데이터시각화이해와실습

Lecture 13. 행복지수 데이터 분석

동덕여자대학교
데이터사이언스 전공
권 범

목차

❖ 01. 분석 대상 데이터 수집

❖ 02. 데이터 가공

❖ 03. 데이터 분석 및 시각화

01. 분석 대상 데이터 수집

02. 데이터 가공

03. 데이터 분석 및 시각화

01. 분석 대상 데이터 수집

❖ 시작하기 전에 (1/3)

- 대한민국은 반세기 동안 엄청난 경제성장을 이뤄낸 반면,
  자살률 및 노인 빈곤률 증가 등 어두운 면이 존재함
- 양적인 경제 번영과 달리 사회의 질적인 발전과 행복은 그렇지 못했음

01. 분석 대상 데이터 수집

❖ 시작하기 전에 (2/3)

- 건강은 정신적으로나 육체적으로 아무 탈없이 튼튼한 상태를 말하는 데,
  행복은 건강에 영향을 주는 중요한 요소임
- 이번 수업에서는 삶에 영향을 주는 중요 요소들을 포함하여 측정한 행복지수 자료를
  분석 대상 데이터로 정하여 수집하고, 가공하여 요소 간의 상관관계를 분석해 보자

01. 분석 대상 데이터 수집

❖ 시작하기 전에 (3/3)

- 고려대학교 정부학연구소 연구팀에서 수행한
  '행복지표체계 구축 기반 연구'를 바탕으로 구축한
  대한민국 행복지도 홈페이지에서 행복지수 데이터를 수집해 보자

01. 분석 대상 데이터 수집

❖ 대한민국 행복지도 사이트에서 수집 (1/7)

- 대한민국 행복지도 사이트(http://happykorea.re.kr/)에 접속해,
  화면 상단 메뉴에서 [2019 행복지도] → [삶의만족도]를 클릭

[사진출처] 데이터 분석을 위한 전처리와 시각화 with 파이썬 (출판사: 길벗캠퍼스)

01. 분석 대상 데이터 수집

❖ 대한민국 행복지도 사이트에서 수집 (2/7)

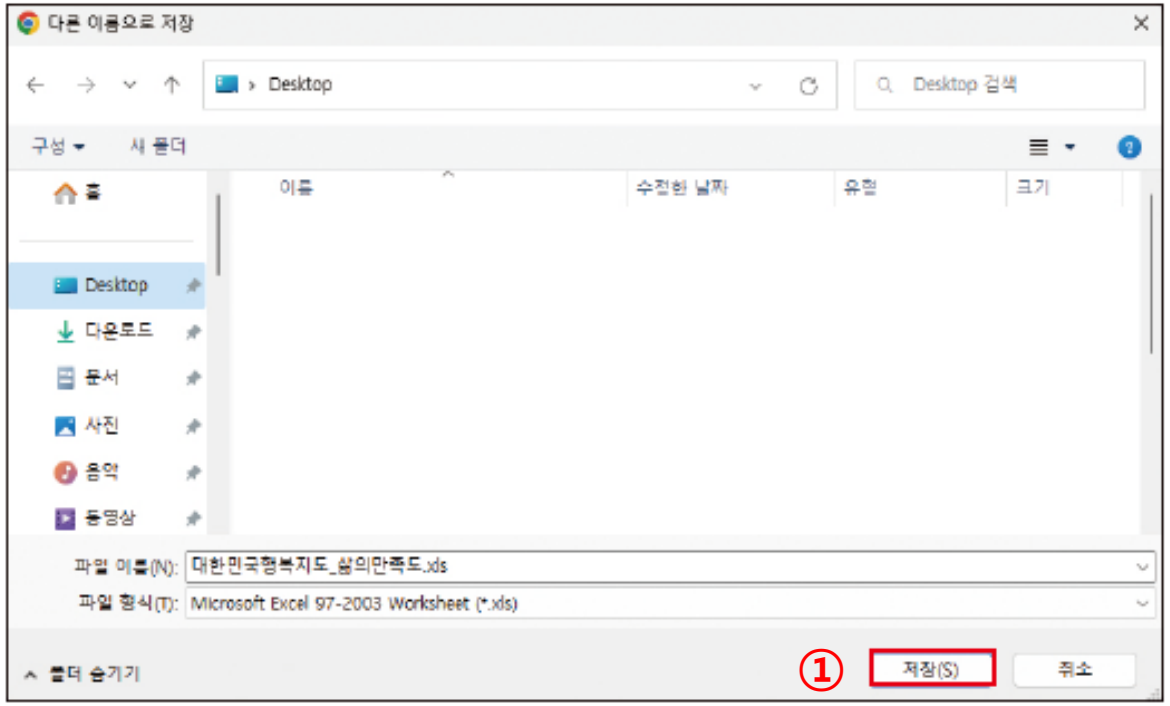
- [삶의만족도] 화면에서 [엑셀다운로드] 버튼을 클릭

[사진출처] 데이터 분석을 위한 전처리와 시각화 with 파이썬 (출판사: 길벗캠퍼스)

01. 분석 대상 데이터 수집

❖ 대한민국 행복지도 사이트에서 수집 (3/7)

- [다른 이름으로 저장] 대화 상자가 표시되면,
  원하는 저장 위치를 지정하고 [저장] 버튼을 클릭하여 저장함



[사진출처] 데이터 분석을 위한 전처리와 시각화 with 파이썬 (출판사: 길벗캠퍼스)

01. 분석 대상 데이터 수집

❖ 대한민국 행복지도 사이트에서 수집 (4/7)

- 엑셀을 실행하고 다운로드한 파일을 불러오면,
  전국의 구군별 삶의 만족도를 확인할 수 있음

	A	B	C	D	E	F	G
1	No	시도	구군	삶의 만족도			
2	1	서울특별시	종로구	0.4437			
3	2	서울특별시	중구	0.4976			
4	3	서울특별시	용산구	0.6161			
5	4	서울특별시	성동구	0.4729			
6	5	서울특별시	광진구	0.4041			
7	6	서울특별시	동대문구	0.5842			
8	7	서울특별시	중랑구	0.1058			
9	8	서울특별시	성북구	0.6382			
10	9	서울특별시	강북구	0.0461			

⋮

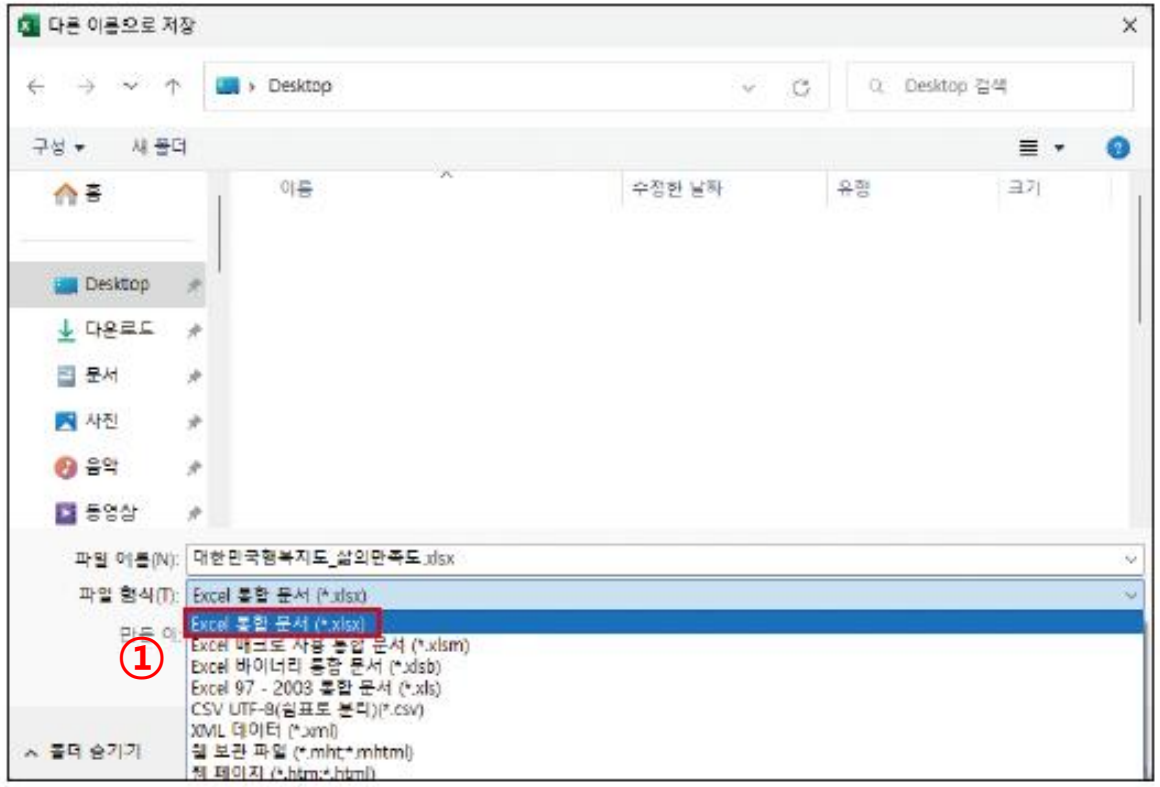
223	222	경상남도	남해군	0.3203
224	223	경상남도	하동군	0.6993
225	224	경상남도	산청군	0.9355
226	225	경상남도	함양군	0.9565
227	226	경상남도	거창군	0.6163
228	227	경상남도	합천군	0.8057
229	228	제주특별자치도	제주시	0.7113
230	229	제주특별자치도	서귀포시	0.7113

[사진출처] 데이터 분석을 위한 전처리와 시각화 with 파이썬 (출판사: 길벗캠퍼스)

01. 분석 대상 데이터 수집

❖ 대한민국 행복지도 사이트에서 수집 (5/7)

- 다운로드한 파일의 형식은 'Microsoft Excel 97-2023 Worksheet.xls(*.xls)'임
- 이 형식의 파일은 코랩에 업로드하여 읽으면 오류가 발생하기 때문에 파일 형식을 'Excel 통합 문서(*.xlsx)'로 변경해야 함



[사진출처] 데이터 분석을 위한 전처리와 시각화 with 파이썬 (출판사: 길벗캠퍼스)

01. 분석 대상 데이터 수집

❖ 대한민국 행복지도 사이트에서 수집 (6/7)

- 동일한 방법으로 '건강', '안전', '환경', '경제', '교육', '관계 및 사회참여', '여가 요소'를 선택하여 각각의 엑셀 파일을 다운로드하자
- 엑셀 파일을 하나씩 열어, 파일 형식을 'Excel 통합 문서(*.xlsx)'로 지정하여 다시 저장하자

01. 분석 대상 데이터 수집

❖ 대한민국 행복지도 사이트에서 수집 (7/7)

- 각 파일의 지표 정보 산출 내용은 대한민국 행복지도 사이트를 참고하자

파일명	열 정보(지표 정보)
대한민국행복지도_삶의만족도.xlsx	자신의 삶(과거와 현재)에 만족하는 정도
대한민국행복지도_건강.xlsx	주관적 건강수준 인지율, 인구 십만 명당 정신건강증진기관 수, 인구 천 명당 의료기관 종사 의사 수, 건강생활 실천률, 인구 천 명당 의료기관 병상 수
대한민국행복지도_안전.xlsx	사회안전에 대한 인식, 인구 천 명당 cctv 대수, 인구 십만 명당 응급의료기관 및 응급실 운영기관 수, 단위면적당 지역경찰관서 수, 지역안전등급 현황 중 '교통사고 및 화재'
대한민국행복지도_환경.xlsx	환경체감도, 인구 천 명당 1일 산업폐수 방류량, 도시지역 중 '녹지지역 비율', 미세먼지(PM2.5), 주민 1인당 생활폐기물 배출량

파일명	열 정보(지표 정보)
대한민국행복지도_경제.xlsx	1인당 지역내총생산(GRDP), 인구 천 명당 사업체 수, 인구 천 명당 종사자 수, 국민기초생활보장 수급자 비율, 종사자 천 명당 영세자영업자 수
대한민국행복지도_교육.xlsx	교원 1인당 학생 수, 영유아 천 명당 보육시설 수, 인구 십만 명당 학교 수, 인구 천 명당 사설학원 수
대한민국행복지도_관계및사회참여.xlsx	인구 십만 명당 자살률, 1인가구(독거노인 제외) 비율, 독거노인가구 비율, 인구 십만 명당 사회적기업 수, 가족관계 만족도
대한민국행복지도_여가.xlsx	여가활용 만족도, 노인 천 명당 노인여가복지시설 수, 인구 십만 명당 도서관 수, 인구 십만 명당 문화기반시설 수, 인구 천 명당 체육관련 여가시설 수

[사진출처] 데이터 분석을 위한 전처리와 시각화 with 파이썬 (출판사: 길벗캠퍼스)

02. 데이터 가공

01. 분석 대상 데이터 수집
03. 데이터 분석 및 시각화

02. 데이터 가공

❖ 데이터를 읽어와서 확인하기: ① 삶의 만족도

- 행복지수 8개 요소 데이터를 가져오기 위해 각각의 엑셀 파일을 읽어 들이고, 첫 5행의 데이터를 출력하여 필요 데이터를 확인하자

```
1 import pandas as pd
2
3 happy_life = pd.read_excel("대한민국행복지도_삶의만족도.xlsx")
4 happy_life.head()
```

실행결과

	No	시도	구군	삶의 만족도
0	1	서울특별시	종로구	0.4437
1	2	서울특별시	중구	0.4976
2	3	서울특별시	용산구	0.6161
3	4	서울특별시	성동구	0.4729
4	5	서울특별시	광진구	0.4041

02. 데이터 가공

❖ 데이터를 읽어와서 확인하기: ② 건강

```
1 happy_health = pd.read_excel("대한민국행복지도_건강.xlsx")
2 happy_health.head()
```

실행결과

	No	시도	구군	평균	주관적 건강수준 인지율	인구 십만명당 정신건강증진기관 수	인구 천명당 의료기관 종사 의사수	건강생활실천율	인구 천명당 의료기관병상수
0	1	서울특별시	종로구	0.9220	0.8424	0.6914	1.0000	0.9697	0.7616
1	2	서울특별시	중구	0.6742	0.5772	0.4106	0.9995	0.9669	0.4043
2	3	서울특별시	용산구	0.5898	0.9819	0.3353	0.6046	0.9844	0.1433
3	4	서울특별시	성동구	0.4794	0.5465	0.3321	0.5783	0.9776	0.2111
4	5	서울특별시	광진구	0.6373	0.8534	0.7393	0.6352	0.8022	0.1936

02. 데이터 가공

❖ 데이터를 읽어와서 확인하기: ③ 안전

```
1 happy_safe = pd.read_excel("대한민국행복지도_안전.xlsx")
2 happy_safe.head()
```

실행결과

	No	시도	구군	평균	사회안전에 대한 인식 b)	인구 천명당 cctv 대수 b)	인구 십만명당 응급의료기관 및 응급실 운영기관수	단위면적당 지역경찰관서 수	지역안전등급 현황 중 '교통사고 및 화재' a)
0	1	서울특별시	종로구	0.7470	0.9965	0.4796	0.7498	0.9998	0.0162
1	2	서울특별시	중구	0.9320	0.9707	0.8214	0.6938	1.0000	0.3014
2	3	서울특별시	용산구	0.5537	0.6641	0.4311	0.1768	0.8923	0.5075
3	4	서울특별시	성동구	0.5347	0.7452	0.4330	0.1562	0.9800	0.3014
4	5	서울특별시	광진구	0.6072	0.7179	0.2196	0.2019	0.9785	0.7117

02. 데이터 가공

❖ 데이터를 읽어와서 확인하기: ④ 환경

```
1 happy_environ = pd.read_excel("대한민국행복지도_환경.xlsx")
2 happy_environ.head()
```

실행결과

	No	시도	구군	평균	환경체감도 b)	인구 천명당 1일 산업폐수 방류량 a)	도시지역 중 '녹지지역 비율'	미세먼지(PM2.5) a) b)	주민 1인당 생활폐기물배출량 a)
0	1	서울특별시	종로구	0.4637	0.4658	0.7223	0.2012	0.7704	0.0006
1	2	서울특별시	중구	0.2865	0.1077	0.6957	0.0013	0.6617	0.0000
2	3	서울특별시	용산구	0.5030	0.3142	0.7185	0.1349	0.6895	0.4571
3	4	서울특별시	성동구	0.4196	0.1480	0.6868	0.0347	0.3757	0.7426
4	5	서울특별시	광진구	0.4992	0.1229	0.6940	0.0574	0.7235	0.7016

02. 데이터 가공

❖ 데이터를 읽어와서 확인하기: ⑤ 경제

```
1 happy_econo = pd.read_excel("대한민국행복지도_경제.xlsx")
2 happy_econo.head()
```

실행결과

	No	시도	구군	평균	1인당 지역내총생산(GRDP)	인구 천명당 사업체수	인구 천명당 종사자수	국민기초생활보장 수급자비율 a)	종사자 천명당 영세자영업자 수 a)
0	1	서울특별시	종로구	1.0000	1.0000	1.0000	1.0000	0.7564	0.9037
1	2	서울특별시	중구	0.9806	1.0000	1.0000	1.0000	0.6718	0.9177
2	3	서울특별시	용산구	0.6915	0.6334	0.5493	0.7282	0.7910	0.8341
3	4	서울특별시	성동구	0.6533	0.5132	0.5368	0.6849	0.8243	0.8377
4	5	서울특별시	광진구	0.4445	0.2906	0.3424	0.3870	0.8681	0.7480

02. 데이터 가공

❖ 데이터를 읽어와서 확인하기: ⑥ 교육

```
1 happy_edu = pd.read_excel("대한민국행복지도_교육.xlsx")
2 happy_edu.head()
```

실행결과

	No	시도	구군	평균	교원 1인당 학생수	영유아 천명당 보육시설 수	인구 십만명당 학교수	인구 천명당 사설학원수
0	1	서울특별시	종로구	0.6839	0.8505	0.6248	0.2249	0.8126
1	2	서울특별시	중구	0.5013	0.8729	0.4147	0.2294	0.2944
2	3	서울특별시	용산구	0.2679	0.6479	0.3804	0.1281	0.1629
3	4	서울특별시	성동구	0.2464	0.7661	0.2270	0.1308	0.3366
4	5	서울특별시	광진구	0.4879	0.9309	0.4263	0.1278	0.4607

02. 데이터 가공

❖ 데이터를 읽어와서 확인하기: ⑦ 관계 및 사회참여

```
1 happy_relation = pd.read_excel("대한민국행복지도_관계및사회참여.xlsx")
2 happy_relation.head()
```

실행결과

	No	시도	구군	평균	인구 십만명당 자살률 a)	1인가구(독거노인 제외) 비율 a)	독거노인가구 비율 a)	인구 십만명당 사회적기업수	가족관계 만족도 b)
0	1	서울특별시	종로구	0.7425	0.8318	0.0057	0.7593	0.9981	0.5949
1	2	서울특별시	중구	0.4608	0.5462	0.0107	0.7254	1.0000	0.2329
2	3	서울특별시	용산구	0.4317	0.7288	0.0211	0.7754	0.7194	0.2009
3	4	서울특별시	성동구	0.4182	0.7288	0.0791	0.8332	0.6783	0.0937
4	5	서울특별시	광진구	0.3519	0.9434	0.0051	0.8541	0.2461	0.2058

02. 데이터 가공

❖ 데이터를 읽어와서 확인하기: ⑧ 여가

```
1 happy_leisure = pd.read_excel("대한민국행복지도_여가.xlsx")
2 happy_leisure.head()
```

실행결과

	No	시도	구군	평균	여가활용 만족도 b)	노인 천명당 노인여가복지시설수	인구 십만명당 도서관수	인구 십만명당 문화기반시설수	인구 천명당 체육관련 여가시설수
0	1	서울특별시	종로구	0.6331	0.8409	0.1573	0.4523	0.9997	0.5559
1	2	서울특별시	중구	0.6691	0.6224	0.1467	0.5369	0.8441	0.9886
2	3	서울특별시	용산구	0.2817	0.6381	0.1628	0.2453	0.3715	0.2934
3	4	서울특별시	성동구	0.3257	0.5657	0.1859	0.3590	0.2768	0.4859
4	5	서울특별시	광진구	0.3313	0.6740	0.1443	0.2333	0.2058	0.6365

행복지수 8개 요소 간의 상관관계 분석을 위해 각 요소에서 평균 수치를 사용하자

02. 데이터 가공

❖ 데이터 병합하기: ① 시도 데이터 추출

- 행복지수 8개 요소 데이터는 각각의 데이터프레임 변수에 저장되어 있음
- 요소마다 데이터프레임 내 평균 수치를 가져와 새로운 데이터프레임에 병합해 보자

```
1 city = list(happy_life["시도"].unique())
2 happy_merge = pd.DataFrame({"시도": city})
3 happy_merge
```

실행결과

	시도
0	서울특별시
1	부산광역시
2	대구광역시
3	인천광역시
4	광주광역시
5	대전광역시
6	울산광역시
7	세종특별자치시
8	경기도
9	강원도
10	충청북도
11	충청남도
12	전라북도
13	전라남도
14	경상북도
15	경상남도
16	제주특별자치도

✓ happy_life 데이터프레임에서 중복되지 않은 '시도' 데이터를, unique( ) 함수를 사용하여 추출하고, 이를 city에 저장하자
✓ 그리고 중복되지 않은 '시도' 데이터로 새로운 데이터프레임을 생성하자

'시도'를 기준으로 데이터프레임을 병합해 보자

02. 데이터 가공

❖ 데이터 병합하기: ② '삶의 만족도' 데이터 병합

- happy_life에서 '삶의 만족도' 열의 데이터를 가져와 mean( ) 함수로 '시도'별 평균을 구한 후, life 변수에 저장하자
- 그다음 '시도'를 기준으로 happy_merge와 life를 병합하자

```
1 life = happy_life["삶의 만족도"].groupby(by=happy_life["시도"]).mean()
2 happy_merge = pd.merge(happy_merge, life, on="시도")
3 happy_merge.head()
```

실행결과

	시도	삶의 만족도
0	서울특별시	0.490972
1	부산광역시	0.362081
2	대구광역시	0.363988
3	인천광역시	0.411480
4	광주광역시	0.484480

02. 데이터 가공

❖ 데이터 병합하기: ③ '건강' 데이터 병합

- happy_health에서 '평균' 열의 데이터를 가져와 mean( ) 함수로 '시도'별 평균을 구한 후, health 변수에 저장하자
- rename( ) 함수를 이용하여 health 변수의 '평균'을 '건강'으로 변경한 다음, '시도'를 기준으로 happy_merge와 life를 병합하자

```
1 health = happy_health["평균"].groupby(by=happy_health["시도"]).mean()
2 happy_merge = pd.merge(happy_merge, health.rename("건강"), on="시도")
3 happy_merge.head()
```

실행결과

	시도	삶의 만족도	건강
0	서울특별시	0.490972	0.569532
1	부산광역시	0.362081	0.511906
2	대구광역시	0.363988	0.482325
3	인천광역시	0.411480	0.339620
4	광주광역시	0.484480	0.632300

02. 데이터 가공

❖ 데이터 병합하기: ④ '안전' 데이터 병합

- happy_safe에서 '평균' 열의 데이터를 가져와 mean( ) 함수로 '시도'별 평균을 구한 후, safe 변수에 저장하자
- rename( ) 함수를 이용하여 safe 변수의 '평균'을 '안전'으로 변경한 다음, '시도'를 기준으로 happy_merge와 safe를 병합하자

```
1 safe = happy_safe["평균"].groupby(by=happy_safe["시도"]).mean()
2 happy_merge = pd.merge(happy_merge, safe.rename("안전"), on="시도")
3 happy_merge.head()
```

실행결과

	시도	삶의 만족도	건강	안전
0	서울특별시	0.490972	0.569532	0.552256
1	부산광역시	0.362081	0.511906	0.404875
2	대구광역시	0.363988	0.482325	0.358429
3	인천광역시	0.411480	0.339620	0.421020
4	광주광역시	0.484480	0.632300	0.266440

02. 데이터 가공

❖ 데이터 병합하기: ⑤ '환경' 데이터 병합

- happy_environ에서 '평균' 열의 데이터를 가져와 mean( ) 함수로 '시도'별 평균을 구한 후, environ 변수에 저장하자
- rename( ) 함수를 이용하여 environ 변수의 '평균'을 '환경'으로 변경한 다음, '시도'를 기준으로 happy_merge와 environ을 병합하자

```
1 environ = happy_environ["평균"].groupby(by=happy_environ["시도"]).mean()
2 happy_merge = pd.merge(happy_merge, environ.rename("환경"), on="시도")
3 happy_merge.head()
```

실행결과

	시도	삶의 만족도	건강	안전	환경
0	서울특별시	0.490972	0.569532	0.552256	0.470712
1	부산광역시	0.362081	0.511906	0.404875	0.448719
2	대구광역시	0.363988	0.482325	0.358429	0.552500
3	인천광역시	0.411480	0.339620	0.421020	0.515020
4	광주광역시	0.484480	0.632300	0.266440	0.607480

02. 데이터 가공

❖ 데이터 병합하기: ⑥ '경제' 데이터 병합

- happy_econo에서 '평균' 열의 데이터를 가져와 mean( ) 함수로 '시도'별 평균을 구한 후, econo 변수에 저장하자
- rename( ) 함수를 이용하여 environ 변수의 '평균'을 '경제'로 변경한 다음, '시도'를 기준으로 happy_merge와 econo를 병합하자

```
1 econo = happy_econo["평균"].groupby(by=happy_econo["시도"]).mean()
2 happy_merge = pd.merge(happy_merge, econo.rename("경제"), on="시도")
3 happy_merge.head()
```

실행결과

	시도	삶의 만족도	건강	안전	환경	경제
0	서울특별시	0.490972	0.569532	0.552256	0.470712	0.532820
1	부산광역시	0.362081	0.511906	0.404875	0.448719	0.438038
2	대구광역시	0.363988	0.482325	0.358429	0.552500	0.393975
3	인천광역시	0.411480	0.339620	0.421020	0.515020	0.410820
4	광주광역시	0.484480	0.632300	0.266440	0.607480	0.387380

02. 데이터 가공

❖ 데이터 병합하기: ⑦ '교육' 데이터 병합

- happy_edu에서 '평균' 열의 데이터를 가져와 mean( ) 함수로 '시도'별 평균을 구한 후, edu 변수에 저장하자
- rename( ) 함수를 이용하여 edu 변수의 '평균'을 '교육'으로 변경한 다음, '시도'를 기준으로 happy_merge와 edu를 병합하자

```
1 edu = happy_edu["평균"].groupby(by=happy_edu["시도"]).mean()
2 happy_merge = pd.merge(happy_merge, edu.rename("교육"), on="시도")
3 happy_merge.head()
```

실행결과

	시도	삶의 만족도	건강	안전	환경	경제	교육
0	서울특별시	0.490972	0.569532	0.552256	0.470712	0.532820	0.399412
1	부산광역시	0.362081	0.511906	0.404875	0.448719	0.438038	0.504594
2	대구광역시	0.363988	0.482325	0.358429	0.552500	0.393975	0.585838
3	인천광역시	0.411480	0.339620	0.421020	0.515020	0.410820	0.502920
4	광주광역시	0.484480	0.632300	0.266440	0.607480	0.387380	0.689680

02. 데이터 가공

❖ 데이터 병합하기: ⑧ '관계 및 사회참여' 데이터 병합

- happy_relation에서 '평균' 열의 데이터를 가져와 mean( ) 함수로 '시도'별 평균을 구한 후, relation 변수에 저장하자
- rename( ) 함수를 이용하여 relation 변수의 '평균'을 '관계'로 변경한 다음, '시도'를 기준으로 happy_merge와 relation을 병합하자

```
1 relation = happy_relation["평균"].groupby(by=happy_relation["시도"]).mean()
2 happy_merge = pd.merge(happy_merge, relation.rename("관계및사회참여"), on="시도")
3 happy_merge.head()
```

실행결과

	시도	삶의 만족도	건강	안전	환경	경제	교육	관계및사회참여
0	서울특별시	0.490972	0.569532	0.552256	0.470712	0.532820	0.399412	0.390656
1	부산광역시	0.362081	0.511906	0.404875	0.448719	0.438038	0.504594	0.294719
2	대구광역시	0.363988	0.482325	0.358429	0.552500	0.393975	0.585838	0.407486
3	인천광역시	0.411480	0.339620	0.421020	0.515020	0.410820	0.502920	0.504920
4	광주광역시	0.484480	0.632300	0.266440	0.607480	0.387380	0.689680	0.637800

02. 데이터 가공

❖ 데이터 병합하기: ⑨ '여가' 데이터 병합

- happy_leisure에서 '평균' 열의 데이터를 가져와 mean( ) 함수로 '시도'별 평균을 구한 후, leisure 변수에 저장하자
- rename( ) 함수를 이용하여 leisure 변수의 '평균'을 '여가'로 변경한 다음, '시도'를 기준으로 happy_merge와 leisure를 병합하자

```
1 leisure = happy_leisure["평균"].groupby(by=happy_leisure["시도"]).mean()
2 happy_merge = pd.merge(happy_merge, leisure.rename("여가"), on="시도")
3 happy_merge
```

실행결과

	시도	삶의 만족도	건강	안전	환경	경제	교육	관계및사회참여	여가
0	서울특별시	0.490972	0.569532	0.552256	0.470712	0.532820	0.399412	0.390656	0.286732
1	부산광역시	0.362081	0.511906	0.404875	0.448719	0.438038	0.504594	0.294719	0.153587
2	대구광역시	0.363988	0.482325	0.358429	0.552500	0.393975	0.585838	0.407486	0.234925
3	인천광역시	0.411480	0.339620	0.421020	0.515020	0.410820	0.502920	0.504920	0.244590
4	광주광역시	0.484480	0.632300	0.266440	0.607480	0.387380	0.689680	0.637800	0.454980

02. 데이터 가공

❖ 데이터 병합하기: ⑩ 병합 결과 확인

```
1 happy_merge
```

실행결과

	시도	삶의 만족도	건강	안전	환경	경제	교육	관계및사회참여	여가
0	서울특별시	0.490972	0.569532	0.552256	0.470712	0.532820	0.399412	0.390656	0.286732
1	부산광역시	0.362081	0.511906	0.404875	0.448719	0.438038	0.504594	0.294719	0.153587
2	대구광역시	0.363988	0.482325	0.358429	0.552500	0.393975	0.585838	0.407486	0.234925
3	인천광역시	0.411480	0.339620	0.421020	0.515020	0.410820	0.502920	0.504920	0.244590
4	광주광역시	0.484480	0.632300	0.266440	0.607480	0.387380	0.689680	0.637800	0.454980
5	대전광역시	0.407580	0.663580	0.206780	0.582380	0.416260	0.840780	0.510820	0.347960
6	울산광역시	0.471980	0.420040	0.477060	0.393440	0.642580	0.763240	0.676440	0.418420
7	세종특별자치시	0.907700	0.232000	0.157800	0.652400	0.511900	0.587000	0.673700	0.447800
8	경기도	0.426023	0.353952	0.325255	0.557977	0.468926	0.668619	0.601532	0.377561
9	강원도	0.619506	0.329506	0.548000	0.639839	0.386861	0.488533	0.563122	0.632750
10	충청북도	0.410100	0.401064	0.573155	0.466045	0.484755	0.592409	0.372782	0.702891
11	충청남도	0.553100	0.347433	0.481313	0.558173	0.402893	0.648047	0.448720	0.625347
12	전라북도	0.608193	0.421300	0.410914	0.670586	0.196543	0.446936	0.487464	0.667900
13	전라남도	0.547709	0.424732	0.565195	0.700573	0.202668	0.420019	0.481323	0.658668
14	경상북도	0.502223	0.264114	0.484835	0.614265	0.303355	0.428252	0.402183	0.490804
15	경상남도	0.530794	0.312722	0.475561	0.735600	0.325017	0.624022	0.395378	0.463750
16	제주특별자치도	0.711300	0.253700	0.446100	0.684200	0.425000	0.646800	0.610700	0.694800

17개 시도별 행복지수 8개 요소의 평균값을 확인할 수 있음

03. 데이터 분석 및 시각화

01. 분석 대상 데이터 수집
02. 데이터 가공

03. 데이터 분석 및 시각화

❖ 데이터 분석 (1/4)

- 앞서 가공한 데이터의 간단한 통계량과 집계 내용을 확인하여, 행복지수 8개 요소 사이의 상관관계를 분석해 보자
- 우선 데이터의 기초 통계량을, describe( ) 함수를 이용하여 확인해 보자

```
1 happy_merge.describe()
```

실행결과

	삶의 만족도	건강	안전	환경	경제	교육	관계및사회참여	여가
count	17.000000	17.000000	17.000000	17.000000	17.000000	17.000000	17.000000	17.000000
mean	0.518189	0.409401	0.420882	0.579406	0.407635	0.578653	0.497632	0.464910
std	0.138257	0.127636	0.123557	0.097299	0.111265	0.125408	0.114394	0.176206
min	0.362081	0.232000	0.157800	0.393440	0.196543	0.399412	0.294719	0.153587
25%	0.411480	0.329506	0.358429	0.515020	0.386861	0.488533	0.402183	0.347960
50%	0.490972	0.401064	0.446100	0.582380	0.410820	0.587000	0.487464	0.454980
75%	0.553100	0.482325	0.484835	0.652400	0.468926	0.648047	0.601532	0.632750
max	0.907700	0.663580	0.573155	0.735600	0.642580	0.840780	0.676440	0.702891

✓ describe( ) 함수를 이용하면 데이터 개수(count), 평균(mean), 표준편차(std), 최솟값(min), 사분위수(25%, 75%), 중앙값(50%), 최댓값(max)을 확인할 수 있음

03. 데이터 분석 및 시각화

❖ 데이터 분석 (2/4)

- mean( ) 함수를 이용하여 행복지수 요소별 평균을 확인해 보자
- 이때 axis(축)의 기준에 따라 행 방향으로 동작하게 할지, 열 방향으로 동작하게 할지를 지정할 수 있음
- mean(axis=0)은 행 방향으로 동작하여 평균을 구함
- mean(axis=1)은 열 방향으로 동작하여 평균을 구함

```
1 happy_merge.iloc[:, 1:].mean(axis=0)
```

실행결과

```
삶의 만족도          0.518189
건강                 0.409401
안전                 0.420882
환경                 0.579406
경제                 0.407635
교육                 0.578653
관계및사회참여       0.497632
여가                 0.464910
dtype: float64
```

행복지수 요소별 평균은 행 방향으로 계산하기 때문에, 여기서는 mean(axis=0)이라고 적음

데이터프레임에서 axis 매개변수의 기본값은 0이기 때문에, axis=0는 생략 가능함

❖ 데이터 분석 (3/4)

- 이번에는 mean( ) 함수를 이용하여 시도별 평균을 확인해 보자
- 시도별 평균은 열 방향으로 계산해야 하므로, mean(axis=1)이라고 적어야 함

```
1 happy_merge.iloc[:, 1:].mean(axis=1)
```

실행결과

```
0     0.461637
1     0.389815
2     0.422433
3     0.418799
4     0.520068
5     0.497017
   … (중략) …
11    0.508128
12    0.488729
13    0.500111
14    0.436254
15    0.482856
16    0.559075
dtype: float64
```

인덱스가 숫자로 되어 있어, '시도' 정보를 한눈에 확인하기 어려움. 어떻게 해야 할까?

❖ 데이터 분석 (4/4)

- set_index( )를 이용하여 인덱스를 '시도'로 지정하고, 시도별 평균을 다시 확인해 보자

```
1 happy_merge.set_index("시도").mean(axis=1)
```

실행결과

```
시도
서울특별시        0.461637
부산광역시        0.389815
대구광역시        0.422433
인천광역시        0.418799
광주광역시        0.520068
대전광역시        0.497017
        … (중략) …
충청북도          0.500400
충청남도          0.508128
전라북도          0.488729
전라남도          0.500111
경상북도          0.436254
경상남도          0.482856
제주특별자치도    0.559075
dtype: float64
```

03. 데이터 분석 및 시각화

❖ 행복지수 요소별 상관관계 확인

- corr( ) 함수를 이용하여 행복지수 요소별 상관관계를 확인해 보자

```
1 happy_merge.iloc[:, 1:].corr()
```

실행결과

	삶의 만족도	건강	안전	환경	경제	교육	관계및사회참여	여가
삶의 만족도	1.000000	-0.570466	-0.191709	0.572262	-0.057815	-0.113522	0.519122	0.497880
건강	-0.570466	1.000000	-0.210138	-0.357841	0.071809	0.242186	-0.175048	-0.405151
안전	-0.191709	-0.210138	1.000000	-0.124581	-0.142781	-0.519065	-0.444827	0.382730
환경	0.572262	-0.357841	-0.124581	1.000000	-0.716187	-0.146646	0.169838	0.494416
경제	-0.057815	0.071809	-0.142781	-0.716187	1.000000	0.451395	0.298562	-0.367461
교육	-0.113522	0.242186	-0.519065	-0.146646	0.451395	1.000000	0.462585	-0.070109
관계및사회참여	0.519122	-0.175048	-0.444827	0.169838	0.298562	0.462585	1.000000	0.216653
여가	0.497880	-0.405151	0.382730	0.494416	-0.367461	-0.070109	0.216653	1.000000

**상관관계는 두 데이터(변인) 사이의 함수 관계를 뜻하는 것이지,
인과관계로 판단해서는 안 됨!**

03. 데이터 분석 및 시각화

❖ 데이터 시각화: ① 선 그래프로 시각화 (1/2)

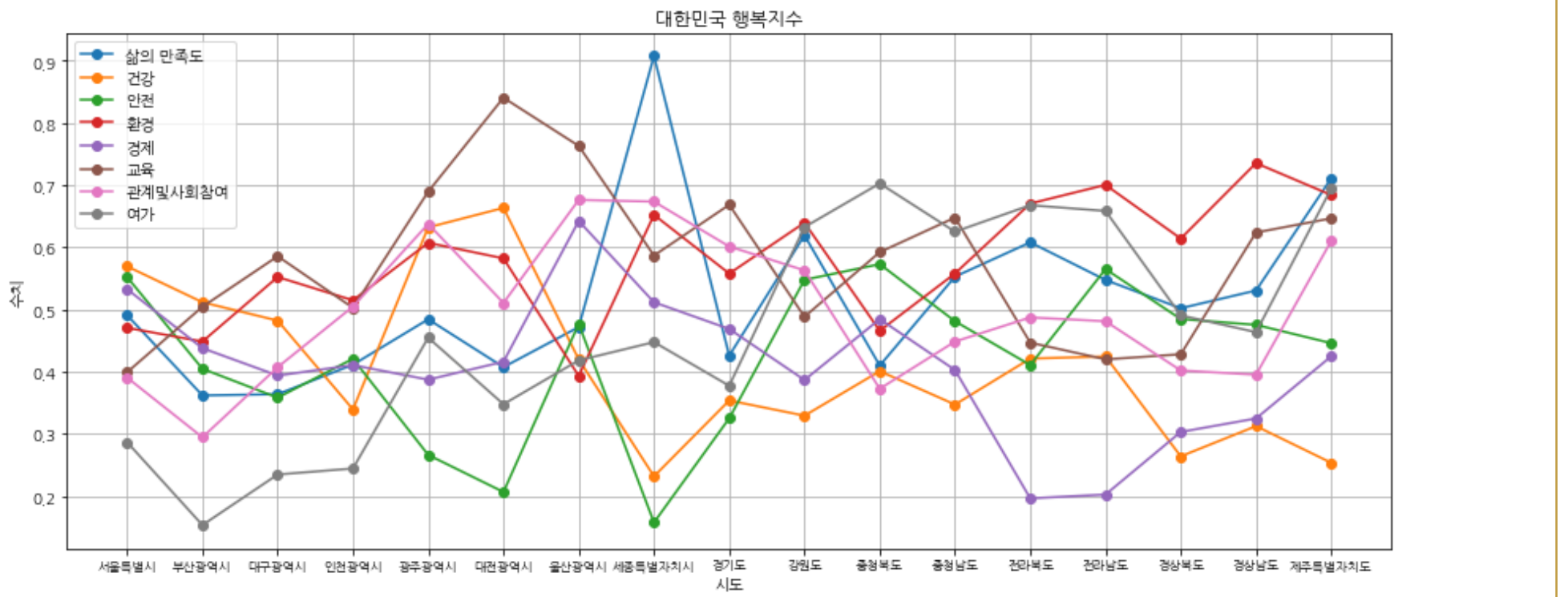
- 시도별 행복지수를 선 그래프로 시각화 해보자

```
1  import matplotlib.pyplot as plt
2
3  plt.rc("font", family="Malgun Gothic")
4
5  plt.figure(figsize=(15, 6))
6  column_name = list(happy_merge.columns[1:9])
7  for name in column_name:
8    data = happy_merge[name]
9    plt.plot(data, marker='o', label=name)
10
11 plt.title("대한민국 행복지수")
12 plt.xlabel("시도")
13 plt.ylabel("수치")
14 plt.xticks(range(0, 17, 1), happy_merge["시도"], fontsize=8)
15 plt.legend()
16 plt.grid()
17 plt.show()
```

03. 데이터 분석 및 시각화

❖ 데이터 시각화: ① 선 그래프로 시각화 (2/2)

실행결과



8개 요인을 그래프 하나에 모두 표시하다 보니, 가독성이 떨어짐

03. 데이터 분석 및 시각화

❖ 데이터 시각화: ② 막대 그래프로 시각화 (1/2)

- 이번에는 8개의 요소들을 구분하여 그래프를 그려보자

```
happy_merge.plot(kind="bar", subplots=True, figsize=(15, 20), grid=True,
                 ylim=(0, 1), xlabel="시도", ylabel="수치")

plt.suptitle("대한민국 행복지수", fontsize=25)
plt.tight_layout(pad=3, h_pad=2)
plt.xticks(range(0, 17, 1), city, rotation=360)
plt.show()
```

✓ DataFrame의 plot( )을 이용하면, DataFrame 내 데이터로부터 그래프를 그릴 수 있음
✓ kind 매개변수에는 그리고자 하는 그래프의 종류를 지정함("line", "bar", "barh", "hist" 등)
✓ suplots 매개변수는 기본값이 False이며, True로 지정하면 DataFrame의 컬럼별로 그래프를 그림
✓ ylim 매개변수에는 그래프 y축의 하한과 상한 값을 튜플로 지정함

✓ DataFrame의 plot(subplots=True)을 이용하여 그래프를 그리면 그래프마다 제목(Title)이 자동으로 표기되며, suptitle( ) 함수를 이용하면 전체 그래프의 제목을 표기할 수 있음

✓ tight_layout( ) 함수의 pad 매개변수에는 그래프 테두리와 subplots의 테두리 사이의 간격을 지정함
✓ h_pad 매개변수에는 인접한 subplots 사이의 간격을 지정함

03. 데이터 분석 및 시각화

❖ 데이터 시각화: ② 막대 그래프로 시각화 (2/2)

실행결과

03. 데이터 분석 및 시각화

❖ 데이터 시각화: ③ 히트맵 그래프로 시각화 (1/2)

- 앞서 corr( ) 함수를 이용하여 행복지수 요소별로 계산한 상관관계 데이터를 히트맵 그래프로 시각화 해보자

```
1  import seaborn as sns
2  import numpy as np
3
4  # plt.rcParams[“axes.unicode_minus”] = False
5  plt.rc(“axes”, unicode_minus=False)
6
7  plt.figure(figsize=(15, 13))
8  plt.title(“대한민국 행복지수 상관관계”, fontsize=25)
9
10 correlation_mat = happy_merge.iloc[:, 1:].corr()
11 upp_mat = np.triu(correlation_mat)        # upper triangular matrix -> triu
12 # low_mat = np.tril(correlation_mat)      # lower triangular matrix -> tril
13
14 sns.heatmap(correlation_mat, cmap=“RdYlGn”, annot=True, mask=upp_mat)
15 plt.show()
```

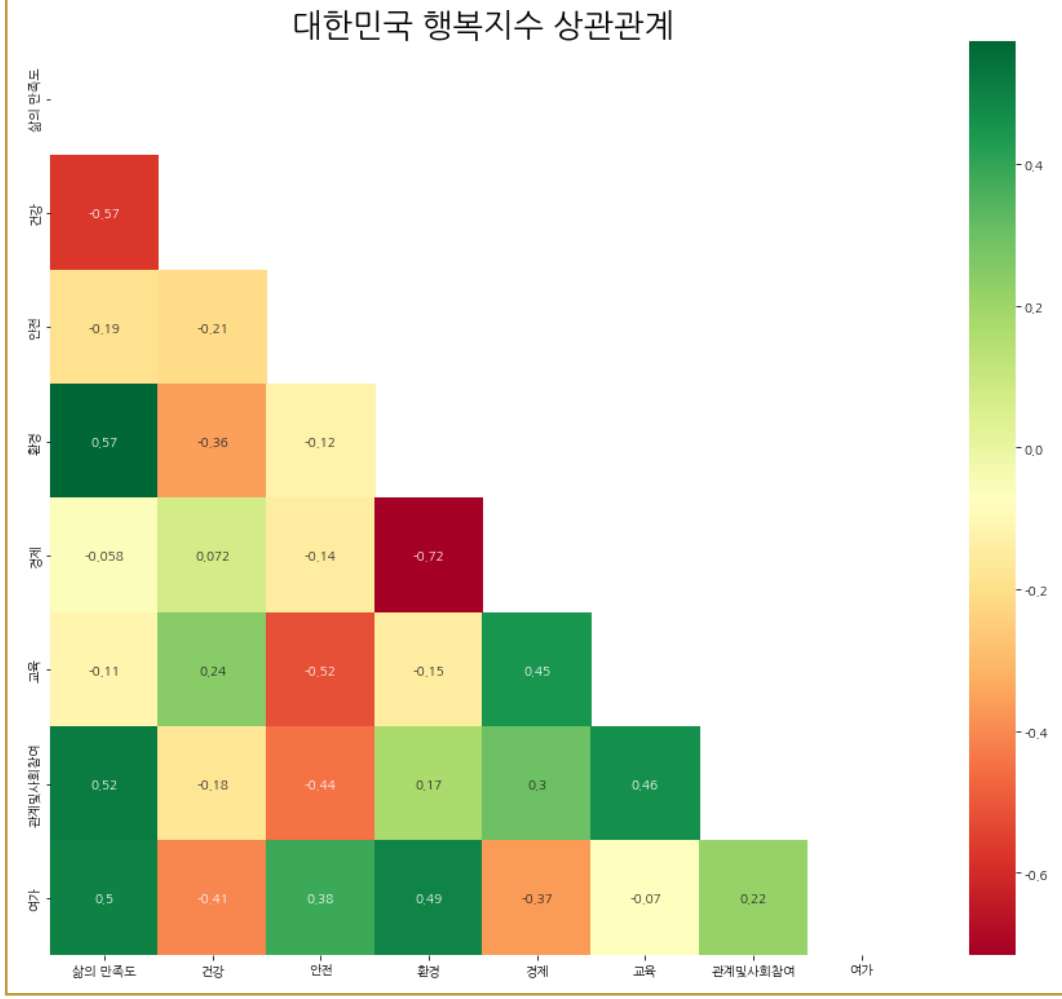
mask로 전달된 배열 내 True로 지정된 부분에 대해서, 히트맵 Cell에서 값을 안 보이게 함

triu( ) 함수를 이용하여 위쪽 상 삼각 행렬을 구하고, 이를 히트맵의 mask 매개변수에 지정함

03. 데이터 분석 및 시각화

❖ 데이터 시각화: ③ 히트맵 그래프로 시각화 (2/2)

실행결과



요소마다 지표들의 평균 값을 이용했기 때문에 이 수치를 토대로 행복지수를 분석하면 안 됨!

- ✓ 예를 들어 '교육' 요소의 경우,
  값이 작을 수록 좋은 '교원 1인당 학생수' 지표와
  값이 높을 수록 좋은 '인구 십만명당 학교수' 지표가 있음
- ✓ 수업에서는 이러한 지표들을 단순 평균하여, 실습하였음

끝맺음

- ❖ 01. 분석 대상 데이터 수집
- ❖ 02. 데이터 가공
- ❖ 03. 데이터 분석 및 시각화

THANK YOU!
Q & A

- Name: 권범
- Office: 동덕여자대학교 인문관 B821호
- Phone: 02-940-4752
- E-mail: bkwon@dongduk.ac.kr