

파이썬 머신러닝 판다스 데이터분석

Lecture (10)



Dr. Heesuk Kim

Part 0. 개발환경 준비

Part 1. 판다스 입문

Part 2. 데이터 입출력

Part 3. 데이터 살펴보기

Part 4. 시각화 도구

Part 5. 데이터 사전처리

Part 6. 데이터프레임의 다양한 응용

Part 7. 머신러닝 데이터 분석



Part 2. 데이터 입출력

1. 외부파일 읽기

1-1. CSV 파일

1-2. Excel 파일

1-3. JSON 파일

2. 웹(web)에서 가져오기

2-1. HTML 웹 페이지에서 표 속성 가져오기

2-2. 웹 스크래핑

3. API 활용하여 데이터 수집하기

4. 데이터 저장하기

4-1. CSV 파일로 저장

4-2. JSON 파일로 저장

4-3. Excel 파일로 저장

4-4. 여러 개의 데이터프레임을 하나의 Excel 파일로 저장



Part 2. 데이터 입출력

1. 외부파일 읽기

• 판다스 데이터 입출력 도구

판다스는 다양한 형태의 외부 파일을 읽어와서 데이터프레임으로 변환하는 함수를 제공한다. 어떤 파일이든 판다스 객체인 데이터프레임으로 변환되고 나면, 판다스의 모든 함수와 기능을 자유롭게 사용할 수 있다.

반대로, 데이터프레임을 다양한 유형의 파일로 저장할 수도 있다. [표 2-1]은 판다스 공식 사이트에서 제공하는 입출력 도구에 관한 자료를 요약한 것이다.

File Format	Reader	Writer
CSV	<code>read_csv</code>	<code>to_csv</code>
JSON	<code>read_json</code>	<code>to_json</code>
HTML	<code>read_html</code>	<code>to_html</code>
Local clipboard	<code>read_clipboard</code>	<code>to_clipboard</code>
MS Excel	<code>read_excel</code>	<code>to_excel</code>
HDF5 Format	<code>read_hdf</code>	<code>to_hdf</code>
SQL	<code>read_sql</code>	<code>to_sql</code>

[표 2-1] 판다스 데이터 입출력 도구(출처: <http://pandas.pydata.org>)



Part 2. 데이터 입출력

1-1. CSV 파일

데이터 값을 쉼표(,)로 구분하고 있다는 의미로 CSV(comma-separated values)라고 부르는 텍스트 파일이다. 쉼표(,)로 열을 구분하고 줄바꿈으로 행을 구분한다.

`read_csv()` 함수에 확장자(.csv)를 포함하여 파일경로(파일명)을 입력하면, CSV 파일을 읽어와서 데이터프레임으로 변환한다.

CSV 파일 → 데이터프레임: `pandas.read_csv("파일 경로 (이름) ")`



Part 2. 데이터 입출력

1-1. CSV 파일

CSV 파일 → 데이터프레임: `pandas.read_csv("파일 경로(이름)")`

〈CSV 파일〉

	0	1	2	3
0	c0	c1	c2	c3
1	0	1	4	7
2	1	2	5	8
3	2	3	6	9

* header 옵션

- '열 이름'이 되는 행을 지정
- `read_csv(file, header=?)`

❶ **header=0** (기본 값: 0행을 열 지정): `df = read_csv(file)`

	0	1	2	3
0	c0	c1	c2	c3
1	0	1	4	7
2	1	2	5	8
3	2	3	6	9

	c0	c1	c2	c3
0	0	1	4	7
1	1	2	5	8
2	2	3	6	9

❷ **header=1** (1행을 열 지정): `df = read_csv(file, header=1)`

	0	1	2	3
0	c0	c1	c2	c3
1	0	1	4	7
2	1	2	5	8
3	2	3	6	9

	0	1	4	7
0	1	2	5	8
1	2	3	6	9

❸ **header=None** (행을 열 지정하지 않음): `df = read_csv(file, header=None)`

	0	1	2	3
0	c0	c1	c2	c3
1	0	1	4	7
2	1	2	5	8
3	2	3	6	9

	0	1	2	3
0	c0	c1	c2	c3
1	0	1	4	7
2	1	2	5	8
3	2	3	6	9

[그림 2-1] CSV 파일 읽기 - header 옵션 비교



Part 2. 데이터 입출력

1-1. CSV 파일

CSV 파일 → 데이터프레임: `pandas.read_csv("파일 경로(이름) ")`

〈CSV 파일〉

	0	1	2	3
0	c0	c1	c2	c3
1	0	1	4	7
2	1	2	5	8
3	2	3	6	9

* index_col 옵션

- '행 주소'가 되는 열을 지정
- `read_csv(file, index_col=?)`

❶ `index_col=False` (인덱스 지정하지 않음)

`: df = read_csv(file, index_col=False)`

	0	1	2	3
0	c0	c1	c2	c3
1	0	1	4	7
2	1	2	5	8
3	2	3	6	9

	c0	c1	c2	c3
0	0	1	4	7
1	1	2	5	8
2	2	3	6	9

❷ `index_col='c0'` ('c0'열을 인덱스 지정)

`: df = read_csv(file, index_col='c0')`

	0	1	2	3
0	c0	c1	c2	c3
1	0	1	4	7
2	1	2	5	8
3	2	3	6	9

	c1	c2	c3
0	1	4	7
1	2	5	8
2	3	6	9

[그림 2-2] CSV 파일 읽기 - index_col 옵션 비교



Part 2. 데이터 입출력

1-1. CSV 파일

CSV 파일 → 데이터프레임: `pandas.read_csv("파일 경로(이름) ")`

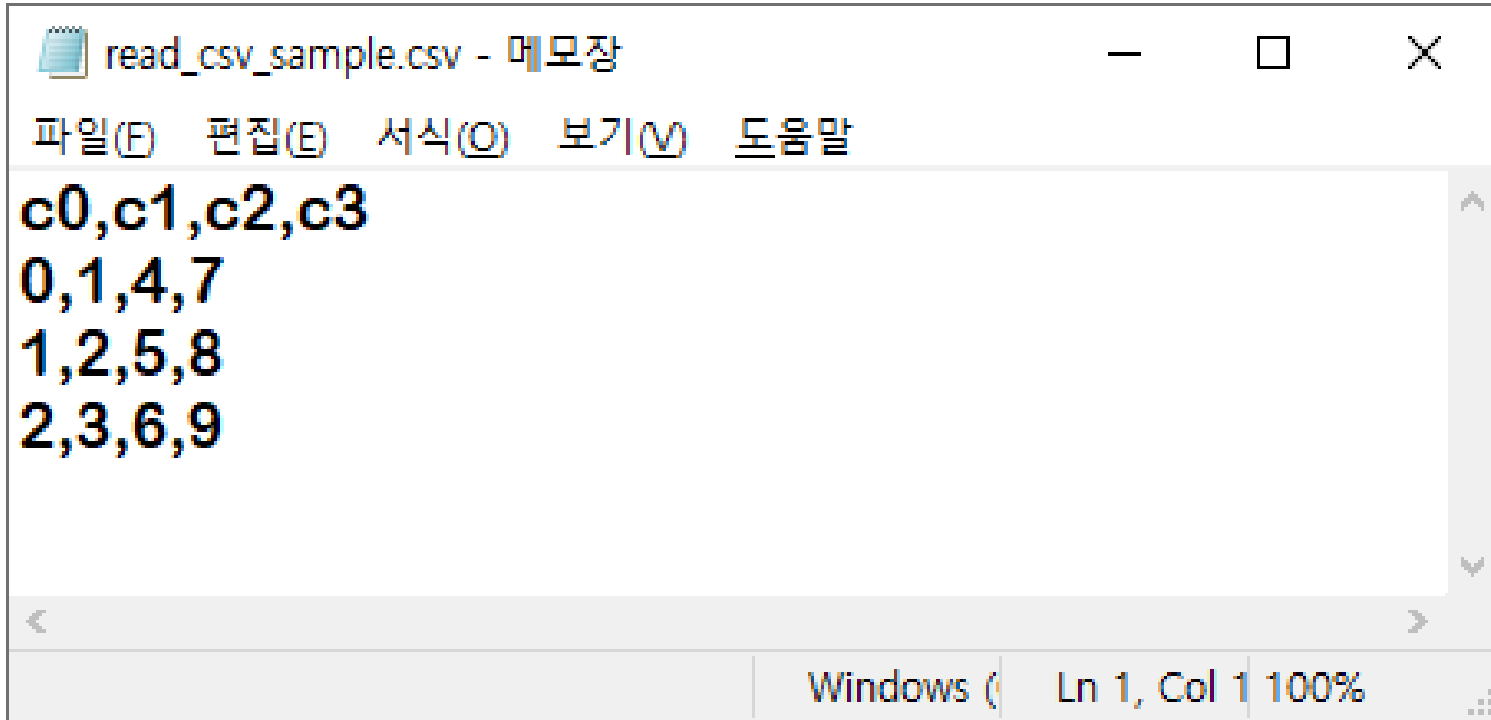
옵션	설명
path	파일의 위치(파일명 포함), URL
sep(또는 delimiter)	텍스트 데이터를 필드별로 구분하는 문자
header	열 이름으로 사용될 행의 번호(기본값은 0) header가 없고 첫 행부터 데이터가 있는 경우 None으로 지정 가능
index_col	행 인덱스로 사용할 열의 번호 또는 열 이름
names	열 이름으로 사용할 문자열의 리스트
skiprows	처음 몇 줄을 skip할 것인지 설정(숫자 입력) skip하려는 행의 번호를 담은 리스트로 설정 가능(예: [1, 3, 5])
parse_dates	날짜 텍스트를 datetime64로 변환할 것인지 설정(기본값은 False)
skip_footer	마지막 몇 줄을 skip할 것인지 설정(숫자 입력)
encoding	텍스트 인코딩 종류를 지정(예: 'utf-8')

[표 2-2] read_csv() 함수의 옵션



Part 2. 데이터 입출력

1-1. CSV 파일



```
read_csv_sample.csv - 메모장
파일(F)  편집(E)  서식(O)  보기(V)  도움말
c0,c1,c2,c3
0,1,4,7
1,2,5,8
2,3,6,9
Windows ( Ln 1, Col 1 100%
```



1-1. CSV 파일

① CSV 파일 미리보기

예제에서 불러올 CSV 파일의 내용을 확인하면, 데이터가 쉼표(,)와 행으로 구분된 것을 확인할 수 있다.

〈CSV 파일〉 미리보기 (File: example/part2/read_csv_sample.csv)

```
1 c0,c1,c2,c3
2 0,1,4,7
3 1,2,5,8
4 2,3,6,9
```

② CSV 파일 읽어오기

〈예제 2-1〉 CSV 파일 읽기 (File: example/part2/2.1_csv_to_df.py)

```
1 #-*- coding: utf-8 -*-
2
3 # 라이브러리 불러오기
4 import pandas as pd
5
6 # 파일 경로(파이션 파일과 같은 폴더)를 찾고, 변수 file_path에 저장
7 file_path = './read_csv_sample.csv'
8
9 # read_csv() 함수로 데이터프레임 변환. 변수 df1에 저장
10 df1 = pd.read_csv(file_path)
11 print(df1)
12 print('\n')
13
14 # read_csv() 함수로 데이터프레임 변환. 변수 df2에 저장. header=None 옵션
15 df2 = pd.read_csv(file_path, header=None)
16 print(df2)
17 print('\n')
18
19 # read_csv() 함수로 데이터프레임 변환. 변수 df3에 저장. index_col=None 옵션
20 df3 = pd.read_csv(file_path, index_col=None)
21 print(df3)
22 print('\n')
23
24 # read_csv() 함수로 데이터프레임 변환. 변수 df4에 저장. index_col='c0' 옵션
25 df4 = pd.read_csv(file_path, index_col='c0')
26 print(df4)
```

〈실행 결과〉 코드 전부 실행

```
c0 c1 c2 c3
0 0 1 4 7
1 1 2 5 8
2 2 3 6 9
```

```
0 1 2 3
0 c0 c1 c2 c3
1 0 1 4 7
2 1 2 5 8
3 2 3 6 9
```

```
c0 c1 c2 c3
0 0 1 4 7
1 1 2 5 8
2 2 3 6 9
```

```
c1 c2 c3
c0
0 1 4 7
1 2 5 8
2 3 6 9
```

header 옵션이 없으면 CSV 파일의 첫 행의 데이터(c0,c1,c2,c3)가 열 이름이 된다.

한편, index_col 옵션을 지정하지 않으면, 행 인덱스는 정수 0, 1, 2가 자동으로 지정된다.

데이터프레임 df4의 경우, index_col='c0' 옵션을 사용하여 'c0' 열이 행 인덱스가 되는 것을 볼 수 있다.



Part 2. 데이터 입출력 [화면 녹화]

The screenshot shows the Spyder Python IDE interface. The main editor displays a script named `2_1_read_csv.py` with the following code:

```
1 # -*- coding: utf-8 -*-
2
3 # 라이브러리 불러오기
4 import pandas as pd
5
6 # 파일경로를 찾고, 변수 file_path에 저장
7 file_path = './read_csv_sample.csv'
8
9 # read_csv() 함수로 데이터프레임 변환. 변수 df1에 저장
10 df1 = pd.read_csv(file_path)
11 print(df1)
12 print('\n')
13
14 # read_csv() 함수로 데이터프레임 변환. 변수 df2에 저장. header=None 옵션
15 df2 = pd.read_csv(file_path, header=None)
16 print(df2)
17 print('\n')
18
19 # read_csv() 함수로 데이터프레임 변환. 변수 df3에 저장. index_col=None 옵션
20 df3 = pd.read_csv(file_path, index_col=None)
21 print(df3)
22 print('\n')
23
24 # read_csv() 함수로 데이터프레임 변환. 변수 df4에 저장. index_col='c0' 옵션
25 df4 = pd.read_csv(file_path, index_col='c0')
26 print(df4)
```

The Variable explorer on the right is empty. The IPython console at the bottom shows the following output:

```
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.6.1 -- An enhanced Interactive Python.

In [1]:
```

The status bar at the bottom indicates: Permissions: RW, End-of-lines: CRLF, Encoding: UTF-8, Line: 3, Column: 13, Memory: 76 %.



Part 2. 데이터 입출력 [화면 녹화]

Spyder (Python 3.7)

File Edit Search Source Run Debug Consoles Projects Tools View Help

untitled0.py 2_1_read_csv.py

```
1 #-*- coding: utf-8 -*-
2
3 # 라이브러리 불러오기
4 import pandas as pd
5
6 # 파일경로를 찾고, 변수 file_path에 저장
7 file_path = './read_csv_sample.csv'
8
9 # read_csv() 함수로 데이터프레임 변환. 변수 df1에 저장
10 df1 = pd.read_csv(file_path)
11 print(df1)
12 print('\n')
13
14 # read_csv() 함수로 데이터프레임 변환. 변수 df2에 저장. header=None 옵션
15 df2 = pd.read_csv(file_path, header=None)
16 print(df2)
17 print('\n')
18
19 # read_csv() 함수로 데이터프레임 변환. 변수 df3에 저장. index_col=None 옵션
20 df3 = pd.read_csv(file_path, index_col=None)
21 print(df3)
22 print('\n')
23
24 # read_csv() 함수로 데이터프레임 변환. 변수 df4에 저장. index_col='c0' 옵션
25 df4 = pd.read_csv(file_path, index_col='c0')
26 print(df4)
```

Variable explorer

Name	Type	Size	Value
df1	DataFrame	(3, 4)	Column names: c0, c1, c2, c3
df2	DataFrame	(4, 4)	Column names: 0, 1, 2, 3
df3	DataFrame	(3, 4)	Column names: c0, c1, c2, c3
df4	DataFrame	(3, 3)	Column names: c1, c2, c3
file_path	str	1	./read_csv_sample.csv

IPython console

Console 6/A

```
0 0 1 2 3
0 c0 c1 c2 c3
1 0 1 4 7
2 1 2 5 8
3 2 3 6 9

c0 c1 c2 c3
0 0 1 4 7
1 1 2 5 8
2 2 3 6 9

c1 c2 c3
c0
0 1 4 7
1 2 5 8
2 3 6 9

In [2]:
```

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 26 Column: 11 Memory: 77 %



Any Question?

Thank you.

