

## [Python] 파이썬 기초 14 - 아주 기초적인 pandas 사용법과 예제 — 나무늘보의 개발 블로그

노트북: 첫 번째 노트북

만든 날짜: 2020-10-24 오후 7:20

URL: <https://continuous-development.tistory.com/74?category=736681>

Python

## [Python] 파이썬 기초 14 - 아주 기초적인 pandas 사용법과 예제

2020. 8. 19. 22:28 수정 삭제 공개

read.csv를 하는 데 있어서 pd.read.csv로 읽는다. 이처럼 읽을 경우 data는 데이터 프레임 형태로 만들어진다.

```
csv_excel.py x parsers.py x service_bmi.csv x
1
2 # csv, excel input / output
3 import pandas as pd
4
5 # print(data.info())
6 # print(data.head())
7 # print(data.tail())
8
9 def load_csv():
10     data = pd.read_csv("../word/service_bmi.csv", encoding='UTF-8')
11     service_bmi(data)
12
13 def service_bmi(data):
14     print(data.head())
```

```
csv_excel_caller.py x
1
2 from service.file.csv_excel import *
3
4 load_csv()
```

```
C:\Users\i\Anaconda3\python.exe C:/Users/i/PycharmProjects/python_base/csv_excel_caller.py
   height  weight  label
0     184     61   thin
1     189     56   thin
2     183     79  normal
3     143     40  normal
4     187     66  normal

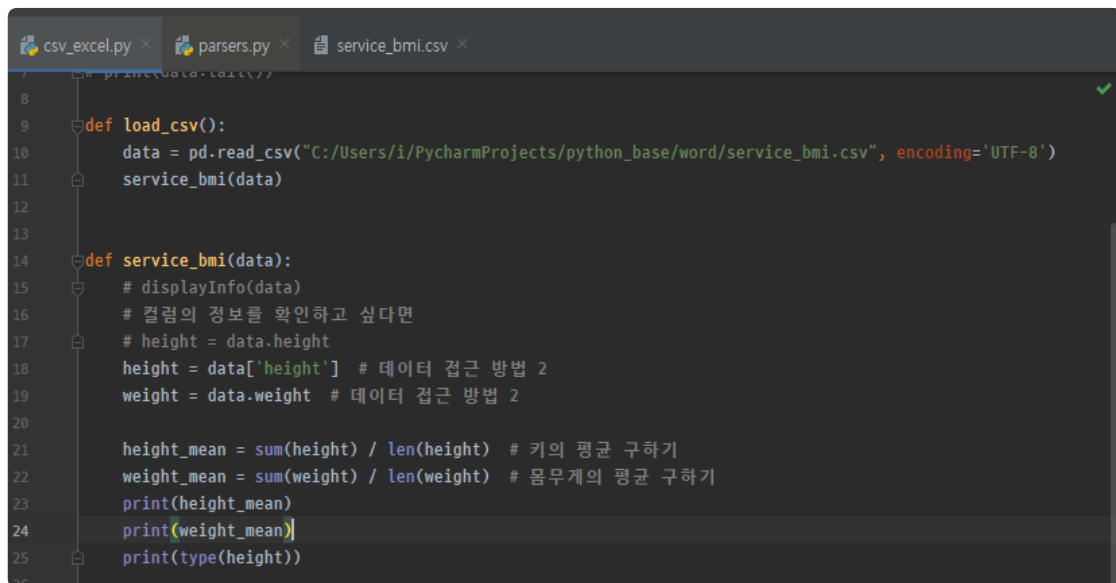
Process finished with exit code 0
```

이렇게 데이터 프레임형태로 생성이 된다. pandas에서는 데이터 프레임 타입과 series 타입을 제공해준다. 데이터 프레임은 위와 같이 행 과열을 가진 형태고 series는 R의 벡터와 같은 개념이다. 하나하나의 값들을 series라고 한다.

```
csv_excel.py x parsers.py x service_bmi.csv x
1 # csv, excel input / output
2 import pandas as pd
3
4
5 # print(data.info())
6 # print(data.head())
7 # print(data.tail())
8
9 def load_csv():
10     data = pd.read_csv("C:/Users/i/PycharmProjects/python_base/word/service_bmi.csv", encoding='UTF-8')
11     service_bmi(data)
12
13
14 def service_bmi(data):
15     # displayInfo(data)
16     # 컬럼의 정보를 확인하고 싶다면
17     height = data.height
18     print(height)
19     print(type(height))
20
21
22 def displayInfo(data):
23     print("head")
24     print(data.head())
```

데이터 프레임의 값 접근법은 2가지가 있다. 하나는 컬럼명을 붙여주는 형태이다. 위에서는 data.height라는 값을 height에 넣어줬다. 이 height를 출력하면 아래와 같이 해당 height의 값들을 나타낸다. 이 값들의 타입은 Series이다.

```
19994    188
19995    168
19996    190
19997    179
19998    148
19999    167
Name: height, Length: 20000, dtype: int64
<class 'pandas.core.series.Series'>
|
Process finished with exit code 0
```



```
7 print(data.tail())
8
9 def load_csv():
10     data = pd.read_csv("C:/Users/i/PycharmProjects/python_base/word/service_bmi.csv", encoding='UTF-8')
11     service_bmi(data)
12
13
14 def service_bmi(data):
15     # displayInfo(data)
16     # 컬럼의 정보를 확인하고 싶다면
17     # height = data.height
18     height = data['height'] # 데이터 접근 방법 2
19     weight = data.weight # 데이터 접근 방법 2
20
21     height_mean = sum(height) / len(height) # 키의 평균 구하기
22     weight_mean = sum(weight) / len(weight) # 몸무게의 평균 구하기
23     print(height_mean)
24     print(weight_mean)
25     print(type(height))
```

두 번째 접근 방법은 data ['컬럼명']을 통해 접근하는 방법이다. 이 두 가지 방법 다 같은 결과를 낸다.

여기서 가져온 값을 통해 평균을 구하고 있다.

```
csv_excel.py x parsers.py x service_bmi.csv x
1  print(data.tail())
2
3
4
5
6
7
8
9  def load_csv():
10     data = pd.read_csv("C:/Users/i/PycharmProjects/python_base/word/service_bmi.csv", encoding='UTF-8')
11     service_bmi(data)
12
13
14  def service_bmi(data):
15     # displayInfo(data)
16     # 컬럼의 정보를 확인하고 싶다면
17     # height = data.height
18     height = data['height'] # 데이터 접근 방법 2
19     weight = data.weight # 데이터 접근 방법 2
20
21     height_mean = sum(height) / len(height) # 키의 평균 구하기
22     weight_mean = sum(weight) / len(weight) # 몸무게의 평균 구하기
23     print("키의 평균 : ", height_mean)
24     print("몸무게의 평균 : ", weight_mean)
25     print("가장 큰 키 : ", max(height))
26     print("가장 작은 키 : ", min(height))
27
28     print(type(height))
29
```

max나 sum , len 등 다양한 함수들을 사용 할 수 있다.

```
C:\Users\i\Anaconda3\python.exe C:/Users/i/PycharmProjects/python_base/csv_excel_caller.py
키의 평균 : 164.9379
몸무게의 평균 : 62.40995
가장 큰 키 : 190
가장 작은 키 : 140
<class 'pandas.core.series.Series'>

Process finished with exit code 0
```

이와 같이 결과가 나온다.

## # 라벨컬럼을 활용하여 각 단어의 빈도수를 출력하는 로직

```
csv_excel.py x dict.py x typing.py x _parser.py x parsers.py x
26  print("가장 큰 키 : ", max(height))
27  print("가장 작은 키 : ", min(height))
28
29  # 라벨 컬럼을 활용하여 빈도수를 출력하는 로직을 만들어 보자
30  print(data.head())
31  label_dict = {}
32  for i in label:
33     count = label_dict.get(i, 0) # get 으로 value 를 가져오는데 뒤에 0 은 기본값을 0으로 잡는 것이다.
34     label_dict[i] = count + 1
35  print(label_dict)
36
37  labelFreq = {}
38  for key in data.label: # 딕셔너리는 key : value 형태이다 아래와 같이 넣으면 이형태로 들어가는데
39     labelFreq[key] = labelFreq.get(key, 0) + 1 # labelFreq에서 key가 가지고 있는 value를 가져온다음 1을 추가하는
40     print(labelFreq) # 로직이다.
41
42  print(type(height))
43  test = {}
44  test['바보야'] = 30
45  print(test)
46
```

맨 처음 data.head()를 통해 데이터의 형태를 확인한다. 그 후 label\_dict = {}이라는 dir 타입의 변수를 만들어준다.

그다음 for 문을 돌려 label에 있는 값을 하나씩 빼낸다.

밑에 로직으로 설명하자면 key 값은 data.labe의 한줄 한 줄을 나타낸다.

그 후 labelFreq 이라는 딕셔너리 타입에 labelFreq[key] = labelFreq.get(key,0) + 1라는 형식으로 넣는다.

labelFreq[key] 는 thin , normal, fat 등등의 값을 가진 키값의 value를 정의하는 부분이다.

labelFreq.get(key,0) + 1 이 부분은 지금 labelFreq.get에서 get 이라는 함수를 통해 labelFreq의 value의 값을 가져온다. 그때 그 value의 키는 (key, 0)에 들어가는 key 값이 되고 0은 해당 값이 없을 시 0으로 초기화한다는 뜻이다.

결론은 labelFreq의 key에 따른 value 값을 가져와서 1씩 더하면서 카운트하고 그 값을 해당 key 값인 labelFreq[key]에 넣어준다. 결과값은 아래와 같다.

```
4      187      66 normal
{'thin': 4898, 'normal': 7677, 'fat': 7425}
{'thin': 4898, 'normal': 7677, 'fat': 7425}
<class 'pandas.core.series.Series'>
{'바보야': 30}
```

## 평균 간단하게 구하기

```
csv_excel.py x dict.py x typing.pyi x _par
1      # csv, excel input / output
2      import pandas as pd
3      from statistics import mean
```

```
def load_xls():
    kospi = pd.ExcelFile('./word/sam_kospi.xlsx')
    # print(kospi) # <pandas.io.excel.ExcelFile object at 0x000001F10B0FA518> (object 형태)
    kospi = kospi.parse("sam_kospi") # 지정된 시트를 지정된 시트를 DataFrame 으로 구문 분석.
    # print(kospi.info())
    print(kospi.head())
    print('High - ', mean(kospi.High)) # High - 1307947.3684210526
    print('Low - ', mean(kospi.Low)) # Low - 1280919.028340081

# def kospi_info():
```

# pandas와 numpy는 추후에 다시 제대로 정리할 예정입니다.

### 'Python' 카테고리의 다른 글

[python] 영화 리뷰에 대한 자연어 처리분석/ 감성분석하기 feat. 스크래핑

[python] BeautifulSoup를 통한 영화리뷰 scraping 하기

**[Python] 파이썬 기초 14 - 아주 기초적인 pandas 사용법과 예제**

[Python] 파이썬 기초 13 - 파이썬을 통한 파일 입출력 사용법

[Python] 파이썬 기초 12 - 예외처리

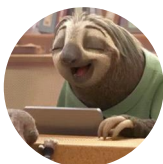
[Python] 파이썬 기초 11 - 객체의 4대 특성 ( 상속화, 캡슐화, 다형성, 추상화)

Python pandas

python 판다스

파이썬 pandas

파이썬 판다스



나무늘보스

혼자 끄적끄적하는 블로그 입니다.