Chapter. 03

재료 준비하기: 데이터 불러오기

I 개요

FAST CAMPUS
ONLINE
데이터 탐색과 전처리 I

강사. 안길승

I 경로 확인하기 및 설정하기

- 파이썬에서 데이터를 불러오려면 반드시 **경로와 확장자까지 포함**시켜서 불러와야 함

A > A-1 > A-1-2
Data.csv

pd.read_csv("Data.csv") (X) # 경로 미포함

pd.read_csv("A/A-1/A-1-2/Data") (X) # 확장자 미포함

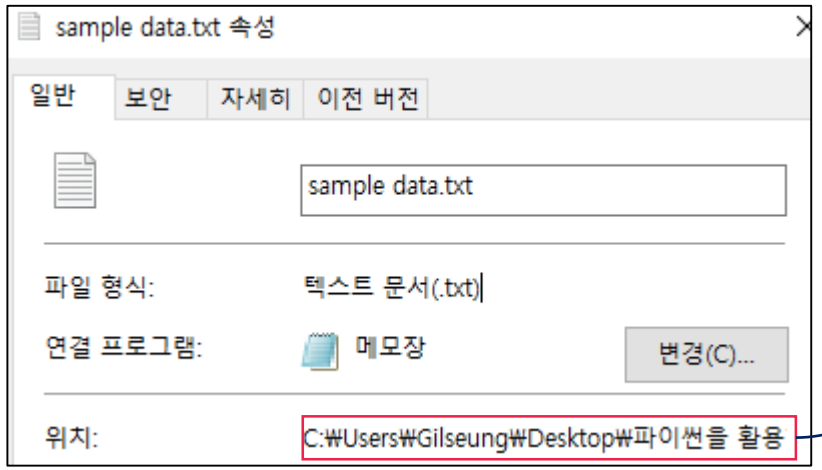
pd.read_csv("A/A-1/A-1-2/Data.csv") (O)

- 경로를 설정함으로써 파일을 불러올 때마다 경로를 포함시키는 번거로움을 해결함

- os 모듈에 속한 함수를 사용하면 경로를 확인하고 설정할 수 있음
  - ➢ os.getcwd(): 현재 경로를 반환
  - ➢ os.chdir(path): 현재 경로를 path로 설정

I 경로 설정 Tip

- 데이터 속성 – 위치 혹은 주소창에 있는 경로를 복사해서 붙여넣어 손쉽게 경로를 설정할 수 있음



"C:\Users\Gilseung\Desktop\파이썬을 활용한 데이터 전처리 Level UP\실습 데이터\1. 데이터 핸들링"

- 단, 역슬래시(\)는 파이썬을 포함한 대부분의 프로그래밍 언어에서 중요한 기능을 하므로 (예: \n: 줄바꿈), 특별한 표현이 아님을 나타내야 함
  - 방법 1. 역슬래시를 두 번 쓰는 방법: "C:\\Users\\Gilseung\\...\\1. 데이터 핸들링"
  - 방법 2. 역슬래시를 슬래시로 바꾸는 방법: "C:/Users/Gilseung/.../1. 데이터 핸들링"
  - 방법 3. r을 사용하여 raw string임을 밝히는 방법: r"C:\Users\Gilseung\...\1. 데이터 핸들링"

Chapter. 03

재료 준비하기: 데이터 불러오기

txt, csv, tsv 데이터 불러오기

FAST CAMPUS
ONLINE
데이터 탐색과 전처리 I

강사. 안길승

I open 함수를 이용한 파일 불러오기

- 파이썬의 내장 함수인 open 함수를 사용하면 파일을 손쉽게 불러올 수 있음

  파일 객체 = open(파일 경로 및 이름, 모드)

- 정제되지 않은 형태의 데이터를 불러오는 경우에 주로 사용함

- 모드에는 크게 r (default), w, a가 있음
  - r: 읽기 (기존 파일을 읽어 옴)
  - w: 쓰기 (새로운 파일을 생성하여 씀. 같은 파일이 있으면 새로운 내용으로 덮어 씀)
  - a: 추가하기 (기존 파일에 새로운 내용을 씀)

- 파일 객체는 사용 후에 close 함수를 사용하여 닫아줘야 함 (참고. with 구문)

I read와 readline을 이용하여 데이터 불러오기

- f.read(): 파일 f에 있는 모든 내용을 불러옴 (f는 반드시 'r'이나 'rb'로 불러와야 함)
- f.readline(): 파일 f에 있는 한 줄(₩n 기준 및 포함)을 불러옴 (f는 반드시 'r'이나 'rb'로 불러와야 함)
- read 및 readline의 결과물은 문자열이므로 문자열 관련 함수를 숙지해야 함
  - ➢ str.split(sep): str을 sep을 기준으로 분할하여 리스트로 변환 (예: 'a-b-c'.split('-') = ['a', 'b', 'c'])
  - ➢ map(함수, L): iterable한 객체 L에 함수를 일괄 적용

I write 함수를 사용하여 내용 쓰기

- f.write(string): string을 파일 f에 씀 (f는 반드시 'w'이나 'a'로 불러와야 함)
- 리스트 등을 string으로 변환하는 join 함수를 활용하면 효율적으로 파일을 쓸 수 있음
  - sep.join(list): list의 문자열 요소들을 sep으로 연결
  - 예시: '-'.join(['a', 'b', 'c']) = 'a-b-c'

I read_csv 함수를 이용한 데이터 불러오기

- Pandas의 read_csv 함수는 테이블 형태의 데이터를 불러오는데 효과적인 함수임

```
pd.read_csv(filepath, sep, header, index_col, usecols, parse_dates, nrows)
```

- filepath: 파일 경로 및 이름
- sep: 구분자 (default: ‘,’)
- header: 헤더의 위치로 None을 입력하면 컬럼명이 0, 1, 2, …로 자동 부여됨 (default: ‘infer’)
- index_col: 인덱스의 위치 (default: None)
- usecols: 사용할 컬럼 목록 및 위치 목록 (데이터가 큰 경우에 주로 사용)
- nrows: 불러올 행의 개수 (데이터가 큰 경우에 주로 사용)

I to_csv 함수를 이용한 데이터 저장하기

- Pandas의 to_csv 함수는 테이블 형태의 데이터를 저장하는데 효과적인 함수임

```
df.to_csv(filepath, sep, index)
```

- filepath: 파일 경로 및 이름
- sep: 구분자 (default: ‘,’)
- index: 인덱스를 저장할지 여부

Chapter. 03

재료 준비하기: 데이터 불러오기

Excel 데이터 불러오기

FAST CAMPUS
ONLINE
데이터 탐색과 전처리 I

강사. 안길승

I read_excel 함수를 이용한 데이터 불러오기

- Pandas의 read_excel 함수는 .xlsx 포맷의 데이터를 불러오는데 효과적인 함수임

```
pd.read_excel(filepath, sheet_name, header, index_col, usecols, parse_dates, nrows)
```

- filepath: 파일 경로 및 이름
- sheet_name: 불러오고자 하는 시트 이름 및 위치
- header: 헤더의 위치로 None을 입력하면 컬럼명이 0, 1, 2, …로 자동 부여됨 (default: 'infer')
- index_col: 인덱스의 위치 (default: None)
- usecols: 사용할 컬럼 목록 및 위치 목록 (데이터가 큰 경우에 주로 사용)
- nrows: 불러올 행의 개수 (데이터가 큰 경우에 주로 사용)
- skiprows: 불러오지 않을 행의 위치 (리스트)

I to_excel 함수를 이용한 데이터 저장하기

- Pandas의 to_excel 함수는 테이블 형태의 데이터를 저장하는데 효과적인 함수임

```
df.to_excel(filepath, index, sheet_name, mode)
```

  - filepath: 파일 경로 및 이름
  - index: 인덱스를 저장할지 여부
  - sheet_name: 시트 명

- 여러 시트를 생성해야 하는 경우에는 ExcelWriter를 사용함

```
writer = pd.ExcelWriter(xlsx file)
df1.to_excel(writer, sheet_name = “sheet1”)
df2.to_excel(writer, sheet_name = “sheet2”)
```