untitled64

November 29, 2024

[11]:
```
filename="https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/
 ↪IBMDeveloperSkillsNetwork-DA0101EN-SkillsNetwork/labs/Data%20files/auto.csv"
headers = ["symboling","normalized-losses","make","fuel-type","aspiration",␣
 ↪"num-of-doors","body-style",
        "drive-wheels","engine-location","wheel-base",␣
 ↪"length","width","height","curb-weight","engine-type",
        "num-of-cylinders",␣
 ↪"engine-size","fuel-system","bore","stroke","compression-ratio","horsepower",
        "peak-rpm","city-mpg","highway-mpg","price"]
```

[13]:
```
import pandas as pd
import matplotlib.pylab as plt
df = pd.read_csv(filename, names = headers)
```

[15]:
```
df
```

[15]:

	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	
..	…	…	…	…	…	
200	-1	95	volvo	gas	std	
201	-1	95	volvo	gas	turbo	
202	-1	95	volvo	gas	std	
203	-1	95	volvo	diesel	turbo	
204	-1	95	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	…	
..	…	…	…	…	…	…	
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    3.40               10.0        102
4            136         mpfi  3.19    3.40                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    3.40               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]
```

[17]:
```
import numpy as  np
df.replace("?",np.nan,inplace=True)
```

[19]:
```
df.head(50)
```

[19]:
```
    symboling normalized-losses         make fuel-type aspiration  \
0           3                NaN  alfa-romero       gas        std
1           3                NaN  alfa-romero       gas        std
2           1                NaN  alfa-romero       gas        std
3           2                164         audi       gas        std
4           2                164         audi       gas        std
5           2                NaN         audi       gas        std
6           1                158         audi       gas        std
7           1                NaN         audi       gas        std
```

```
8           1                158         audi       gas      turbo
9           0                NaN         audi       gas      turbo
10          2                192          bmw       gas        std
11          0                192          bmw       gas        std
12          0                188          bmw       gas        std
13          0                188          bmw       gas        std
14          1                NaN          bmw       gas        std
15          0                NaN          bmw       gas        std
16          0                NaN          bmw       gas        std
17          0                NaN          bmw       gas        std
18          2                121    chevrolet       gas        std
19          1                 98    chevrolet       gas        std
20          0                 81    chevrolet       gas        std
21          1                118        dodge       gas        std
22          1                118        dodge       gas        std
23          1                118        dodge       gas      turbo
24          1                148        dodge       gas        std
25          1                148        dodge       gas        std
26          1                148        dodge       gas        std
27          1                148        dodge       gas      turbo
28         -1                110        dodge       gas        std
29          3                145        dodge       gas      turbo
30          2                137        honda       gas        std
31          2                137        honda       gas        std
32          1                101        honda       gas        std
33          1                101        honda       gas        std
34          1                101        honda       gas        std
35          0                110        honda       gas        std
36          0                 78        honda       gas        std
37          0                106        honda       gas        std
38          0                106        honda       gas        std
39          0                 85        honda       gas        std
40          0                 85        honda       gas        std
41          0                 85        honda       gas        std
42          1                107        honda       gas        std
43          0                NaN        isuzu       gas        std
44          1                NaN        isuzu       gas        std
45          0                NaN        isuzu       gas        std
46          2                NaN        isuzu       gas        std
47          0                145       jaguar       gas        std
48          0                NaN       jaguar       gas        std
49          0                NaN       jaguar       gas        std

   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	…
5	two	sedan	fwd	front	99.8	…
6	four	sedan	fwd	front	105.8	…
7	four	wagon	fwd	front	105.8	…
8	four	sedan	fwd	front	105.8	…
9	two	hatchback	4wd	front	99.5	…
10	two	sedan	rwd	front	101.2	…
11	four	sedan	rwd	front	101.2	…
12	two	sedan	rwd	front	101.2	…
13	four	sedan	rwd	front	101.2	…
14	four	sedan	rwd	front	103.5	…
15	four	sedan	rwd	front	103.5	…
16	two	sedan	rwd	front	103.5	…
17	four	sedan	rwd	front	110.0	…
18	two	hatchback	fwd	front	88.4	…
19	two	hatchback	fwd	front	94.5	…
20	four	sedan	fwd	front	94.5	…
21	two	hatchback	fwd	front	93.7	…
22	two	hatchback	fwd	front	93.7	…
23	two	hatchback	fwd	front	93.7	…
24	four	hatchback	fwd	front	93.7	…
25	four	sedan	fwd	front	93.7	…
26	four	sedan	fwd	front	93.7	…
27	NaN	sedan	fwd	front	93.7	…
28	four	wagon	fwd	front	103.3	…
29	two	hatchback	fwd	front	95.9	…
30	two	hatchback	fwd	front	86.6	…
31	two	hatchback	fwd	front	86.6	…
32	two	hatchback	fwd	front	93.7	…
33	two	hatchback	fwd	front	93.7	…
34	two	hatchback	fwd	front	93.7	…
35	four	sedan	fwd	front	96.5	…
36	four	wagon	fwd	front	96.5	…
37	two	hatchback	fwd	front	96.5	…
38	two	hatchback	fwd	front	96.5	…
39	four	sedan	fwd	front	96.5	…
40	four	sedan	fwd	front	96.5	…
41	four	sedan	fwd	front	96.5	…
42	two	sedan	fwd	front	96.5	…
43	four	sedan	rwd	front	94.3	…
44	two	sedan	fwd	front	94.5	…
45	four	sedan	fwd	front	94.5	…
46	two	hatchback	rwd	front	96.0	…
47	four	sedan	rwd	front	113.0	…
48	four	sedan	rwd	front	113.0	…
49	two	sedan	rwd	front	102.0	…

	engine-size	fuel-system	bore	stroke	compression-ratio	horsepower	\
0	130	mpfi	3.47	2.68	9.00	111	
1	130	mpfi	3.47	2.68	9.00	111	
2	152	mpfi	2.68	3.47	9.00	154	
3	109	mpfi	3.19	3.40	10.00	102	
4	136	mpfi	3.19	3.40	8.00	115	
5	136	mpfi	3.19	3.40	8.50	110	
6	136	mpfi	3.19	3.40	8.50	110	
7	136	mpfi	3.19	3.40	8.50	110	
8	131	mpfi	3.13	3.40	8.30	140	
9	131	mpfi	3.13	3.40	7.00	160	
10	108	mpfi	3.50	2.80	8.80	101	
11	108	mpfi	3.50	2.80	8.80	101	
12	164	mpfi	3.31	3.19	9.00	121	
13	164	mpfi	3.31	3.19	9.00	121	
14	164	mpfi	3.31	3.19	9.00	121	
15	209	mpfi	3.62	3.39	8.00	182	
16	209	mpfi	3.62	3.39	8.00	182	
17	209	mpfi	3.62	3.39	8.00	182	
18	61	2bbl	2.91	3.03	9.50	48	
19	90	2bbl	3.03	3.11	9.60	70	
20	90	2bbl	3.03	3.11	9.60	70	
21	90	2bbl	2.97	3.23	9.41	68	
22	90	2bbl	2.97	3.23	9.40	68	
23	98	mpfi	3.03	3.39	7.60	102	
24	90	2bbl	2.97	3.23	9.40	68	
25	90	2bbl	2.97	3.23	9.40	68	
26	90	2bbl	2.97	3.23	9.40	68	
27	98	mpfi	3.03	3.39	7.60	102	
28	122	2bbl	3.34	3.46	8.50	88	
29	156	mfi	3.60	3.90	7.00	145	
30	92	1bbl	2.91	3.41	9.60	58	
31	92	1bbl	2.91	3.41	9.20	76	
32	79	1bbl	2.91	3.07	10.10	60	
33	92	1bbl	2.91	3.41	9.20	76	
34	92	1bbl	2.91	3.41	9.20	76	
35	92	1bbl	2.91	3.41	9.20	76	
36	92	1bbl	2.92	3.41	9.20	76	
37	110	1bbl	3.15	3.58	9.00	86	
38	110	1bbl	3.15	3.58	9.00	86	
39	110	1bbl	3.15	3.58	9.00	86	
40	110	1bbl	3.15	3.58	9.00	86	
41	110	mpfi	3.15	3.58	9.00	101	
42	110	2bbl	3.15	3.58	9.10	100	
43	111	2bbl	3.31	3.23	8.50	78	
44	90	2bbl	3.03	3.11	9.60	70	

45	90	2bbl	3.03	3.11	9.60	70
46	119	spfi	3.43	3.23	9.20	90
47	258	mpfi	3.63	4.17	8.10	176
48	258	mpfi	3.63	4.17	8.10	176
49	326	mpfi	3.54	2.76	11.50	262

	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
5	5500	19	25	15250
6	5500	19	25	17710
7	5500	19	25	18920
8	5500	17	20	23875
9	5500	16	22	NaN
10	5800	23	29	16430
11	5800	23	29	16925
12	4250	21	28	20970
13	4250	21	28	21105
14	4250	20	25	24565
15	5400	16	22	30760
16	5400	16	22	41315
17	5400	15	20	36880
18	5100	47	53	5151
19	5400	38	43	6295
20	5400	38	43	6575
21	5500	37	41	5572
22	5500	31	38	6377
23	5500	24	30	7957
24	5500	31	38	6229
25	5500	31	38	6692
26	5500	31	38	7609
27	5500	24	30	8558
28	5000	24	30	8921
29	5000	19	24	12964
30	4800	49	54	6479
31	6000	31	38	6855
32	5500	38	42	5399
33	6000	30	34	6529
34	6000	30	34	7129
35	6000	30	34	7295
36	6000	30	34	7295
37	5800	27	33	7895
38	5800	27	33	9095
39	5800	27	33	8845

```
40      5800        27           33  10295
41      5800        24           28  12945
42      5500        25           31  10345
43      4800        24           29   6785
44      5400        38           43    NaN
45      5400        38           43    NaN
46      5000        24           29  11048
47      4750        15           19  32250
48      4750        15           19  35550
49      5000        13           17  36000

[50 rows x 26 columns]
```

[21]:
```
missing_data = df.isnull()
missing_data.head(5)
```

[21]:
```
   symboling  normalized-losses   make  fuel-type  aspiration  num-of-doors  \
0      False              True  False      False       False         False
1      False              True  False      False       False         False
2      False              True  False      False       False         False
3      False             False  False      False       False         False
4      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

   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

   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

[5 rows x 26 columns]
```

[23]:
```
for column in missing_data.columns.values.tolist():
    print(column)
    print (missing_data[column].value_counts())
    print("")
```

```
symboling
symboling
False    205
Name: count, dtype: int64

normalized-losses
normalized-losses
False    164
True      41
Name: count, dtype: int64

make
make
False    205
Name: count, dtype: int64

fuel-type
fuel-type
False    205
Name: count, dtype: int64

aspiration
aspiration
False    205
Name: count, dtype: int64

num-of-doors
num-of-doors
False    203
True       2
Name: count, dtype: int64

body-style
body-style
False    205
Name: count, dtype: int64

drive-wheels
drive-wheels
False    205
Name: count, dtype: int64
```

```
engine-location
engine-location
False    205
Name: count, dtype: int64

wheel-base
wheel-base
False    205
Name: count, dtype: int64

length
length
False    205
Name: count, dtype: int64

width
width
False    205
Name: count, dtype: int64

height
height
False    205
Name: count, dtype: int64

curb-weight
curb-weight
False    205
Name: count, dtype: int64

engine-type
engine-type
False    205
Name: count, dtype: int64

num-of-cylinders
num-of-cylinders
False    205
Name: count, dtype: int64

engine-size
engine-size
False    205
Name: count, dtype: int64

fuel-system
fuel-system
False    205
```

```
Name: count, dtype: int64

bore
bore
False    201
True       4
Name: count, dtype: int64

stroke
stroke
False    201
True       4
Name: count, dtype: int64

compression-ratio
compression-ratio
False    205
Name: count, dtype: int64

horsepower
horsepower
False    203
True       2
Name: count, dtype: int64

peak-rpm
peak-rpm
False    203
True       2
Name: count, dtype: int64

city-mpg
city-mpg
False    205
Name: count, dtype: int64

highway-mpg
highway-mpg
False    205
Name: count, dtype: int64

price
price
False    201
True       4
Name: count, dtype: int64
```

[25]:
```
avg_norm_loss = df["normalized-losses"].astype("float").mean(axis=0)
print("Average of normalized-losses:", avg_norm_loss)
```

```
Average of normalized-losses: 122.0
```

[27]:
```
df["normalized-losses"].replace(np.nan, avg_norm_loss, inplace=True)
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_13824\2599940699.py:1: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained
assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


  df["normalized-losses"].replace(np.nan, avg_norm_loss, inplace=True)
```

[29]:
```
avg_bore=df['bore'].astype('float').mean(axis=0)
print("Average of bore:", avg_bore)
```

```
Average of bore: 3.3297512437810943
```

[31]:
```
df["bore"].replace(np.nan, avg_bore, inplace=True)
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_13824\1952189479.py:1: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained
assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


  df["bore"].replace(np.nan, avg_bore, inplace=True)
```

[33]:
```
df["stroke"].replace(np.nan,df['stroke'].astype('float').mean(axis=0))
```

[33]:
```
0      2.68
1      2.68
2      3.47
3      3.40
```

```
4      3.40
       …
200    3.15
201    3.15
202    2.87
203    3.40
204    3.15
Name: stroke, Length: 205, dtype: object
```

[35]:
```
df["stroke"]
```

[35]:
```
0      2.68
1      2.68
2      3.47
3      3.40
4      3.40
       …
200    3.15
201    3.15
202    2.87
203    3.40
204    3.15
Name: stroke, Length: 205, dtype: object
```

[37]:
```
df
```

[37]:
```
     symboling normalized-losses         make fuel-type aspiration  \
0            3             122.0  alfa-romero       gas        std
1            3             122.0  alfa-romero       gas        std
2            1             122.0  alfa-romero       gas        std
3            2               164         audi       gas        std
4            2               164         audi       gas        std
..         …               …           …        …         …
200         -1                95        volvo       gas        std
201         -1                95        volvo       gas      turbo
202         -1                95        volvo       gas        std
203         -1                95        volvo    diesel      turbo
204         -1                95        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  …
..           …          …          …              …          …  …
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    3.40               10.0        102
4            136         mpfi  3.19    3.40                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    3.40               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]
```

[39]:
```
df["horsepower"].astype('float').mean(axis=0)
```

[39]:
```
104.25615763546799
```

[41]:
```
df['horsepower'].replace(np.nan,df["horsepower"].astype('float').mean(axis=0) ,␣
↪inplace=True)
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_13824\3745643968.py:1: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained
assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.
```

```
For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


```

```
  df['horsepower'].replace(np.nan,df["horsepower"].astype('float').mean(axis=0)
, inplace=True)
```

[43]:
```
avg_peakrpm=df['peak-rpm'].astype('float').mean(axis=0)
print("Average peak rpm:", avg_peakrpm)
```

```
Average peak rpm: 5125.369458128079
```

[45]:
```
df['peak-rpm'].replace(np.nan, avg_peakrpm, inplace=True)
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_13824\2061375298.py:1: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained
assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


```

```
  df['peak-rpm'].replace(np.nan, avg_peakrpm, inplace=True)
```

[47]:
```
df['num-of-doors'].value_counts()
```

[47]:
```
num-of-doors
four    114
two      89
Name: count, dtype: int64
```

[49]:
```
df['num-of-doors'].value_counts().idxmax()
```

[49]:
```
'four'
```

[51]:
```
#replace the missing 'num-of-doors' values by the most frequent
df["num-of-doors"].replace(np.nan, "four", inplace=True)
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_13824\2406474689.py:2: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained
assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
```

```
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


  df["num-of-doors"].replace(np.nan, "four", inplace=True)
```

[53]:
```
# simply drop whole row with NaN in "price" column
df.dropna(subset=["price"], axis=0, inplace=True)

# reset index, because we droped two rows
df.reset_index(drop=True, inplace=True)
```

[74]:
```
df.head()
```

[74]:

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	\
0	3	122.0	alfa-romero	gas	std	two	
1	3	122.0	alfa-romero	gas	std	two	
2	1	122.0	alfa-romero	gas	std	two	
3	2	164	audi	gas	std	four	
4	2	164	audi	gas	std	four	

	body-style	drive-wheels	engine-location	wheel-base	…	engine-size	\
0	convertible	rwd	front	88.6	…	130	
1	convertible	rwd	front	88.6	…	130	
2	hatchback	rwd	front	94.5	…	152	
3	sedan	fwd	front	99.8	…	109	
4	sedan	4wd	front	99.4	…	136	

	fuel-system	bore	stroke	compression-ratio	horsepower	peak-rpm	city-mpg	\
0	mpfi	3.47	2.68	9.0	111	5000	21	
1	mpfi	3.47	2.68	9.0	111	5000	21	
2	mpfi	2.68	3.47	9.0	154	5000	19	
3	mpfi	3.19	3.40	10.0	102	5500	24	
4	mpfi	3.19	3.40	8.0	115	5500	18	

	highway-mpg	price
0	27	13495
1	27	16500
2	26	16500
3	30	13950
4	22	17450

[5 rows x 26 columns]

[55]:
```
df.dtypes
```

[55]:
```
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
```

[57]:
```
df[["bore", "stroke"]] = df[["bore", "stroke"]].astype("float")
df[["normalized-losses"]] = df[["normalized-losses"]].astype("int")
df[["price"]] = df[["price"]].astype("float")
df[["peak-rpm"]] = df[["peak-rpm"]].astype("float")
```

[80]:
```
df.dtypes
```

[80]:
```
symboling              int64
normalized-losses      int32
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                 float64
stroke               float64
compression-ratio    float64
horsepower            object
peak-rpm             float64
city-mpg               int64
highway-mpg            int64
price                float64
dtype: object
```

[82]:
```
df.head()
```

[82]:

	symboling	normalized-losses	make	fuel-type	aspiration	…
0	3	122	alfa-romero	gas	std	
1	3	122	alfa-romero	gas	std	
2	1	122	alfa-romero	gas	std	
3	2	164	audi	gas	std	
4	2	164	audi	gas	std	

	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	…

	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	3.40	10.0	102
4	136	mpfi	3.19	3.40	8.0	115

	peak-rpm	city-mpg	highway-mpg	price
0	5000.0	21	27	13495.0
1	5000.0	21	27	16500.0
2	5000.0	19	26	16500.0
3	5500.0	24	30	13950.0
4	5500.0	18	22	17450.0

```
[5 rows x 26 columns]
```

[59]:
```
df["highway-mpg"].replace("mpg","L/100km",inplace=True)
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_13824\460410175.py:1: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained
assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


  df["highway-mpg"].replace("mpg","L/100km",inplace=True)
```

[86]:
```
df
```

[86]:

	symboling	normalized-losses	make	fuel-type	aspiration	\
0	3	122	alfa-romero	gas	std	
1	3	122	alfa-romero	gas	std	
2	1	122	alfa-romero	gas	std	
3	2	164	audi	gas	std	
4	2	164	audi	gas	std	
..	…	…	…	…	…	
196	-1	95	volvo	gas	std	
197	-1	95	volvo	gas	turbo	
198	-1	95	volvo	gas	std	
199	-1	95	volvo	diesel	turbo	
200	-1	95	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	…	
..	…	…	…	…	…	…	
196	four	sedan	rwd	front	109.1	…	
197	four	sedan	rwd	front	109.1	…	
198	four	sedan	rwd	front	109.1	…	
199	four	sedan	rwd	front	109.1	…	
200	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    3.40               10.0        102
4            136         mpfi  3.19    3.40                8.0        115
..           …           …   …     …                  …         …
196          141         mpfi  3.78    3.15                9.5        114
197          141         mpfi  3.78    3.15                8.7        160
198          173         mpfi  3.58    2.87                8.8        134
199          145          idi  3.01    3.40               23.0        106
200          141         mpfi  3.78    3.15                9.5        114

     peak-rpm city-mpg  highway-mpg    price
0      5000.0       21           27  13495.0
1      5000.0       21           27  16500.0
2      5000.0       19           26  16500.0
3      5500.0       24           30  13950.0
4      5500.0       18           22  17450.0
..        …       …           …      …
196    5400.0       23           28  16845.0
197    5300.0       19           25  19045.0
198    5500.0       18           23  21485.0
199    4800.0       26           27  22470.0
200    5400.0       19           25  22625.0

[201 rows x 26 columns]
```

[61]:
```
# transform mpg to L/100km by mathematical operation (235 divided by mpg)
df["highway-mpg"] = 235/df["highway-mpg"]

# rename column name from "highway-mpg" to "highway-L/100km"
df.rename(columns={'"highway-mpg"':'highway-L/100km'}, inplace=True)

# check your transformed data
df.head()
df["height"]
```

[61]:
```
0      48.8
1      48.8
2      52.4
3      54.3
4      54.3
       …
196    55.5
197    55.5
198    55.5
```

```
199    55.5
200    55.5
Name: height, Length: 201, dtype: float64
```

[63]:
```
# Write your code below and press Shift+Enter to execute
df["height"]=df["height"]/df["height"].max()
```

[91]:
```
df
```

[91]:

	symboling	normalized-losses	make	fuel-type	aspiration	\
0	3	122	alfa-romero	gas	std	
1	3	122	alfa-romero	gas	std	
2	1	122	alfa-romero	gas	std	
3	2	164	audi	gas	std	
4	2	164	audi	gas	std	
..	…	…	…	…	…	
196	-1	95	volvo	gas	std	
197	-1	95	volvo	gas	turbo	
198	-1	95	volvo	gas	std	
199	-1	95	volvo	diesel	turbo	
200	-1	95	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	…	
..	…	…	…	…	…	…	
196	four	sedan	rwd	front	109.1	…	
197	four	sedan	rwd	front	109.1	…	
198	four	sedan	rwd	front	109.1	…	
199	four	sedan	rwd	front	109.1	…	
200	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	3.40	10.0	102	
4	136	mpfi	3.19	3.40	8.0	115	
..	…	…	…	…	…	…	
196	141	mpfi	3.78	3.15	9.5	114	
197	141	mpfi	3.78	3.15	8.7	160	
198	173	mpfi	3.58	2.87	8.8	134	
199	145	idi	3.01	3.40	23.0	106	
200	141	mpfi	3.78	3.15	9.5	114	

```
     peak-rpm city-mpg  highway-mpg    price
0      5000.0       21     8.703704  13495.0
1      5000.0       21     8.703704  16500.0
2      5000.0       19     9.038462  16500.0
3      5500.0       24     7.833333  13950.0
4      5500.0       18    10.681818  17450.0
..        …       …           …       …
196    5400.0       23     8.392857  16845.0
197    5300.0       19     9.400000  19045.0
198    5500.0       18    10.217391  21485.0
199    4800.0       26     8.703704  22470.0
200    5400.0       19     9.400000  22625.0

[201 rows x 26 columns]
```

[65]:
```
df["height"]
```

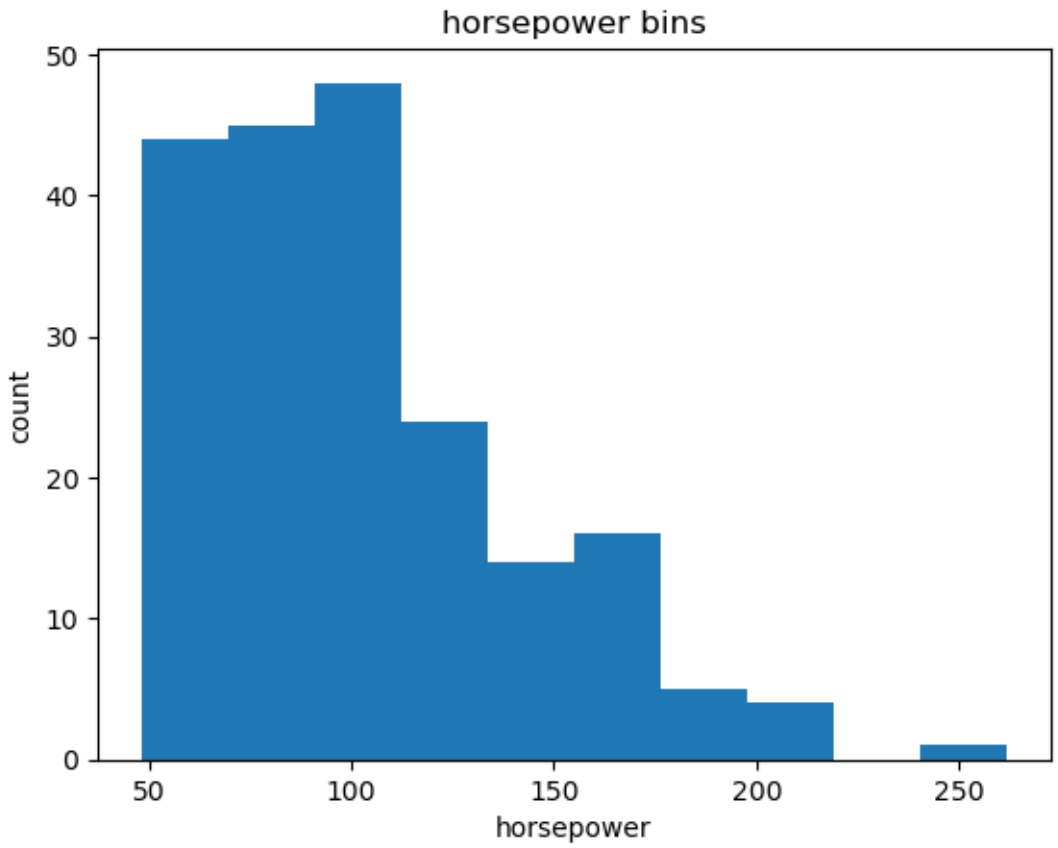
[65]:
```
0      0.816054
1      0.816054
2      0.876254
3      0.908027
4      0.908027
         …
196    0.928094
197    0.928094
198    0.928094
199    0.928094
200    0.928094
Name: height, Length: 201, dtype: float64
```

[67]:
```
df["horsepower"]=df["horsepower"].astype(int, copy=True)
```

[69]:
```
%matplotlib inline
import matplotlib as plt
from matplotlib import pyplot
plt.pyplot.hist(df["horsepower"])

# set x/y labels and plot title
plt.pyplot.xlabel("horsepower")
plt.pyplot.ylabel("count")
plt.pyplot.title("horsepower bins")
```

[69]:
```
Text(0.5, 1.0, 'horsepower bins')
```



[71]:
```
bins = np.linspace(min(df["horsepower"]), max(df["horsepower"]), 4)
bins
```

[71]:
```
array([ 48.        , 119.33333333, 190.66666667, 262.        ])
```

[73]:
```
group_names = ['Low', 'Medium', 'High']
```

[75]:
```
df['horsepower-binned'] = pd.cut(df['horsepower'], bins, labels=group_names,␣
 ↪include_lowest=True )
df[['horsepower','horsepower-binned']].head(20)
```

[75]:

	horsepower	horsepower-binned
0	111	Low
1	111	Low
2	154	Medium
3	102	Low
4	115	Low
5	110	Low
6	110	Low
7	110	Low

```
8          140            Medium
9          101               Low
10         101               Low
11         121            Medium
12         121            Medium
13         121            Medium
14         182            Medium
15         182            Medium
16         182            Medium
17          48               Low
18          70               Low
19          70               Low
```

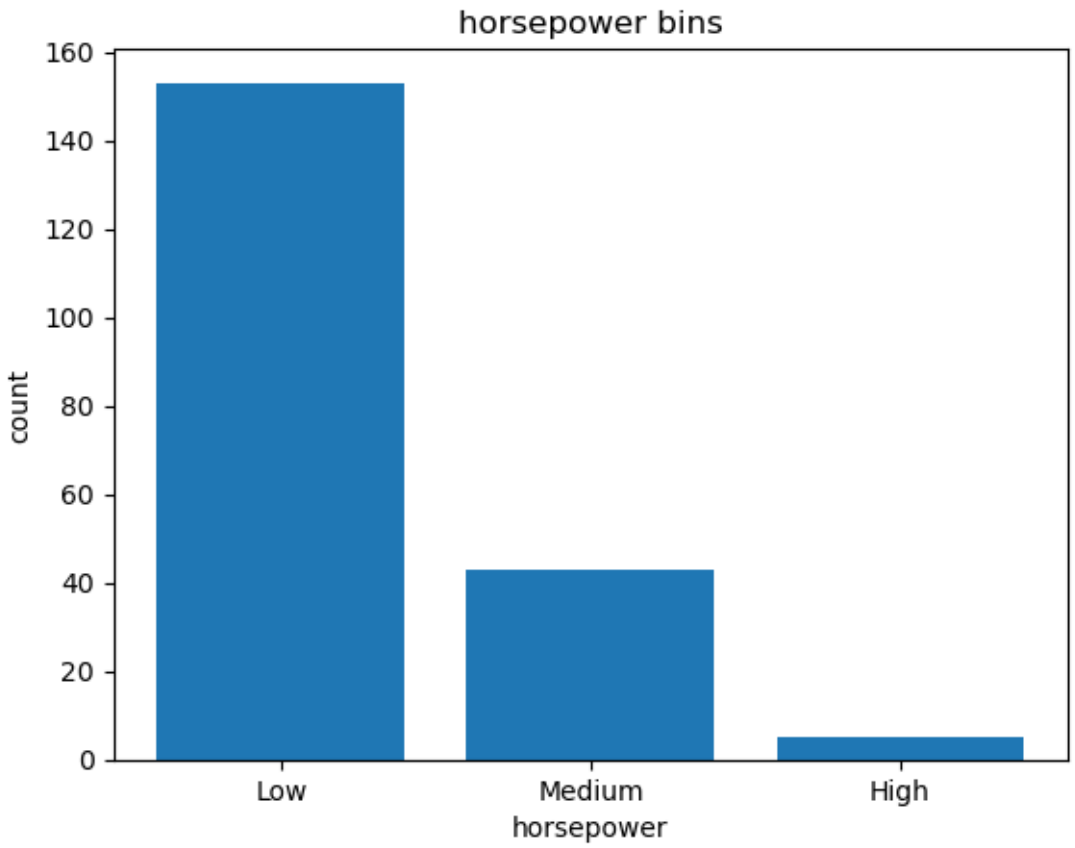
[77]:
```
df["horsepower-binned"].value_counts()
```

[77]:
```
horsepower-binned
Low       153
Medium     43
High        5
Name: count, dtype: int64
```

[103]:
```
%matplotlib inline
import matplotlib as plt
from matplotlib import pyplot
pyplot.bar(group_names, df["horsepower-binned"].value_counts())

# set x/y labels and plot title
plt.pyplot.xlabel("horsepower")
plt.pyplot.ylabel("count")
plt.pyplot.title("horsepower bins")
```

[103]:
```
Text(0.5, 1.0, 'horsepower bins')
```

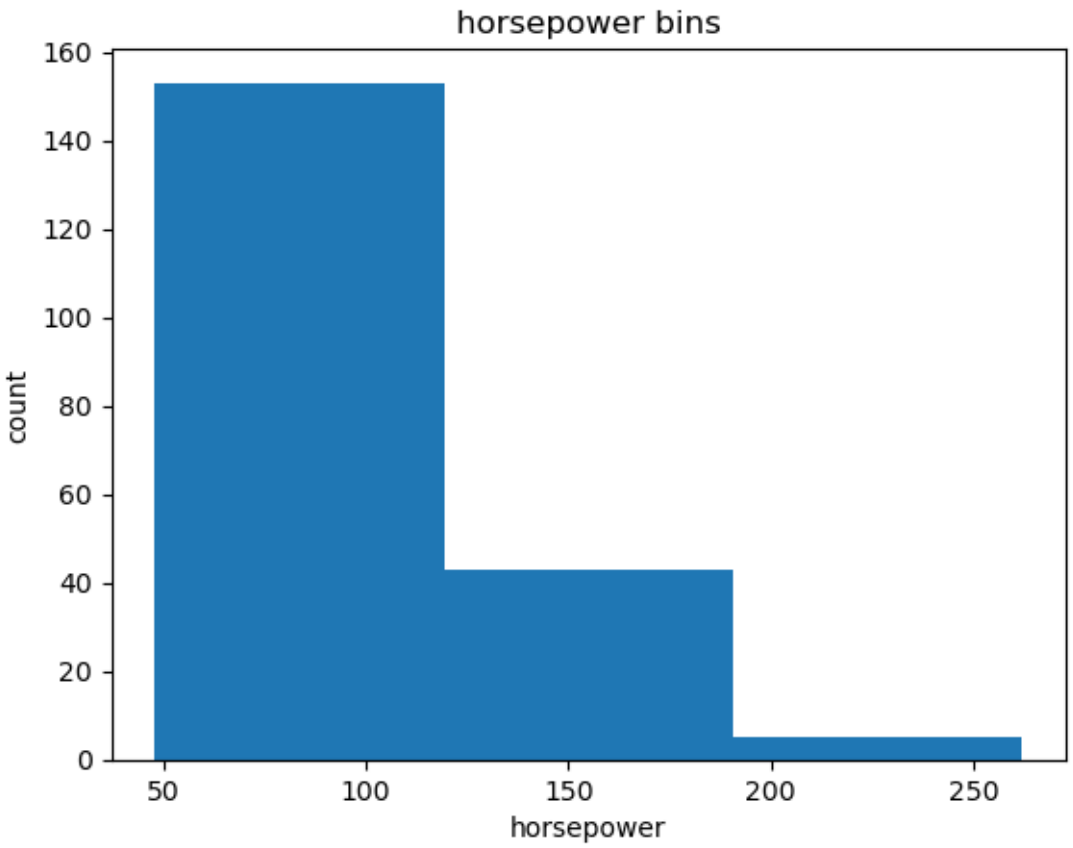


[79]:
```
%matplotlib inline
import matplotlib as plt
from matplotlib import pyplot


# draw historgram of attribute "horsepower" with bins = 3
plt.pyplot.hist(df["horsepower"], bins = 3)

# set x/y labels and plot title
plt.pyplot.xlabel("horsepower")
plt.pyplot.ylabel("count")
plt.pyplot.title("horsepower bins")
```

[79]:
```
Text(0.5, 1.0, 'horsepower bins')
```



[105]:
```
df.columns
```

[105]:
```
Index(['symboling', 'normalized-losses', 'make', 'fuel-type', 'aspiration',
       'num-of-doors', 'body-style', 'drive-wheels', 'engine-location',
       'wheel-base', 'length', 'width', 'height', 'curb-weight', 'engine-type',
       'num-of-cylinders', 'engine-size', 'fuel-system', 'bore', 'stroke',
       'compression-ratio', 'horsepower', 'peak-rpm', 'city-mpg',
       'highway-mpg', 'price', 'horsepower-binned'],
      dtype='object')
```

[81]:
```
dummy_variable_1 = pd.get_dummies(df["fuel-type"])
dummy_variable_1.head()
```

[81]:

	diesel	gas
0	False	True
1	False	True
2	False	True
3	False	True
4	False	True

[83]:
```
dummy_variable_1.rename(columns={'gas':'fuel-type-gas', 'diesel':
 ↪'fuel-type-diesel'}, inplace=True)
dummy_variable_1.head()
```

[83]:

	fuel-type-diesel	fuel-type-gas
0	False	True
1	False	True
2	False	True
3	False	True
4	False	True

[85]:
```
# merge data frame "df" and "dummy_variable_1"
df = pd.concat([df, dummy_variable_1], axis=1)

# drop original column "fuel-type" from "df"
df.drop("fuel-type", axis = 1, inplace=True)
```

[87]:
```
# get indicator variables of aspiration and assign it to data frame␣
 ↪"dummy_variable_2"
dummy_variable_2 = pd.get_dummies(df['aspiration'])

# change column names for clarity
dummy_variable_2.rename(columns={'std':'aspiration-std', 'turbo':␣
 ↪'aspiration-turbo'}, inplace=True)

# show first 5 instances of data frame "dummy_variable_1"
dummy_variable_2.head()
```

[87]:

	aspiration-std	aspiration-turbo
0	True	False
1	True	False
2	True	False
3	True	False
4	True	False

[89]:
```
# merge the new dataframe to the original datafram
df = pd.concat([df, dummy_variable_2], axis=1)

# drop original column "aspiration" from "df"
df.drop('aspiration', axis = 1, inplace=True)
```

[123]:
```
df.to_csv('clean_df.csv')
```

[91]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```