



오늘의 학습

학습내용

■ 인구구조 데이터 분석 실습



Python DataFrame 문자열 포함 확인 str.contains()

DataFrame에 저장한 데이터에 문자열이 검색 문자열이 포함되어 있는지 판단

pandas.Series 문자열 메서드인 str.contains()를 사용하면 지정한 문자열이 포함되어 있는지 확인

검색 문자열이 포함되어 있는 경우에는 True를 반환





str.contains() 사용 방법

contains에 지정한 '김'문자열이 포함된 행은 True를 반환하고 문자열이 포함되지 않은 행은 False를 반환

반환된 결과값은 pandas.Series 형태로 반환

str.contains는 지정한 문자열이 완전히 일치하는 값이 아닌 부분 일치 하는 값을 찾는다.

```
import pandas as pd
    df = pd.DataFrame({'name': ['김길동','홍사림','미지수','김지희','나진수','미채리'],
                       'age': [20,40,17,40,30,50],
                       'point': [60,20,70,70,80,55]}
    print(df['name'].str.contains('김'))
      True
     False
     False
     True
     False
     False
Name: name, dtype: bool
print(uit name 1.5th.contains) - 11
```



DataFrame으로 추출하기 -> name컬럼에 '김' 문자열이 포함된 행의 값을 전부 표시

```
1 | import pandas as pd
2 | df = pd.DataFrame({'name': ['김길동','홍사림','이지수','김지희','나진수','이채리'],
4 | 'age': [20,40,17,40,30,50],
5 | 'point': [60,20,70,70,80,55]}
6 | )
7 | df[df['name'].str.contains('김')]
```

	name	age	point
0	김길동	20	60
3	김지희	40	70



NaN 값이 존재하는 경우 -> 문자열에 검색할 컬럼에 NaN값이 존재하는 경우 True/False가 아닌 NaN반환

데이터를 수집하면 전산오류나 사람의 실수로 결측치(NaN)가 발생

pandas.Series로 결과를 받아 행을 취득하려고 하면 에러가 발생

```
import pandas as pd
   df = pd.DataFrame({'name': ['김길동','홍사림','이지수',1,'나진수','이채리'],
                      'age': [20,40,17,40,30,50],
                      point': [60,20,70,70,80,55]}
   |df[df['name'].str.contains('김')]
ValueError
                                        Traceback (most recent call last)
C:#Users#Public#Documents#ESTsoft#CreatorTemp/ipykernel_18576/1308088444.py in <module>
----> 8 df[df['name'].str.contains('김')]
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\frame.py in __getitem__(self, key)
  3446
  3447
              # Do we have a (boolean) 1d indexer?
-> 3448
                if com.is_bool_indexer(kev):
  3449
                  return self._getitem_bool_array(key)
  3450
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\common.py in is_bool_indexer(key)
                         # Don't raise on e.g. ["A", "B", np.nan], see
   137
   138
                         # test_loc_getitem_list_of_labels_categoricalindex_with_na
--> 139
                             raise ValueError(na_msg)
   1.40
                      return False
   141
                  return True
ValueError: Cannot mask with non-boolean array containing NA / NaN values
```



NaN이 존재하는 경우에는 na로 NaN 결과를 치환해 주도록 설정

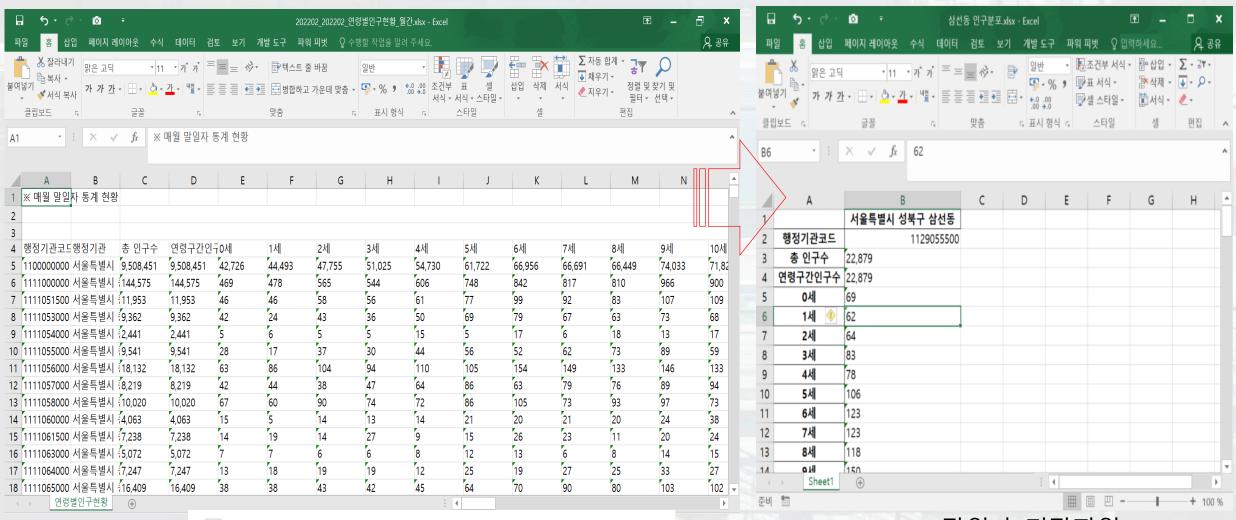
```
import pandas as pd
    df = pd.DataFrame({'name': ['김길동','홍사림','이지수',1 ,'나진수','이채리'],
                      age': [20,40,17,40,30,50],
                      'point': [60,20,70,70,80,55]}
    # NaNO/ 존재하는 경우 False로 치환
   print(df['name'].str.contains('김', na=False))
0
    True
    False
    False
3
    False
    False
    False
Name: name, dtype: bool
    import pandas as pd
    df = pd.DataFrame({'name': ['김길동','홍사림','미지수',1 ,'나진수','미채리'],
                      'age': [20,40,17,40,30,50],
                      'point': [60,20,70,70,80,55]}
    # NaNOI 존재하는 경우 True로 치환
    print(df['name'].str.contains('김', na=True))
     True
0
    False
    False
    True
    False
    False
Name: name, dtype: bool
```

[실습]인구구조 분석



__ [실습 결과물]

[문제1] '2022년 2월 연령별 인구현황_월간'파일에서 우리동네 연령별 인구 분포 분석 후 엑셀 파일로 저장하기



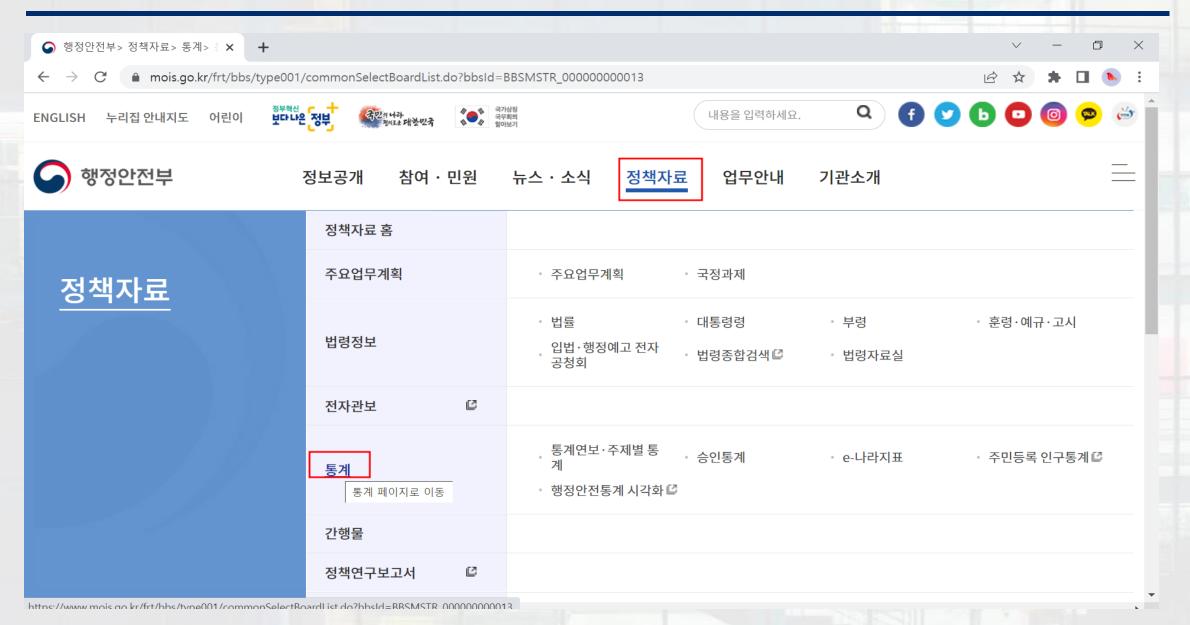
파일 원본

хa,

작업 후 저장파일

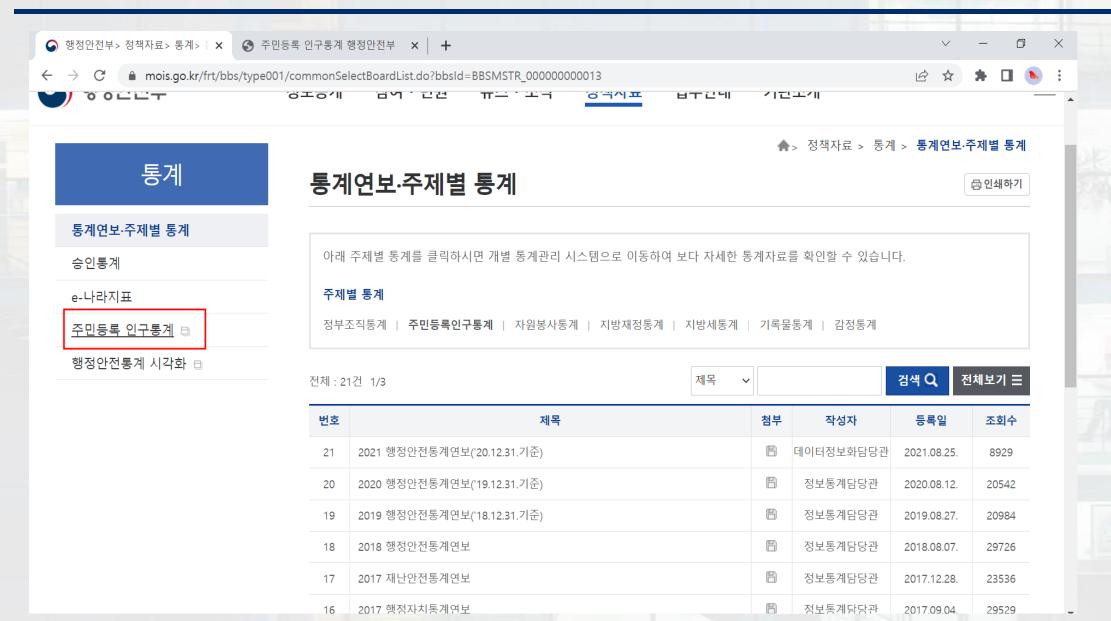


[파일준비] 행정안전부 홈페이지-> 정책자료->통계



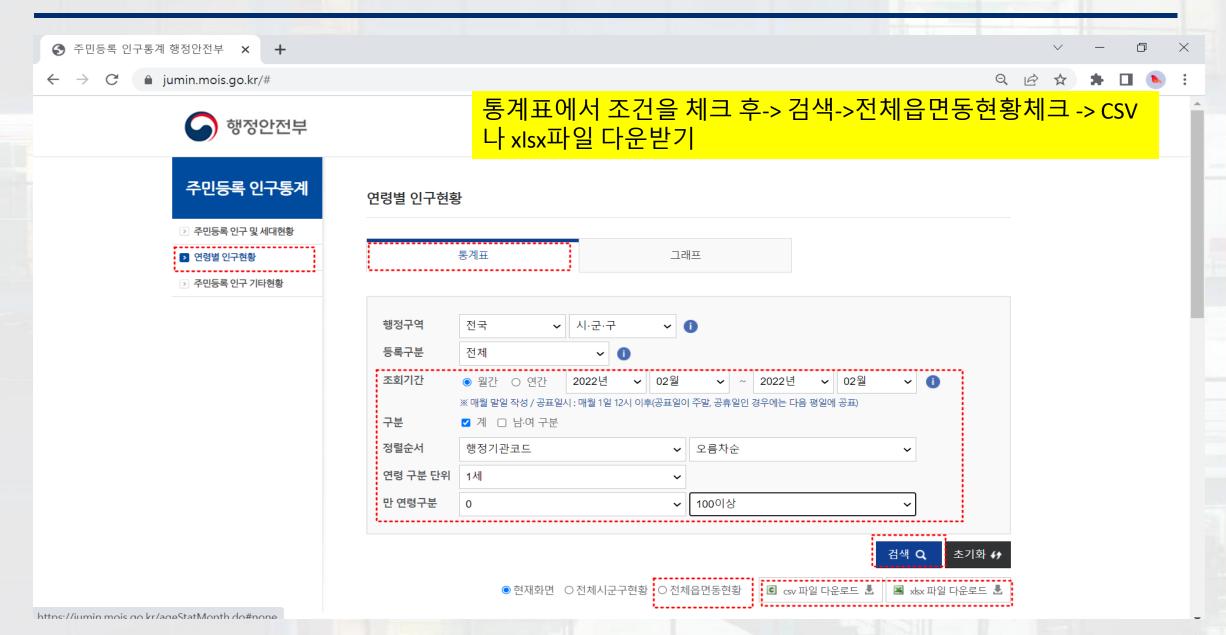


[파일준비]행정안전부 홈페이지-> 정책자료->통계->주민등록 인구통계





[파일준비] 주민등록 인구통계->[연령별 인구현황]



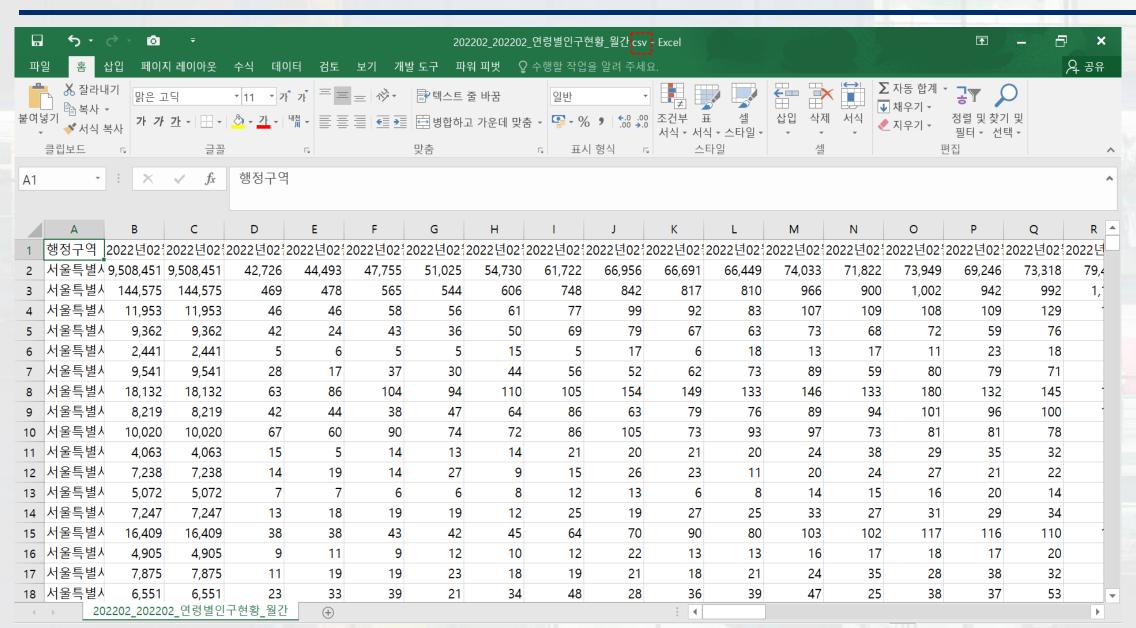


다운로드 받은 파일 확인하기

CSV파일



💶 202202_202202_연령별인구현황_월간.csv



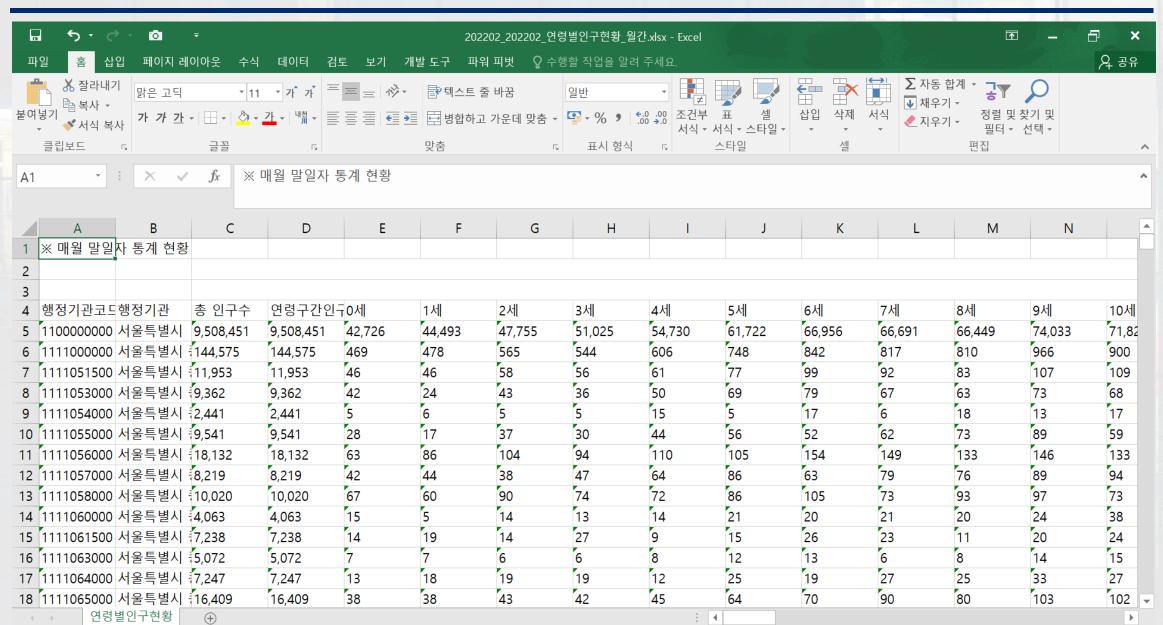


다운로드 받은 파일 확인하기

XIsx 파일



🛍 202202_202202_연령별인구현황_월간.xlsx





다운로드 받은 파일 이름 변경하기

🛍 202202_202202_연령별인구현황_월간.csv

👊 202202_202202_연령별인구현황_월간.xlsx

파일명 간단하게 수정해서 사용하기

population.csv

Population.xlsx



다운로드 받은 파일 Pandas 라이브러리를 사용하여 불러오기

- import pandas as pd df = pd.read csv('population.csv', encoding = 'cp949', low memory=**False**, index col = 0) df.head(n=3) 2022 2022 2022 2022 2022 2022 2022 2022 20 2022 2022 2022 2022 2022 2022 2022 2022 년02 년02 년02 년02 년02 년02 년02 년02 2022년02 년 2022년02 년02 년02 년02 월_ 월 계 연 년02 년02 년02 년02 년02 월_계_총 월_계 월_계 월_계 월_계 월_계 월_계 월_계 월_계 령구간인 계 인구수 _92 94 _98 _2세 _3세 _4세 _6세 _0세 _7세 세 행정구역 서울특별시 9,508,451 9,508,451 42,726 44,493 47,755 51,025 54,730 61,722 66,956 66,691 ... 8,025 6,853 5,356 4,119 2,851 1,867 1,357 1,202 9 (1100000000) 서울특별시 144,575 144.575 469 544 606 748 842 817 ... 170 52 32 22 143 113
- 서울특별시 종로구 청운 11.953 11,953 46 61 77 92 ... 9 10 3 3 효자동 (1111051500)

3 rows × 103 columns

종로구

(1111000000)



다운로드 받은 파일 Pandas 라이브러리를 사용하여 불러오기

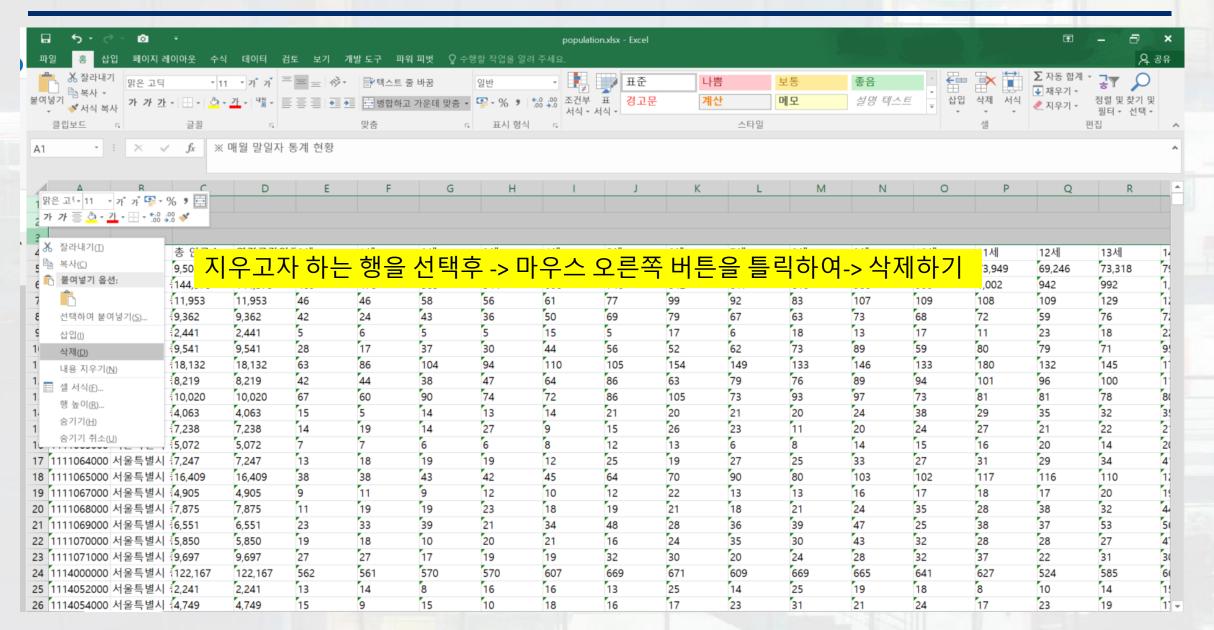
1. Pandas로 csv 파일과 excel 파일 불러오기(엑셀 파일)

- import pandas as pd df = pd.read_excel('population.xlsx') df.head(n=3) C:\ProgramData\Anaconda3\Iib\site-packages\openpyx|\styles\stylesheet.py:226: User\arning: Workbook contains no default style, appl y openpyxl's default warn("Workbook contains no default style, apply openpyxl's default") Ж 매월 Unnamed: Unnamed 97 5 자 통계 현황 0 NaN NaN NaN NaN NaN NaN NaN NaN NaN ... NaN Na NaN NaN 1 NaN NaN NaN NaN NaN NaN NaN NaN NaN ... NaN NaN NaN Na 행정 연령구간 2 기관 행정기관 총 인구수 0세 1세 2세 3세 4세 5세 ... 93세 91세 92세 941 인구수
- 3 rows × 105 columns

코드



엑셀에서 불필요한 행 삭제하기





다운로드 받은 파일 Pandas 라이브러리를 사용하여 불러오기

엑셀파일에서 3행을 제거후 엑셀 파일 다시 불러오기

```
1 import pandas as pd
 2 df = pd.read_excel('population.xlsx')
3 df.head(n=3)
   행정기관코
                                                                                                                                   세이
             서
0 11000000000 특
                9.508.451 9.508.451 42.726 44.493 47.755 51.025 54.730 61.722 ... 8.025 6.853 5.356 4.119 2.851 1.867 1.357 1.202 900 1.708
             시
1 1111000000
                  144.575
                                                                        748 ...
                           144.575
                                                                                170
                                                                                                                                     37
             로
2 1111051500
                   11,953
                            11,953
                                                                        77 ...
```



파일에서 필요없는 열 삭제하기

2. 엑셀 파일에서 필요없는 열 삭제하기

```
      1 import pandas as pd

      2 df = df = pd.read_excel('population.xlsx')

      3 del df['행정기관코드'], df['총 인구수'], df['연령구간인구수']
      # 행정기관코드, 총 인구수, 연령구간인구수 삭제하기

      4 df.head(n=5)
```

	행정기관	0세	1세	2세	3세	4세	5세	6세	7세	8세	 91세	92세	93세	94세	95세	96세	97세	98세	99 세	100세 이상
0	서울특별시	42,726	44,493	47,755	51,025	54,730	61,722	66,956	66,691	66,449	 8,025	6,853	5,356	4,119	2,851	1,867	1,357	1,202	900	1,708
1	서울특별시 종로구	469	478	565	544	606	748	842	817	810	 170	143	113	89	80	52	32	22	26	37
2	서울특별시 종로구 청 운효자동	46	46	58	56	61	77	99	92	83	 9	10	8	7	4	5	3	3	2	2
3	서울특별시 종로구 사 직동	42	24	43	36	50	69	79	67	63	 13	16	10	9	6	7	4	2	2	1
4	서울특별시 종로구 삼 청동	5	6	5	5	15	5	17	6	18	 4	6	2	3	3	0	2	0	2	2

5 rows × 102 columns



궁금한 지역의 연령별 인구 추출하기

3. 궁금한 지역 이름 입력받고 해당 지역의 인구 추출하기

```
l limport pandas as pd
 2 df = df = pd.read_excel('population.xlsx', index_col = 1)
   name = input('원하는 지역의 이름을 입력=>') #input()활수를 통해 입력
   la = df.index.str.contains(name)
   df1 = df[a]
  df1
원하는 지역의 이름을 입력=>참선동
       행정기관
  서울특별시 성북구
               1129055500
                        22.879
                                  22,879 69 62 64 83 78 106 123 ... 26 14 13 14
         삼선동
```

1 rows × 104 columns



___ [개인 실습하기]

[문제 풀기] 본인이 궁금한 지역(살고 있는 지역)의 연령별 인구 추출하기



전체열과 전체행 보이기

row 생략 없이 출력 pd.set_option('display.max_rows', None) col 생략 없이 출력 pd.set_option('display.max_columns', None)

0	서울특별시 (1100000000)	9,508,451	9,508,451	42,726	44,493	47,755	51,025	54,730	61,722	66,956	 5,901	5,042	4,031	3,178	2,227	1,479	1,042	898	685
1	서울특별시 종로구 (1111000000)	144,575	144,575	469	478	565	544	606	748	842	 117	104	81	68	58	34	20	14	14
2	서울특별시 종로구 청운 효자동 (1111051500)	11,953	11,953	46	46	58	56	61	77	99	 6	8	6	4	3	3	1	2	1
3	서울특별시 종로구 사직 동 (1111053000)	9,362	9,362	42	24	43	36	50	69	79	 10	10	5	4	4	4	0	2	1
4	서울특별시 종로구 삼청 동 (1111054000)	2,441	2,441	5	6	5	5	15	5	17	 2	4	2	2	0	0	1	0	1
3861	제주특별자치 도 서귀포시 서홍동 (5013058000)	11,192	11,192	79	72	89	98	117	143	167	 5	3	8	1	1	2	0	0	2



데이터 정제

분석하기 좋은 데이터 : 분석하기 좋은 데이터는 데이터 집합을 분석하기 좋은 상태로 만들어 놓은것

데이터 분석단계에서 데이터 정리가 중요(데이터 분석 작업의 70% 이상이 데이터 정리작업)

분석하기 좋은 데이터 조건

- 데이터 분석 목적에 맞는 데이터를 모아 새로운 표(Table)을 만들기
- 측정한 값은 행(row)을 구성
- 변수는 열(column)로 구성



데이터프레임의 행과 열 바꾸기

데이터프레임의 행과 열을 바꾸기위해 df.transpose()사용

0세 1세 2세 3세 4세 5세 6세 7세 8세 9세 ... 91세 92세 93세 94세 95세 96세 97세 98세 99세 100세 이상 행정기관 서울특별시 성북구 삼선동 69 62 64 83 78 106 123 123 118 150 ... 26 14 13 14 7 5 3 0 4 10 1 rows × 101 columns

1 ### 데이터를 행과 열 바꾸기

1 filt = df1.transpose() 2 filt.head(n=TT)

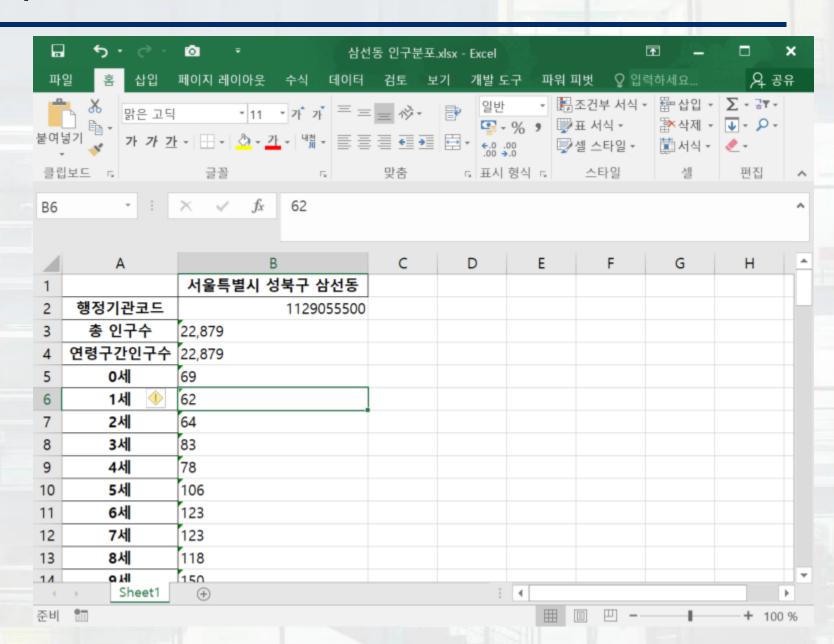
#행과 열 바꾸기

행정기관	서울특별시 성북구 삼선동
행정기관코드	1129055500
총 인구수	22,879
연령구간인구수	22,879
0세	69
1세	62
2세	64
3세	83
4세	78
5세	106
6세	123
7세	123



엑셀로 저장하고 확인하기

filt.to_excel('삼선동 인구분포.xlsx')





[참고] CSV 파일에서 lloc로 열과 행 추출하기

```
import pandas as pd
df = pd.read_csv('age.csv',encoding = 'cp949', low_memory=False)
filt = df.iloc[0:5,0:10]
filt
```

	행정구역	2022년02월_계_ 총인구수	2022년02월_계_연 령구간인구수	2022년02월 _계_0세	2022년02월 _계_1세	2022년02월 _계_2세	2022년02월 _계_3세	2022년02월 _계_4세	2022년02월 _계_5세	2022년02월 _계_6세
0	서울특별시 (1100000000)	9,508,451	9,508,451	42,726	44,493	47,755	51,025	54,730	61,722	66,956
1	서울특별시 종로구 (1111000000)	144,575	144,575	469	478	565	544	606	748	842
2	서울특별시 종로구 청운효 자동(1111051500)	11,953	11,953	46	46	58	56	61	77	99
3	서울특별시 종로구 사직동 (1111053000)	9,362	9,362	42	24	43	36	50	69	79
4	서울특별시 종로구 삼청동 (1111054000)	2,441	2,441	5	6	5	5	15	5	17