



Introduction To Data Analysis

# 데이터분석입문

Lecture 07. 서울 기온 데이터분석 기초

인공지능소프트웨어학과 강환수 교수

### 학습개요

- ✓ 기상자료개방포털에서 서울 일별 기온 데이터 준비  
➤ <https://data.kma.go.kr/>
- ✓ 서울 일별 기온 데이터의 행과 열 이해하고 결측치(누락된 데이터) 파악
- ✓ 모듈 csv로 서울 일별 기온 데이터 읽어서 출력



### 학습목표

- ✓ 기상자료개방포털에서 필요한 기온 데이터를 내려 받을 수 있다.
- ✓ 파일 포맷 CSV(comma separated value) 이해할 수 있다.
- ✓ 서울 일별 기온 데이터의 행과 열 이해하고 결측치(누락된 데이터) 파악할 수 있다.
- ✓ 모듈 csv를 사용해 전체를 읽어와 행 별로 출력할 수 있다.
- ✓ 모듈 csv를 사용해 결측치(누락된 데이터)가 있는 행만 출력할 수 있다.

## LESSON 01

# 기온 데이터 준비



## 01. 기온 데이터 분석 시작하기


### → 기온 공공데이터 살펴보기 (1/5)

 기상 관련 데이터 수집 → 기상자료개방포털 (<https://data.kma.go.kr/>)



## 01. 기온 데이터 분석 시작하기

### → 기온 공공데이터 살펴보기 (2 / 5)

 [기후통계분석] → [조건별통계] 버튼 클릭



## 01. 기온 데이터 분석 시작하기

### 기온 공공데이터 살펴보기 (3 / 5)

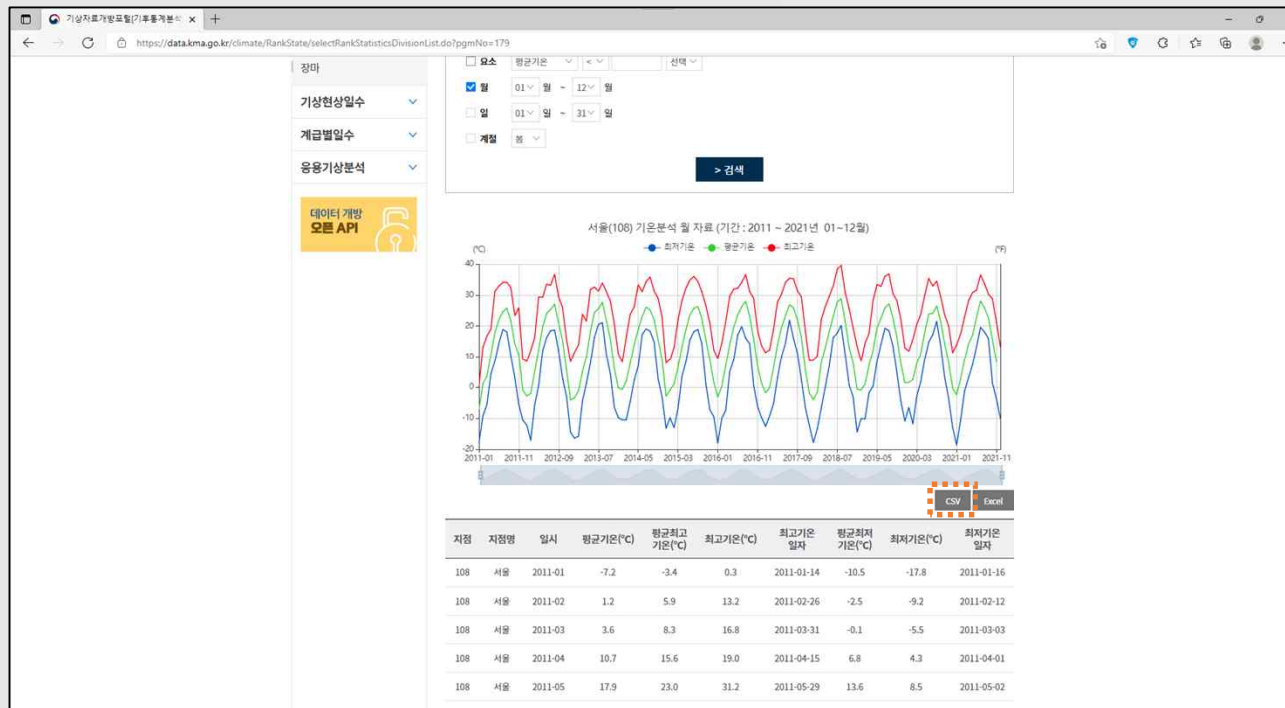
검색조건에서 조건을 지정하고 검색 버튼 클릭

The screenshot shows the '기후통계분석' (Climate Statistics Analysis) page on the KMA website. The left sidebar contains a menu with options like '평년값', '등계분석', '조건별통계', '기온분석', '강수량분석', '다중지점통계', '24절기', '순위값', '장마', '기상현상일수', '계급별일수', and '응용기상분석'. The main content area is titled '조건별통계' (Statistics by Condition) and includes a '자료설명' (Data Description) section. Below this, the '검색조건' (Search Conditions) section is highlighted with a red dashed box. It contains several dropdown menus and checkboxes for specifying search criteria: '분류' (Classification) set to '기온' (Temperature), '지역/지점' (Location/Point) set to '전국' (All Korea), '시월' (Start Month) set to '01' (January), '종월' (End Month) set to '12' (December), and '기간' (Period) set to '월' (Monthly). A red dashed box also highlights the '검색' (Search) button at the bottom right of the search conditions section.

## 01. 기온 데이터 분석 시작하기

### 기온 공공데이터 살펴보기 (4 / 5)

그래프 오른쪽 아래쪽에 위치한 CSV 버튼 클릭

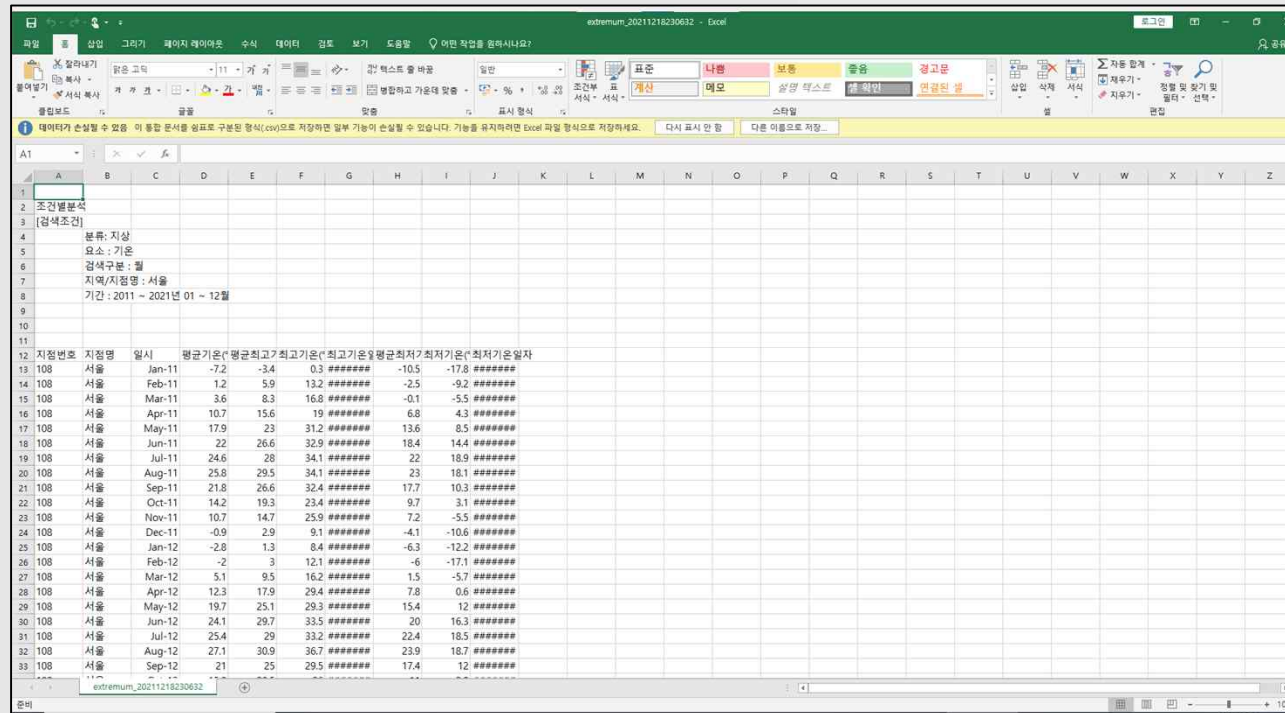




## 01. 기온 데이터 분석 시작하기

### 기온 공공데이터 살펴보기 (5 / 5)

#### 다운로드 (Download) 폴더에 저장된 CSV 파일 열기



지점번호	지점명	일시	평균기온(°C)	최고기온(°C)	최저기온(°C)
108	서울	Jan-11	-7.2	-3.4	0.3
108	서울	Feb-11	1.2	5.9	13.2
108	서울	Mar-11	3.6	8.3	16.8
108	서울	Apr-11	10.7	15.6	19.0
108	서울	May-11	17.9	23.0	31.2
108	서울	Jun-11	22.0	26.6	32.9
108	서울	Jul-11	24.6	28.0	34.1
108	서울	Aug-11	25.8	29.5	34.1
108	서울	Sep-11	21.8	26.6	32.4
108	서울	Oct-11	14.2	19.3	23.4
108	서울	Nov-11	10.7	14.7	25.9
108	서울	Dec-11	-0.9	2.9	9.1
108	서울	Jan-12	-2.8	1.3	8.4
108	서울	Feb-12	-2.0	3.0	12.1
108	서울	Mar-12	5.1	9.5	16.2
108	서울	Apr-12	12.3	17.9	29.4
108	서울	May-12	19.7	25.1	29.3
108	서울	Jun-12	24.1	29.7	33.5
108	서울	Jul-12	25.4	29.0	33.2
108	서울	Aug-12	27.1	30.9	36.7
108	서울	Sep-12	21.0	25.0	29.5

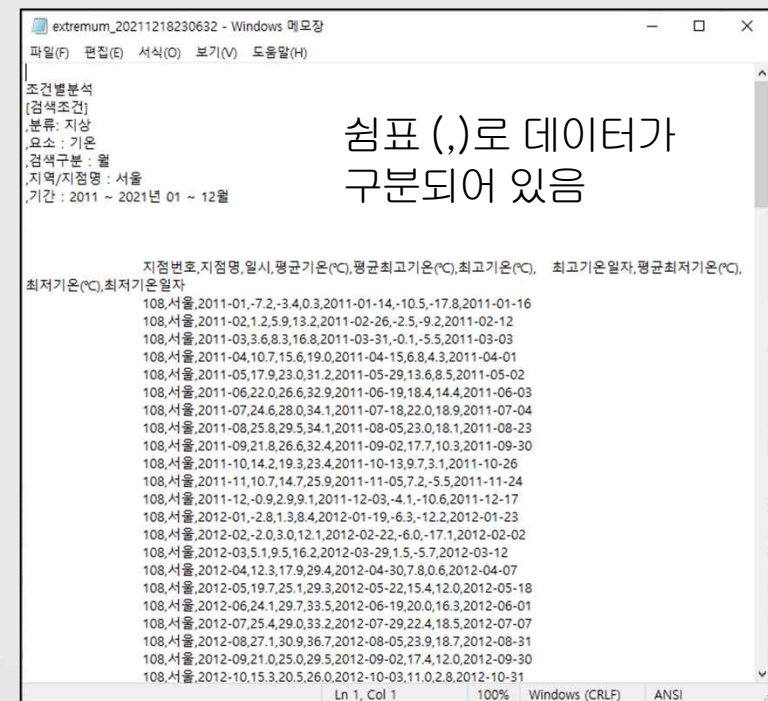
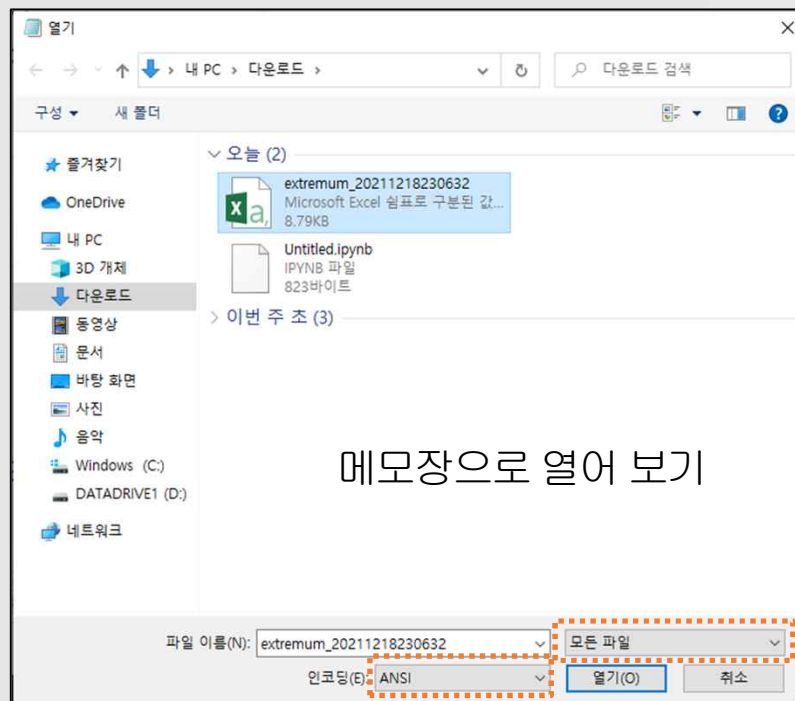


## 01. 기온 데이터 분석 시작하기

### → CSV 파일이란?

CSV (= Comma-Separated Values)

각 데이터를 쉼표 (,)로 구분하여 저장하는 파일 형식



## LESSON 02

# 서울 기온 데이터에서 누락된 기온 파악



## 02. 서울의 기온 데이터 분석하기

### → 상황 가정

○○님, 제가 방금 메일로  
seoul.csv 파일을 보냈어요.  
서울에 기온 데이터가 기록되어  
있어요. 언제 가장 더웠었는지  
그리고 그때 온도가 몇 도였는지  
확인해서 오늘 퇴근하기 전까지  
알려주세요.



팀장



나

네, 팀장님 알겠습니다.

## 02. 서울의 기온 데이터 분석하기

### → CSV 파일 다운로드

기상자료개방포털 (<https://data.kma.go.kr/>)

[기후통계분석] – [기온분석] – [검색조건 설정] – [CSV 다운로드]

The screenshot shows the '기온분석' (Temperature Analysis) page on the KMA Data Portal. The left sidebar contains navigation links: '기후통계분석', '평년값', '통계분석', '조건별통계', '기온분석' (highlighted), '강수량분석', '다중지점통계', '24절기', '순위값', '장마', '기상현상일수', '계급별일수', '응용기상분석', and '데이터 개방 오픈 API'. The main content area is titled '기온분석 - 그래프' and includes a '자료설명' (Data Description) section. Below this is the '검색조건' (Search Conditions) section, which contains filters for '관측구분' (Observation Area) set to '서울' (Seoul), '자료형태' (Data Type) set to '기온' (Temperature), and '기간' (Period) set to '19040101' to '20211223'. A '> 검색' (Search) button is visible. At the bottom, there is a 'CSV' button and an 'Excel' button. A line graph titled '기온분석 기본 서울(108) 일자료 기간: 19040101 ~ 20211223' is displayed, showing daily temperature data with a legend for '최저기온' (Minimum Temperature), '평균기온' (Average Temperature), and '최고기온' (Maximum Temperature).

## 02. 서울의 기온 데이터 분석하기

### → CSV 파일 편집

다운로드한 CSV 파일을 엑셀 프로그램으로 열고  
데이터 분석에 불필요한 1~7행을 삭제합니다.

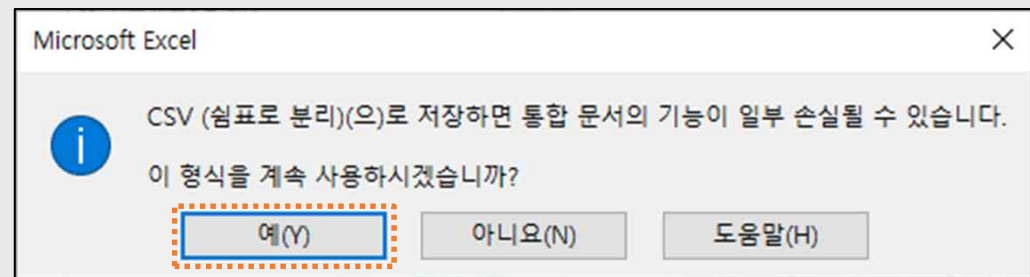
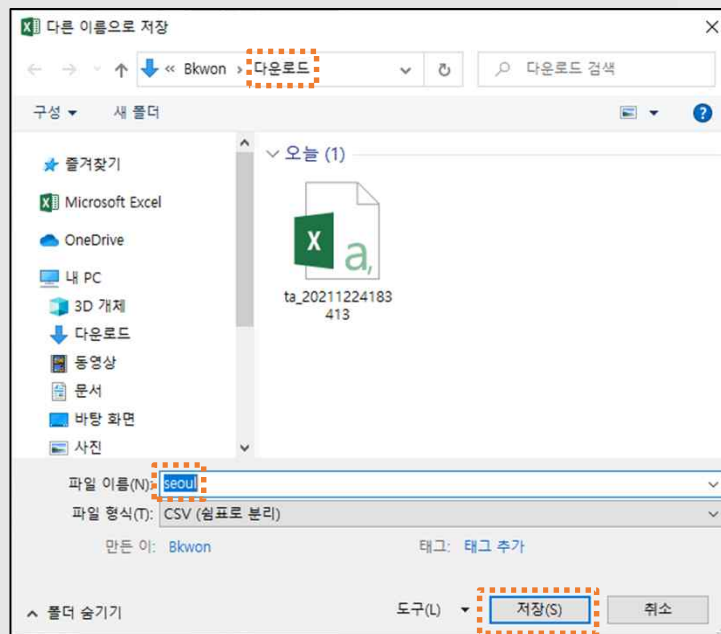
삭제

기온분석							
[검색조건]							
자료구분 : 일							
기온(°C)	최저기온(°C)	최고기온(°C)					
13.5	7.9	20.7					
16.2	7.9	22					
16.2	13.1	21.3					
16.5	11.2	22					
17.6	10.9	25.4					
13	11.2	21.3					
11.3	6.3	16.1					
8.9	3.9	14.9					
11.6	3.8	21.1					
14.2	6.4	24.1					
15.4	10.1	20.4					
13.9	11.1	17.4					
108	13.8	8.3	21.3				
108	13	6.1	20.6				
108	13.1	5.7	20.9				
108	14.1	8.2	20.2				
108	16.4	10.3	21.6				
108	14.3	9.8	20.9				
108	13.9	6.7	21.3				
108	12.4	22.7					

## 02. 서울의 기온 데이터 분석하기

### → CSV 파일 다운로드

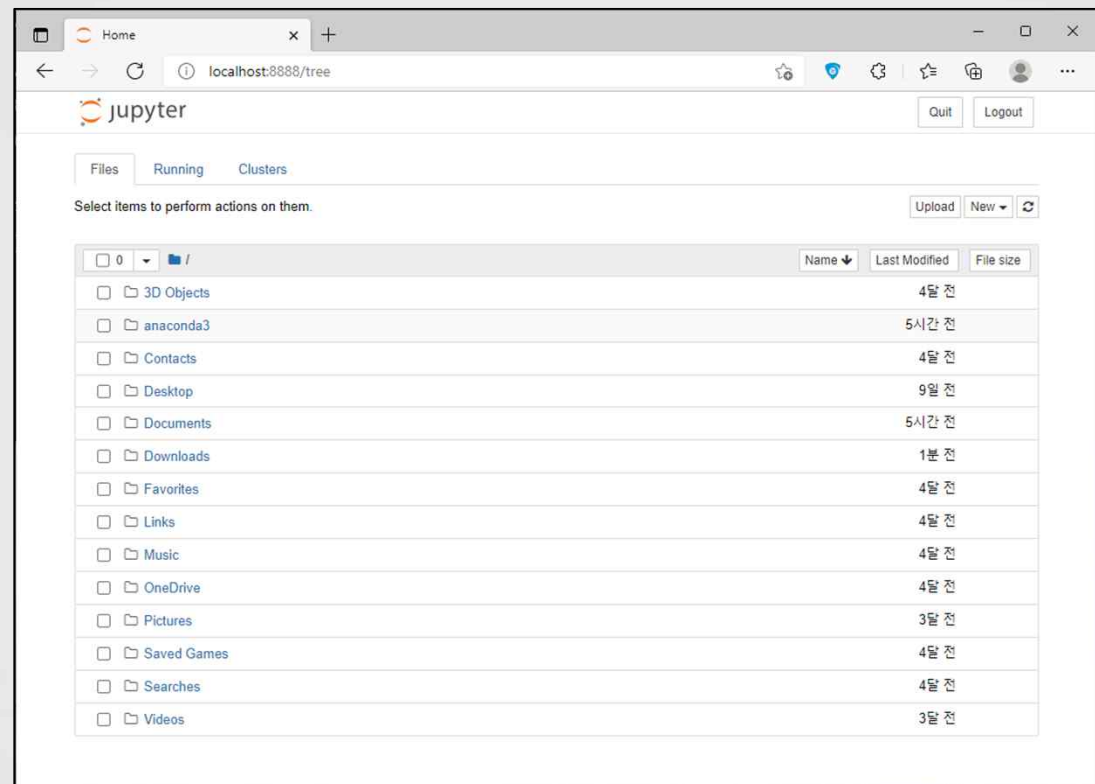
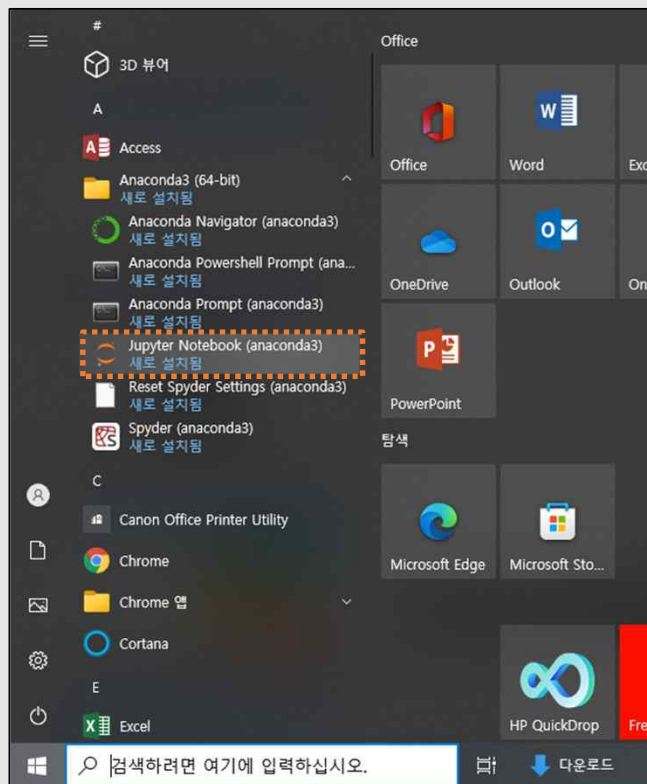
- 메뉴에서 [파일] – [다른 이름으로 저장]을 눌러 파일 이름을 seoul로 변경하고 다운로드 폴더에 저장합니다. 이때 경고창이 뜨면 “예(Y)” 버튼을 클릭합니다.



## 02. 서울의 기온 데이터 분석하기

### → CSV 파일에서 데이터 읽어오기 (1/5)

주피터 노트북을 실행합니다.

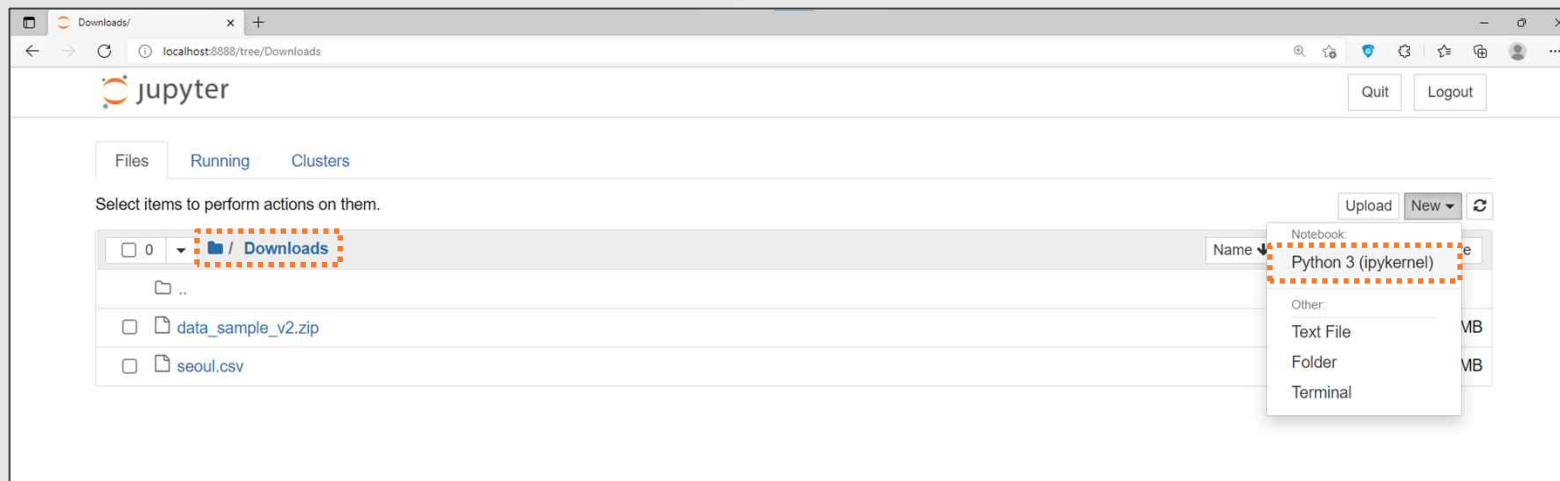




## 02. 서울의 기온 데이터 분석하기

### → CSV 파일에서 데이터 읽어오기 (2 / 5)

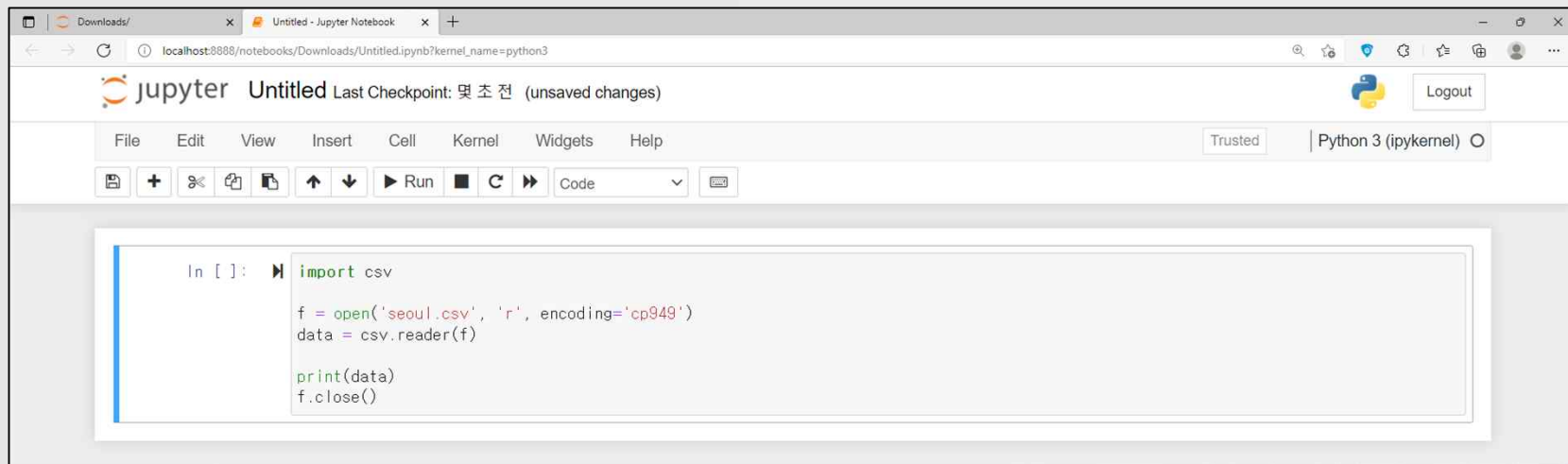
- 브라우저에서 Downloads 폴더를 선택한 후 오른쪽에 있는 [New] – [Python 3] 버튼을 클릭하여 새 파이썬 노트북을 생성합니다.



## 02. 서울의 기온 데이터 분석하기

### → CSV 파일에서 데이터 읽어오기 (3 / 5)

📌 노트북의 빈 셀 (cell)에 다음과 같이 코드를 작성합니다.



```
In [ ]: import csv

f = open('seoul.csv', 'r', encoding='cp949')
data = csv.reader(f)

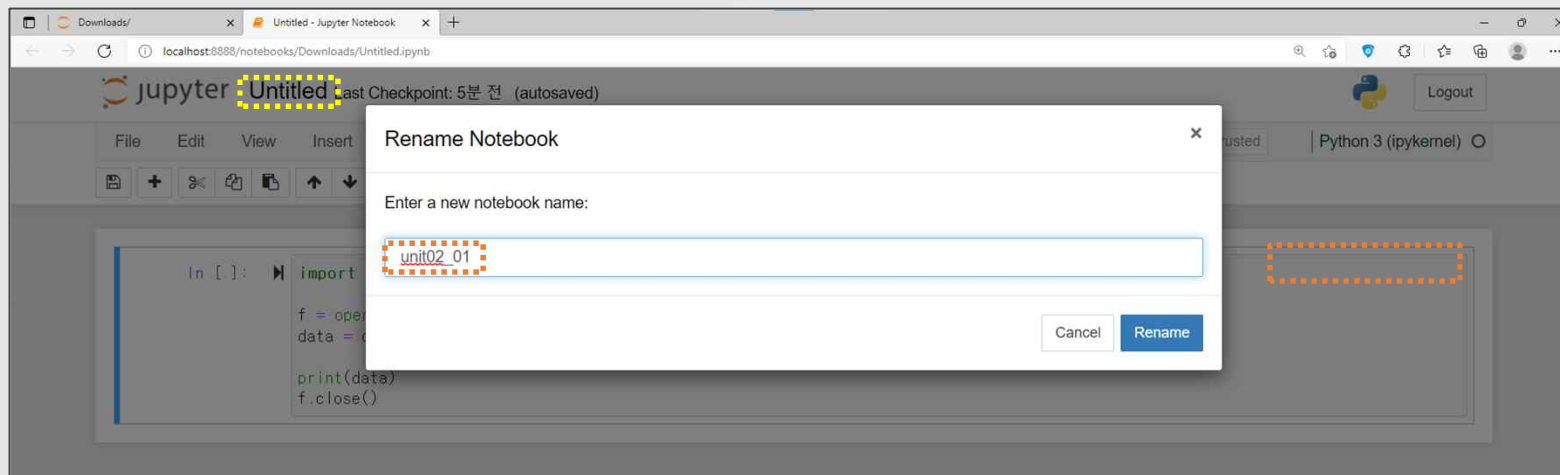
print(data)
f.close()
```

만약 윈도우가 아닌 다른 운영체제 (macOS, 리눅스 등)를 사용하고 있다면,  
encoding='cp949'를 반드시 입력해야 합니다.  
특히, vs code에서 노트북으로 실행하려면 encoding='cp949'를 반드시 입력

## 02. 서울의 기온 데이터 분석하기

### → CSV 파일에서 데이터 읽어오기 (4 / 5)

- “Untitled”라고 적혀 있는 노트북의 이름을 클릭하여 “unit02\_01”로 변경한 후 Rename버튼을 클릭하여 저장합니다.

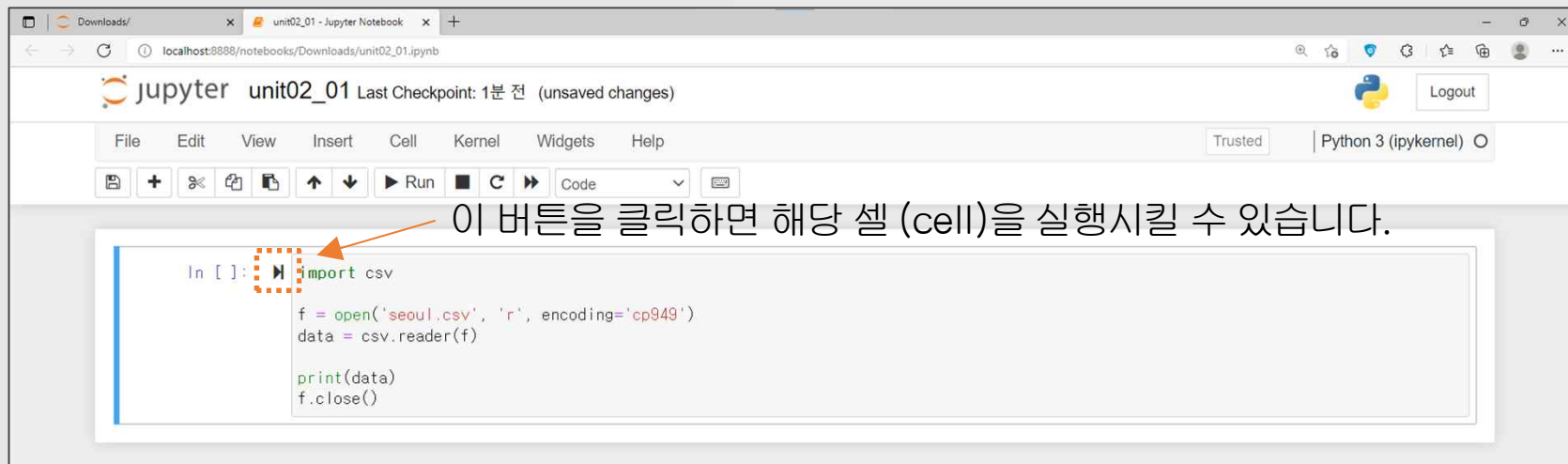


lecture07-08-09-seoul-temperature-mpl-np.ipynb

## 02. 서울의 기온 데이터 분석하기

### → CSV 파일에서 데이터 읽어오기 (5 / 5)

📌 작성한 코드를 실행하겠습니다.



단축키를 통해서도 실행할 수 있습니다.

- [Ctrl] + [Enter]: Run Cells
- [Shift] + [Enter]: Run Cells and Select Cell
- [Alt] + [Enter]: Run Cells and Insert Cell

## 02. 서울의 기온 데이터 분석하기

### → 데이터 출력하기 (1/6)

 셀 (cell)을 하나 추가하여 아래의 코드 내용을 작성하고 실행합니다.

```
In [ ]: import csv

f = open('seoul.csv', 'r', encoding='cp949')
data = csv.reader(f)

for row in data:
    print(row)

f.close()
```

```
['날짜', '지점', '평균기온(℃)', '최저기온(℃)', '최고기온(℃)']
['1907-10-01', '108', '13.5', '7.9', '20.7']
['1907-10-02', '108', '16.2', '7.9', '22']
['1907-10-03', '108', '16.2', '13.1', '21.3']
['1907-10-04', '108', '16.5', '11.2', '22']
['1907-10-05', '108', '17.6', '10.9', '25.4']
['1907-10-06', '108', '13', '11.2', '21.3']
['1907-10-07', '108', '11.3', '6.3', '16.1']
['1907-10-08', '108', '8.9', '3.9', '14.9']
['1907-10-09', '108', '11.6', '3.8', '21.1']
['1907-10-10', '108', '14.2', '6.4', '24.1']
['1907-10-11', '108', '15.4', '10.1', '20.4']
['1907-10-12', '108', '13.9', '11.1', '17.4']
['1907-10-13', '108', '13.8', '8.3', '21.3']
```

## 02. 서울의 기온 데이터 분석하기

### → 데이터 출력하기 (2 / 6)

 출력된 결과화면을 살펴봅니다.

```
[ '1952-01-02', '108', '', '', '' ]  
[ '1952-01-03', '108', '', '', '' ]  
[ '1952-01-04', '108', '', '', '' ]  
[ '1952-01-05', '108', '', '', '' ]  
[ '1952-01-06', '108', '', '', '' ]  
[ '1952-01-07', '108', '', '', '' ]  
[ '1952-01-08', '108', '', '', '' ]  
[ '1952-01-09', '108', '', '', '' ]  
[ '1952-01-10', '108', '', '', '' ]  
[ '1952-01-11', '108', '', '', '' ]  
[ '1952-01-12', '108', '', '', '' ]  
[ '1952-01-13', '108', '', '', '' ]  
[ '1952-01-14', '108', '', '', '' ]  
[ '1952-01-15', '108', '', '', '' ]  
[ '1952-01-16', '108', '', '', '' ]  
[ '1952-01-17', '108', '', '', '' ]  
[ '1952-01-18', '108', '', '', '' ]  
[ '1952-01-19', '108', '', '', '' ]  
[ '1952-01-20', '108', '', '', '' ]  
[ '1952-01-21', '108', '', '', '' ]
```

누락된 데이터가 있습니다.  
6.25전쟁 (1950년 6월 25일 - 1953년 7월 27일)

## 02. 서울의 기온 데이터 분석하기

### → 데이터 출력하기 (3 / 6)

 출력된 결과화면을 살펴봅니다.

```
['2017-10-01', '108', '18.2', '15.5', '21.2']  
['2017-10-02', '108', '22', '15.6', '29.4']  
['2017-10-03', '108', '17.6', '13.4', '23.6']  
['2017-10-04', '108', '16.7', '10.7', '24.3']  
['2017-10-05', '108', '18.7', '13.9', '23.4']  
['2017-10-06', '108', '18.9', '16.2', '23.3']  
['2017-10-07', '108', '21.9', '16.9', '28.8']  
['2017-10-08', '108', '23', '19.3', '28.7']  
['2017-10-09', '108', '22.5', '19.8', '27.6']  
['2017-10-10', '108', '21.4', '18.6', '24.8']  
['2017-10-11', '108', '15.5', '12.2', '21.7']  
['2017-10-12', '108', '11.4', '8.8', '18.9']  
['2017-10-13', '108', '12.8', '6.1', '20.5']  
['2017-10-14', '108', '14.4', '9', '23']  
['2017-10-15', '108', '15.8', '9', '23']  
['2017-10-16', '108', '16.6', '13.6', '22']  
['2017-10-17', '108', '16.2', '9.2', '23.9']  
['2017-10-18', '108', '16.5', '14.2', '19.1']  
['2017-10-19', '108', '17', '11.9', '23.2']  
['2017-10-20', '108', '17', '11.1', '24.2']
```

2017년 10월12일의 최고기온 데이터도  
누락되어 있습니다.



## 02. 서울의 기온 데이터 분석하기

→ with open as f

 f.close() 필요 없음

```
In [9]: import csv

with open('seoul.csv', 'r', encoding='cp949') as f:
    data = csv.reader(f)
    for row in data:
        print(row)
```

['날짜', '지점', '평균기온(℃)', '최저기온(℃)', '최고기온(℃)']  
['#t1907-10-01', '108', '13.5', '7.9', '20.7']  
['#t1907-10-02', '108', '16.2', '7.9', '22']  
['#t1907-10-03', '108', '16.2', '13.1', '21.3']  
['#t1907-10-04', '108', '16.5', '11.2', '22']  
['#t1907-10-05', '108', '17.6', '10.9', '25.4']  
['#t1907-10-06', '108', '13', '11.2', '21.3']  
['#t1907-10-07', '108', '11.3', '6.3', '16.1']  
['#t1907-10-08', '108', '8.9', '3.9', '14.9']  
['#t1907-10-09', '108', '11.6', '3.8', '21.1']

### → 데이터 출력하기 (4 / 6)

앞에서 살펴본 것처럼 전체 데이터에서 누락된 값 (= 결측치, Missing Value)이 있는지 여부를 데이터 분석 전에 확인해 보는 습관을 갖도록 합니다.

만약 결측치가 있다는 것이 확인되면, 결측치를 어떻게 처리해야 할까요?

✓ 결측치 대체

+ 해당 결측치를 평균 값이나 바로 앞 (또는 뒤) 데이터 값으로 대체하는 등 여러 방법들이 존재합니다.

✓ 결측치 제거

+ 결측치가 존재하는 행 (Row) 또는 열 (Column)을 제거합니다.

## 02. 서울의 기온 데이터 분석하기

### → 데이터 출력하기 (5 / 6)

- 서울 기온 데이터에는 결측치가 빈 문자열 ('') 형태로 존재하니, 결측치를 확인할 수 있는 기능을 아래와 같이 구현해 봅니다.

```
In [14]: import csv

f = open('seoul.csv', 'r', encoding='cp949')
data = csv.reader(f)

for row in data:
    if '' in row:
        print(row)
    #     break

f.close()
```

## 02. 서울의 기온 데이터 분석하기

### → 데이터 출력하기 (6 / 6)

📊 실행하여 보면 아래와 같이 결측치를 포함하는 데이터들만 출력됩니다.

```
[ '1953-11-15', '108', '', '', '' ]  
[ '1953-11-16', '108', '', '', '' ]  
[ '1953-11-17', '108', '', '', '' ]  
[ '1953-11-18', '108', '', '', '' ]  
[ '1953-11-19', '108', '', '', '' ]  
[ '1953-11-20', '108', '', '', '' ]  
[ '1953-11-21', '108', '', '', '' ]  
[ '1953-11-22', '108', '', '', '' ]  
[ '1953-11-23', '108', '', '', '' ]  
[ '1953-11-24', '108', '', '', '' ]  
[ '1953-11-25', '108', '', '', '' ]  
[ '1953-11-26', '108', '', '', '' ]  
[ '1953-11-27', '108', '', '', '' ]  
[ '1953-11-28', '108', '', '', '' ]  
[ '1953-11-29', '108', '', '', '' ]  
[ '1953-11-30', '108', '', '', '' ]  
[ '1967-02-19', '108', '-1.7', '', '' ]  
[ '1973-10-16', '108', '12.3', '', '' ]  
[ '2017-10-12', '108', '11.4', '8.8', '' ]
```

### → 헤더 저장하기 (1/2)

● 헤더 (Header)란 데이터 파일에서 각 값이 어떤 의미를 갖는지 표시한 행 (Row)을 의미합니다.

● 헤더를 별도로 저장하기 위해서 next( ) 함수를 사용할 수 있습니다.

✓ next( ) 함수

+ 첫 번째 데이터 행을 읽어오면 데이터의 탐색 위치를 다음 행으로 이동시킵니다.

```
In [15]: import csv

f = open('seoul.csv', 'r', encoding='cp949')
data = csv.reader(f)

header = next(data)
print(header)

f.close()

['날짜', '지점', '평균기온(℃)', '최저기온(℃)', '최고기온(℃)']
```

### → 헤더 저장하기 (2 / 2)

📊 header = next(data) 코드가 있는 경우와 없는 경우의 출력을 비교하면 next( ) 함수의 기능을 보다 쉽게 이해할 수 있습니다.

```
In [16]: import csv

f = open('seoul.csv', 'r', encoding='cp949')
data = csv.reader(f)

header = next(data)
print(header)

for row in data:
    print(row)

f.close()
```

# Summary 학습정리





기상자료개방포털에서 필요한 기온 데이터를 내려 받기

파일 포맷 CSV (comma separated value)

value0, value1, value2, value3, ...

결측치(누락된 데이터)를 이해하고 처리

모듈 csv를 사용해 전체를 읽어오기

읽어 온 전체를 행 별로 출력

결측치가 있는 행 출력

```
import csv

f = open('seoul.csv', 'r', encoding='cp949')
data = csv.reader(f)

for row in data:
    print(row)

f.close()
```

```
import csv

f = open('seoul.csv', 'r', encoding='cp949')
data = csv.reader(f)

for row in data:
    if '' in row:
        print(row)

f.close()
```

