.1 in',
            '60.2 in', '61.2 in', '80.2 in', '70.5 in', '52.3 in', '77.8 in',
            '74.5 in', '73.3 in', '49.6 in', '53.6 in', '62.1 in', '73.2 in',
            '74.9 in', '71.4 in', '52.8 in', '48.2 in', '50.4 in', '69.2 in',
            '75.4 in', '46.1 in', '46 in', '88 in', '75 in', '77.6 in',
            '49.8 in', '61.4 in', '76.7 in', '46.3 in', '72.3 in', '50 in',
            '48.1 in', '52.7 in', '49.7 in', '63.8 in', '77.1 in', '47.4 in',
            '44 in', '52.2 in', '49.3 in', '52.4 in', '72.1 in', '48.4 in',
            '71.2 in', '74.8 in', '47 in', '47.9 in', '44.3 in', '73.1 in',
            '59.9 in', '47.1 in', '76 in', '60.3 in', '78.4 in', '49.2 in',
            '49.5 in', '48 in', '51.7 in', '46.5 in', '46.6 in', '74.7 in',
            '46.7 in', '46.9 in', '64.3 in', '52.5 in', '53.4 in', '47.6 in',
            '77.5 in', '50.3 in', '50.5 in', '72.9 in'], dtype=object)
```

```
from pandas.core.api import notnull
for i in range(0, len(df)):
  temp = df.at[i,"height"]
  if(temp != '--' and temp is not None ):
    x= type("ojmlk")
    if(type(temp) == x):
      temp1 = temp.replace(" in", '')
      df['height'] = df['height'].replace([temp],float(temp1))

  elif (temp == "--"):
      df['height'] = df['height'].replace([temp],None)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200000 entries, 0 to 199999
Data columns (total 56 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   Unnamed: 0            200000 non-null  int64  
 1   vin                   200000 non-null  object 
 2   back_legroom          192430 non-null  float64
 3   body_type             200000 non-null  object 
 4   city                  200000 non-null  object 
 5   city_fuel_economy     176380 non-null  float64
 6   daysonmarket          200000 non-null  int64  
 7   dealer_zip            200000 non-null  object 
 8   engine_cylinders      194920 non-null  object 
 9   engine_displacement   191357 non-null  float64
 10  engine_type           194920 non-null  object 
 11  exterior_color        199996 non-null  object 
 12  fleet                 108332 non-null  object 
 13  frame_damaged         108332 non-null  object 
 14  franchise_dealer      200000 non-null  bool   
 15  franchise_make        160488 non-null  object 
 16  front_legroom         192479 non-null  float64
 17  fuel_tank_volume      192536 non-null  float64
 18  fuel_type             196297 non-null  object 
 19  has_accidents         108332 non-null  object 
 20  height                192538 non-null  float64
 21  highway_fuel_economy  176380 non-null  float64
 22  horsepower            191357 non-null  float64
 23  interior_color        199990 non-null  object 
 24  isCab                 108332 non-null  object 
 25  is_new                200000 non-null  bool   
 26  latitude              200000 non-null  float64
 27  length                192540 non-null  object 
 28  listed_date           200000 non-null  object 
 29  listing_color         200000 non-null  object 
 30  listing_id            200000 non-null  int64  
 31  longitude             200000 non-null  float64
 32  major_options         188278 non-null  object 
```

```
 33  make_name             200000 non-null  object
 34  maximum_seating       192540 non-null  object
 35  mileage               191049 non-null  float64
 36  model_name            200000 non-null  object
 37  owner_count           103263 non-null  float64
 38  power                 170366 non-null  object
 39  price                 200000 non-null  float64
 40  salvage               108332 non-null  object
 41  savings_amount        200000 non-null  int64
 42  seller_rating         197294 non-null  float64
 43  sp_id                 199993 non-null  float64
 44  sp_name               200000 non-null  object
 45  theft_title           108332 non-null  object
 46  torque                167437 non-null  object
 47  transmission          195802 non-null  object
 48  transmission_display  195802 non-null  object
 49  trimId                195391 non-null  object
 50  trim_name             195368 non-null  object
 51  wheel_system          193140 non-null  object
```

Statistics of the height feature after converting it to float

```
print("Number of missing values = ", df["fuel_tank_volume"].isna().sum())
```

Length Feature:

As we can see, the length feature describes the length of the car in inches. So, it is mainly a distance so, it should be a numeric value. However, it is a string object in the data because the unit is included. As a result, the first step in cleaning the data in this feature will be removing the units and converting the data from strings to floats

```
df["length"].unique()
```

```
array(['192.2 in', '204.3 in', '182.5 in', '182.3 in', '192.4 in',
       '191.1 in', '182.6 in', '201.4 in', '174.7 in', '200.4 in',
       '174.4 in', '196.2 in', '192.9 in', '179.1 in', '176.4 in', nan,
       '182.7 in', '183.1 in', '189.8 in', '182 in', '193.8 in',
       '166.6 in', '220.8 in', '201.3 in', '198.4 in', '195 in',
       '173.4 in', '182.1 in', '159 in', '193.4 in', '192.5 in',
       '206.5 in', '180.9 in', '191.8 in', '178.3 in', '195.5 in',
       '198.3 in', '188.4 in', '210.7 in', '180.6 in', '171.4 in',
       '187.3 in', '173 in', '224.4 in', '185.1 in', '200.1 in',
       '190.7 in', '191.6 in', '190.9 in', '191.7 in', '194.4 in',
       '181.5 in', '192.1 in', '188.3 in', '183.6 in', '178.1 in',
       '184.5 in', '198.8 in', '183.8 in', '181.9 in', '180.5 in',
       '178.7 in', '201.2 in', '205.2 in', '187.6 in', '196.5 in',
       '210 in', '169.5 in', '192.8 in', '203.7 in', '188.1 in',
       '180.7 in', '200.2 in', '203.2 in', '203.6 in', '179.4 in',
       '191.4 in', '177 in', '168.4 in', '179 in', '176.5 in', '169.1 in',
       '145.6 in', '193.9 in', '178.2 in', '196.9 in', '187.8 in',
       '181.1 in', '191.3 in', '167.6 in', '171.7 in', '189.7 in',
       '197.9 in', '202.8 in', '176 in', '191 in', '171.2 in', '200.6 in',
       '179.2 in', '176.7 in', '222.4 in', '173.8 in', '185 in', '175 in',
       '179.9 in', '187.4 in', '179.5 in', '159.7 in', '183.9 in',
       '195.3 in', '194.5 in', '172.6 in', '183.3 in', '169.3 in',
       '195.9 in', '175.2 in', '183.5 in', '164 in', '195.2 in',
       '175.4 in', '199.3 in', '185.7 in', '192.7 in', '198.5 in',
       '210.2 in', '193.6 in', '184.8 in', '183.7 in', '161.3 in',
       '190.2 in', '204 in', '179.3 in', '184.4 in', '172 in', '200.7 in',
       '197.1 in', '194.9 in', '139.6 in', '181 in', '189.1 in',
       '193.3 in', '173.9 in', '174.9 in', '225.7 in', '161.6 in',
       '201.9 in', '190.6 in', '185.2 in', '184.2 in', '195.6 in',
       '199.4 in', '165.2 in', '197.7 in', '202.9 in', '221.9 in',
       '200.8 in', '186.8 in', '175.6 in', '194.3 in', '154.7 in',
       '200 in', '198.6 in', '203.9 in', '198.1 in', '175.3 in',
       '158.7 in', '189.6 in', '189 in', '194.2 in', '188.8 in',
       '177.4 in', '172.8 in', '197.2 in', '163 in', '186.6 in',
       '197.5 in', '191.5 in', '182.2 in', '191.2 in', '180 in', '205 in',
       '196.8 in', '184.3 in', '180.2 in', '173.7 in', '189.9 in',
       '185.5 in', '164.6 in', '185.9 in', '171.6 in', '172.5 in',
       '194 in', '202.5 in', '176.9 in', '191.9 in', '190 in', '177.9 in',
       '188.5 in', '201 in', '195.7 in', '196.4 in', '206.9 in',
       '199.6 in', '170.4 in', '173.6 in', '160.1 in', '177.3 in',
       '176.2 in', '167.5 in', '190.5 in', '175.5 in', '165.9 in',
       '178.9 in', '173.1 in', '167 in', '193.2 in', '184 in', '172.2 in',
       '186.7 in', '152.8 in', '189.2 in', '202 in', '154.1 in',
       '173.5 in', '194.1 in', '192.6 in', '178.4 in', '170.3 in',
       '189.3 in', '185.6 in', '201.8 in', '208.9 in', '196.1 in',
       '197.6 in', '207.8 in', '194.6 in', '175.1 in', '187.2 in',
       '219.3 in', '162.2 in', '189.5 in', '190.4 in', '180.4 in',
```

```
            '174 in', '182.8 in', '174.8 in', '196.6 in', '184.6 in',
            '167.2 in', '203.8 in', '143.1 in', '161.4 in', '171.9 in',
            '188.2 in', '186.3 in', '188 in', '157.3 in', '170.5 in',
            '163.8 in', '186.2 in', '157.2 in', '187.5 in', '173.2 in',
            '149.4 in', '202.1 in', '206 in', '168.9 in', '168 in', '207.2 in',
            '188.6 in', '184.9 in', '174.3 in', '193.1 in', '172.1 in',
            '180.3 in', '174.6 in', '224 in', '178 in', '192 in', '162.4 in',
            '184.7 in', '179.7 in', '182.9 in', '177.5 in', '175.8 in',
            '179.8 in', '205.1 in', '222.9 in', '169.8 in', '144.4 in',
            '178.5 in', '190.8 in', '204.7 in', '168.2 in', '186.1 in',
            '171.5 in', '150.6 in', '149 in', '157.4 in', '162 in', '183.4 in'
```

```
from pandas.core.api import notnull
for i in range(0, len(df)):
  temp = df.at[i,"length"]
  if(temp != '--' and temp is not None ):
    x= type("ojmlk")
    if(type(temp) == x):
      temp1 = temp.replace(" in", '')
      df['length'] = df['length'].replace([temp],float(temp1))

  elif (temp == "--"):
      df['length'] = df['length'].replace([temp],None)


df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200000 entries, 0 to 199999
Data columns (total 56 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   Unnamed: 0            200000 non-null  int64
 1   vin                   200000 non-null  object
 2   back_legroom          192430 non-null  float64
 3   body_type             200000 non-null  object
 4   city                  200000 non-null  object
 5   city_fuel_economy     176380 non-null  float64
 6   daysonmarket          200000 non-null  int64
 7   dealer_zip            200000 non-null  object
 8   engine_cylinders      194920 non-null  object
 9   engine_displacement   191357 non-null  float64
 10  engine_type           194920 non-null  object
 11  exterior_color        199996 non-null  object
 12  fleet                 108332 non-null  object
 13  frame_damaged         108332 non-null  object
 14  franchise_dealer      200000 non-null  bool
 15  franchise_make        160488 non-null  object
 16  front_legroom         192479 non-null  float64
 17  fuel_tank_volume      192536 non-null  float64
 18  fuel_type             196297 non-null  object
 19  has_accidents         108332 non-null  object
 20  height                192538 non-null  float64
 21  highway_fuel_economy  176380 non-null  float64
 22  horsepower            191357 non-null  float64
 23  interior_color        199990 non-null  object
 24  isCab                 108332 non-null  object
 25  is_new                200000 non-null  bool
 26  latitude              200000 non-null  float64
 27  length                192538 non-null  float64
 28  listed_date           200000 non-null  object
 29  listing_color         200000 non-null  object
 30  listing_id            200000 non-null  int64
 31  longitude             200000 non-null  float64
 32  major_options         188278 non-null  object
 33  make_name             200000 non-null  object
 34  maximum_seating       192540 non-null  object
 35  mileage               191049 non-null  float64
 36  model_name            200000 non-null  object
 37  owner_count           103263 non-null  float64
 38  power                 170366 non-null  object
 39  price                 200000 non-null  float64
 40  salvage               108332 non-null  object
 41  savings_amount        200000 non-null  int64
 42  seller_rating         197294 non-null  float64
 43  sp_id                 199993 non-null  float64
 44  sp_name               200000 non-null  object
 45  theft_title           108332 non-null  object
 46  torque                167437 non-null  object
 47  transmission          195802 non-null  object
 48  transmission_display  195802 non-null  object
 49  trimId                195391 non-null  object
 50  trim_name             195368 non-null  object
```

```
 51  wheel_system          193140 non-null  object
```

Maximum Seating Feature:

As we can see, the maximum seating feature describes the number of seats of the car. So, it is mainly a number so, it should be a numeric value. However, it is a string object in the data because the unit is included. As a result, the first step in cleaning the data in this feature will be removing the units and converting the data from strings to integers.

```
df["maximum_seating"].unique()
```

```
array(['5 seats', '8 seats', '7 seats', '4 seats', '6 seats', nan,
       '2 seats', '9 seats', '--', '3 seats'], dtype=object)
```

```
from pandas.core.api import notnull
for i in range(0, len(df)):
  temp = df.at[i,"maximum_seating"]
  if(temp != '--' and temp is not None ):
    x= type("ojmlk")
    if(type(temp) == x):
      temp1 = temp.replace(" seats", '')
      df['maximum_seating'] = df['maximum_seating'].replace([temp],int(temp1))

  elif (temp == "--"):
      df['maximum_seating'] = df['maximum_seating'].replace([temp],None)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200000 entries, 0 to 199999
Data columns (total 56 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   Unnamed: 0            200000 non-null  int64
 1   vin                   200000 non-null  object
 2   back_legroom          192430 non-null  float64
 3   body_type             200000 non-null  object
 4   city                  200000 non-null  object
 5   city_fuel_economy     176380 non-null  float64
 6   daysonmarket          200000 non-null  int64
 7   dealer_zip            200000 non-null  object
 8   engine_cylinders      194920 non-null  object
 9   engine_displacement   191357 non-null  float64
 10  engine_type           194920 non-null  object
 11  exterior_color        199996 non-null  object
 12  fleet                 108332 non-null  object
 13  frame_damaged         108332 non-null  object
 14  franchise_dealer      200000 non-null  bool
 15  franchise_make        160488 non-null  object
 16  front_legroom         192479 non-null  float64
 17  fuel_tank_volume      192536 non-null  float64
 18  fuel_type             196297 non-null  object
 19  has_accidents         108332 non-null  object
 20  height                192538 non-null  float64
 21  highway_fuel_economy  176380 non-null  float64
 22  horsepower            191357 non-null  float64
 23  interior_color        199990 non-null  object
 24  isCab                 108332 non-null  object
 25  is_new                200000 non-null  bool
 26  latitude              200000 non-null  float64
 27  length                192538 non-null  float64
 28  listed_date           200000 non-null  object
 29  listing_color         200000 non-null  object
 30  listing_id            200000 non-null  int64
 31  longitude             200000 non-null  float64
 32  major_options         188278 non-null  object
 33  make_name             200000 non-null  object
 34  maximum_seating       192538 non-null  float64
 35  mileage               191049 non-null  float64
 36  model_name            200000 non-null  object
 37  owner_count           103263 non-null  float64
 38  power                 170366 non-null  object
 39  price                 200000 non-null  float64
 40  salvage               108332 non-null  object
 41  savings_amount        200000 non-null  int64
 42  seller_rating         197294 non-null  float64
 43  sp_id                 199993 non-null  float64
 44  sp_name               200000 non-null  object
 45  theft_title           108332 non-null  object
 46  torque                167437 non-null  object
```

```
 47  transmission          195802 non-null  object
 48  transmission_display  195802 non-null  object
 49  trimId                195391 non-null  object
 50  trim_name             195368 non-null  object
 51  wheel_system          193140 non-null  object
 52  wheel_system_display  193140 non-null  object
```

Wheel Base Feature:

The wheel base feature describes the distance between the centre of the front wheels and the centre of the rear wheels. So, it is mainly a length so, it should be a numeric value. However, it is a string object in the data because the unit is included. As a result, the first step in cleaning the data in this feature will be removing the units and converting the data from strings to floats.

```
df["wheelbase"].unique()
```

```
array(['111.4 in', '120.9 in', '104.3 in', '107.3 in', '113.8 in',
       '110.4 in', '110.6 in', '120.5 in', '103.1 in', '110.5 in',
       '111 in', '111.8 in', '106.3 in', '105.1 in', nan, '114.8 in',
       '106.6 in', '115.7 in', '101.2 in', '131 in', '114.7 in',
       '106.2 in', '111.7 in', '120.2 in', '117.5 in', '104.8 in',
       '99.2 in', '112.5 in', '109.8 in', '119 in', '105.9 in',
       '112.2 in', '117.3 in', '112.8 in', '118.4 in', '104.7 in',
       '102.2 in', '111.2 in', '103.8 in', '130 in', '105.7 in', '120 in',
       '109.3 in', '117.9 in', '103.9 in', '110.7 in', '106.5 in',
       '114.2 in', '110 in', '119.1 in', '113.1 in', '106.7 in',
       '119.8 in', '118.1 in', '100.4 in', '121.2 in', '107.9 in',
       '119.3 in', '121.6 in', '109.2 in', '100 in', '103.5 in',
       '104.9 in', '97.1 in', '106.9 in', '108.9 in', '109.4 in',
       '108.1 in', '100.6 in', '116.2 in', '103.7 in', '103.2 in',
       '103 in', '98 in', '122.5 in', '111.6 in', '114.5 in', '101.6 in',
       '104.4 in', '113 in', '102.4 in', '112.7 in', '121.1 in',
       '106.8 in', '115.1 in', '108.3 in', '107.8 in', '116 in',
       '112.6 in', '90.6 in', '134.1 in', '107.1 in', '118.9 in',
       '105.6 in', '115.3 in', '131.6 in', '98.8 in', '112.9 in',
       '96.5 in', '104.2 in', '97.6 in', '97.2 in', '113.2 in', '118 in',
       '124.6 in', '102.8 in', '101.4 in', '101.5 in', '96.7 in',
       '99.4 in', '103.4 in', '108.7 in', '114.6 in', '115.5 in',
       '107 in', '109.5 in', '95.4 in', '115.6 in', '90.9 in', '117.1 in',
       '113.3 in', '112.4 in', '107.4 in', '113.4 in', '129 in',
       '112.3 in', '106.1 in', '115 in', '93.9 in', '97 in', '99.6 in',
       '108 in', '107.5 in', '95.1 in', '110.2 in', '114 in', '111.9 in',
       '103.6 in', '124.8 in', '114.4 in', '104.1 in', '98.4 in',
       '104.5 in', '117.4 in', '113.6 in', '123.2 in', '110.1 in',
       '116.9 in', '122 in', '122.9 in', '104 in', '105.3 in', '113.7 in',
       '96.9 in', '118.5 in', '103.3 in', '101.8 in', '90.5 in',
       '108.2 in', '95.7 in', '122.8 in', '124.4 in', '129.7 in',
       '89.4 in', '118.2 in', '123.4 in', '116.1 in', '132.5 in',
       '107.7 in', '106.4 in', '115.8 in', '105.5 in', '98.2 in', '--',
       '126.4 in', '92.5 in', '111.3 in', '121.7 in', '118.3 in',
       '122.4 in', '102.5 in', '99.8 in', '119.2 in', '123 in',
       '101.3 in', '73.5 in', '100.9 in', '112 in', '108.8 in',
       '102.9 in', '108.5 in', '115.2 in', '102.7 in', '96.2 in',
       '91.7 in', '101.1 in', '96.8 in', '114.9 in', '123.1 in',
       '101.7 in', '115.9 in', '137 in', '107.2 in', '97.4 in', '99.9 in',
       '100.5 in', '137.1 in', '116.5 in', '124.3 in', '120.1 in',
       '93.4 in', '116.4 in', '116.7 in', '100.3 in', '122.2 in',
       '98.6 in', '109.7 in', '117.7 in', '117.8 in', '120.7 in',
       '109 in', '100.8 in', '120.4 in', '102 in', '73.7 in', '100.1 in',
       '93.5 in', '116.3 in', '98.1 in', '111.5 in', '93 in', '92.6 in',
       '121.5 in', '94.4 in', '106 in', '105.2 in', '123.6 in', '93.3 in',
       '110.9 in', '95.9 in', '112.1 in', '111.1 in', '95.2 in',
       '109.9 in', '97.8 in', '92.7 in', '110.8 in', '120.8 in',
       '94.5 in', '113.9 in', '96.1 in', '118.8 in', '122.7 in',
       '119.4 in', '89.2 in', '96.6 in', '89.5 in', '121 in', '98.7 in',
       '126.6 in', '105 in', '88.6 in', '119.9 in', '98.9 in', '120.6 in',
       '115.4 in', '117.2 in', '124 in', '99 in', '108.6 in', '98.3 in',
       '92.9 in', '96.3 in', '94.9 in', '101.9 in', '133.5 in',
       '123.7 in', '140.6 in', '125.7 in', '102.6 in', '93.7 in',
       '96.4 in', '89.8 in', '131.5 in', '117 in', '95.6 in', '78.7 in',
       '128.6 in', '102.1 in', '101 in', '102.3 in', '136.4 in',
       '130.7 in'], dtype=object)
```

```
from pandas.core.api import notnull
for i in range(0, len(df)):
  temp = df.at[i,"wheelbase"]
  if(temp != '--' and temp is not None ):
    x= type("ojmlk")
    if(type(temp) == x):
      temp1 = temp.replace(" in", '')
      df['wheelbase'] = df['wheelbase'].replace([temp],float(temp1))
```

```
  elif (temp == "--"):
      df['wheelbase'] = df['wheelbase'].replace([temp],None)

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200000 entries, 0 to 199999
Data columns (total 56 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   Unnamed: 0            200000 non-null  int64  
 1   vin                   200000 non-null  object 
 2   back_legroom          192430 non-null  float64
 3   body_type             200000 non-null  object 
 4   city                  200000 non-null  object 
 5   city_fuel_economy     176380 non-null  float64
 6   daysonmarket          200000 non-null  int64  
 7   dealer_zip            200000 non-null  object 
 8   engine_cylinders      194920 non-null  object 
 9   engine_displacement   191357 non-null  float64
 10  engine_type           194920 non-null  object 
 11  exterior_color        199996 non-null  object 
 12  fleet                 108332 non-null  object 
 13  frame_damaged         108332 non-null  object 
 14  franchise_dealer      200000 non-null  bool   
 15  franchise_make        160488 non-null  object 
 16  front_legroom         192479 non-null  float64
 17  fuel_tank_volume      192536 non-null  float64
 18  fuel_type             196297 non-null  object 
 19  has_accidents         108332 non-null  object 
 20  height                192538 non-null  float64
 21  highway_fuel_economy  176380 non-null  float64
 22  horsepower            191357 non-null  float64
 23  interior_color        199990 non-null  object 
 24  isCab                 108332 non-null  object 
 25  is_new                200000 non-null  bool   
 26  latitude              200000 non-null  float64
 27  length                192538 non-null  float64
 28  listed_date           200000 non-null  object 
 29  listing_color         200000 non-null  object 
 30  listing_id            200000 non-null  int64  
 31  longitude             200000 non-null  float64
 32  major_options         188278 non-null  object 
 33  make_name             200000 non-null  object 
 34  maximum_seating       192538 non-null  float64
 35  mileage               191049 non-null  float64
 36  model_name            200000 non-null  object 
 37  owner_count           103263 non-null  float64
 38  power                 170366 non-null  object 
 39  price                 200000 non-null  float64
 40  salvage               108332 non-null  object 
 41  savings_amount        200000 non-null  int64  
 42  seller_rating         197294 non-null  float64
 43  sp_id                 199993 non-null  float64
 44  sp_name               200000 non-null  object 
 45  theft_title           108332 non-null  object 
 46  torque                167437 non-null  object 
 47  transmission          195802 non-null  object 
 48  transmission_display  195802 non-null  object 
 49  trimId                195391 non-null  object 
 50  trim_name             195368 non-null  object 
 51  wheel_system          193140 non-null  object 
 52  wheel_system_display  193140 non-null  object 
```

Width Feature:

The width feature maiinly describes the width of the car. So, it is mainly a length so, it should be a numeric value. However, it is a string object in the data because the unit is included. As a result, the first step in cleaning the data in this feature will be removing the units and converting the data from strings to floats.

```
df["width"].unique()
```

```
array(['73.3 in', '78.6 in', '73.6 in', '72.4 in', '83.7 in', '73.4 in',
       '79.4 in', '89.3 in', '81.3 in', '72.9 in', '71.6 in', '77.7 in',
       '73.2 in', '69.9 in', '73 in', nan, '70.9 in', '84.8 in',
       '79.6 in', '91.8 in', '83.3 in', '84.3 in', '75 in', '71.1 in',
       '69.7 in', '75.4 in', '74.6 in', '83.5 in', '78.4 in', '68.7 in',
       '90.2 in', '73.8 in', '81 in', '71.4 in', '79.8 in', '80 in',
       '80.5 in', '70.8 in', '74.5 in', '70.7 in', '72.2 in', '71.7 in',
       '84.9 in', '74.7 in', '82.2 in', '81.8 in', '77.2 in', '72 in',
```

```
              '82.5 in', '85.6 in', '70.1 in', '85.5 in', '92.3 in', '71.3 in',
              '65.7 in', '72.8 in', '88.5 in', '75.8 in', '85.8 in', '78.2 in',
              '90.4 in', '70.6 in', '78.5 in', '68.5 in', '71.2 in', '81.7 in',
              '71 in', '69.3 in', '72.6 in', '68 in', '80.9 in', '74.4 in',
              '74.2 in', '82 in', '75.2 in', '83.9 in', '84.1 in', '72.5 in',
              '69.2 in', '79.1 in', '79.5 in', '67.9 in', '80.3 in', '79 in',
              '67.8 in', '93.4 in', '68.1 in', '70 in', '67.5 in', '89.9 in',
              '81.4 in', '77.3 in', '79.9 in', '71.5 in', '87.4 in', '73.1 in',
              '82.3 in', '76 in', '73.5 in', '71.8 in', '81.1 in', '74.9 in',
              '70.3 in', '86.7 in', '87.5 in', '78.9 in', '84.2 in', '77.5 in',
              '75.7 in', '78.3 in', '91.5 in', '74.8 in', '83 in', '66.7 in',
              '85.7 in', '70.4 in', '70.5 in', '74 in', '72.3 in', '73.7 in',
              '78.1 in', '86.1 in', '83.1 in', '80.1 in', '81.5 in', '81.9 in',
              '82.7 in', '87.1 in', '69.8 in', '75.6 in', '77.8 in', '69.1 in',
              '69.4 in', '76.7 in', '68.3 in', '80.8 in', '86 in', '80.4 in',
              '82.6 in', '82.8 in', '87.3 in', '88.8 in', '85.4 in', '82.9 in',
              '79.3 in', '68.8 in', '75.9 in', '78.8 in', '75.5 in', '93.8 in',
              '62.8 in', '67 in', '80.6 in', '85.2 in', '65.6 in', '77.1 in',
              '76.3 in', '85 in', '67.1 in', '72.7 in', '80.2 in', '69 in',
              '86.4 in', '69.5 in', '85.1 in', '78.7 in', '75.1 in', '79.7 in',
              '72.1 in', '70.2 in', '66.9 in', '65 in', '84.4 in', '77.9 in',
              '67.7 in', '76.1 in', '77.4 in', '68.9 in', '78 in', '74.1 in',
              '84 in', '88 in', '--', '86.2 in', '67.3 in', '73.9 in', '81.2 in',
              '67.4 in', '87.2 in', '86.6 in', '76.4 in', '82.1 in', '83.8 in',
              '61.4 in', '63 in', '83.4 in', '87.6 in', '84.5 in', '71.9 in',
              '69.6 in', '66.3 in', '83.2 in', '66.4 in', '77.6 in', '86.5 in',
              '76.2 in', '57 in', '76.9 in', '64.1 in', '66.5 in', '75.3 in',
              '79.2 in', '66.1 in', '87.7 in', '76.5 in', '66.2 in', '86.9 in',
              '65.5 in', '86.3 in', '76.8 in', '62.9 in', '76.6 in', '67.2 in',
              '82.4 in', '66 in', '74.3 in', '66.8 in', '77 in', '65.4 in',
              '81.6 in', '64 in', '68.2 in', '64.2 in', '68.6 in', '89.5 in',
              '66.6 in', '85.9 in', '62 in', '65.9 in', '89.1 in', '63.5 in',
              '89.2 in', '67.6 in'], dtype=object)
```

```
from pandas.core.api import notnull
for i in range(0, len(df)):
  temp = df.at[i,"width"]
  if(temp != '--' and temp is not None ):
    x= type("ojmlk")
    if(type(temp) == x):
      temp1 = temp.replace(" in", '')
      df['width'] = df['width'].replace([temp],float(temp1))

  elif (temp == "--"):
      df['width'] = df['width'].replace([temp],None)
```

```
df.info()
```

```
    <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 200000 entries, 0 to 199999
    Data columns (total 56 columns):
     #   Column                Non-Null Count   Dtype
    ---  ------                --------------   -----
     0   Unnamed: 0            200000 non-null  int64
     1   vin                   200000 non-null  object
     2   back_legroom          192430 non-null  float64
     3   body_type             200000 non-null  object
     4   city                  200000 non-null  object
     5   city_fuel_economy     176380 non-null  float64
     6   daysonmarket          200000 non-null  int64
     7   dealer_zip            200000 non-null  object
     8   engine_cylinders      194920 non-null  object
     9   engine_displacement   191357 non-null  float64
     10  engine_type           194920 non-null  object
     11  exterior_color        199996 non-null  object
     12  fleet                 108332 non-null  object
     13  frame_damaged         108332 non-null  object
     14  franchise_dealer      200000 non-null  bool
     15  franchise_make        160488 non-null  object
     16  front_legroom         192479 non-null  float64
     17  fuel_tank_volume      192536 non-null  float64
     18  fuel_type             196297 non-null  object
     19  has_accidents         108332 non-null  object
     20  height                192538 non-null  float64
     21  highway_fuel_economy  176380 non-null  float64
     22  horsepower            191357 non-null  float64
     23  interior_color        199990 non-null  object
     24  isCab                 108332 non-null  object
     25  is_new                200000 non-null  bool
     26  latitude              200000 non-null  float64
     27  length                192538 non-null  float64
```

```
28  listed_date           200000 non-null  object
29  listing_color         200000 non-null  object
30  listing_id            200000 non-null  int64
31  longitude             200000 non-null  float64
32  major_options         188278 non-null  object
33  make_name             200000 non-null  object
34  maximum_seating       192538 non-null  float64
35  mileage               191049 non-null  float64
36  model_name            200000 non-null  object
37  owner_count           103263 non-null  float64
38  power                 170366 non-null  object
39  price                 200000 non-null  float64
40  salvage               108332 non-null  object
41  savings_amount        200000 non-null  int64
42  seller_rating         197294 non-null  float64
43  sp_id                 199993 non-null  float64
44  sp_name               200000 non-null  object
45  theft_title           108332 non-null  object
46  torque                167437 non-null  object
47  transmission          195802 non-null  object
48  transmission_display  195802 non-null  object
49  trimId                195391 non-null  object
50  trim_name             195368 non-null  object
51  wheel_system          193140 non-null  object
```

✓ 0s completed at 7:34 PM