review-introduction

August 16, 2019

```
< a href="https://cocl.us/corsera_da0101en_notebook_top"> \\ < img src="https://s3-api.us-geo.objectstorage.softlayer.net/cf-courses-data/CognitiveClass/DA0101EN/Images </a> <math display="block">< /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

Data Acquisition

Basic Insight of Dataset

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.

data source: https://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports-85.data

data type: csv

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.

```
[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.

```
[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-geo.objectstorage.softlayer.net/cf-courses-data/CognitiveClass/
→DA0101EN/auto.csv"

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.

```
[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

```
? alfa-romero gas std
                              two convertible rwd front 88.6 ...
      ? alfa-romero gas std
                              two convertible rwd front 88.6 ...
2
      ? alfa-romero gas std two
                                    hatchback rwd front 94.5 ...
3
  2 164
               audi gas std four
                                      sedan fwd front 99.8 ...
  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 \text{ rows x } 26 \text{ columns}]
```

Question #1:

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

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

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.

```
[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]: \frac{1}{\text{df.columns}} = \text{headers}
\frac{1}{\text{df.head}(10)}
```

[7]:		symboling n	ormalized-	losses	make fuel-type aspiration num-of-doors			
	0	3	?	${\it alfa-romero}$	$_{ m gas}$	std	two	
	1	3	?	${\it alfa-romero}$	$_{ m gas}$	std	two	
	2	1	?	${\it alfa-romero}$	$_{ m gas}$	std	two	
	3	2	164	audi	gas	std	four	
	4	2	164	audi	gas	std	four	
	5	2	?	audi	$_{ m gas}$	std	two	
	6	1	158	audi	$_{ m gas}$	std	four	
	7	1	?	audi	$_{ m gas}$	std	four	
	8	1	158	audi	$_{ m gas}$	turbo	four	
	9	0	?	audi	${ m gas}$	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 \dots
                                                               152
3
       sedan
                     fwd
                                 front
                                            99.8 ...
                                                             109
4
       sedan
                     4 wd
                                 front
                                             99.4 ...
                                                              136
                                            99.8 ...
5
       sedan
                     fwd
                                 front
                                                              136
6
       sedan
                     fwd
                                 front
                                            105.8 ...
                                                              136
7
                                            105.8 ...
       wagon
                     fwd
                                  front
                                                              136
8
       sedan
                     fwd
                                 front
                                            105.8 \dots
                                                              131
    hatchback
                      4 wd
                                              99.5 \dots
                                                               131
                                   front
```

fuel-system bore stroke compression-ratio horsepower peak-rpm city-mpg \setminus 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
                                                                  24
                                                102
                                                        5500
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
                                                                  17
                                                       5500
9
                                      7.0
                                                                  16
        mpfi 3.13
                      3.40
                                               160
                                                       5500
```

highway-mpg price

- $\begin{array}{cccc} 0 & & 27 & 13495 \\ 1 & & 27 & 16500 \end{array}$
- 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

```
df.dropna(subset=["price"], axis=0)
[8]:
          symboling normalized-losses
                                                    make fuel-type aspiration \
     0
                                  ? alfa-romero
                                                                    \operatorname{std}
                 3
                                                         gas
     1
                 3
                                     alfa-romero
                                                         gas
                                                                    \operatorname{std}
     2
                                  ?
                 1
                                     alfa-romero
                                                                    \operatorname{std}
                                                         gas
     3
                 2
                                 164
                                             audi
                                                                   \operatorname{std}
                                                        gas
     4
                 2
                                 164
                                             audi
                                                        gas
                                                                   std
                                             . . .
                                                       . . .
     200
                 -1
                                  95
                                            volvo
                                                                    \operatorname{std}
                                                         gas
     201
                 -1
                                  95
                                            volvo
                                                                  turbo
                                                         gas
     202
                 -1
                                  95
                                            volvo
                                                                    \operatorname{std}
                                                         gas
     203
                 -1
                                  95
                                            volvo
                                                      diesel
                                                                  turbo
     204
                 -1
                                  95
                                            volvo
                                                                  turbo
                                                         gas
         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
                         hatchback
                                              rwd
                                                             front
                                                                          94.5 \dots
                 two
     3
                             sedan
                                                           front
                                                                        99.8 ...
                 four
                                            fwd
                                                                        99.4 ...
     4
                 four
                            sedan
                                             4 wd
                                                           front
                                                                      . . . . . . .
                 . . .
                                                                        109.1 ...
     200
                 four
                             sedan
                                             rwd
                                                            front
                                                                        109.1 ...
     201
                             sedan
                                                            front
                 four
                                             rwd
```

202	four	sedar	1	rwd		front	109.1						
203	four	sedar	1	rwd		front	109.1						
204	four	sedar	1	rwd		front	109.1						
engine-size fuel-system bore stroke compression-ratio horsepower													
0	130	-	3.47	2.68		9.0	111						
1	130	-	3.47	2.68		9.0	111						
2	152	_	2.68			9.0	154						
3	109	${ m mpfi}$	3.19	3.40		10.0	102						
4	136	${ m mpfi}$	3.19	3.40		8.0	115						
200	141	-	i 3.78			9.5	114						
201	141	mpf	3.78	3.15		8.7	160						
202	173	-	3.58			8.8	134						
203	145	idi	3.01	3.40		23.0	106						
204	141	mpf	i 3.78	3.15		9.5	114						
$\mathbf{p}\epsilon$	eak-rpm ci	ty-mpg hi			rice								
0	5000	21	27 1	3495									
1	5000	21	27 1	6500									
2	5000	19	$26 \ 1$	6500									
3	5500	24	30 1	3950									
4	5500	18	$22 \ 1$	7450									
200	5400	23	28	16845									
201	5300	19	25	19045									
202	5500	18	23 2	21485									

[205 rows x 26 columns]

4800

5400

26

19

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

Ouestion #2:

203

204

Find the name of the columns of the dataframe

27 22470

 $25 \ 22625$

[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.