

<4장> 행복지수 데이터 분석

학습 목표

- 행복지수 데이터 분석을 위한 분석 대상 데이터를 수집한다.
- 수집한 데이터를 목적에 따라 가공하고 데이터를 분석한다.
- 한글 폰트 설정 후 시각화하여 분석 결과를 해석한다.

목차

01 분석 대상 데이터 수집

02 데이터 가공

03 데이터 분석 및 시각화

01

분석 대상 데이터 수집

1. 데이터 시각화 이해

■ 대한민국 행복지도 사이트에서 수집

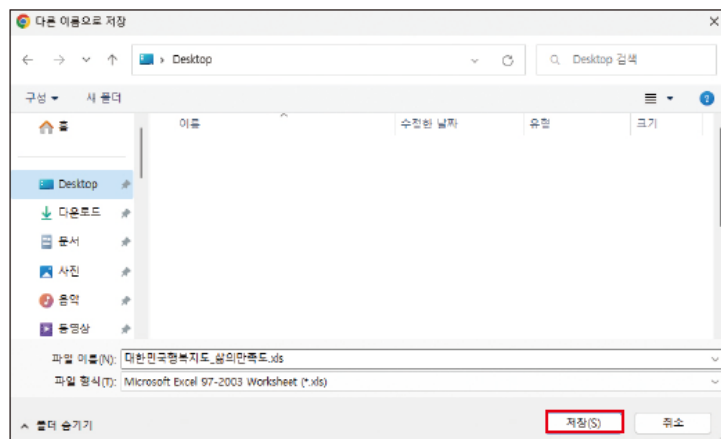
- (1) 대한민국 행복지도 사이트(<http://happykorea.re.kr/>)에 접속한다.
- 화면 상단 메뉴에서 '2019 행복지도' → '삶의만족도'를 클릭한다.
- (2) 삶의만족도의 <엑셀다운로드> 버튼을 클릭한다.



1. 데이터 시각화 이해

■ 대한민국 행복지도 사이트에서 수집

- (3) 저장 위치를 지정하고 <저장> 버튼을 클릭하여 저장한다.
- (4) 엑셀에서 파일을 불러오면 전국의 구군별 삶의 만족도를 확인할 수 있다.



	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	종랑구	0.1058			
9	8	서울특별시	성북구	0.6382			
10	9	서울특별시	강북구	0.0461			

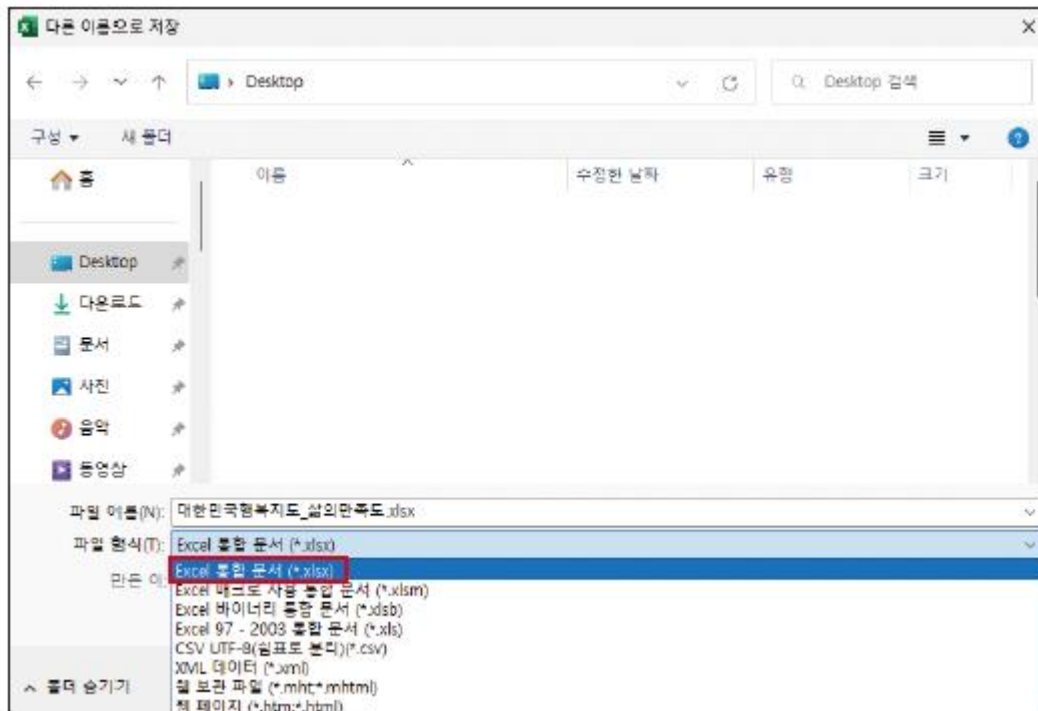
⋮

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			

1. 데이터 시각화 이해

■ 대한민국 행복지도 사이트에서 수집

- (5) 다운로드한 엑셀 파일을 'Excel 통합 문서(*.xlsx)'로 변경한다.
- (6) 같은 방법으로 건강, 안전, 환경, 경제, 교육, 관계 및 사회참여, 여가 요소를 선택하여 각각의 엑셀 파일을 다운로드



1. 데이터 시각화 이해

■ 대한민국 행복지도 사이트에서 수집

■ 데이터 정보

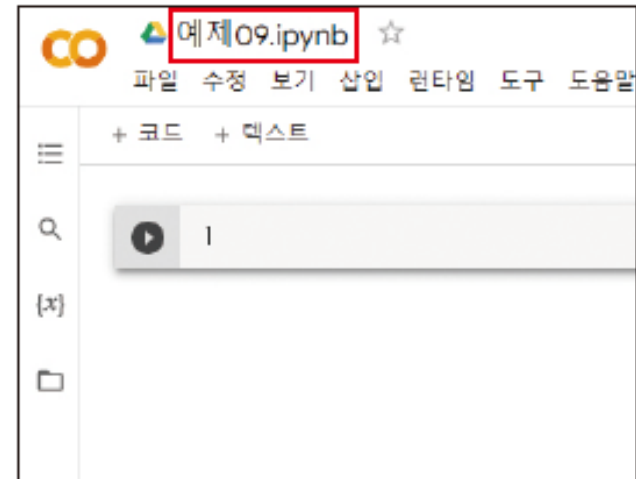
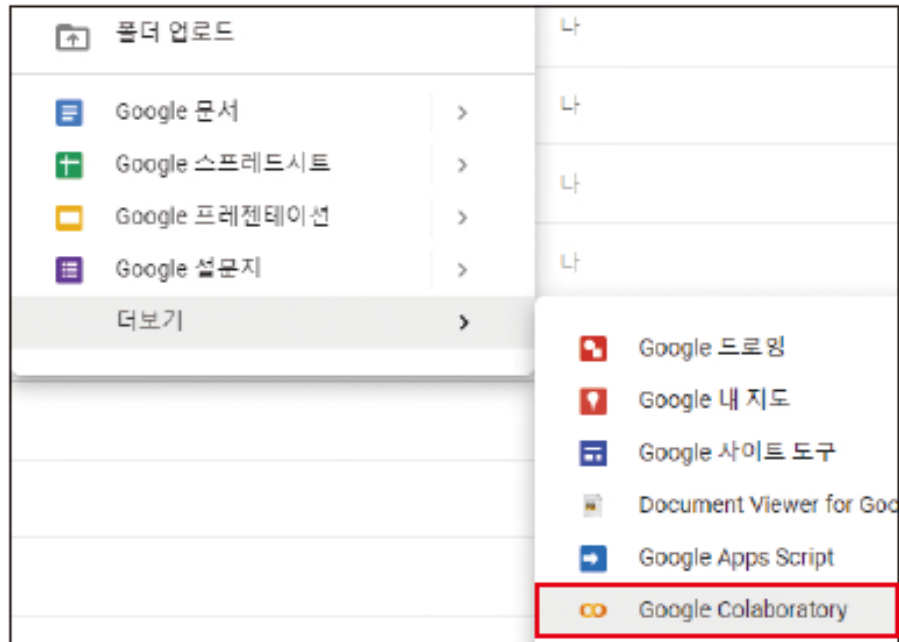
표 9-1. 데이터 정보

파일명	열 정보(지표 정보)		
대한민국행복지도_삶의만족도.xlsx	자신의 삶(과거와 현재)에 만족하는 정도	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_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_교육.xlsx	교원 1인당 학생 수, 영유아 천 명당 보육시설 수, 인구 십만 명당 학교 수, 인구 천 명당 사설학원 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_관계및사회참여.xlsx	인구 십만 명당 자살률, 1인가구(독거노인 제외) 비율, 독거노인가구 비율, 인구 십만 명당 사회적기업 수, 가족관계 만족도
대한민국행복지도_환경.xlsx	환경체감도, 인구 천 명당 1일 산업폐수 방류량, 도시지역 중 '녹지지역 비율', 미세먼지(PM2.5), 주민 1인당 생활폐기물 배출량	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

1. 데이터 시각화 이해

■ 세션 저장소에 업로드

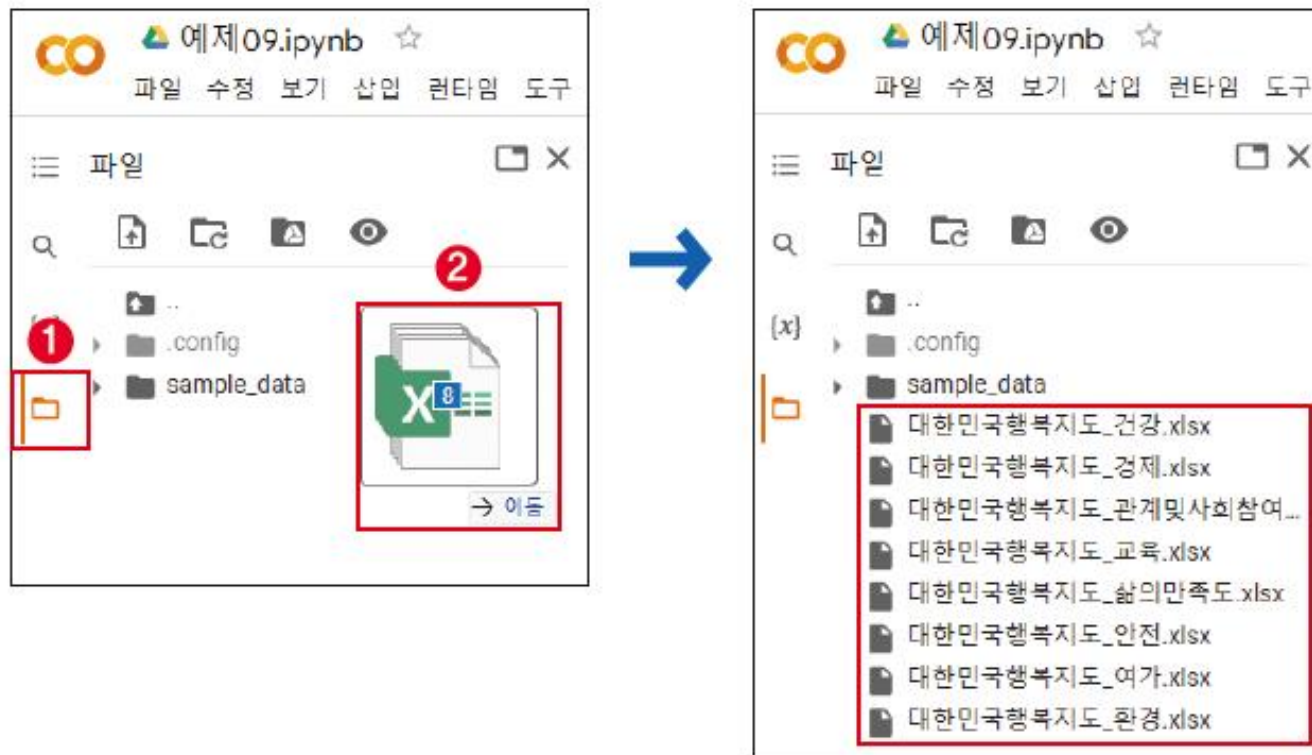
- (1) 구글 드라이브에서 마우스 오른쪽 버튼을 클릭하여 '더보기' → 'Google Colaboratory'를 선택해 코랩을 실행



1. 데이터 시각화 이해

■ 세션 저장소에 업로드

- (2) 코랩 노트의 왼쪽 메뉴에서 '파일' 아이콘을 클릭한다.
- 파일 화면이 열리면 수집한 8개의 엑셀 파일을 드래그하여 업로드한다.



02

데이터 가공

2. 데이터 가공

■ 데이터 읽어와서 확인

‘삶의 만족도’ 엑셀 파일 읽어오기

```
1 from pandas import *
2
3 happy_life=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

‘건강’ 엑셀 파일 읽어오기

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

〈실행결과〉

No	시도	구군	평균	주관적 건강수준	인지율	인구	선만화당	결산건강증진기관 수	영구 천문당	의료기관	종사 의사수	건강생활지표	영구 천문당	의료기관	종사 의사수
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	0.5808	0.0819		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			0.6152	0.8022		0.1936			

■ 데이터 읽어와서 확인

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

№	세도	구분	평균	사과전통음 (재배 2회 x 3)	사과 전통음 (cctv 재배 x 3)	사과 싹타전통 음(재배 2회 x 3) 음(재배 2회 x 3) 음(재배 2회 x 3)	사과전통음 재배 2회 x 3	사과전통음 재배 2회 x 3 음(재배 2회 x 3) 음(재배 2회 x 3)
0	1	서울특별시 중구	0.7670	0.9905	0.8796	0.7498	0.9998	0.9163
1	2	서울특별시 중구	0.9320	0.9707	0.5214	0.6930	1.0000	0.5014
2	3	서울특별시 용인구	0.5537	0.6641	0.4311	0.1768	0.6923	0.5075
3	4	서울특별시 연천군	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

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

№	시도	구군	평균	환경저감도 b)	연구 전염영 1원 산업폐수 영향량 a)	도시지역 중 '독자지역' 비율	대세면적(PM2.5) a) b)	주변 1인당 생활폐기물배출량 a)
0	1 서울특별시	영등포구	0.4637	0.4658	0.7223	0.2912	0.7704	0.6006
1	2 서울특별시	영구	0.2865	0.1077	0.6957	0.0915	0.6617	0.0000
2	3 서울특별시	영남구	0.5030	0.3142	0.7105	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	강진구	0.4892	0.1229	0.6940	0.0574	0.7235	0.7016

2. 데이터 가공

■ 데이터 읽어와서 확인

‘경제’ 엑셀 파일 읽어오기

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

〈실행결과〉

No	시도	구군	평균	1인당 지역내총생산(GRDP)	인구 천명당 사업체수	인구 천명당 종사자수	국민거주총생산보장 수급자비율 a)	종사자 천명당 일제자영업자 수 a)
0	1	서울특별시 종로구	1.0000	1.0000	1.0000	1.0000	0.7564	0.9037
1	2	서울특별시 중구	0.9906	1.0000	1.0000	1.0000	0.6718	0.9177
2	3	서울특별시 동산구	0.6915	0.6334	0.5493	0.7282	0.7910	0.6341
3	4	서울특별시 성북구	0.6533	0.5132	0.5368	0.6849	0.8243	0.6377
4	5	서울특별시 광진구	0.4445	0.2906	0.3424	0.3070	0.8681	0.7400

‘교육’ 엑셀 파일 읽어오기

```
1 happy_edu=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 사립학원수
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 동산구	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

2. 데이터 가공

■ 데이터 읽어와서 확인

‘관계 및 사회참여’ 엑셀 파일 읽어오기

```
1 happy_relation=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 사회적거립수	가족관계 만족도 b)
0	1	서울특별시 용마루구	0.7425	0.8318	0.0057	0.7593	0.9981	0.5949
1	2	서울특별시 용마루구	0.4608	0.5462	0.0107	0.7254	1.0000	0.2329
2	3	서울특별시 용마루구	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

‘여가’ 엑셀 파일 읽어오기

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

〈실행결과〉

No	시도	구군	평균	여가활동 만족도 b)	노년 권영당 노인여가복지시설수	인구 십만명당 도서관수	인구 십만명당 문화기반시설수	인구 권영당 체육관련 여가시설수
0	1	서울특별시 용마루구	0.6331	0.8409	0.1573	0.4523	0.9997	0.5559
1	2	서울특별시 용마루구	0.6691	0.6224	0.1467	0.5369	0.8441	0.9886
2	3	서울특별시 용마루구	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

2. 데이터 가공

■ 데이터 병합

‘시도’ 데이터 추출

```
1 city=list(happy_life['시도'].unique())
2 happy_merge=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

‘삶의 만족도’ 데이터 병합

```
1 life=happy_life['삶의 만족도'].groupby(by=happy_life['시도']).mean()
2 happy_merge=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

2. 데이터 가공

■ 데이터 병합

‘건강’ 데이터 병합

```
1 health=happy_health['평균'].groupby(by=happy_health['시도']).mean()
2 happy_merge=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

‘안전’ 데이터 병합

```
1 safe=happy_safe['평균'].groupby(by=happy_safe['시도']).mean()
2 happy_merge=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

2. 데이터 가공

■ 데이터 병합

‘환경’ 데이터 병합

```
1 environ=happy_environ['평균'].groupby(by=happy_environ['시도']).mean()
2 happy_merge=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

‘경제’ 데이터 병합

```
1 econo=happy_econo['평균'].groupby(by=happy_econo['시도']).mean()
2 happy_merge=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	0.438037
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

2. 데이터 가공

■ 데이터 병합

‘교육’ 데이터 병합

```
1 edu=happy_edu['평균'].groupby(by=happy_edu['시도']).mean()
2 happy_merge=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	0.438037	0.504594
2	대구광역시	0.363988	0.482325	0.358429	0.552500	0.393975	0.585837
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

‘관계 및 사회참여’ 데이터 병합

```
1 relation=happy_relation['평균'].groupby(by=happy_relation['시도']).mean()
2 happy_merge=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	0.438037	0.504594	0.294719
2	대구광역시	0.363988	0.482325	0.358429	0.552500	0.393975	0.585837	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

2. 데이터 가공

■ 데이터 병합

'여가' 데이터 병합

```
1 leisure=happy_leisure['평균'].groupby(by=happy_leisure['시도']).mean()
2 happy_merge=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
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

2. 데이터 가공

■ 데이터 검산

‘삶의 만족도’ 데이터 검산

	〈코드〉	〈실행결과〉
1	<code>print(happy_merge['삶의 만족도'].mean())</code>	0.5181887062335198
2	<code>print(happy_life['삶의 만족도'].mean())</code>	0.4937618421052632

‘안전’ 데이터 검산

	〈코드〉	〈실행결과〉
1	<code>print(happy_merge['안전'].mean())</code>	0.42088164252386323
2	<code>print(happy_safe['평균'].mean())</code>	0.4569065789473684

‘건강’ 데이터 검산

	〈코드〉	〈실행결과〉
1	<code>print(happy_merge['건강'].mean())</code>	0.4094014744072605
2	<code>print(happy_health['평균'].mean())</code>	0.40443508771929826

‘환경’ 데이터 검산

	〈코드〉	〈실행결과〉
1	<code>print(happy_merge['환경'].mean())</code>	0.5794064414748389
2	<code>print(happy_enviro['평균'].mean())</code>	0.5806381578947368

2. 데이터 가공

■ 데이터 검산

‘경제’ 데이터 검산

	〈코드〉	〈실행결과〉
1	<code>print(happy_merge['경제'].mean())</code>	0.4076346792607561
2	<code>print(happy_econo['평균'].mean())</code>	0.3894403508771929

‘관계 및 사회참여’ 데이터 검산

	〈코드〉	〈실행결과〉
1	<code>print(happy_merge['관계및사회참여'].mean())</code>	0.49763200954204206
2	<code>print(happy_relation['평균'].mean())</code>	0.468419298245614

‘교육’ 데이터 검산

	〈코드〉	〈실행결과〉
1	<code>print(happy_merge['교육'].mean())</code>	0.5786529913992841
2	<code>print(happy_edu['평균'].mean())</code>	0.5396557017543859

‘여가’ 데이터 검산

	〈코드〉	〈실행결과〉
1	<code>print(happy_merge['여가'].mean())</code>	0.46490970955241717
2	<code>print(happy_leisure['평균'].mean())</code>	0.4623894736842106

2. 데이터 가공

■ 데이터 검산

데이터 결손 확인		
	〈코드〉	〈실행결과〉
1	<pre>happy_merge.isnull().sum()</pre>	<pre>시도 0 삶의 만족도 0 건강 0 안전 0 환경 0 경제 0 교육 0 관계및사회참여 0 여가 0 dtype: int64</pre>

03

데이터 분석 및 시각화

3. 데이터 분석 및 시각화

■ 데이터 부선

데이터 집계 내용 확인

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

행복지수 요소별 평균 확인

<코드>

1 `happy_merge.mean(
numeric_only=True,
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

3. 데이터 분석 및 시각화

■ 데이터 분석

시도별 평균 확인 (1)

〈코드〉

```
happy_merge.mean(numeric_only  
=True,axis=1)
```

〈실행결과〉

```
0    0.461637  
1    0.389815  
2    0.422433  
3    0.418799  
4    0.520068  
5    0.497017  
6    0.532900  
7     0.21288  
8    0.472481  
9    0.526015  
10   0.500400  
11   0.508128  
12   0.488729  
13   0.500111  
14   0.436254  
15   0.482856  
16   0.559075  
dtype: float64
```

시도별 평균 확인 (2)

〈코드〉

```
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  
울산광역시    0.532900  
세종특별자치시  0.21288  
경기도        0.472481  
강원도        0.526015  
충청북도      0.500400  
충청남도      0.508128  
전라북도      0.488729  
전라남도      0.500111  
경상북도      0.436254  
경상남도      0.482856  
제주특별자치도 0.559075  
dtype: float64
```

3. 데이터 분석 및 시각화

■ 데이터 분석

행복지수 요소별 상관관계 확인

1 happy_merge.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

3. 데이터 분석 및 시각화

■ 한글 폰트 설치

- 코랩에서는 맷플롯립으로 그래프를 그릴 때 한글이 깨지기 때문에 다음의 한글 폰트 설치 코드를 실행하여 설치를 진행해야 한다.
- 주의할 점은 코랩이라는 가상환경에 설치한 한글 폰트는 세션이 종료되면 매번 재설치를 해야 한다.
- 설치 후에는 반드시 '런타임 다시시작'을 클릭한다.
- 한글 폰트 설치 부분을 제외한 데이터를 읽어와서 시각화하는 모든 코드를 재실행한다.

한글 폰트 설치

1	<code>!sudo apt-get install -y fonts-nanum</code>	# 나눔 폰트 설치
2	<code>!sudo fc-cache -fv</code>	# 시스템의 폰트 업데이트
3	<code>!rm ~/.cache/matplotlib -rf</code>	# 기존 설정된 글꼴 캐시 삭제

3. 데이터 분석 및 시각화

■ 한글 폰트 설정

- 1행에서는 그래프를 그리기 위한 맷플롯립(matplotlib) 라이브러리를 импорт(Import)
- 2행에서는 한글 폰트 설정을 위한 폰트 매니저(matplotlib.font_manager)를 импорт(Import)
- 3행에서는 시본(Seaborn) 라이브러리를 импорт(Import)
- 5행에서는 나눔고딕 폰트 경로를 변수 font_path에 저장
- 6행에서는 나눔고딕 폰트, 글자 크기 10의 폰트 정보를 변수 font_name에 저장
- 8행에서는 5행과 6행에서 지정한 한글 폰트 정보를 rc()를 이용하여 적용

한글 폰트 설정

```
1 import matplotlib.pyplot as plt
2 import matplotlib.font_manager as fm
3 import seaborn as sns
4
5 font_path='/usr/share/fonts/truetype/nanum/NanumGothic.ttf'
6 font_name=fm.FontProperties(fname=font_path, size=10).get_name()
7
8 plt.rc('font', family=font_name)
```

3. 데이터 분석 및 시각화

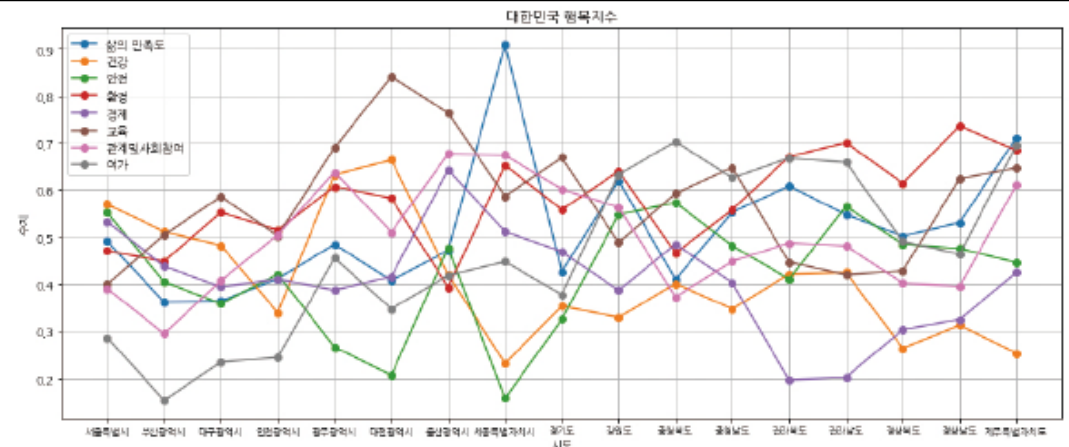
■ 선 그래프로 시각화

- 행복지수가 '시도'별 지역에 따라 얼마나 차이가 나는지 선 그래프로 시각화하여 출력

```
import matplotlib.font_manager as fm
import matplotlib.pyplot as plt
import seaborn as sns
plt.rcParams['font.family']="Malgun Gothic"
plt.rcParams['axes.unicode_minus']=False
```

선 그래프로 시각화

```
plt.figure(figsize=(15,6))
items=list(happy_merge.columns[1:9])
for a in items:
    chartdata=happy_merge[a]
    plt.plot(chartdata, marker='o', label=a)
plt.xlabel('시도')
plt.ylabel('수치')
plt.xticks(range(17), happy_merge['시도'],
            fontsize=8)
plt.title('대한민국 행복지수')
plt.legend()
plt.grid()
plt.show()
```



3. 데이터 분석 및 시각화

■ 막대 그래프로 시각화

- 세로 막대 그래프의 서브플롯을 활성화하여 8개의 요소들을 구분하여 출력

막대 그래프로 시각화

```
import numpy as np
happy_merge.plot(kind='bar', xlabel='시도',
ylabel='수치', figsize=(15,20), grid=True,
subplots=True)

plt.suptitle('대한민국 행복지수', fontsize=25)
plt.tight_layout(pad=8, h_pad=2)
plt.xticks(np.arange(17), city, rotation=360)
plt.show()
```

〈실행결과〉



3. 데이터 분석 및 시각화

■ 히트맵 그래프로 시각화

- 시본(Seaborn) 라이브러리를 추가로 이용하여 데이터 범위를 다양한 색상으로 출력

히트맵 그래프로 시각화

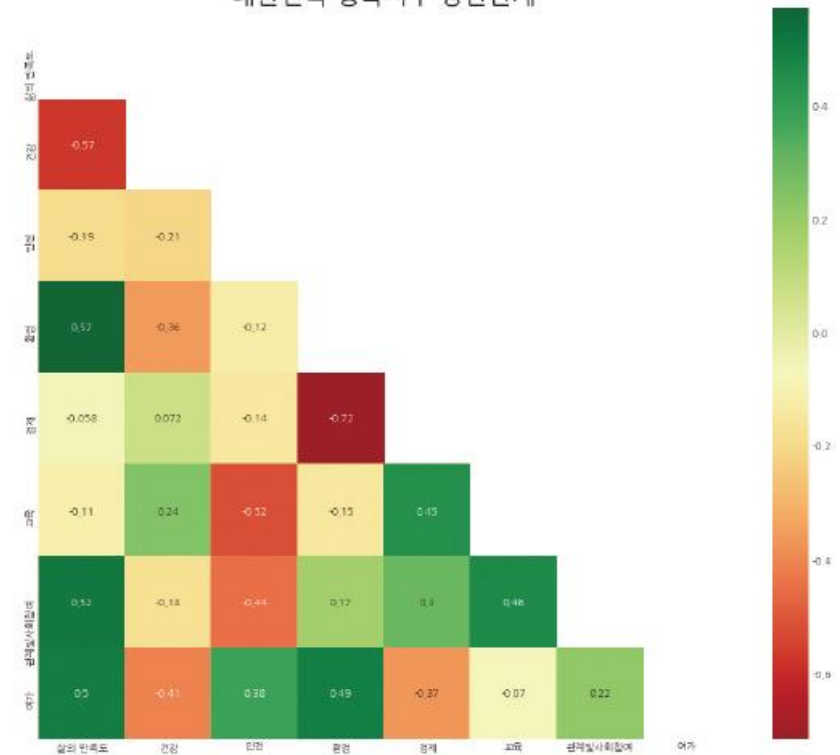
```
plt.figure(figsize=(15,13))
plt.title('대한민국 행복지수 상관관계', fontsize=25)

correlation_mat=happy_merge.corr(numeric_only=True)
upp_mat=np.triu(correlation_mat)
sns.heatmap(correlation_mat, mask=upp_mat,
            annot=True, cmap='RdYlGn')

plt.show()
```

〈실행결과〉

대한민국 행복지수 상관관계



Q&A