```
pip install matplotlib
```

```
Looking in indexes: https://pypi.org/simple, https://us-
python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: matplotlib in
/usr/local/lib/python3.10/dist-packages (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.0.7)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (4.39.3)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.4)
Requirement already satisfied: numpy>=1.20 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.22.4)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (23.1)
Requirement already satisfied: pillow>=6.2.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (8.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7-
>matplotlib) (1.16.0)
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
data=pd.read_csv('Automobile_data.csv')
```

```
type(data)
```

```
pandas.core.frame.DataFrame
```

```
data.dtypes
```

```
symboling              int64
normalized-losses     object
make                  object
fuel-type             object
aspiration            object
num-of-doors          object
body-style            object
drive-wheels          object
engine-location       object
wheel-base           float64
length               float64
```

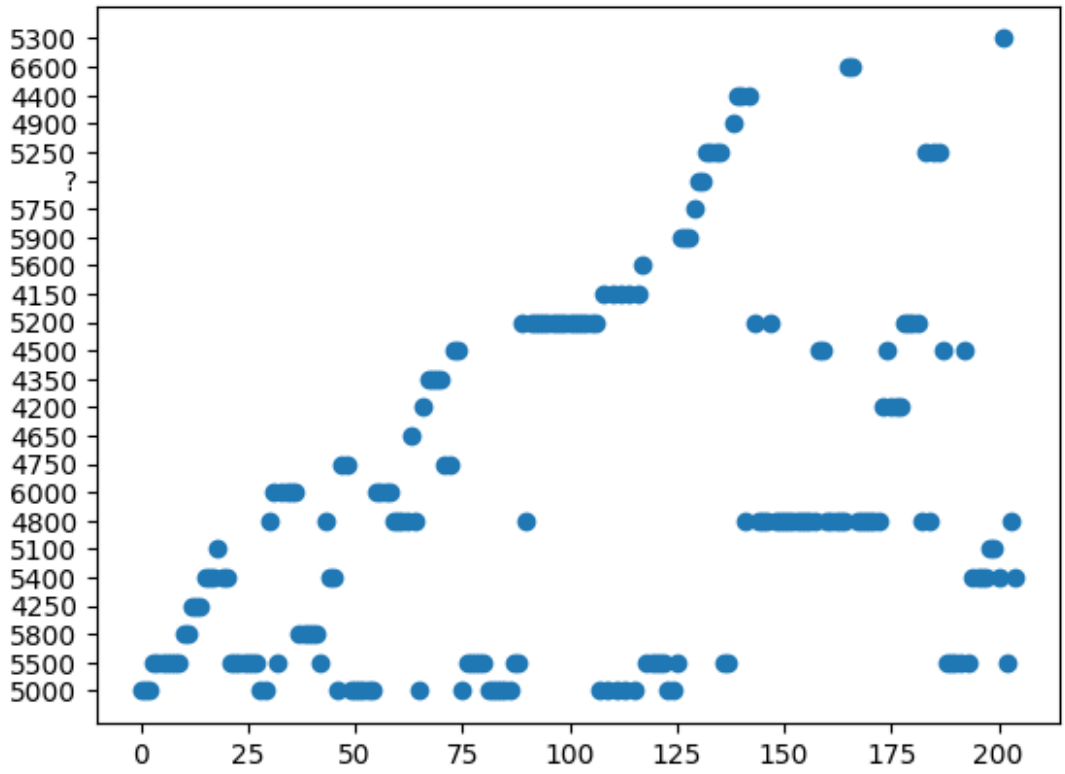
```
width                float64
height               float64
curb-weight            int64
engine-type           object
num-of-cylinders      object
engine-size            int64
fuel-system           object
bore                  object
stroke                object
compression-ratio    float64
horsepower            object
peak-rpm              object
city-mpg               int64
highway-mpg            int64
price                 object
dtype: object
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 26 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   symboling          205 non-null    int64  
 1   normalized-losses  205 non-null    object 
 2   make               205 non-null    object 
 3   fuel-type          205 non-null    object 
 4   aspiration         205 non-null    object 
 5   num-of-doors       205 non-null    object 
 6   body-style         205 non-null    object 
 7   drive-wheels       205 non-null    object 
 8   engine-location    205 non-null    object 
 9   wheel-base         205 non-null    float64
 10  length             205 non-null    float64
 11  width              205 non-null    float64
 12  height             205 non-null    float64
 13  curb-weight        205 non-null    int64  
 14  engine-type        205 non-null    object 
 15  num-of-cylinders   205 non-null    object 
 16  engine-size        205 non-null    int64  
 17  fuel-system        205 non-null    object 
 18  bore               205 non-null    object 
 19  stroke             205 non-null    object 
 20  compression-ratio  205 non-null    float64
 21  horsepower         205 non-null    object 
 22  peak-rpm           205 non-null    object 
 23  city-mpg           205 non-null    int64  
 24  highway-mpg        205 non-null    int64  
 25  price              205 non-null    object 
```

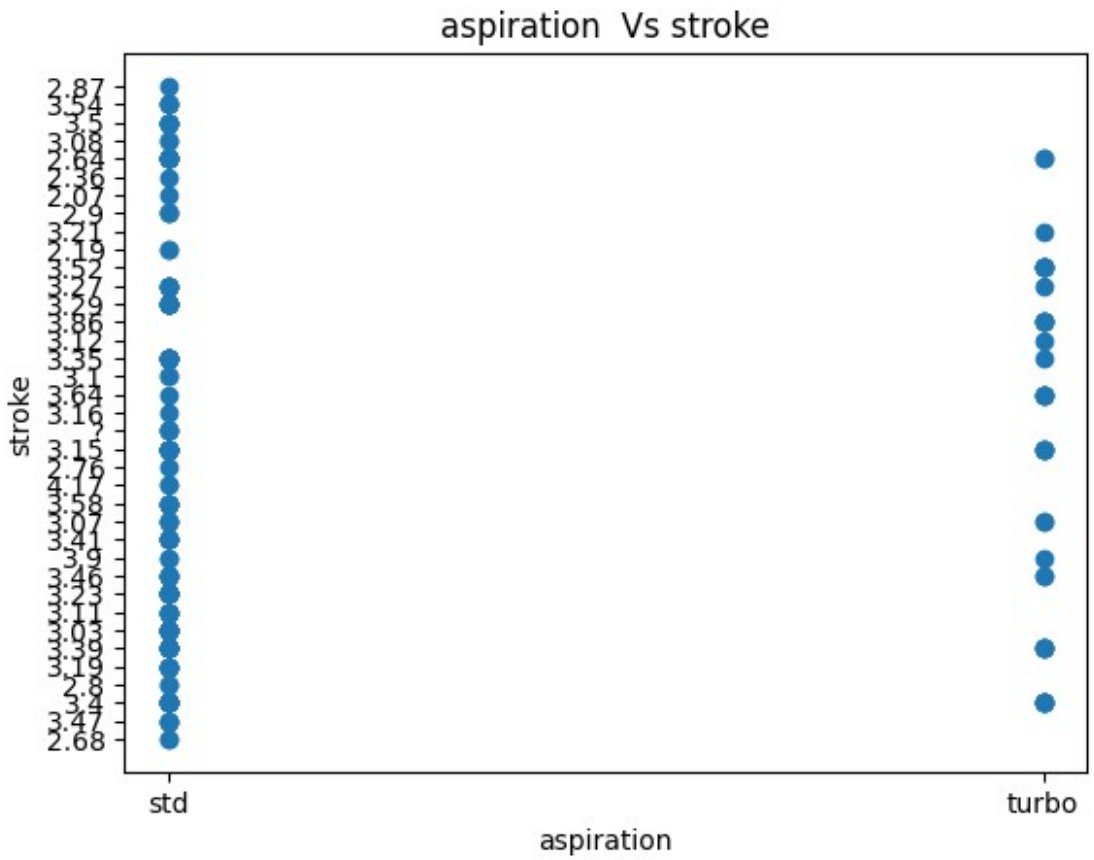
```
dtypes: float64(5), int64(5), object(16)
memory usage: 41.8+ KB
```

```
plt.scatter(data.index,data['peak-rpm'])
plt.show()
```



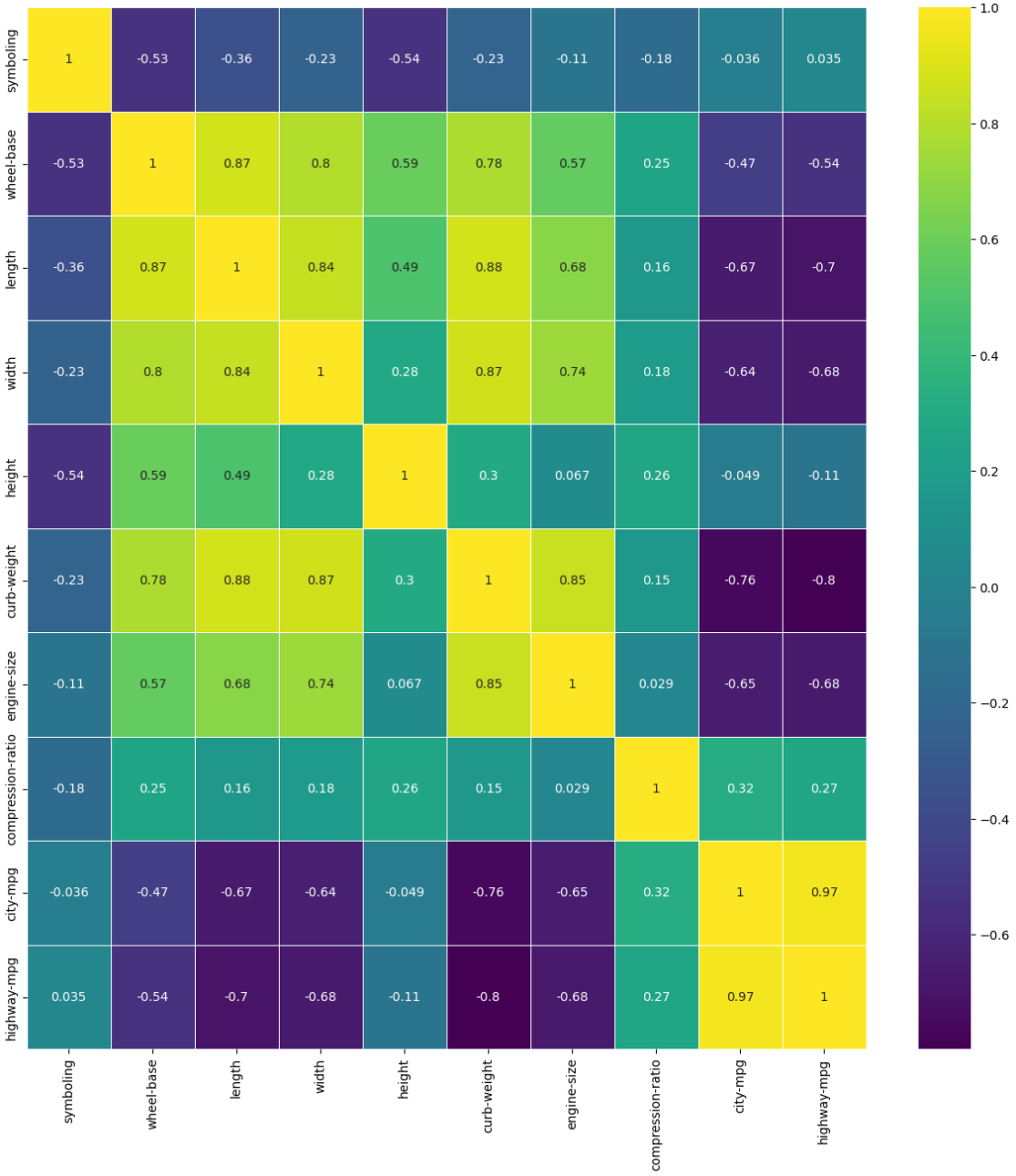

```
plt.scatter(data.aspiration, data.stroke)
plt.title('aspiration  Vs stroke')
plt.xlabel('aspiration')
plt.ylabel('stroke')
```

```
Text(0, 0.5, 'stroke')
```




```
plt.figure(figsize=(15,15))#multivariate analysis
sns.heatmap(data.corr(),linewidths=0.5,annot=True,cmap='viridis')
plt.show()
```

```
<ipython-input-12-6f01439add53>:2: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it
will default to False. Select only valid columns or specify the value
of numeric_only to silence this warning.
  sns.heatmap(data.corr(),linewidths=0.5,annot=True,cmap='viridis')
```




```
data.isnull()#handling missing values
```

```
     symboling  normalized-losses   make  fuel-type  aspiration  num-
of-doors  \
0        False              False  False      False       False
False
1        False              False  False      False       False
False
2        False              False  False      False       False
False
3        False              False  False      False       False
False
```

```
4        False              False  False      False       False
False
..         ...                ...    ...        ...         ...
...
200      False              False  False      False       False
False
201      False              False  False      False       False
False
202      False              False  False      False       False
False
203      False              False  False      False       False
False
204      False              False  False      False       False
False

     body-style  drive-wheels  engine-location  wheel-base  ...
engine-size  \
0         False         False            False       False  ...
False
1         False         False            False       False  ...
False
2         False         False            False       False  ...
False
3         False         False            False       False  ...
False
4         False         False            False       False  ...
False
..          ...           ...              ...         ...  ...
...
200       False         False            False       False  ...
False
201       False         False            False       False  ...
False
202       False         False            False       False  ...
False
203       False         False            False       False  ...
False
204       False         False            False       False  ...
False

     fuel-system   bore  stroke  compression-ratio  horsepower  peak-
rpm  \
0          False  False   False              False       False
False
1          False  False   False              False       False
False
2          False  False   False              False       False
False
3          False  False   False              False       False
False
```

```
4          False  False   False              False       False
False
..           ...    ...     ...                ...         ...       .
..
200        False  False   False              False       False
False
201        False  False   False              False       False
False
202        False  False   False              False       False
False
203        False  False   False              False       False
False
204        False  False   False              False       False
False

     city-mpg  highway-mpg  price
0       False        False  False
1       False        False  False
2       False        False  False
3       False        False  False
4       False        False  False
..        ...          ...    ...
200     False        False  False
201     False        False  False
202     False        False  False
203     False        False  False
204     False        False  False

[205 rows x 26 columns]
```

```
data.isnull().sum()
```

```
symboling            0
normalized-losses    0
make                 0
fuel-type            0
aspiration           0
num-of-doors         0
body-style           0
drive-wheels         0
engine-location      0
wheel-base           0
length               0
width                0
height               0
curb-weight          0
engine-type          0
num-of-cylinders     0
engine-size          0
fuel-system          0
```

```
bore                0
stroke              0
compression-ratio   0
horsepower          0
peak-rpm            0
city-mpg            0
highway-mpg         0
price               0
dtype: int64
```

```
data.head(200)
```

```
     symboling normalized-losses         make fuel-type aspiration  \
0            3                 ?  alfa-romero       gas        std
1            3                 ?  alfa-romero       gas        std
2            1                 ?  alfa-romero       gas        std
3            2               164         audi       gas        std
4            2               164         audi       gas        std
..         ...               ...          ...       ...        ...
195         -1                74        volvo       gas        std
196         -2               103        volvo       gas        std
197         -1                74        volvo       gas        std
198         -2               103        volvo       gas      turbo
199         -1                74        volvo       gas      turbo

    num-of-doors   body-style drive-wheels engine-location  wheel-base
...  \
0            two  convertible          rwd           front        88.6
...
1            two  convertible          rwd           front        88.6
...
2            two    hatchback          rwd           front        94.5
...
3           four        sedan          fwd           front        99.8
...
4           four        sedan          4wd           front        99.4
...
..           ...          ...          ...             ...         ...
...
195         four        wagon          rwd           front       104.3
...
196         four        sedan          rwd           front       104.3
...
197         four        wagon          rwd           front       104.3
...
198         four        sedan          rwd           front       104.3
...
199         four        wagon          rwd           front       104.3
...
```

```
     engine-size  fuel-system  bore  stroke compression-ratio
horsepower  \
0            130         mpfi  3.47    2.68               9.0
111
1            130         mpfi  3.47    2.68               9.0
111
2            152         mpfi  2.68    3.47               9.0
154
3            109         mpfi  3.19     3.4              10.0
102
4            136         mpfi  3.19     3.4               8.0
115
..           ...          ...   ...     ...               ...        .
..
195          141         mpfi  3.78    3.15               9.5
114
196          141         mpfi  3.78    3.15               9.5
114
197          141         mpfi  3.78    3.15               9.5
114
198          130         mpfi  3.62    3.15               7.5
162
199          130         mpfi  3.62    3.15               7.5
162

     peak-rpm city-mpg highway-mpg  price
0        5000       21          27  13495
1        5000       21          27  16500
2        5000       19          26  16500
3        5500       24          30  13950
4        5500       18          22  17450
..        ...      ...         ...    ...
195      5400       23          28  13415
196      5400       24          28  15985
197      5400       24          28  16515
198      5100       17          22  18420
199      5100       17          22  18950

[200 rows x 26 columns]
```

```
fuel_mapping = {'gas': 1,
                'diesel': 2}

data['fuel-type'] = data['fuel-type'].map(fuel_mapping)

data
```

```
     symboling normalized-losses         make  fuel-type aspiration  \
0            3                 ?  alfa-romero          1        std
1            3                 ?  alfa-romero          1        std
```

```
2            1                ?  alfa-romero          1        std
3            2              164         audi          1        std
4            2              164         audi          1        std
..         ...              ...          ...        ...        ...
200         -1               95        volvo          1        std
201         -1               95        volvo          1      turbo
202         -1               95        volvo          1        std
203         -1               95        volvo          2      turbo
204         -1               95        volvo          1      turbo

    num-of-doors   body-style drive-wheels engine-location  wheel-base
...  \
0            two  convertible          rwd           front        88.6
...
1            two  convertible          rwd           front        88.6
...
2            two    hatchback          rwd           front        94.5
...
3           four        sedan          fwd           front        99.8
...
4           four        sedan          4wd           front        99.4
...
..           ...          ...          ...             ...         ...
...
200         four        sedan          rwd           front       109.1
...
201         four        sedan          rwd           front       109.1
...
202         four        sedan          rwd           front       109.1
...
203         four        sedan          rwd           front       109.1
...
204         four        sedan          rwd           front       109.1
...

     engine-size  fuel-system  bore  stroke compression-ratio
horsepower  \
0            130         mpfi  3.47    2.68               9.0
111
1            130         mpfi  3.47    2.68               9.0
111
2            152         mpfi  2.68    3.47               9.0
154
3            109         mpfi  3.19     3.4              10.0
102
4            136         mpfi  3.19     3.4               8.0
115
..           ...          ...   ...     ...               ...        .
..
200          141         mpfi  3.78    3.15               9.5
```

```
114
201          141         mpfi  3.78    3.15              8.7
160
202          173         mpfi  3.58    2.87              8.8
134
203          145          idi  3.01     3.4             23.0
106
204          141         mpfi  3.78    3.15              9.5
114

     peak-rpm city-mpg highway-mpg  price
0        5000       21          27  13495
1        5000       21          27  16500
2        5000       19          26  16500
3        5500       24          30  13950
4        5500       18          22  17450
..        ...      ...         ...    ...
200      5400       23          28  16845
201      5300       19          25  19045
202      5500       18          23  21485
203      4800       26          27  22470
204      5400       19          25  22625

[205 rows x 26 columns]
```

```python
# from sklearn.preprocessing import StandardScaler
# # Initialise the Scaler
# scaler = StandardScaler()

# # To scale data
# data=scaler.fit_transform(data)
data_normalized = (data-data.mean())/data.std()
print(data_normalized)
```

```
    aspiration body-style bore  city-mpg  compression-ratio  curb-
weight  \
0          NaN        NaN  NaN -0.644974          -0.287645    -
0.014531
1          NaN        NaN  NaN -0.644974          -0.287645    -
0.014531
2          NaN        NaN  NaN -0.950684          -0.287645
0.513625
3          NaN        NaN  NaN -0.186409          -0.035885    -
0.419770
4          NaN        NaN  NaN -1.103540          -0.539405
0.515545
..         ...        ...  ...       ...                ...          .
..
200        NaN        NaN  NaN -0.339264          -0.161765
0.761377
```

```
201        NaN        NaN  NaN -0.950684          -0.363173
0.947672
202        NaN        NaN  NaN -1.103540          -0.337997
0.876611
203        NaN        NaN  NaN  0.119302           3.236992
1.270327
204        NaN        NaN  NaN -0.950684          -0.161765
0.972640

    drive-wheels engine-location  engine-size engine-type  ... make  \
0            NaN             NaN     0.074267         NaN  ...  NaN
1            NaN             NaN     0.074267         NaN  ...  NaN
2            NaN             NaN     0.602571         NaN  ...  NaN
3            NaN             NaN    -0.430023         NaN  ...  NaN
4            NaN             NaN     0.218350         NaN  ...  NaN
..           ...             ...          ...         ...  ...  ...
200          NaN             NaN     0.338419         NaN  ...  NaN
201          NaN             NaN     0.338419         NaN  ...  NaN
202          NaN             NaN     1.106861         NaN  ...  NaN
203          NaN             NaN     0.434474         NaN  ...  NaN
204          NaN             NaN     0.338419         NaN  ...  NaN

     normalized-losses  num-of-cylinders  num-of-doors peak-rpm  price
stroke  \
0                  NaN               NaN           NaN      NaN    NaN
NaN
1                  NaN               NaN           NaN      NaN    NaN
NaN
2                  NaN               NaN           NaN      NaN    NaN
NaN
3                  NaN               NaN           NaN      NaN    NaN
NaN
4                  NaN               NaN           NaN      NaN    NaN
NaN
..                 ...               ...           ...      ...    ...
...
200                NaN               NaN           NaN      NaN    NaN
NaN
201                NaN               NaN           NaN      NaN    NaN
NaN
202                NaN               NaN           NaN      NaN    NaN
NaN
203                NaN               NaN           NaN      NaN    NaN
NaN
204                NaN               NaN           NaN      NaN    NaN
NaN

    symboling wheel-base     width
0    1.739213  -1.686643 -0.842719
1    1.739213  -1.686643 -0.842719
```

```
2    0.133183  -0.706865 -0.190101
3    0.936198   0.173274  0.136209
4    0.936198   0.106848  0.229440
..        ...        ...       ...
200 -1.472847   1.717669  1.394830
201 -1.472847   1.717669  1.348215
202 -1.472847   1.717669  1.394830
203 -1.472847   1.717669  1.394830
204 -1.472847   1.717669  1.394830

[205 rows x 26 columns]
```

```
<ipython-input-18-aca9df9d4b81>:7: FutureWarning: The default value of
numeric_only in DataFrame.mean is deprecated. In a future version, it
will default to False. In addition, specifying 'numeric_only=None' is
deprecated. Select only valid columns or specify the value of
numeric_only to silence this warning.
  data_normalized = (data-data.mean())/data.std()
<ipython-input-18-aca9df9d4b81>:7: FutureWarning: The default value of
numeric_only in DataFrame.std is deprecated. In a future version, it
will default to False. In addition, specifying 'numeric_only=None' is
deprecated. Select only valid columns or specify the value of
numeric_only to silence this warning.
  data_normalized = (data-data.mean())/data.std()
```

```
data.corr()
```

```
<ipython-input-19-c44ded798807>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it
will default to False. Select only valid columns or specify the value
of numeric_only to silence this warning.
  data.corr()
```

```
                  symboling  fuel-type  wheel-base    length
width  \
symboling          1.000000  -0.194311   -0.531954 -0.357612 -
0.232919
fuel-type         -0.194311   1.000000    0.308346  0.212679
0.233880
wheel-base        -0.531954   0.308346    1.000000  0.874587
0.795144
length            -0.357612   0.212679    0.874587  1.000000
0.841118
width             -0.232919   0.233880    0.795144  0.841118
1.000000
height            -0.541038   0.284631    0.589435  0.491029
0.279210
curb-weight       -0.227691   0.217275    0.776386  0.877728
0.867032
engine-size       -0.105790   0.069594    0.569329  0.683360
0.735433
```

```
compression-ratio  -0.178515   0.984356    0.249786  0.158414
0.181129
city-mpg           -0.035823   0.255963   -0.470414 -0.670909 -
0.642704
highway-mpg         0.034606   0.191392   -0.544082 -0.704662 -
0.677218

                     height  curb-weight  engine-size  compression-
ratio  \
symboling         -0.541038    -0.227691    -0.105790          -
0.178515
fuel-type          0.284631     0.217275     0.069594
0.984356
wheel-base         0.589435     0.776386     0.569329
0.249786
length             0.491029     0.877728     0.683360
0.158414
width              0.279210     0.867032     0.735433
0.181129
height             1.000000     0.295572     0.067149
0.261214
curb-weight        0.295572     1.000000     0.850594
0.151362
engine-size        0.067149     0.850594     1.000000
0.028971
compression-ratio  0.261214     0.151362     0.028971
1.000000
city-mpg          -0.048640    -0.757414    -0.653658
0.324701
highway-mpg       -0.107358    -0.797465    -0.677470
0.265201

                   city-mpg  highway-mpg
symboling         -0.035823     0.034606
fuel-type          0.255963     0.191392
wheel-base        -0.470414    -0.544082
length            -0.670909    -0.704662
width             -0.642704    -0.677218
height            -0.048640    -0.107358
curb-weight       -0.757414    -0.797465
engine-size       -0.653658    -0.677470
compression-ratio  0.324701     0.265201
city-mpg           1.000000     0.971337
highway-mpg        0.971337     1.000000
```

```
data.describe()
#descriptive statistic
```

```
        symboling   fuel-type  wheel-base      length       width
height  \
```

```
count  205.000000  205.000000  205.000000  205.000000  205.000000
205.000000
mean     0.834146    1.097561   98.756585  174.049268   65.907805
53.724878
std      1.245307    0.297446    6.021776   12.337289    2.145204
2.443522
min     -2.000000    1.000000   86.600000  141.100000   60.300000
47.800000
25%      0.000000    1.000000   94.500000  166.300000   64.100000
52.000000
50%      1.000000    1.000000   97.000000  173.200000   65.500000
54.100000
75%      2.000000    1.000000  102.400000  183.100000   66.900000
55.500000
max      3.000000    2.000000  120.900000  208.100000   72.300000
59.800000

       curb-weight  engine-size  compression-ratio    city-mpg
highway-mpg
count   205.000000   205.000000         205.000000  205.000000
205.000000
mean   2555.565854   126.907317          10.142537   25.219512
30.751220
std     520.680204    41.642693           3.972040    6.542142
6.886443
min    1488.000000    61.000000           7.000000   13.000000
16.000000
25%    2145.000000    97.000000           8.600000   19.000000
25.000000
50%    2414.000000   120.000000           9.000000   24.000000
30.000000
75%    2935.000000   141.000000           9.400000   30.000000
34.000000
max    4066.000000   326.000000          23.000000   49.000000
54.000000
```

```
data = data.replace(r'^\s*$', np.nan, regex=True)
```

```
#depandent and independant variable
x=data.iloc[:,1:2]
```

```
x
```

```
     symboling
0            3
1            3
2            1
3            2
4            2
..         ...
200         -1
```

```
201         -1
202         -1
203         -1
204         -1

[205 rows x 1 columns]
```

```
y=data.iloc[:,1:]
y
```

```
    normalized-losses         make  fuel-type aspiration num-of-doors
\
0                   ?  alfa-romero          1        std          two

1                   ?  alfa-romero          1        std          two

2                   ?  alfa-romero          1        std          two

3                 164         audi          1        std         four

4                 164         audi          1        std         four

..                ...          ...        ...        ...          ...

200                95        volvo          1        std         four

201                95        volvo          1      turbo         four

202                95        volvo          1        std         four

203                95        volvo          2      turbo         four

204                95        volvo          1      turbo         four


      body-style drive-wheels engine-location  wheel-base  length  ...
\
0    convertible          rwd           front        88.6   168.8  ...

1    convertible          rwd           front        88.6   168.8  ...

2      hatchback          rwd           front        94.5   171.2  ...

3          sedan          fwd           front        99.8   176.6  ...

4          sedan          4wd           front        99.4   176.6  ...

..           ...          ...             ...         ...     ...  ...
```

```
200        sedan          rwd           front       109.1   188.8  ...

201        sedan          rwd           front       109.1   188.8  ...

202        sedan          rwd           front       109.1   188.8  ...

203        sedan          rwd           front       109.1   188.8  ...

204        sedan          rwd           front       109.1   188.8  ...


     engine-size  fuel-system  bore stroke compression-ratio
horsepower  \
0            130         mpfi  3.47   2.68               9.0
111
1            130         mpfi  3.47   2.68               9.0
111
2            152         mpfi  2.68   3.47               9.0
154
3            109         mpfi  3.19    3.4              10.0
102
4            136         mpfi  3.19    3.4               8.0
115
..           ...          ...   ...    ...               ...         .
..
200          141         mpfi  3.78   3.15               9.5
114
201          141         mpfi  3.78   3.15               8.7
160
202          173         mpfi  3.58   2.87               8.8
134
203          145          idi  3.01    3.4              23.0
106
204          141         mpfi  3.78   3.15               9.5
114

    peak-rpm city-mpg highway-mpg  price
0       5000       21          27  13495
1       5000       21          27  16500
2       5000       19          26  16500
3       5500       24          30  13950
4       5500       18          22  17450
..       ...      ...         ...    ...
200     5400       23          28  16845
201     5300       19          25  19045
202     5500       18          23  21485
203     4800       26          27  22470
204     5400       19          25  22625
```

```
[205 rows x 25 columns]
```

```
#split data into train and test
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,
random_state = 0)

print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(164, 1)
(41, 1)
(164, 25)
(41, 25)
```

```
x_test
```

```
     symboling
52           1
181         -1
5            2
18           2
188          2
170          2
76           2
154          0
104          3
33           1
12           0
129          1
55           3
66           0
45           0
169          2
130          0
7            1
37           0
152          1
80           3
111          0
131          2
171          2
179          3
138          2
156          0
113          0
161          0
89           1
```

```
183          2
193          0
125          3
173         -1
92           1
16           0
189          3
136          3
22           1
74           1
44           1
```

```
x_train
```

```
     symboling
4            2
71          -1
134          3
145          0
122          1
..         ...
67          -1
192          0
117          0
47           0
172          2
```

```
[164 rows x 1 columns]
```

```
print(str(f"{len(x_train):,}"))
print(str(f"{len(y_train):,}"))
print(str(f"{len(x_test):,}"))
print(str(f"{len(y_test):,}"))
```

```
164
164
41
41
```

```
import statsmodels.api as sm
x_train_sm = sm.add_constant(x_train)
```

```
print(str(x_train_sm)) #intercept added to the linear regression model
```

```
     const  symboling
4      1.0          2
71     1.0         -1
134    1.0          3
145    1.0          0
122    1.0          1
..     ...        ...
67     1.0         -1
```

```
192    1.0          0
117    1.0          0
47     1.0          0
172    1.0          2

[164 rows x 2 columns]
```

```
import numpy as np

# from sklearn.linear_model import LinearRegression
# lr = LinearRegression()

#fitting the regression line using "OLS"
lr = sm.OLS(y_train,x_train_sm).fit()
lr.params
lr.summary()
```

```
---------------------------------------------------------------------------
-----
ValueError                                Traceback (most recent call
last)
<ipython-input-61-70cd6a61859c> in <cell line: 2>()
      1 #fitting the regression line using "OLS"
----> 2 lr = sm.OLS(y_train,x_train_sm).fit()
      3 lr.params
      4 lr.summary()

/usr/local/lib/python3.10/dist-packages/statsmodels/regression/linear_
model.py in __init__(self, endog, exog, missing, hasconst, **kwargs)
    904                    "An exception will be raised in the next
version.")
    905             warnings.warn(msg, ValueWarning)
--> 906         super(OLS, self).__init__(endog, exog,
missing=missing,
    907                                   hasconst=hasconst, **kwargs)
    908         if "weights" in self._init_keys:

/usr/local/lib/python3.10/dist-packages/statsmodels/regression/linear_
model.py in __init__(self, endog, exog, weights, missing, hasconst,
**kwargs)
    731         else:
    732             weights = weights.squeeze()
--> 733         super(WLS, self).__init__(endog, exog,
missing=missing,
    734                                   weights=weights,
hasconst=hasconst, **kwargs)
    735         nobs = self.exog.shape[0]

/usr/local/lib/python3.10/dist-packages/statsmodels/regression/linear_
model.py in __init__(self, endog, exog, **kwargs)
    188     """
```

```
    189     def __init__(self, endog, exog, **kwargs):
--> 190         super(RegressionModel, self).__init__(endog, exog,
**kwargs)
    191         self._data_attr.extend(['pinv_wexog', 'wendog',
'wexog', 'weights'])
    192

/usr/local/lib/python3.10/dist-packages/statsmodels/base/model.py in
__init__(self, endog, exog, **kwargs)
    265
    266     def __init__(self, endog, exog=None, **kwargs):
--> 267         super().__init__(endog, exog, **kwargs)
    268         self.initialize()
    269

/usr/local/lib/python3.10/dist-packages/statsmodels/base/model.py in
__init__(self, endog, exog, **kwargs)
     90         missing = kwargs.pop('missing', 'none')
     91         hasconst = kwargs.pop('hasconst', None)
---> 92         self.data = self._handle_data(endog, exog, missing,
hasconst,
     93                                       **kwargs)
     94         self.k_constant = self.data.k_constant

/usr/local/lib/python3.10/dist-packages/statsmodels/base/model.py in
_handle_data(self, endog, exog, missing, hasconst, **kwargs)
    130
    131     def _handle_data(self, endog, exog, missing, hasconst,
**kwargs):
--> 132         data = handle_data(endog, exog, missing, hasconst,
**kwargs)
    133         # kwargs arrays could have changed, easier to just
attach here
    134         for key in kwargs:

/usr/local/lib/python3.10/dist-packages/statsmodels/base/data.py in
handle_data(endog, exog, missing, hasconst, **kwargs)
    698
    699     klass = handle_data_class_factory(endog, exog)
--> 700     return klass(endog, exog=exog, missing=missing,
hasconst=hasconst,
    701                  **kwargs)

/usr/local/lib/python3.10/dist-packages/statsmodels/base/data.py in
__init__(self, endog, exog, missing, hasconst, **kwargs)
     82             self.orig_endog = endog
     83             self.orig_exog = exog
---> 84             self.endog, self.exog =
self._convert_endog_exog(endog, exog)
     85
```

```
     86         self.const_idx = None

/usr/local/lib/python3.10/dist-packages/statsmodels/base/data.py in
_convert_endog_exog(self, endog, exog)
    528             else:
    529                 exog_dtype = None
--> 530             raise ValueError(
    531                 "Pandas data cast to numpy dtype of object.
Check input data "
    532                 "with np.asarray(data). The types seen were"

ValueError: Pandas data cast to numpy dtype of object. Check input
data with np.asarray(data). The types seen werenormalized-losses
object
make                  object
fuel-type              int64
aspiration            object
num-of-doors          object
body-style            object
drive-wheels          object
engine-location       object
wheel-base           float64
length               float64
width                float64
height               float64
curb-weight            int64
engine-type           object
num-of-cylinders      object
engine-size            int64
fuel-system           object
bore                  object
stroke                object
compression-ratio    float64
horsepower            object
peak-rpm              object
city-mpg               int64
highway-mpg            int64
price                 object
dtype: object and const        float64
symboling      int64
dtype: object. The data was
    normalized-losses           make  fuel-type aspiration num-of-
doors  \
4                 164           audi          1        std
four
71                  ?  mercedes-benz          1        std
four
134               150           saab          1        std
two
145               102         subaru          1      turbo
```

```
four
122               154       plymouth          1        std
four
..                ...            ...        ...        ...          ..
.
67                 93  mercedes-benz          2      turbo
four
192                 ?     volkswagen          2      turbo
four
117               161         peugot          1      turbo
four
47                145         jaguar          1        std
four
172               134         toyota          1        std
two

      body-style drive-wheels engine-location  wheel-base  length  ...
\
4          sedan          4wd           front        99.4   176.6  ...

71         sedan          rwd           front       115.6   202.6  ...

134    hatchback          fwd           front        99.1   186.6  ...

145        sedan          4wd           front        97.0   172.0  ...

122        sedan          fwd           front        93.7   167.3  ...

..           ...          ...             ...         ...     ...  ...

67         sedan          rwd           front       110.0   190.9  ...

192        sedan          fwd           front       100.4   180.2  ...

117        sedan          rwd           front       108.0   186.7  ...

47         sedan          rwd           front       113.0   199.6  ...

172  convertible          rwd           front        98.4   176.2  ...


     engine-size  fuel-system  bore stroke compression-ratio
horsepower  \
4            136         mpfi  3.19    3.4               8.0
115
71           234         mpfi  3.46    3.1               8.3
155
134          121         mpfi  2.54   2.07               9.3
110
```

```
145          108         mpfi  3.62   2.64              7.7
111
122           98         2bbl  2.97   3.23              9.4
68
..           ...          ...   ...    ...              ...         .
..
67           183          idi  3.58   3.64             21.5
123
192           97          idi  3.01    3.4             23.0
68
117          134         mpfi  3.61   3.21              7.0
142
47           258         mpfi  3.63   4.17              8.1
176
172          146         mpfi  3.62    3.5              9.3
116

    peak-rpm city-mpg highway-mpg  price
4       5500       18          22  17450
71      4750       16          18  34184
134     5250       21          28  15040
145     4800       24          29  11259
122     5500       31          38   7609
..       ...      ...         ...    ...
67      4350       22          25  25552
192     4500       33          38  13845
117     5600       18          24  18150
47      4750       15          19  32250
172     4800       24          30  17669

[164 rows x 25 columns]
and
      const  symboling
4      1.0          2
71     1.0         -1
134    1.0          3
145    1.0          0
122    1.0          1
..     ...        ...
67     1.0         -1
192    1.0          0
117    1.0          0
47     1.0          0
172    1.0          2

[164 rows x 2 columns]
before. After,
[['164' 'audi' 1 ... 18 22 '17450']
 ['?' 'mercedes-benz' 1 ... 16 18 '34184']
 ['150' 'saab' 1 ... 21 28 '15040']
```

```
 ...
 ['161' 'peugot' 1 ... 18 24 '18150']
 ['145' 'jaguar' 1 ... 15 19 '32250']
 ['134' 'toyota' 1 ... 24 30 '17669']]
[[ 1.  2.]
 [ 1. -1.]
 [ 1.  3.]
 [ 1.  0.]
 [ 1.  1.]
 [ 1.  1.]
 [ 1.  3.]
 [ 1.  0.]
 [ 1.  2.]
 [ 1.  1.]
 [ 1.  1.]
 [ 1.  1.]
 [ 1.  1.]
 [ 1.  1.]
 [ 1.  1.]
 [ 1.  0.]
 [ 1. -2.]
 [ 1.  0.]
 [ 1.  1.]
 [ 1.  2.]
 [ 1.  2.]
 [ 1.  0.]
 [ 1.  0.]
 [ 1.  3.]
 [ 1.  0.]
 [ 1.  1.]
 [ 1.  1.]
 [ 1.  0.]
 [ 1.  1.]
 [ 1.  1.]
 [ 1.  2.]
 [ 1.  1.]
 [ 1.  2.]
 [ 1.  0.]
 [ 1.  0.]
 [ 1.  1.]
 [ 1.  0.]
 [ 1.  0.]
 [ 1.  0.]
 [ 1.  3.]
 [ 1.  1.]
 [ 1.  1.]
 [ 1.  0.]
 [ 1.  2.]
 [ 1.  3.]
 [ 1.  2.]
```

```
[ 1.  0.]
[ 1. -1.]
[ 1.  0.]
[ 1.  0.]
[ 1.  1.]
[ 1.  1.]
[ 1. -1.]
[ 1.  3.]
[ 1.  0.]
[ 1.  2.]
[ 1.  0.]
[ 1.  0.]
[ 1. -2.]
[ 1.  2.]
[ 1.  1.]
[ 1.  2.]
[ 1.  2.]
[ 1.  1.]
[ 1.  1.]
[ 1.  0.]
[ 1. -1.]
[ 1.  1.]
[ 1.  3.]
[ 1.  1.]
[ 1.  0.]
[ 1.  1.]
[ 1.  0.]
[ 1.  0.]
[ 1.  0.]
[ 1.  0.]
[ 1.  0.]
[ 1.  0.]
[ 1.  1.]
[ 1. -1.]
[ 1.  0.]
[ 1.  0.]
[ 1.  1.]
[ 1.  3.]
[ 1.  2.]
[ 1.  0.]
[ 1.  0.]
[ 1.  2.]
[ 1.  0.]
[ 1.  1.]
[ 1. -1.]
[ 1.  1.]
[ 1. -1.]
[ 1.  3.]
[ 1.  1.]
[ 1.  0.]
```

```
[ 1.  2.]
[ 1.  0.]
[ 1.  1.]
[ 1. -1.]
[ 1.  0.]
[ 1.  1.]
[ 1.  0.]
[ 1.  0.]
[ 1.  3.]
[ 1.  0.]
[ 1.  3.]
[ 1.  1.]
[ 1.  1.]
[ 1.  1.]
[ 1.  3.]
[ 1.  3.]
[ 1. -1.]
[ 1.  0.]
[ 1.  0.]
[ 1.  2.]
[ 1.  1.]
[ 1.  0.]
[ 1.  3.]
[ 1.  1.]
[ 1. -1.]
[ 1.  0.]
[ 1.  1.]
[ 1.  2.]
[ 1.  2.]
[ 1.  3.]
[ 1. -1.]
[ 1.  1.]
[ 1.  2.]
[ 1.  0.]
[ 1. -1.]
[ 1.  0.]
[ 1.  3.]
[ 1.  0.]
[ 1.  3.]
[ 1. -1.]
[ 1.  1.]
[ 1.  0.]
[ 1.  0.]
[ 1. -1.]
[ 1.  3.]
[ 1.  2.]
[ 1.  1.]
[ 1.  1.]
[ 1.  3.]
[ 1. -1.]
```

```
[ 1. -1.]
[ 1. -2.]
[ 1.  0.]
[ 1.  3.]
[ 1.  2.]
[ 1. -1.]
[ 1. -1.]
[ 1.  1.]
[ 1.  0.]
[ 1.  1.]
[ 1.  0.]
[ 1.  0.]
[ 1. -1.]
[ 1. -1.]
[ 1.  0.]
[ 1.  0.]
[ 1.  0.]
[ 1.  2.]].
```