

p2

October 14, 2023

```
[ ]: import pandas as pd
```

```
df = pd.read_csv('./automobileEDA.csv')
df
```

```
[ ]:      symboling  normalized-losses      make aspiration num-of-doors \
0           3           122  alfa-romero      std          two
1           3           122  alfa-romero      std          two
2           1           122  alfa-romero      std          two
3           2           164      audi      std          four
4           2           164      audi      std          four
..      ...      ...      ...      ...      ...
196        -1           95    volvo      std          four
197        -1           95    volvo    turbo          four
198        -1           95    volvo      std          four
199        -1           95    volvo    turbo          four
200        -1           95    volvo    turbo          four

      body-style drive-wheels engine-location  wheel-base  length  ... \
0  convertible      rwd      front      88.6  0.811148  ...
1  convertible      rwd      front      88.6  0.811148  ...
2   hatchback      rwd      front      94.5  0.822681  ...
3      sedan      fwd      front      99.8  0.848630  ...
4      sedan      4wd      front      99.4  0.848630  ...
..      ...      ...      ...      ...      ...
196      sedan      rwd      front     109.1  0.907256  ...
197      sedan      rwd      front     109.1  0.907256  ...
198      sedan      rwd      front     109.1  0.907256  ...
199      sedan      rwd      front     109.1  0.907256  ...
200      sedan      rwd      front     109.1  0.907256  ...

      compression-ratio  horsepower  peak-rpm  city-mpg  highway-mpg  price \
0           9.0      111.0     5000.0      21      27  13495.0
1           9.0      111.0     5000.0      21      27  16500.0
2           9.0      154.0     5000.0      19      26  16500.0
3          10.0      102.0     5500.0      24      30  13950.0
4           8.0      115.0     5500.0      18      22  17450.0
..      ...      ...      ...      ...      ...
```

196	9.5	114.0	5400.0	23	28	16845.0
197	8.7	160.0	5300.0	19	25	19045.0
198	8.8	134.0	5500.0	18	23	21485.0
199	23.0	106.0	4800.0	26	27	22470.0
200	9.5	114.0	5400.0	19	25	22625.0

	city-L/100km	horsepower-binned	diesel	gas
0	11.190476	Medium	0	1
1	11.190476	Medium	0	1
2	12.368421	Medium	0	1
3	9.791667	Medium	0	1
4	13.055556	Medium	0	1
..
196	10.217391	Medium	0	1
197	12.368421	High	0	1
198	13.055556	Medium	0	1
199	9.038462	Medium	1	0
200	12.368421	Medium	0	1

[201 rows x 29 columns]

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

```
[ ]:      symboling  normalized-losses      make aspiration num-of-doors \
0         3         122  alfa-romero      std         two
1         3         122  alfa-romero      std         two
2         1         122  alfa-romero      std         two
3         2         164      audi      std         four
4         2         164      audi      std         four
```

	body-style	drive-wheels	engine-location	wheel-base	length	...	\
0	convertible	rwd	front	88.6	0.811148	...	
1	convertible	rwd	front	88.6	0.811148	...	
2	hatchback	rwd	front	94.5	0.822681	...	
3	sedan	fwd	front	99.8	0.848630	...	
4	sedan	4wd	front	99.4	0.848630	...	

	compression-ratio	horsepower	peak-rpm	city-mpg	highway-mpg	price	\
0	9.0	111.0	5000.0	21	27	13495.0	
1	9.0	111.0	5000.0	21	27	16500.0	
2	9.0	154.0	5000.0	19	26	16500.0	
3	10.0	102.0	5500.0	24	30	13950.0	
4	8.0	115.0	5500.0	18	22	17450.0	

	city-L/100km	horsepower-binned	diesel	gas
0	11.190476	Medium	0	1
1	11.190476	Medium	0	1

2	12.368421	Medium	0	1
3	9.791667	Medium	0	1
4	13.055556	Medium	0	1

[5 rows x 29 columns]

Code Diatas mengambil 5 data teratas dari data frame yang telah dibuat sebelumnya

```
[ ]: df.tail()
```

```
[ ]:      symboling  normalized-losses  make aspiration num-of-doors body-style \
196         -1             95  volvo      std         four      sedan
197         -1             95  volvo    turbo         four      sedan
198         -1             95  volvo      std         four      sedan
199         -1             95  volvo    turbo         four      sedan
200         -1             95  volvo    turbo         four      sedan
```

```
      drive-wheels engine-location  wheel-base  length  ... \
196          rwd         front      109.1  0.907256  ...
197          rwd         front      109.1  0.907256  ...
198          rwd         front      109.1  0.907256  ...
199          rwd         front      109.1  0.907256  ...
200          rwd         front      109.1  0.907256  ...
```

```
      compression-ratio  horsepower  peak-rpm  city-mpg  highway-mpg  price \
196             9.5        114.0    5400.0      23         28  16845.0
197             8.7        160.0    5300.0      19         25  19045.0
198             8.8        134.0    5500.0      18         23  21485.0
199            23.0        106.0    4800.0      26         27  22470.0
200             9.5        114.0    5400.0      19         25  22625.0
```

```
      city-L/100km  horsepower-binned  diesel  gas
196    10.217391         Medium        0     1
197    12.368421          High        0     1
198    13.055556         Medium        0     1
199     9.038462         Medium        1     0
200    12.368421         Medium        0     1
```

[5 rows x 29 columns]

Code Diatas mengambil 5 data terakhir dari data frame yang telah dibuat sebelumnya

```
[ ]: type(df.dtypes), len(df.dtypes), df.dtypes
```

```
[ ]: (pandas.core.series.Series,
29,
symboling          int64
normalized-losses  int64
make              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          float64
peak-rpm           float64
city-mpg            int64
highway-mpg         int64
price              float64
city-L/100km        float64
horsepower-binned   object
diesel             int64
gas                int64
dtype: object)

```

`type(df.dtypes)` untuk mengetahui tipe fungsi dari `df.dtypes`

`len(df.dtypes)` untuk mengetahui jumlah kolom dari `df.dtypes`

`df.dtypes` untuk mengetahui tipe data dari setiap kolom

```

[ ]: df_noid = df.iloc[:,2:]
     df_noid

```

```

[ ]:
      make aspiration num-of-doors  body-style drive-wheels \
0   alfa-romero      std         two convertible         rwd
1   alfa-romero      std         two convertible         rwd
2   alfa-romero      std         two  hatchback         rwd
3       audi      std         four      sedan         fwd
4       audi      std         four      sedan         4wd
..      ...      ...      ...      ...      ...
196  volvo      std         four      sedan         rwd
197  volvo  turbo         four      sedan         rwd
198  volvo      std         four      sedan         rwd
199  volvo  turbo         four      sedan         rwd

```

200	volvo	turbo	four	sedan	rwd	
-----	-------	-------	------	-------	-----	--

	engine-location	wheel-base	length	width	height	...	\
0	front	88.6	0.811148	0.890278	48.8	...	
1	front	88.6	0.811148	0.890278	48.8	...	
2	front	94.5	0.822681	0.909722	52.4	...	
3	front	99.8	0.848630	0.919444	54.3	...	
4	front	99.4	0.848630	0.922222	54.3	...	
..	
196	front	109.1	0.907256	0.956944	55.5	...	
197	front	109.1	0.907256	0.955556	55.5	...	
198	front	109.1	0.907256	0.956944	55.5	...	
199	front	109.1	0.907256	0.956944	55.5	...	
200	front	109.1	0.907256	0.956944	55.5	...	

	compression-ratio	horsepower	peak-rpm	city-mpg	highway-mpg	price	\
0	9.0	111.0	5000.0	21	27	13495.0	
1	9.0	111.0	5000.0	21	27	16500.0	
2	9.0	154.0	5000.0	19	26	16500.0	
3	10.0	102.0	5500.0	24	30	13950.0	
4	8.0	115.0	5500.0	18	22	17450.0	
..	
196	9.5	114.0	5400.0	23	28	16845.0	
197	8.7	160.0	5300.0	19	25	19045.0	
198	8.8	134.0	5500.0	18	23	21485.0	
199	23.0	106.0	4800.0	26	27	22470.0	
200	9.5	114.0	5400.0	19	25	22625.0	

	city-L/100km	horsepower-binned	diesel	gas
0	11.190476	Medium	0	1
1	11.190476	Medium	0	1
2	12.368421	Medium	0	1
3	9.791667	Medium	0	1
4	13.055556	Medium	0	1
..
196	10.217391	Medium	0	1
197	12.368421	High	0	1
198	13.055556	Medium	0	1
199	9.038462	Medium	1	0
200	12.368421	Medium	0	1

[201 rows x 27 columns]

`df.iloc[:,2:]` untuk mengambil semua baris dan kolom ke 2 sampai akhir

```
[ ]: df1 = df_noid.sort_values(by=['horsepower'], ascending=True)
      df1.head(10)
```

```

[ ]:      make aspiration num-of-doors body-style drive-wheels \
17      chevrolet      std          two  hatchback      fwd
178     volkswagen     std          two    sedan        fwd
180     volkswagen     std          four   sedan        fwd
87      nissan          std          two    sedan        fwd
155     toyota         std          four  hatchback     fwd
154     toyota         std          four    sedan       fwd
29      honda          std          two   hatchback     fwd
31      honda          std          two   hatchback     fwd
149     toyota         std          four    wagon       fwd
150     toyota         std          four    wagon       4wd

      engine-location  wheel-base   length    width  height  ... \
17      front          88.4  0.678039  0.837500  53.2  ...
178     front          97.3  0.825084  0.909722  55.7  ...
180     front          97.3  0.825084  0.909722  55.7  ...
87      front          94.5  0.794330  0.886111  54.5  ...
155     front          95.7  0.799135  0.894444  52.8  ...
154     front          95.7  0.799135  0.894444  53.0  ...
29      front          86.6  0.694858  0.887500  50.8  ...
31      front          93.7  0.720807  0.888889  52.6  ...
149     front          95.7  0.815473  0.883333  59.1  ...
150     front          95.7  0.815473  0.883333  59.1  ...

      compression-ratio horsepower peak-rpm  city-mpg highway-mpg  price \
17      9.5          48.0   5100.0      47          53  5151.0
178     23.0         52.0   4800.0      37          46  7775.0
180     23.0         52.0   4800.0      37          46  7995.0
87      21.9         55.0   4800.0      45          50  7099.0
155     22.5         56.0   4500.0      38          47  7788.0
154     22.5         56.0   4500.0      34          36  7898.0
29      9.6          58.0   4800.0      49          54  6479.0
31      10.1         60.0   5500.0      38          42  5399.0
149     9.0          62.0   4800.0      31          37  6918.0
150     9.0          62.0   4800.0      27          32  7898.0

      city-L/100km  horsepower-binned  diesel  gas
17      5.000000          Low          0    1
178     6.351351          Low          1    0
180     6.351351          Low          1    0
87      5.222222          Low          1    0
155     6.184211          Low          1    0
154     6.911765          Low          1    0
29      4.795918          Low          0    1
31      6.184211          Low          0    1
149     7.580645          Low          0    1
150     8.703704          Low          0    1

```

[10 rows x 27 columns]

code diatas untuk mengurutkan data berdasarkan kolom 'horsepower' secara ascending dan mengambil 10 data teratas

```
[ ]: df1 = df_noid.sort_values(by=['horsepower', 'body-style'],
    ↪ascending=[False, True])
df1.head(10)
```

```
[ ]:
      make aspiration num-of-doors  body-style drive-wheels \
46      jaguar      std          two      sedan          rwd
125     porsche      std          two  convertible          rwd
123     porsche      std          two    hardtop          rwd
124     porsche      std          two    hardtop          rwd
102      nissan    turbo          two  hatchback          rwd
71  mercedes-benz      std          two    hardtop          rwd
70  mercedes-benz      std          four    sedan          rwd
14         bmw      std          four    sedan          rwd
15         bmw      std          two    sedan          rwd
16         bmw      std          four    sedan          rwd

      engine-location  wheel-base  length  width  height  ... \
46          front      102.0  0.921192  0.980556  47.8  ...
125          rear      89.5  0.811629  0.902778  51.6  ...
123          rear      89.5  0.811629  0.902778  51.6  ...
124          rear      89.5  0.811629  0.902778  51.6  ...
102          front      91.3  0.820279  0.943056  49.7  ...
71          front     112.0  0.957232  1.000000  55.4  ...
70          front     120.9  1.000000  0.995833  56.7  ...
14          front     103.5  0.908217  0.929167  55.7  ...
15          front     103.5  0.931283  0.943056  53.7  ...
16          front     110.0  0.946660  0.984722  56.3  ...

      compression-ratio  horsepower  peak-rpm  city-mpg  highway-mpg  price \
46             11.5         262.0    5000.0        13         17  36000.0
125             9.5         207.0    5900.0        17         25  37028.0
123             9.5         207.0    5900.0        17         25  32528.0
124             9.5         207.0    5900.0        17         25  34028.0
102             7.8         200.0    5200.0        17         23  19699.0
71              8.0         184.0    4500.0        14         16  45400.0
70              8.0         184.0    4500.0        14         16  40960.0
14              8.0         182.0    5400.0        16         22  30760.0
15              8.0         182.0    5400.0        16         22  41315.0
16              8.0         182.0    5400.0        15         20  36880.0

      city-L/100km  horsepower-binned  diesel  gas
46      18.076923              NaN        0     1
```

125	13.823529	High	0	1
123	13.823529	High	0	1
124	13.823529	High	0	1
102	13.823529	High	0	1
71	16.785714	High	0	1
70	16.785714	High	0	1
14	14.687500	High	0	1
15	14.687500	High	0	1
16	15.666667	High	0	1

[10 rows x 27 columns]

code diatas untuk mengurutkan data berdasarkan kolom 'body-style' secara descending dan mengambil 10 data teratas

```
[ ]: df_noid.describe()
```

```
[ ]:
count    wheel-base    length    width    height    curb-weight  \
mean      98.797015    0.837102    0.915126    53.766667    2555.666667
std        6.066366    0.059213    0.029187     2.447822    517.296727
min       86.600000    0.678039    0.837500    47.800000    1488.000000
25%       94.500000    0.801538    0.890278    52.000000    2169.000000
50%       97.000000    0.832292    0.909722    54.100000    2414.000000
75%      102.400000    0.881788    0.925000    55.500000    2926.000000
max      120.900000    1.000000    1.000000    59.800000    4066.000000

count    engine-size    bore    stroke    compression-ratio    horsepower  \
mean      126.875622    3.330692    3.256904         10.164279    103.405534
std        41.546834    0.268072    0.319256         4.004965     37.365700
min        61.000000    2.540000    2.070000         7.000000     48.000000
25%        98.000000    3.150000    3.110000         8.600000     70.000000
50%       120.000000    3.310000    3.290000         9.000000     95.000000
75%       141.000000    3.580000    3.410000         9.400000    116.000000
max       326.000000    3.940000    4.170000        23.000000    262.000000

count    peak-rpm    city-mpg    highway-mpg    price    city-L/100km  \
mean    5117.665368    25.179104    30.686567    13207.129353     9.944145
std      478.113805     6.423220     6.815150     7947.066342     2.534599
min     4150.000000    13.000000    16.000000    5118.000000     4.795918
25%     4800.000000    19.000000    25.000000    7775.000000     7.833333
50%     5125.369458    24.000000    30.000000    10295.000000     9.791667
75%     5500.000000    30.000000    34.000000    16500.000000    12.368421
max     6600.000000    49.000000    54.000000    45400.000000    18.076923
```

diesel gas

count	201.000000	201.000000
mean	0.099502	0.900498
std	0.300083	0.300083
min	0.000000	0.000000
25%	0.000000	1.000000
50%	0.000000	1.000000
75%	0.000000	1.000000
max	1.000000	1.000000

`df_noid.describe()` untuk mengetahui statistik dari data frame `df_noid`

```
[ ]: df_noid.describe(include='all')
```

```
[ ]:      make aspiration num-of-doors body-style drive-wheels \
count      201         201         201         201         201
unique      22          2          2          5          3
top    toyota      std      four      sedan      fwd
freq        32       165       115        94       118
mean        NaN       NaN       NaN       NaN       NaN
std         NaN       NaN       NaN       NaN       NaN
min         NaN       NaN       NaN       NaN       NaN
25%         NaN       NaN       NaN       NaN       NaN
50%         NaN       NaN       NaN       NaN       NaN
75%         NaN       NaN       NaN       NaN       NaN
max         NaN       NaN       NaN       NaN       NaN
```

	engine-location	wheel-base	length	width	height	...	\
count	201	201.000000	201.000000	201.000000	201.000000	...	
unique	2	NaN	NaN	NaN	NaN	...	
top	front	NaN	NaN	NaN	NaN	...	
freq	198	NaN	NaN	NaN	NaN	...	
mean	NaN	98.797015	0.837102	0.915126	53.766667	...	
std	NaN	6.066366	0.059213	0.029187	2.447822	...	
min	NaN	86.600000	0.678039	0.837500	47.800000	...	
25%	NaN	94.500000	0.801538	0.890278	52.000000	...	
50%	NaN	97.000000	0.832292	0.909722	54.100000	...	
75%	NaN	102.400000	0.881788	0.925000	55.500000	...	
max	NaN	120.900000	1.000000	1.000000	59.800000	...	

	compression-ratio	horsepower	peak-rpm	city-mpg	highway-mpg	...	\
count	201.000000	201.000000	201.000000	201.000000	201.000000	...	
unique	NaN	NaN	NaN	NaN	NaN	...	
top	NaN	NaN	NaN	NaN	NaN	...	
freq	NaN	NaN	NaN	NaN	NaN	...	
mean	10.164279	103.405534	5117.665368	25.179104	30.686567	...	
std	4.004965	37.365700	478.113805	6.423220	6.815150	...	
min	7.000000	48.000000	4150.000000	13.000000	16.000000	...	
25%	8.600000	70.000000	4800.000000	19.000000	25.000000	...	

50%	9.000000	95.000000	5125.369458	24.000000	30.000000
75%	9.400000	116.000000	5500.000000	30.000000	34.000000
max	23.000000	262.000000	6600.000000	49.000000	54.000000

	price	city-L/100km	horsepower-binned	diesel	gas
count	201.000000	201.000000	200	201.000000	201.000000
unique	NaN	NaN	3	NaN	NaN
top	NaN	NaN	Low	NaN	NaN
freq	NaN	NaN	115	NaN	NaN
mean	13207.129353	9.944145	NaN	0.099502	0.900498
std	7947.066342	2.534599	NaN	0.300083	0.300083
min	5118.000000	4.795918	NaN	0.000000	0.000000
25%	7775.000000	7.833333	NaN	0.000000	1.000000
50%	10295.000000	9.791667	NaN	0.000000	1.000000
75%	16500.000000	12.368421	NaN	0.000000	1.000000
max	45400.000000	18.076923	NaN	1.000000	1.000000

[11 rows x 27 columns]

`df_noid.describe(include='all')` untuk mengetahui statistik dari data frame `df_noid` termasuk data kategorikal (object)

perbedaan dari `df.describe()` dan `df.describe(include='all')` adalah `df.describe()` hanya menampilkan statistik dari data numerikal saja sedangkan `df.describe(include='all')` menampilkan statistik dari data numerikal dan kategorikal

`df_noid.mean()` untuk mengetahui rata-rata dari setiap kolom pada data frame `df_noid` (hanya data numerikal)

```
[ ]: testdata = df_noid.describe()
      testdata.mean()
```

```
[ ]: wheel-base      100.907923
      length         25.761246
      width          25.813352
      height         65.801811
      curb-weight    2042.120424
      engine-size    139.427807
      bore           27.639845
      stroke         27.078270
      compression-ratio 34.021156
      horsepower     116.596404
      peak-rpm       3996.518579
      city-mpg       45.950291
      highway-mpg    49.687715
      price          13305.399462
      city-L/100km   33.293126
      diesel         25.299948
```

```
gas                25.775073
dtype: float64
```

untuk mengetahui rata-rata dari kolom 'length','width','height'

```
[ ]: df_noid[['length','width','height']].mean()
```

```
[ ]: length    0.837102
      width    0.915126
      height   53.766667
      dtype: float64
```

untuk mengetahui median dari kolom 'length','width','height','price'

```
[ ]: df_noid[['length','width','height','price']].median()
```

```
[ ]: length    0.832292
      width    0.909722
      height   54.100000
      price    10295.000000
      dtype: float64
```

untuk mengetahui modus dari kolom 'make'

```
[ ]: df_noid[['make']].mode()
```

```
[ ]:      make
      0  toyota
```

mengambil max dari kolom 'price' dari data frame df_noid

```
[ ]: df_noid[['price']].max()
```

```
[ ]: price    45400.0
      dtype: float64
```

untuk mengetahui quartile 3 dari kolom 'price'

```
[ ]: df_noid[['price']].quantile(0.75)
```

```
[ ]: price    16500.0
      Name: 0.75, dtype: float64
```

df_noid[['price']].var() untuk mengetahui variansi dari kolom 'price'

df_noid[['price']].std() untuk mengetahui standar deviasi dari kolom 'price'

```
[ ]: df_noid[['price']].var(), df_noid[['price']].std()
```

```
[ ]: (price    6.315586e+07
      dtype: float64,
      price    7947.066342
```

```
dtype: float64)
```

`df_noid[['make']].value_counts()` untuk mengetahui jumlah data dari setiap kategori pada kolom 'make'

```
[ ]: df_noid[['make']].value_counts()
```

```
[ ]: make
      toyota      32
      nissan      18
      mazda       17
      mitsubishi  13
      honda       13
      volkswagen  12
      subaru       12
      peugot       11
      volvo        11
      dodge        9
      mercedes-benz 8
      bmw          8
      plymouth      7
      audi          6
      saab          6
      porsche       4
      jaguar        3
      chevrolet     3
      alfa-romero   3
      isuzu         2
      renault       2
      mercury       1
      Name: count, dtype: int64
```

untuk mengetahui rata-rata harga dari setiap kategori pada kolom 'make' dan diurutkan secara descending

```
[ ]: df_noid.groupby('make')[['price']].mean().sort_values(by='price',
    ↪ascending=False)
```

```
[ ]:           price
make
jaguar      34600.000000
mercedes-benz 33647.000000
porsche     31400.500000
bmw         26118.750000
volvo       18063.181818
audi        17859.166667
mercury     16503.000000
alfa-romero 15498.333333
peugot      15489.090909
```

```

saab      15223.333333
mazda     10652.882353
nissan     10415.666667
volkswagen 10077.500000
toyota     9885.812500
renault    9595.000000
mitsubishi 9239.769231
isuzu      8916.500000
subaru     8541.250000
honda      8184.692308
plymouth   7963.428571
dodge      7875.444444
chevrolet  6007.000000

```

```
[ ]: testdata.corr()
```

```

[ ]:
      wheel-base   length   width   height   curb-weight  \
wheel-base      1.000000  0.765341  0.765552  0.927062    0.014183
length          0.765341  1.000000  1.000000  0.949543   -0.582144
width           0.765552  1.000000  1.000000  0.949667   -0.582196
height          0.927062  0.949543  0.949667  1.000000   -0.343002
curb-weight     0.014183 -0.582144 -0.582196 -0.343002    1.000000
engine-size     0.619345  0.279265  0.278996  0.434452    0.549092
bore            0.773015  0.999926  0.999929  0.953216   -0.573169
stroke          0.773202  0.999915  0.999916  0.953225   -0.571979
compression-ratio 0.795959  0.996824  0.996805  0.960213   -0.523654
horsepower      0.704200  0.442268  0.441997  0.571158    0.400702
peak-rpm       -0.006893 -0.646461 -0.646276 -0.380106    0.949942
city-mpg        0.846448  0.980887  0.980847  0.973319   -0.416829
highway-mpg     0.861176  0.976338  0.976306  0.977996   -0.394226
price           0.001457 -0.379830 -0.380195 -0.249811    0.832007
city-L/100km    0.796429  0.997811  0.997797  0.961943   -0.528167
diesel          0.763158  0.999983  0.999979  0.948294   -0.583084
gas             0.764825  0.999991  0.999989  0.949271   -0.581817

```

```

      engine-size   bore   stroke   compression-ratio  \
wheel-base      0.619345  0.773015  0.773202    0.795959
length          0.279265  0.999926  0.999915    0.996824
width           0.278996  0.999929  0.999916    0.996805
height          0.434452  0.953216  0.953225    0.960213
curb-weight     0.549092 -0.573169 -0.571979   -0.523654
engine-size     1.000000  0.287009  0.288907    0.353982
bore            0.287009  1.000000  0.999996    0.997363
stroke          0.288907  0.999996  1.000000    0.997502
compression-ratio 0.353982  0.997363  0.997502    1.000000
horsepower      0.983376  0.449262  0.451031    0.511160
peak-rpm        0.348223 -0.637201 -0.636750   -0.601865

```

city-mpg	0.456145	0.982680	0.983012	0.992375
highway-mpg	0.471049	0.978484	0.978846	0.989133
price	0.760621	-0.374210	-0.372313	-0.307267
city-L/100km	0.339274	0.998385	0.998492	0.999430
diesel	0.280474	0.999867	0.999864	0.996926
gas	0.279678	0.999915	0.999913	0.996819

	horsepower	peak-rpm	city-mpg	highway-mpg	price \
wheel-base	0.704200	-0.006893	0.846448	0.861176	0.001457
length	0.442268	-0.646461	0.980887	0.976338	-0.379830
width	0.441997	-0.646276	0.980847	0.976306	-0.380195
height	0.571158	-0.380106	0.973319	0.977996	-0.249811
curb-weight	0.400702	0.949942	-0.416829	-0.394226	0.832007
engine-size	0.983376	0.348223	0.456145	0.471049	0.760621
bore	0.449262	-0.637201	0.982680	0.978484	-0.374210
stroke	0.451031	-0.636750	0.983012	0.978846	-0.372313
compression-ratio	0.511160	-0.601865	0.992375	0.989133	-0.307267
horsepower	1.000000	0.194302	0.603703	0.615845	0.646757
peak-rpm	0.194302	1.000000	-0.511446	-0.486941	0.644306
city-mpg	0.603703	-0.511446	1.000000	0.999466	-0.207219
highway-mpg	0.615845	-0.486941	0.999466	1.000000	-0.195749
price	0.646757	0.644306	-0.207219	-0.195749	1.000000
city-L/100km	0.497820	-0.602950	0.991549	0.988394	-0.324534
diesel	0.443520	-0.648757	0.980893	0.976229	-0.377605
gas	0.442655	-0.646811	0.980977	0.976449	-0.379417

	city-L/100km	diesel	gas
wheel-base	0.796429	0.763158	0.764825
length	0.997811	0.999983	0.999991
width	0.997797	0.999979	0.999989
height	0.961943	0.948294	0.949271
curb-weight	-0.528167	-0.583084	-0.581817
engine-size	0.339274	0.280474	0.279678
bore	0.998385	0.999867	0.999915
stroke	0.998492	0.999864	0.999913
compression-ratio	0.999430	0.996926	0.996819
horsepower	0.497820	0.443520	0.442655
peak-rpm	-0.602950	-0.648757	-0.646811
city-mpg	0.991549	0.980893	0.980977
highway-mpg	0.988394	0.976229	0.976449
price	-0.324534	-0.377605	-0.379417
city-L/100km	1.000000	0.997782	0.997853
diesel	0.997782	1.000000	0.999978
gas	0.997853	0.999978	1.000000

`df_noid.corr()` untuk mengetahui korelasi antar kolom pada data frame `df_noid` (hanya untuk data numerikal)