hci+d lab.

Week 04 • 데이터저널리즘

Data Analysis Using NumPy and Pandas 1

Joonhwan Lee
human-computer interaction + design lab.

오늘 다룰 내용

- Data Processing
- NumPy

Data Processing

Data Analysis Process

Data Analysis Process

- Question Phase
  - Characteristics of students who finish MOOC lectures
  - Age and gender distribution of people who spend money in Gangnam area

Data Analysis Process

- Wrangling Phase
  - Data acquisition - where to get data to answer the questions
  - Data cleaning - (in most case) data need to be cleaned - we spend most of our time for this…(80~90%)

Data Analysis Process

- Explore Phase
  - Build intuition by exploratory data analysis
    - information visualization
    - find patterns

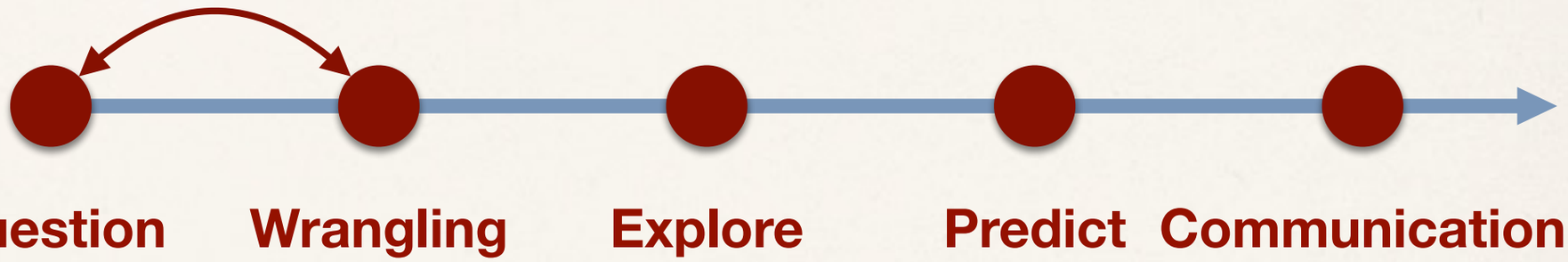
Data Analysis Process

- Prediction Phase
  - Predict results of out question
    - eg. Age and gender distribution of people who spend money in Gangnam area => According to our data analysis, 20-30 women spend more money in this area. => marketing insights
  - Usually requires statistics or machine learning

Data Analysis Process

- Communication Phase
- Data Journalisms
- Blog Posts
- Data Visualizations
- Papers

Data Analysis Process



Question Wrangling Explore Predict Communication

Data Acquisition

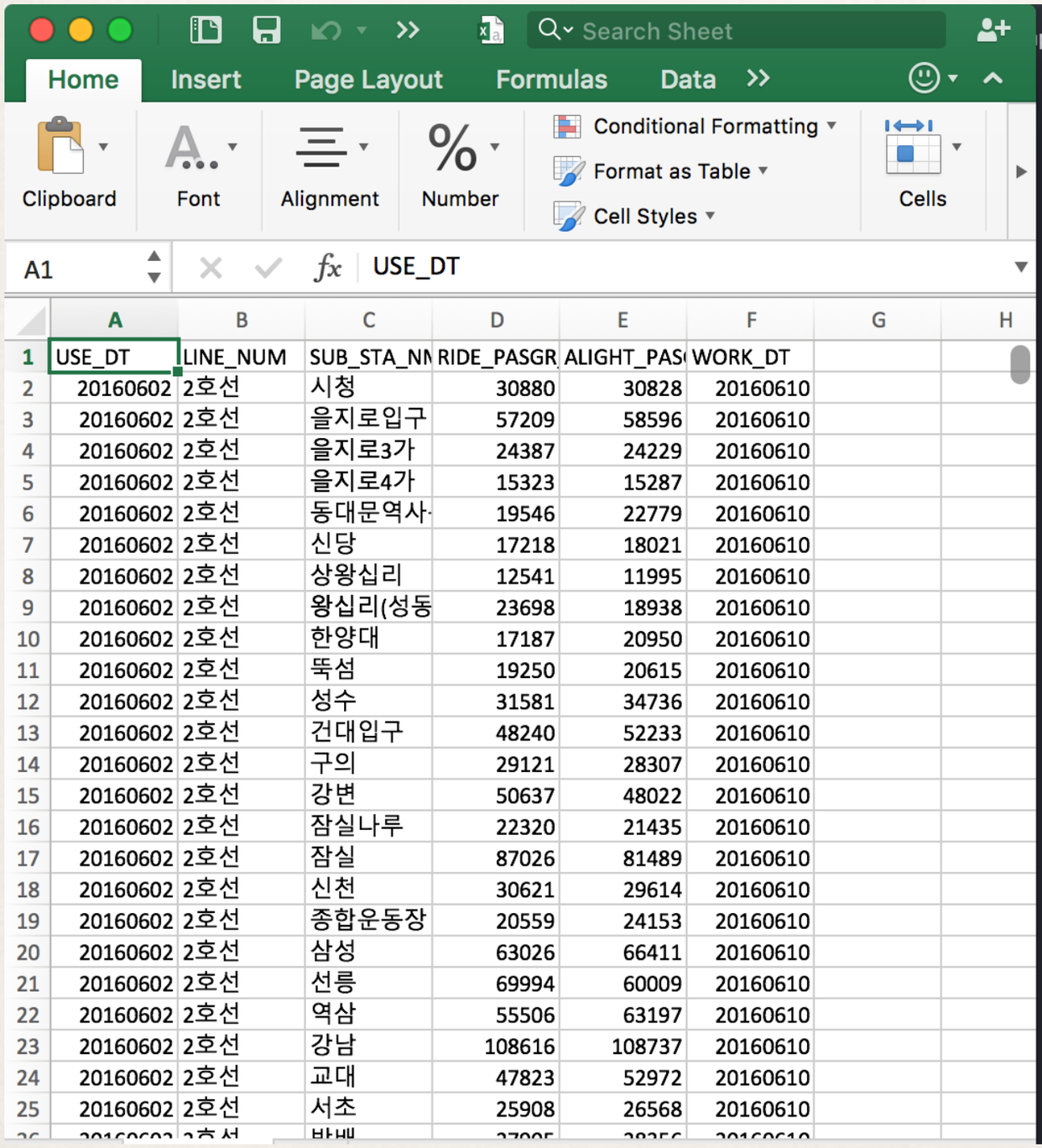
- **Downloading files**
- Accessing an API
- Scraping a web page

→ will do these later

Data Format

- CSV: Comma Separated Values
  - data column separated by comma
  - text file format (xls is binary format) → can read from text editors

Data Format



	A	B	C	D	E	F	G	H
1	USE_DT	LINE_NUM	SUB_STA_NM	RIDE_PASGR	ALIGHT_PAS	WORK_DT		
2	20160602	2호선	시청	30880	30828	20160610		
3	20160602	2호선	을지로입구	57209	58596	20160610		
4	20160602	2호선	을지로3가	24387	24229	20160610		
5	20160602	2호선	을지로4가	15323	15287	20160610		
6	20160602	2호선	동대문역사	19546	22779	20160610		
7	20160602	2호선	신당	17218	18021	20160610		
8	20160602	2호선	상왕십리	12541	11995	20160610		
9	20160602	2호선	왕십리(성동	23698	18938	20160610		
10	20160602	2호선	한양대	17187	20950	20160610		
11	20160602	2호선	뚝섬	19250	20615	20160610		
12	20160602	2호선	성수	31581	34736	20160610		
13	20160602	2호선	건대입구	48240	52233	20160610		
14	20160602	2호선	구의	29121	28307	20160610		
15	20160602	2호선	강변	50637	48022	20160610		
16	20160602	2호선	잠실나루	22320	21435	20160610		
17	20160602	2호선	잠실	87026	81489	20160610		
18	20160602	2호선	신천	30621	29614	20160610		
19	20160602	2호선	종합운동장	20559	24153	20160610		
20	20160602	2호선	삼성	63026	66411	20160610		
21	20160602	2호선	선릉	69994	60009	20160610		
22	20160602	2호선	역삼	55506	63197	20160610		
23	20160602	2호선	강남	108616	108737	20160610		
24	20160602	2호선	교대	47823	52972	20160610		
25	20160602	2호선	서초	25908	26568	20160610		

ple-2.csv

```
USE_DT,LINE_NUM,SUB_STA_NM,RIDE_PASGR_NUM,ALIGHT_PASGR_NUM,
  WORK_DT
20160602,2호선,시청,30880,30828,20160610
20160602,2호선,을지로입구,57209,58596,20160610
20160602,2호선,을지로3가,24387,24229,20160610
20160602,2호선,을지로4가,15323,15287,20160610
20160602,2호선,동대문역사문화공원,19546,22779,20160610
20160602,2호선,신당,17218,18021,20160610
20160602,2호선,상왕십리,12541,11995,20160610
20160602,2호선,왕십리(성동구청),23698,18938,20160610
20160602,2호선,한양대,17187,20950,20160610
20160602,2호선,뚝섬,19250,20615,20160610
20160602,2호선,성수,31581,34736,20160610
20160602,2호선,건대입구,48240,52233,20160610
20160602,2호선,구의,29121,28307,20160610
20160602,2호선,강변,50637,48022,20160610
20160602,2호선,잠실나루,22320,21435,20160610
20160602,2호선,잠실,87026,81489,20160610
20160602,2호선,신천,30621,29614,20160610
20160602,2호선,종합운동장,20559,24153,20160610
20160602,2호선,삼성,63026,66411,20160610
20160602,2호선,선릉,69994,60009,20160610
20160602,2호선,역삼,55506,63197,20160610
20160602,2호선,강남,108616,108737,20160610
20160602,2호선,교대,47823,52972,20160610
20160602,2호선,서초,25908,26568,20160610
```

Using NumPy and Pandas

- NumPy와 Pandas는 수치분석 및 데이터 분석을 위한 쉬운 도구를 제공해준다. (=> compared to R or Matlab)

```
import pandas as pd
daily_engagement =
    pd.read_csv('daily_engagement.csv')
len(daily_engagement['acct'].unique())
```

Using NumPy and Pandas

- NumPy는 데이터의 연산에 도움을 준다

```
import numpy as np
total_minutes =
    total_minutes_by_account.values()
print('Mean:',
    np.mean(list(total_minutes)))
print('Standard deviation:',
    np.std(list(total_minutes)))
```

NumPy

- Numpy는 Numerical Python의 약자로 이름에서 알수 있듯이 파이썬에서 과학적 계산을 하기 위해 수치연산기능을 제공함.
- 고성능 다차원 배열 객체와 이들과 함께 사용할 수 있는 다양한 수치연산 메소드를 제공하여 파이썬에서 Matlab 혹은 R 과 같은 기능을 사용할 수 있게 함.

NumPy

- NumPy는 고성능 연산을 위해 자체적으로 데이터구조를 제공하는데 파이썬이 기본적으로 제공하는 데이터구조와 유사점/차이점은 다음과 같다.
- 유사점
  - index를 사용하여 요소에 접근할 수 있다.

    ```
    a = ['a', 'b', 'c', 'd', 'e']
    a[3] → ‘d’
    ```

  - range를 사용하여 요소에 접근할 수 있다.

    ```
    a[1:3] → ['b', 'c']
    ```

  - loop를 사용할 수 있다

    ```
    for x in a:
    ```

NumPy

- 차이점
  - 하나의 array에는 같은 type의 데이터만 담을 수 있다.
    - array can holds `string`, `int`, `float64`, `boolean`, etc.
- array와 함께 사용할 수 있는 손쉬운 수치연산 메소드 들을 제공한다.
  - `std()`, `mean()`, `log()`, `sin()`, etc.
- 다차원의 array를 만들 수 있다.
  - 2D Array, 3D Array, etc.

Questions…?