

# review-introduction

August 16, 2019

<a href="https://cocl.us/corsera\_da0101en\_notebook\_top">  


Thus, we have to add headers manually.

Firstly, we create a list "headers" that include all column names in order. Then, we use `dataframe.columns = headers` to replace the headers by the list we created.

```
[6]: # create headers list
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"]
print("headers\n", headers) #
```

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']
```

We replace headers and recheck our data frame

```
[7]: df.columns = headers
df.head(10)
```

```
[7]: symboling normalized-losses      make fuel-type aspiration num-of-doors \
0      3          ?  alfa-romero    gas      std        two
1      3          ?  alfa-romero    gas      std        two
2      1          ?  alfa-romero    gas      std        two
3      2        164    audi      gas      std        four
4      2        164    audi      gas      std        four
5      2          ?    audi      gas      std        two
6      1        158    audi      gas      std        four
7      1          ?    audi      gas      std        four
8      1        158    audi      gas      turbo       four
9      0          ?    audi      gas      turbo       two

      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
5      sedan      fwd      front      99.8 ...      136
6      sedan      fwd      front     105.8 ...      136
7      wagon      fwd      front     105.8 ...      136
8      sedan      fwd      front     105.8 ...      131
9   hatchback      4wd      front      99.5 ...      131

      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
5	mpfi	3.19	3.40	8.5	110	5500	19
6	mpfi	3.19	3.40	8.5	110	5500	19
7	mpfi	3.19	3.40	8.5	110	5500	19
8	mpfi	3.13	3.40	8.3	140	5500	17
9	mpfi	3.13	3.40	7.0	160	5500	16

	highway-mpg	price
0	27	13495
1	27	16500
2	26	16500
3	30	13950
4	22	17450
5	25	15250
6	25	17710
7	25	18920
8	20	23875
9	22	?

[10 rows x 26 columns]

we can drop missing values along the column “price” as follows

```
[8]: df.dropna(subset=["price"], axis=0)
```

```
[8]:      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]

Now, we have successfully read the raw dataset and add the correct headers into the data frame.

Question #2:

Find the name of the columns of the dataframe

[9]: `# Write your code below and press Shift+Enter to execute`  
`print(df.columns)`

```
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'],
      dtype='object')
```

Double-click here for the solution.