파이썬을 활용한 빅데이터 분석 기초

- awekim@handong.edu

파이썬 실습 환경 만들기

- Untitled0.ipynb를 확인하세요!

1. 파이썬 활용을 위한 핵심 개념

```
# 1과 '1'은 다르다.
1+1
```

```
'1'+'1'
```

```
1+'1'
```

```
# 문자 a와 변수 a는 다르다
'a'
```

```
a
```

```
# True와 False의 차이
True == 1
```

```
False == 0
```

```
sum([True, True, False])
```

```
# 매트릭스와 데이터프레임의 차이
import numpy as np
mat = np.arange(0,9).reshape(3,3)
mat
```

```
import pandas as pd
dataframe = pd.DataFrame({'변수1':[0,3,6],
                          '변수2':[1,4,7],
                          '변수3':[2,5,8]})

dataframe
```

```
# 행렬 전체 값 중에서 최대값
mat.max()
```

```
# 데이터프레임 열별 최대값
dataframe.max()
```

```
# 행렬 전체 값의 총합
mat.sum()
```

```
# 데이터프레임 열별 총합
dataframe.sum()
```

```
# 인덱싱과 슬라이싱
a = [0, 1, 2, 3, 4, 5]
a
```

```
# a의 0번째부터 2번째 값
a[0:3]
```

```
# a의 처음부터 2번째 값
a[:3]
```

```
# a의 2번째부터 끝까지
a[2:]
```

```
# a의 값 전체
a[:]
```

```
# 컴퓨터가 코드를 작동시키는 방식 = 사람이 글을 읽는 방식
# <Version 1>
a = 1 + 1
print(a)
```

```
# <Version 2>
print(b)
b = 1 + 1
```

```
# 파이썬을 잘 사용하는 사람 = 파이썬에서 제공되는 패키지를 잘 사용하는 사람
import pandas
```

```
import pandas as pd
```

```
from pandas import DataFrame
```

```
a = pd
b = a
a = a.DataFrame
```

```
a
```

```
b
```

```
a == b
```

2. 빅데이터 분석 1단계: 데이터 수집

```
import pandas as pd
kor_df = pd.read_csv("/content/drive/MyDrive/CoWeek_BigDataPython/Korea_data/KoreaIncomeWelfare.c
```

```
# 첫 5줄 확인하기
kor_df.head()
```

```
# 데이터프레임의 행과 열 수 확인하기
# 92857 개의 행과 14개의 열
kor_df.shape
```

```
# 데이터프레임의 열 수 확인하기
kor_df.columns
```

2. 빅데이터 분석 2단계: 데이터 전처리

```
# 데이터 탐색
# 연도별로 묶기
kor_df_yr = kor_df.groupby('year')
kor_df_yr
```

```
kor_df_yr.size()
```

```
# 지역별로 묶기
kor_df_reg = kor_df.groupby('region')
kor_df_reg
```

```
kor_df_reg.size()
```

```
# 연도별 평균 가족 수 살펴보기
kor_df_yr_sum = kor_df_yr['family_member'].mean().reset_index()
kor_df_yr_sum
```

```
import seaborn as sns
sns.lineplot(x='year', y='family_member', data=kor_df_yr_sum)
```

```
# 교육 수준과 기업 크기간의
kor_df_edu = kor_df.groupby('education_level')
kor_df_edu_sum = kor_df_edu['company_size'].mean().reset_index()
kor_df_edu_sum
```

```
import seaborn as sns
sns.barplot(x='education_level', y='company_size', data=kor_df_edu_sum)
```

```
# 결측치 존재 여부 확인
# pd.isnull(): 데이터프레임 내 전체 값을 True 또는 False로 반환
kor_df.isnull()
```

```
kor_df.isnull().any()
```

```
kor_df.isnull().sum()
```

```
# occupation 열의 결측치 확인
kor_df['occupation'].isnull()
```

```
# occupation 열의 결측치를 포함한 행 확인
kor_df[kor_df['occupation'].isnull()]
```

```
# reason_none_worker 열의 결측치를 포함한 행 확인
kor_df[kor_df['reason_none_worker'].isnull()]
```

```
# 결측치 처리: 제외하기
# 결측치 있는 열 제외하기
kor_df.dropna(axis=1)
```

```
# 결측치 있는 행 제외하기
kor_df.dropna(axis=0)
```

```
# occupation 열에 결측치 있는 행 제외하기
kor_df.dropna(axis=0, subset=['occupation'])
```

```
# 결측치 처리: 채워넣기
import numpy as np
kor_df.replace(to_replace = np.nan, value = -99)
```

```
# 이상치 검색
kor_df[['income','family_member','year_born']].describe()
```

```
import seaborn as sns
sns.displot(kor_df['income'], kde=False, bins=13, color='red')
```

```
import seaborn as sns
sns.histplot(kor_df['year_born'])
```

```
# 연수입 최대값: 11,600,000,000 만
kor_df['income'].max()
```

```
# 연수입 최대값을 포함하는 행 찾기
kor_df.loc[kor_df['income']==kor_df['income'].max()]
```

```
# 연수입 최대값을 포함하는 id의 값 찾기
kor_df.loc[kor_df['id']==98000701]
```

```
# 전처리 완료된 최종 데이터 셋
import pandas as pd
kor_df = pd.read_csv("/content/drive/MyDrive/CoWeek_BigDataPython/Korea_data/KoreaIncomeWelfare.c

kor_df_ed = kor_df.dropna(axis=0, subset=['occupation'])
kor_df_ed.dropna(axis=1, inplace=True)
kor_df_ed = kor_df_ed.loc[kor_df_ed['id']!=98000701]
```

4. 빅데이터 분석 3단계: 데이터 분석

```
kor_df_ed
```