

데이터 랭글링 과정

다음주 수요일 잠실경기장 야구 관중수는
어느 정도될까?

➤ **데이터 분석은 문제의식에서 출발**

- ✓ 이 문제를 해결하기 위해 필요한 데이터는?
- ✓ 단순 과거 데이터가 아니라 관중수와 상관관계가 있는 데이터에 대한 연구부터 시작해야 한다.
- ✓ 예를 들면, 날씨와의 상관관계 등 -> EDA

➤ **분석 전에 양질의 데이터 확보 필요**

- ✓ 데이터 랭글링을 위해 단순 질문에서 시작해보자.

서울이 가장 더웠던 날은 언제였을까?

➤ 이 문제를 해결하기 위해 필요한 데이터는?

- ✓ 우선, 좀 더 구체적이고 명확한 질문이 필요!
- ✓ 언제부터 – 대한제국? 해방전? -> 기상관측 이래
- ✓ 기온, 습도, 풍속을 모두 고려할 것인가? -> 기온

➤ 질문 다듬기

- ✓ 기상 관측 이래, 서울의 최고 **기온**이 가장 높았던 날은 언제였고, 몇 도였을까?

기온데이터는 어떻게 얻을 수 있을까?

➤ 데이터 수집의 문제

- ✓ 데이터 수집 방법 -> 누군가에게 전화를 해서?
- ✓ 데이터의 형식 -> 수기 자료도 가능?
- ✓ 파이썬으로 처리하기 위해서는 디지털 자료로



① 데이터 수집

➤ 데이터 수집 방법

- ✓ 파일 다운로드가 가능한가?
- ✓ API로 제공되는가?
: 구글 API 등
- ✓ 웹스크래핑 등



➤ 데이터의 형식

- ✓ 기계가 읽을 수 있는 데이터인가?
- ✓ csv파일 등

기온 공공데이터

➤ 기상자료개방포털

- ✓ <https://data.kma.go.kr/>
- ✓ 기후통계분석 / 통계분석 / 기온분석
- ✓ 1904년 데이터부터 검색 가능

➤ 실습1:

- ✓ 기온 그래프를 관찰하기
- ✓ csv다운로드
- ✓ 코드로 csv파일 읽고 내용 출력하기

csv reading code

■ csv 읽고 출력하기

```
import csv
f = open('seoul.csv')           # f : file handler
data = csv.reader(f)           # data : csv reader object
for row in data:                #      -> iterate over lines
    print( row )
f.close()
```

■ Python Standard Library

- <https://docs.python.org/3/library/index.html>

csv.reader()

CSV — CSV File Reading and Writing

Source code: [Lib/csv.py](#)

The so-called **CSV** (Comma Separated Values) format is the most common import and export format for spreadsheets and databases. **CSV** format was used for many years prior to attempts to describe the format in a standardized way in [RFC 4180](#). The lack of a well-defined standard means that subtle differences often exist in the data produced and consumed by different applications. These differences can make it annoying to process CSV files from multiple sources. Still, while the delimiters and quoting characters vary, the overall format is similar enough that it is possible to write a single module which can efficiently manipulate such data, hiding the details of reading and writing the data from the programmer.

The **csv** module implements classes to read and write tabular data in **CSV** format. It allows programmers to say, “write this data in the format preferred by Excel,” or “read data from this file which was generated by Excel,” without knowing the precise details of the CSV format used by Excel. Programmers can also describe the CSV formats understood by other applications or define their own special-purpose CSV formats.

The **csv** module’s **reader** and **writer** objects read and write sequences. Programmers can also read and write data in dictionary form using the **DictReader** and **DictWriter** classes.

See also:

[PEP 305](#) - CSV File API

The Python Enhancement Proposal which proposed this addition to Python.

Module Contents

The **csv** module defines the following functions:

csv.reader(csvfile, dialect='excel', **fmtparams)

Return a **reader object** which will iterate over lines in the given **csvfile**. **csvfile** can be any object which supports the **iterator** protocol and returns a string each time its `__next__()` method is called — **file objects** and list objects are both suitable. If **csvfile** is a file object, it should be opened with `newline=''`. [1] An optional **dialect** parameter can be given which is used to define a set of parameters specific to a particular **CSV** dialect. It may be an instance of a subclass of the **Dialect** class or one of the strings returned by the

② 데이터 전처리

➤ 데이터 전처리

- ✓ 데이터 클리닝
- ✓ 데이터 통합 (Integration) – 표준화 등
- ✓ 데이터 변환 (Transformation) – 정규화 등

➤ 데이터 클리닝

- ✓ 결측값 처리
- ✓ 중복값 처리
- ✓ 이상치 확인 및 제거

데이터 전처리 실습

■ Header 전처리 실습

- 불필요한 헤더 정리
- 헤더는 연산과정에 필요 없으므로 그 다음 줄부터 연산

`header = next( data )` *# retrieve the next item from the iterator*

■ 데이터 변환 실습

- 최고 기온의 데이터타입이 문자열이므로 비교 연산을 할 수 없음
- 문자열 타입 -> 실수 타입으로 변환
- 파이썬 내장함수 : `float( )`

Built-in Functions (파이썬)

Built-in Functions

The Python interpreter has a number of functions and types built into it that are always available. They are listed here in alphabetical order.

Built-in Functions				
<code>abs()</code>	<code>delattr()</code>	<code>hash()</code>	<code>memoryview()</code>	<code>set()</code>
<code>all()</code>	<code>dict()</code>	<code>help()</code>	<code>min()</code>	<code>setattr()</code>
<code>any()</code>	<code>dir()</code>	<code>hex()</code>	<code>next()</code>	<code>slice()</code>
<code>ascii()</code>	<code>divmod()</code>	<code>id()</code>	<code>object()</code>	<code>sorted()</code>
<code>bin()</code>	<code>enumerate()</code>	<code>input()</code>	<code>oct()</code>	<code>staticmethod()</code>
<code>bool()</code>	<code>eval()</code>	<code>int()</code>	<code>open()</code>	<code>str()</code>
<code>breakpoint()</code>	<code>exec()</code>	<code>isinstance()</code>	<code>ord()</code>	<code>sum()</code>
<code>bytearray()</code>	<code>filter()</code>	<code>issubclass()</code>	<code>pow()</code>	<code>super()</code>
<code>bytes()</code>	<code>float()</code>	<code>iter()</code>	<code>print()</code>	<code>tuple()</code>
<code>callable()</code>	<code>format()</code>	<code>len()</code>	<code>property()</code>	<code>type()</code>
<code>chr()</code>	<code>frozenset()</code>	<code>list()</code>	<code>range()</code>	<code>vars()</code>
<code>classmethod()</code>	<code>getattr()</code>	<code>locals()</code>	<code>repr()</code>	<code>zip()</code>
<code>compile()</code>	<code>globals()</code>	<code>map()</code>	<code>reversed()</code>	<code>__import__()</code>
<code>complex()</code>	<code>hasattr()</code>	<code>max()</code>	<code>round()</code>	

<https://docs.python.org/3/library/functions.html?highlight=float#float>

데이터 클리닝 실습

■ 데이터 변환 중 오류

- 오류 위치: 1950년8월31일 이후
ValueError: could not convert string to float:
- 원인 : ?

■ 누락데이터 클리닝 실습

- 결측값 처리 방식
- 최고 기온으로 나오기 힘든 아주 작은 값으로 대체
: 누락된 데이터인 빈 문자열 “을 -999로

데이터 분석

- 알고리즘 실습
 - 최고 기온 구하기

EDA

➤ Exploratory Data Analysis

- ✓ 탐색적 데이터 분석
- ✓ 수집한 데이터를 다양한 각도에서 관찰하고 이해하는 과정
- ✓ 데이터를 분석하기 전에 그래프나 통계적인 방법으로 자료를 직관적으로 바라보는 과정

➤ EDA 과정

- ✓ '데이터 설명서(data description) 읽기'
- ✓ 시각화 방법으로

EDA 사례

➤ 다음주 수요일 잠실경기장 야구 관중수는 어느 정도될까?

✓ 날씨가 관중수와 상관관계가 있는가?

✓ 상대팀에 따라 관중수에 영향이 있는가?

=> 시각화, 통계분석을 통해 데이터를 선별

참고도서

➤reference

[1] 모두의 데이터분석 with 파이썬

- 길벗

[2] Python Standard Library

<https://docs.python.org/3/library/index.html>

End