

# DA0101EN-Review-Introduction

July 21, 2019

```
<a href="http://cocl.us/DA0101EN_NotbookLink_Top">  
    
</a>
```

Data Analysis with Python

Introduction

Welcome!

In this section, you will learn how to approach data acquisition in various ways, and obtain necessary insights from a dataset. By the end of this lab, you will successfully load the data into Jupyter Notebook, and gain some fundamental insights via Pandas Library.

Table of Contents

```
<li><a href="#data_acquisition">Data Acquisition</a>  
<li><a href="#basic_insight">Basic Insight of Dataset</a></li>
```

Estimated Time Needed: 10 min

Data Acquisition

There are various formats for a dataset, .csv, .json, .xlsx etc. The dataset can be stored in different places, on your local machine or sometimes online. In this section, you will learn how to load a dataset into our Jupyter Notebook. In our case, the Automobile Dataset is an online source, and it is in CSV (comma separated value) format. Let's use this dataset as an example to practice data reading.

```
<li>data source: <a href="https://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports.csv">https://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports.csv</a>  
<li>data type: csv</li>
```

The Pandas Library is a useful tool that enables us to read various datasets into a data frame; our Jupyter notebook platforms have a built-in Pandas Library so that all we need to do is import Pandas without installing.

```
In [1]: # import pandas library  
import pandas as pd
```

Read Data

We use `pandas.read_csv()` function to read the csv file. In the bracket, we put the file path along with a quotation mark, so that pandas will read the file into a data frame from that address. The file path can be either an URL or your local file address. Because the data does not include headers, we can add an argument `headers = None` inside the `read_csv()` method, so that pandas will not automatically set the first row as a header. You can also assign the dataset to any variable you create.

This dataset was hosted on IBM Cloud object click [HERE](#) for free storage.

```
In [2]: # Import pandas library
import pandas as pd

# Read the online file by the URL provides above, and assign it to variable "df"
other_path = "https://s3-api.us-gso.objectstorage.softlayer.net/cf-courses-data/Cognit
df = pd.read_csv(other_path, header=None)
```

After reading the dataset, we can use the dataframe.head(n) method to check the top n rows of the dataframe; where n is an integer. Contrary to dataframe.head(n), dataframe.tail(n) will show you the bottom n rows of the dataframe.

```
In [3]: # show the first 5 rows using dataframe.head() method
print("The first 5 rows of the dataframe")
df.head(5)
```

The first 5 rows of the dataframe

```
Out[3]:
```

	0	1	2	3	4	5	6	7	8	9	...	\
0	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	...	
1	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	...	
2	1	?	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	...	
3	2	164	audi	gas	std	four	sedan	fwd	front	99.8	...	
4	2	164	audi	gas	std	four	sedan	4wd	front	99.4	...	

  

	16	17	18	19	20	21	22	23	24	25
0	130	mpfi	3.47	2.68	9.0	111	5000	21	27	13495
1	130	mpfi	3.47	2.68	9.0	111	5000	21	27	16500
2	152	mpfi	2.68	3.47	9.0	154	5000	19	26	16500
3	109	mpfi	3.19	3.40	10.0	102	5500	24	30	13950
4	136	mpfi	3.19	3.40	8.0	115	5500	18	22	17450

[5 rows x 26 columns]

Question #1:

check the bottom 10 rows of data frame "df".

```
In [4]: # Write your code below and press Shift+Enter to execute

print("The last 10 rows of the dataframe")
df.tail(10)
```

The last 10 rows of the dataframe

```
Out[4]:
```

	0	1	2	3	4	5	6	7	8	9	...	\
195	-1	74	volvo	gas	std	four	wagon	rwd	front	104.3	...	
196	-2	103	volvo	gas	std	four	sedan	rwd	front	104.3	...	
197	-1	74	volvo	gas	std	four	wagon	rwd	front	104.3	...	

198	-2	103	volvo	gas	turbo	four	sedan	rwd	front	104.3	...
199	-1	74	volvo	gas	turbo	four	wagon	rwd	front	104.3	...
200	-1	95	volvo	gas	std	four	sedan	rwd	front	109.1	...
201	-1	95	volvo	gas	turbo	four	sedan	rwd	front	109.1	...
202	-1	95	volvo	gas	std	four	sedan	rwd	front	109.1	...
203	-1	95	volvo	diesel	turbo	four	sedan	rwd	front	109.1	...
204	-1	95	volvo	gas	turbo	four	sedan	rwd	front	109.1	...

	16	17	18	19	20	21	22	23	24	25
195	141	mpfi	3.78	3.15	9.5	114	5400	23	28	13415
196	141	mpfi	3.78	3.15	9.5	114	5400	24	28	15985
197	141	mpfi	3.78	3.15	9.5	114	5400	24	28	16515
198	130	mpfi	3.62	3.15	7.5	162	5100	17	22	18420
199	130	mpfi	3.62	3.15	7.5	162	5100	17	22	18950
200	141	mpfi	3.78	3.15	9.5	114	5400	23	28	16845
201	141	mpfi	3.78	3.15	8.7	160	5300	19	25	19045
202	173	mpfi	3.58	2.87	8.8	134	5500	18	23	21485
203	145	idi	3.01	3.40	23.0	106	4800	26	27	22470
204	141	mpfi	3.78	3.15	9.5	114	5400	19	25	22625

[10 rows x 26 columns]

Question #1 Answer:

Run the code below for the solution!

Double-click here for the solution.

Add Headers

Take a look at our dataset; pandas automatically set the header by an integer from 0.

To better describe our data we can introduce a header, this information is available at:  
<https://archive.ics.uci.edu/ml/datasets/Automobile>

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.

```
In [5]: # create headers list
```

```
headers = ["symboling", "normalized-losses", "make", "fuel-type", "aspiration", "num-of-doors",
           "drive-wheels", "engine-location", "wheel-base", "length", "width", "height", "curb-weight",
           "num-of-cylinders", "engine-size", "fuel-system", "bore", "stroke", "compression-ratio",
           "peak-rpm", "city-mpg", "highway-mpg", "price"]
print("headers\n", headers)
```

```
headers
```

```
['symboling', 'normalized-losses', 'make', 'fuel-type', 'aspiration', 'num-of-doors', 'body-style',
```

We replace headers and recheck our data frame

```
In [6]: df.columns = headers
```

```
df.head(10)
```

```

Out[6]:      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

```
In [7]: # Drop the missing values in the column
df.dropna(subset=["price"], axis=0)
df.head(20)
```

```
Out[7]:
```

	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	
5	2	?	audi	gas	std	
6	1	158	audi	gas	std	
7	1	?	audi	gas	std	
8	1	158	audi	gas	turbo	
9	0	?	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	?	bmw	gas	std	
15	0	?	bmw	gas	std	
16	0	?	bmw	gas	std	
17	0	?	bmw	gas	std	
18	2	121	chevrolet	gas	std	
19	1	98	chevrolet	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	...

	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	
5	136	mpfi	3.19	3.40	8.5	110	
6	136	mpfi	3.19	3.40	8.5	110	
7	136	mpfi	3.19	3.40	8.5	110	
8	131	mpfi	3.13	3.40	8.3	140	
9	131	mpfi	3.13	3.40	7.0	160	
10	108	mpfi	3.50	2.80	8.8	101	
11	108	mpfi	3.50	2.80	8.8	101	
12	164	mpfi	3.31	3.19	9.0	121	
13	164	mpfi	3.31	3.19	9.0	121	
14	164	mpfi	3.31	3.19	9.0	121	
15	209	mpfi	3.62	3.39	8.0	182	
16	209	mpfi	3.62	3.39	8.0	182	
17	209	mpfi	3.62	3.39	8.0	182	
18	61	2bbl	2.91	3.03	9.5	48	
19	90	2bbl	3.03	3.11	9.6	70	

	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	?
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