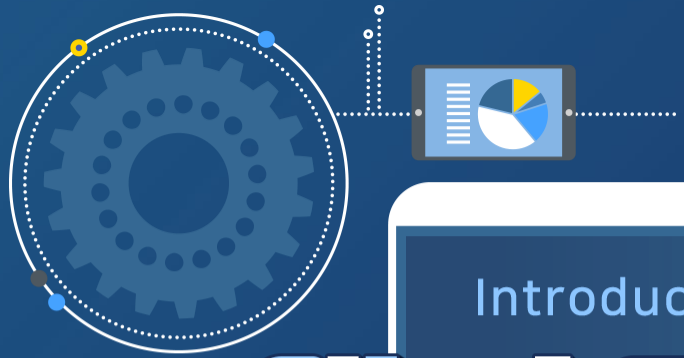


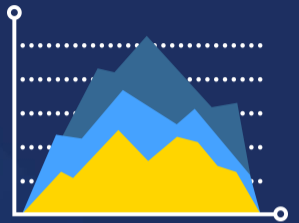

Introduction To Data Analysis

데이터분석입문

Lecture 08. 서울 최고기온 데이터분석




인공지능소프트웨어학과 강환수 교수

학습개요

- 데이터에서 헤더와 결측치 파악
- CSV 파일 읽어 행 출력
- 특정 열의 최대 값 또는 최소 값 출력

학습목표

- 서울 최고 기온, 최저 기온을 찾아 출력할 수 있다.
- 서울 최고 기온, 최저 기온 10개를 찾아 출력할 수 있다.

LESSON 01

서울 기온 데이터분석 기초

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

- 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.

데이터 출력하기

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()
```

데이터 출력하기

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

- 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

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()
```

LESSON 02

서울이 가장 더웠던 날은 언제 였을까?

질문 다듬기

○○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

- 가장 덥다?
  → “최고 기온이 가장 높다”

나

네, 팀장님 알겠습니다.

기상 관측 이래, 서울의 최고 기온이 가장 높았던 날은 언제였고, 몇 도였을까?

문제 해결 방법 구상하기

문제를 해결하기 위한 절차 (= 알고리즘, Algorithm)는?

- Step 1) 데이터를 읽어온다.
- Step 2) 순차적으로 최고 기온을 확인한다.
- Step 3) 최고 기온이 가장 높았던 날짜의 데이터를 저장한다.
- Step 4) 최종 저장된 데이터를 출력한다.

파이썬 코드로 구현하기 (1/6)

Step 1) 데이터를 읽어온다.

- 주피터 노트북을 열고 [File] – [New Notebook] – [Python 3] 버튼을 클릭하여 새 파일을 만듭니다.
- 파일의 이름을 lecture08-seoul-temperature로 수정하고 다음과 같은 코드를 작성합니다.

```
In [3]:
import csv

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

print(header)
for row in data:
    print(row)
    break

f.close()
```

```
['날짜', '지점', '평균기온(℃)', '최저기온(℃)', '최고기온(℃)']
['1907-10-01', '108', '13.5', '7.9', '20.7']
```

우리가 관심있는 “최고기온” 데이터는
리스트(list)의 가장 마지막에 위치하고 있으며 자료형이 문자열(string)입니다.

파이썬 코드로 구현하기 (2/6)

Step 1) 데이터를 읽어온다.

- 최고 기온 값을 찾기 위해서는 기온이 높고 낮다는 대소 관계를 비교해야 합니다.
- 따라서 자료형을 문자열(string)에서 실수(float)로 변환해 주어야 합니다.
  → float( ) 함수를 사용하면 됩니다.

In [5]:

```
import csv

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

print(header)
for row in data:
    row[4] = float(row[4])      # row[4]와 row[-1]은 같습니다.
    print(row)
    break

f.close()
```

```
['날짜', '지점', '평균기온(℃)', '최저기온(℃)', '최고기온(℃)']
['1907-10-01', '108', '13.5', '7.9', 20.7]
```

작은 따옴표 (')가 사라졌습니다.
더 이상 자료형이 문자열이 아닙니다.

파이썬 코드로 구현하기 (3/6)

Step 1) 데이터를 읽어온다.

- 서울 기온 데이터에는 빈 문자열 (‘’)로 표현되는 결측치 (Missing Value)가 있었습니다.
- break 명령어를 주석 처리하고 코드를 실행하면 아래와 같이 오류가 발생합니다.

```
['1950-08-24', '108', '21.6', '15.2', 29.6]
['1950-08-25', '108', '22', '16', 28.7]
['1950-08-26', '108', '21.6', '17.1', 28.6]
['1950-08-27', '108', '20.4', '15.5', 26.4]
['1950-08-28', '108', '21.2', '16.4', 28.4]
['1950-08-29', '108', '23.1', '16.8', 30.4]
['1950-08-30', '108', '24.6', '18', 32.6]
['1950-08-31', '108', '25.4', '20.1', 32.5]

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~#AppData#Local#Temp/ipykernel_8928/1401348546.py in <module>
      7 print(header)
      8 for row in data:
----> 9     row[4] = float(row[4])     # row[4]와 row[-1]은 같습니다.
     10     print(row)
     11

ValueError: could not convert string to float: ''
```

빈 문자열 (‘’)을 어떤 실수 값으로 바꿔야 할지 몰라서 오류가 발생한 것입니다.

파이썬 코드로 구현하기 (4/6)

Step 1) 데이터를 읽어온다.

- 결측치를 다른 값으로 대체하는 전략을 사용하겠습니다.
- 최고 기온을 찾는 문제이므로 최고 기온으로 나오기 어려운 값인 -999으로 결측치를 대체합니다.

In [6]:

```
import csv

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

print(header)
for row in data:
    if row[4] == '':      # 만약 최고기온 데이터가 빈 문자열이라면
        row[4] = -999     # -999를 대입

    row[4] = float(row[4])      # row[4]와 row[-1]은 같습니다.
    print(row)

f.close()
```

파이썬 코드로 구현하기 (5/6)

Step 1) 데이터를 읽어온다.

- [x] 코드를 다시 실행시켜보면, 오류 없이 수행됩니다.

```
['2021-12-05', '108', '2.9', '-2.3', 9.6]
['2021-12-06', '108', '5.4', '-0.7', 12.1]
['2021-12-07', '108', '6.8', '2.6', 13.3]
['2021-12-08', '108', '6.7', '1.9', 13.3]
['2021-12-09', '108', '6.5', '2.7', 9.7]
['2021-12-10', '108', '7', '5.8', 8.2]
['2021-12-11', '108', '6', '4.4', 9.8]
['2021-12-12', '108', '1.2', '-3.8', 5.1]
['2021-12-13', '108', '-2.2', '-5.9', 1.9]
['2021-12-14', '108', '3.7', '-0.7', 8.2]
['2021-12-15', '108', '7.2', '5.4', 10.0]
['2021-12-16', '108', '6.4', '3.3', 9.8]
['2021-12-17', '108', '-5.6', '-10.1', 3.2]
['2021-12-18', '108', '-5.7', '-11.2', -1.8]
['2021-12-19', '108', '-0.8', '-5.8', 4.2]
['2021-12-20', '108', '5.2', '-0.9', 11.1]
['2021-12-21', '108', '5.2', '0.8', 8.9]
['2021-12-22', '108', '2.2', '-2.6', 8.2]
['2021-12-23', '108', '2.7', '-1.5', 8.3]
```

파이썬 코드로 구현하기 (6/6)

In [13]:

```
import csv

max_temp = -999     # 최고 기온 값을 저장할 변수
max_date = ''       # 최고 기온이 가장 높았던 날짜를 저장할 변수

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

# print(header)
for row in data:
    if row[4] == '':      # 만약 최고기온 데이터가 빈 문자열이라면
        row[4] = -999     # -999를 대입

    row[4] = float(row[4])      # row[4]와 row[-1]은 같습니다.

    if max_temp < row[4]:
        max_temp = row[4]       # 최고 기온 값 업데이트
        max_date = row[0]       # 최고 기온 날짜 업데이트

f.close()

print('기상 관측 이래 서울의 최고 기온이 가장 높았던 날은',
      max_date + '로,', max_temp, '도 였습니다.')
```

기상 관측 이래 서울의 최고 기온이 가장 높았던 날은 2018-08-01로, 39.6 도 였습니다.

Step 2) 순차적으로 최고 기온을 확인한다.

Step 3) 최고 기온이 가장 높았던 날짜의 데이터를 저장한다.

Step 4) 최종 저장된 데이터를 출력한다.

with open

```
import csv

max_temp = -999
max_date = ''

with open('seoul.csv', 'r', encoding='cp949') as f:
    data = csv.reader(f)
    header = next(data)

    for row in data:
        if row[4] == '':
            row[4] = -999

        row[4] = float(row[4])

        if max_temp < row[4]:
            max_temp = row[4]
            max_date = row[0].strip()

print(f'서울 최고 기온, 일자: {max_date}, 기온: {max_temp}')
```

```
서울 최고 기온, 일자: 2018-08-01, 기온: 39.6
```

가장 추웠던 날은?

```
import csv

min_temp = 999
min_date = ''

with open('seoul.csv', 'r', encoding='cp949') as f:
    data = csv.reader(f)
    header = next(data)

    for row in data:
        if row[3] == '':
            row[3] = 999

        row[3] = float(row[3])

        if min_temp > row[3]:
            min_temp = row[3]
            min_date = row[0].strip()

print(f'서울 최저 기온, 일자: {min_date}, 기온: {min_temp}')
```

```
서울 최저 기온, 일자: 1927-12-31, 기온: -23.1
```

서울 최고 온도 10개, 서울 최저 온도 10개

```
import csv

max_temp = []
min_temp = []

with open('seoul.csv', 'r', encoding='cp949') as f:
    data = csv.reader(f)
    header = next(data)

    for row in data:
        if row[3] == '':
            row[3] = 999
        if row[4] == '':
            row[4] = -999

        row[3] = float(row[3])
        row[4] = float(row[4])

        min_temp.append(row[3])
        max_temp.append(row[4])

print(f'서울 최고 기온 10개')
max_temp.sort(reverse=True)
print(max_temp[:10])

print(f'서울 최저 기온 10개')
min_temp.sort()
print(min_temp[:10])
```

```
서울 최고 기온 10개
[39.6, 38.4, 38.3, 38.2, 38.2, 38.2, 38.0, 38.0, 37.9, 37.9]
서울 최저 기온 10개
[-23.1, -22.5, -22.3, -22.2, -21.9, -21.8, -21.7, -21.5, -21.3, -21.3]
```

SUMMARY

학습정리

서울 최고 기온, 최저 기온을 찾아 출력

```
for row in data:
    if row[4] == '':
        row[4] = -999

    row[4] = float(row[4])

    if max_temp < row[4]:
        max_temp = row[4]
        max_date = row[0].strip()
```

서울 최고 기온, 최저 기온 10개를 찾아 출력

```
for row in data:
    if row[3] == '':
        row[3] = 999
    if row[4] == '':
        row[4] = -999

    row[3] = float(row[3])
    row[4] = float(row[4])

    min_temp.append(row[3])
    max_temp.append(row[4])
```