

Ch09 데이터 분석 프로젝트

- 한국인의 삶을 파악하라!

09-1 '한국복지패널 데이터' 분석 준비하기

데이터 분석 준비하기

1. 데이터 분석 환경 준비하기

```
In [2]: # 그래프 해상도 설정
import matplotlib.pyplot as plt
plt.rcParams.update({'figure.dpi': '100'})
%config InlineBackend.figure_format = 'retina' #Jupyter 환경 정의: 그래프나 그림의 출력 품질을 설정
```

2. 패키지 설치 및 로드하기

Anaconda Prompt에서 install

SPSS, SAS, STATA 등 다른 툴에서 사용하는 통계 파일을 불러올 수 있음.

pip install pyreadstat

```
In [3]: import pandas as pd
import numpy as np
import seaborn as sns
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\scipy\_\_init\_\_.py:155: UserWarning: A NumPy version >=1.18.5 and <1.25.0 is required for
this version of SciPy (detected version 1.25.2
... warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

3. 데이터 불러오기

Koweps_hpwc14_2019_beta2.sav 파일을 워킹 디렉터리에 삽입

> 2020년 발간 복지패널 데이터. 6,331가구, 14,418명의 정보를 담고 있음

> *.sav: 통계 분석 소프트웨어 SPSS 전용 파일

In [149...]

```
# 데이터 불러오기  
raw_welfare = pd.read_spss('Koweps_hpwc14_2019_beta2.sav')  
  
# 복사본 만들기  
welfare = raw_welfare.copy()  
welfare
```

Out[149]:

	h14_id	h14_ind	h14_sn	h14_merkey	h_new	h14_cobf	p14_wsc	p14_wsl	p14_wgc	p14_wgl	...	wc14_64	wc14_65	wc14_5aq4	wc14_
0	2.0	1.0	1.0	20101.0	0.0	NaN	0.291589	0.291589	1307.764781	1307.764781	...	NaN	NaN	NaN	NaN
1	3.0	1.0	1.0	30101.0	0.0	NaN	0.419753	0.419753	1882.570960	1882.570960	...	NaN	NaN	NaN	NaN
2	4.0	1.0	1.0	40101.0	0.0	NaN	0.265263	0.265980	1189.691668	1192.908537	...	NaN	NaN	NaN	NaN
3	6.0	1.0	1.0	60101.0	0.0	NaN	0.494906	0.495941	2219.630833	2224.273816	...	1.0	.	2.0	
4	6.0	1.0	1.0	60101.0	0.0	NaN	1.017935	1.017935	4565.389177	4565.389177	...	1.0	.	1.0	
...
14413	9800.0	7.0	1.0	98000701.0	1.0	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
14414	9800.0	7.0	1.0	98000701.0	1.0	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
14415	9800.0	7.0	1.0	98000701.0	1.0	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
14416	9800.0	7.0	1.0	98000701.0	1.0	NaN	NaN	NaN	NaN	NaN	...	5.0	.	4.0	
14417	9800.0	7.0	1.0	98000701.0	1.0	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	

14418 rows × 830 columns

4. 데이터 검토하기

In [5]:

```
welfare.shape # 행, 열 개수 출력
```

Out[5]:

```
(14418, 830)
```

In [6]:

```
welfare.info() # 변수 속성 출력
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14418 entries, 0 to 14417
Columns: 830 entries, h14_id to h14_pers_income5
dtypes: float64(826), object(4)
memory usage: 91.3+ MB
```

In [7]: `welfare.describe() # 요약 통계량`

	h14_id	h14_ind	h14_sn	h14_merkey	h_new	h14_cobf	p14_wsc	p14_wsl	p14_wgc	p14_wgl	...
count	14418.000000	14418.000000	14418.000000	1.441800e+04	14418.000000	121.000000	11513.000000	11513.000000	11513.000000	11513.000000	...
mean	4672.108406	3.121723	1.004855	4.672140e+07	0.201484	2.256198	1.000000	1.000000	4484.952219	4484.952541	...
std	2792.998128	3.297963	0.143205	2.793014e+07	0.401123	1.675952	0.906021	1.016782	4063.459773	4560.218659	...
min	2.000000	1.000000	1.000000	2.010100e+04	0.000000	1.000000	0.001998	0.000000	8.960093	0.000000	...
25%	2356.000000	1.000000	1.000000	2.356030e+07	0.000000	1.000000	0.341814	0.269286	1533.021553	1207.736094	...
50%	4535.000000	1.000000	1.000000	4.535010e+07	0.000000	2.000000	0.726304	0.704045	3257.436901	3157.609630	...
75%	6616.000000	7.000000	1.000000	6.616010e+07	0.000000	2.000000	1.366071	1.390045	6126.762919	6234.287538	...
max	9800.000000	14.000000	9.000000	9.800070e+07	1.000000	7.000000	4.727006	5.790039	21200.393903	25968.049029	...

8 rows × 826 columns

In []:

[실습-1] 한국복지패널 제공 관련 CodeBook 파일 로딩

- > 'Koweps_Codebook_2019.xlsx' 파일을 읽어들여서 데이터 프레임 구성 후 출력
- > Koweps_hpwc14_2019_beta2.sav 파일의 변수들에 대한 데이터 구성 내용을 설명하고 있음

In [5]: `## [Data Frame] Excel 파일 Reading: 한국복지패널 제공 관련 CodeBook 파일
import pandas as pd`

```
df_code = pd.read_excel('Koweps_Codebook_2019.xlsx')
```

#header가 있을 때, 첫 행이 데이터 프레임의 열 index 역할을 함

Out[5]:	변수명	설명	내용	범위	모름/무응답	출처 조사설계서
0	h14_g3	성별	1.남 2.여	N(1~2)	모름/무응답 =9	14차 머지데이터_변수명.xlsx\n(2019년 14차 한국복지패널조사) 조사설계서...
1	h14_g4	태어난 연도	년	N(1900~2014)	모름/무응답 =9999	14차 머지데이터_변수명.xlsx\n(2019년 14차 한국복지패널조사) 조사설계서...
2	h14_g10	혼인상태	0.비해당(18세 미만)\n1.유배우 2.사별 3.이혼...	N(0~6)	모름/무응답 =9	14차 머지데이터_변수명.xlsx\n(2019년 14차 한국복지패널조사) 조사설계서...
3	h14_g11	종교	1.있음 2.없음	N(1~2)	모름/무응답 =9	14차 머지데이터_변수명.xlsx\n(2019년 14차 한국복지패널조사) 조사설계서...
4	p1402_8aq1	일한달의 월 평균 임금	만원	N(1~9998)	모름/무응답 =9999	(2019년 14차 한국복지패널조사) 조사설계서-가구원용(beta2).xlsx
5	h14_eco9	직종	직종 코드표 참조	N(직종코드 시트참조)	모름/무응답 =9999	14차 머지데이터_변수명.xlsx\n(2019년 14차 한국복지패널조사) 조사설계서...
6	h14_reg7	7개 권역별 지역구분	1. 서울 2. 수도권(인천/경기) 3. 부산/경남/울산 ...	N(1~7)	Nan	(2019년 14차 한국복지패널조사) 조사설계서-가구용(beta2).xlsx

5. 변수명 바꾸기

> 난해한 변수명의 의미있는 이름으로 변경

[실습-2] 필요한 컬럼만으로 데이터 프레임 구성

변수명 바꾸기

```
> {'h14_g3' : 'sex', 'h14_g4' : 'birth', 'h14_g10' : 'marriage_type', 'h14_g11' : 'religion', 'p1402_8aq1' : 'income', 'h14_eco9' : code_job', 'h14_reg7' : 'code_region'}
```

이름이 바뀐 변수 만으로 newwel 데이터 프레임 구성

In [150...]

```
## 변수명 바꾸기
welfare = welfare.rename(columns = {'h14_g3' : 'sex', # 성별
                                     'h14_g4' : 'birth', # 태어난 연도
                                     'h14_g10' : 'marriage_type', # 혼인 상태
                                     'h14_g11' : 'religion', # 종교
                                     'p1402_8aq1' : 'income', # 월급
                                     'h14_eco9' : 'code_job', # 직업 코드
                                     'h14_reg7' : 'code_region'}) # 지역 코드

welfare.head()
```

Out[150]:

	h14_id	h14_ind	h14_sn	h14_merkey	h_new	h14_cobf	p14_wsc	p14_wsl	p14_wgc	p14_wgl	...	wc14_64	wc14_65	wc14_5aq4	wc14_5aq5
0	2.0	1.0	1.0	20101.0	0.0	NaN	0.291589	0.291589	1307.764781	1307.764781	...	NaN	NaN	NaN	NaN
1	3.0	1.0	1.0	30101.0	0.0	NaN	0.419753	0.419753	1882.570960	1882.570960	...	NaN	NaN	NaN	NaN
2	4.0	1.0	1.0	40101.0	0.0	NaN	0.265263	0.265980	1189.691668	1192.908537	...	NaN	NaN	NaN	NaN
3	6.0	1.0	1.0	60101.0	0.0	NaN	0.494906	0.495941	2219.630833	2224.273816	...	1.0	.	2.0	4.0
4	6.0	1.0	1.0	60101.0	0.0	NaN	1.017935	1.017935	4565.389177	4565.389177	...	1.0	.	1.0	5.0

5 rows × 830 columns

In [151...]

```
## 바꾼 변수명으로만 데이터 프레임 구성
cols = ['sex', 'birth', 'marriage_type', 'religion', 'income', 'code_job', 'code_region']
newwel = welfare[cols]
newwel.head()
```

Out[151]:

	sex	birth	marriage_type	religion	income	code_job	code_region	
0	2.0	1945.0		2.0	1.0	NaN	NaN	1.0
1	1.0	1948.0		2.0	2.0	NaN	NaN	1.0
2	1.0	1942.0		3.0	1.0	107.0	762.0	1.0
3	1.0	1962.0		1.0	1.0	192.0	855.0	1.0
4	2.0	1963.0		1.0	1.0	NaN	NaN	1.0

In []:

09-2 성별에 따른 월급 차이 - 성별에 따라 월급이 다를까?

성별 변수 검토 및 전처리하기

1. 변수 검토하기

```
In [12]: ## 변수의 데이터 타입 확인  
welfare['sex'].dtypes # 변수 타입 출력
```

```
Out[12]: dtype('float64')
```

```
In [13]: ## 성별 값 및 빈도 확인  
welfare['sex'].value_counts() # 빈도 구하기
```

```
Out[13]: sex  
2.0    7913  
1.0    6505  
Name: count, dtype: int64
```

```
In [14]: ## 성별 값 및 빈도 확인  
sum(welfare['sex'].value_counts()) # 빈도 구하기
```

```
Out[14]: 14418
```

```
In [17]: ## sex = 9 빈도 구하기: 9는 무응답  
welfare[welfare['sex'] == 9].value_counts() # 빈도 구하기
```

```
Out[17]: Series([], Name: count, dtype: int64)
```

```
In [16]: # 결측치 확인  
welfare['sex'].isna().sum()
```

```
Out[16]: 0
```

2. 전처리하기

```
In [18]: # 이상치 확인: 'sex'  
welfare['sex'].value_counts()
```

```
Out[18]: sex  
2.0    7913  
1.0    6505  
Name: count, dtype: int64
```

```
In [17]: # 이상치 결측 처리  
welfare['sex'] = np.where(welfare['sex'] == 9, np.nan, welfare['sex'])
```

```
# 결측치 확인  
welfare['sex'].isna().sum()
```

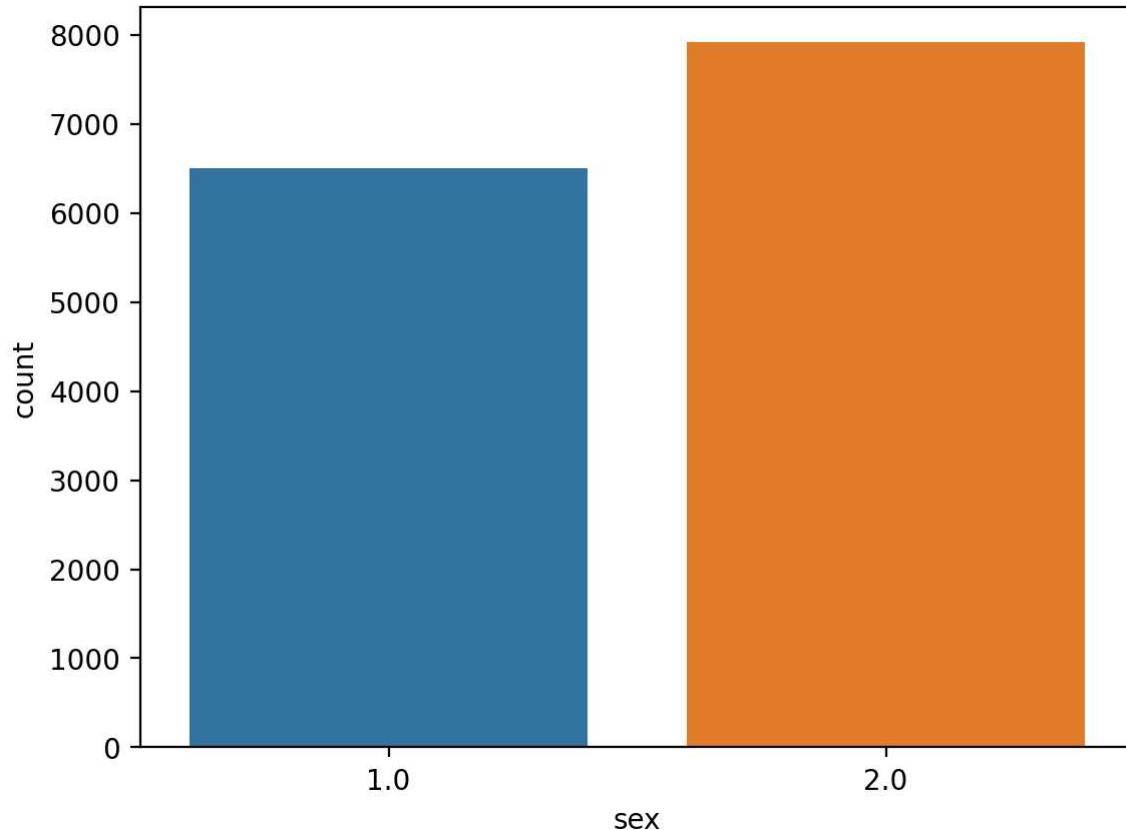
Out[17]: 0

'sex' 빈도 막대 그래프 그리기

```
In [8]: # 빈도 막대 그래프 만들기  
sns.countplot(data = welfare, x = 'sex')
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be  
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead  
    if pd.api.types.is_categorical_dtype(vector):  
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be  
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead  
    if pd.api.types.is_categorical_dtype(vector):  
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be  
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead  
    if pd.api.types.is_categorical_dtype(vector):
```

Out[8]: <Axes: xlabel='sex', ylabel='count'>



[실습-3] 성(sex)별 인원 수 비율 분석하기

> 새로 열 추출 생성한 데이터 프레임 사용

0. 'sex' 열의 숫자 성별 값을 의미있는 문자 값으로 변경

>> {1 : 'male', 2 : 'female'}으로 변경

1. 성별 인원 수 데이터 프레임 구성

2. 파이 그래프로 표현하기

>> 성별 인원 수 비율

>> 그래프의 시작을 상단으로 조정

```
In [9]: # 성별 항목 이름 부여
```

```
## 성별 빈도수 열 만들기
```

```
#### 파이그래프 그리기 : 'sex'별 빈도수에 대한 비율
```

```
C:\Users\ADMIN\AppData\Local\Temp\ipykernel_21212\841750372.py:2: SettingWithCopyWarning:
```

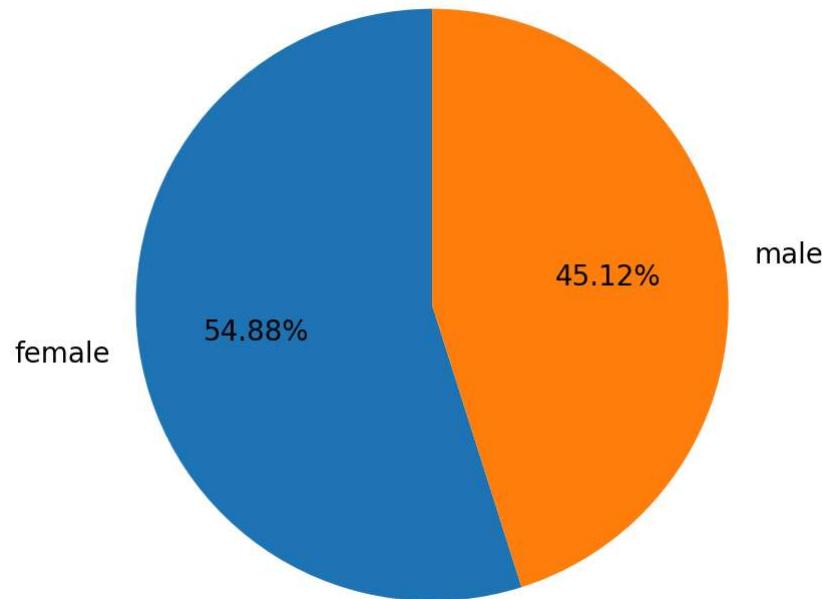
```
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
```

```
newwei['sex'] = np.where(newwei['sex'] == 1, 'male', 'female')
```

```
Out[9]: ([<matplotlib.patches.Wedge at 0x21822a2fca0>,
<matplotlib.patches.Wedge at 0x21822a2fbe0>],
[Text(-1.0870834818683026, -0.1680758859833504, 'female'),
Text(1.087083497604698, 0.1680757842032485, 'male')],
[Text(-0.5929546264736195, -0.09167775599091839, '54.88%'),
Text(0.5929546350571079, 0.09167770047449918, '45.12%')])
```



In []:

월급 변수 검토 및 전처리하기

1. 변수 검토하기

In [21]: ## 변수의 데이터 타입 확인

```
welfare['income'].dtypes # 변수 타입 출력
```

Out[21]: dtype('float64')

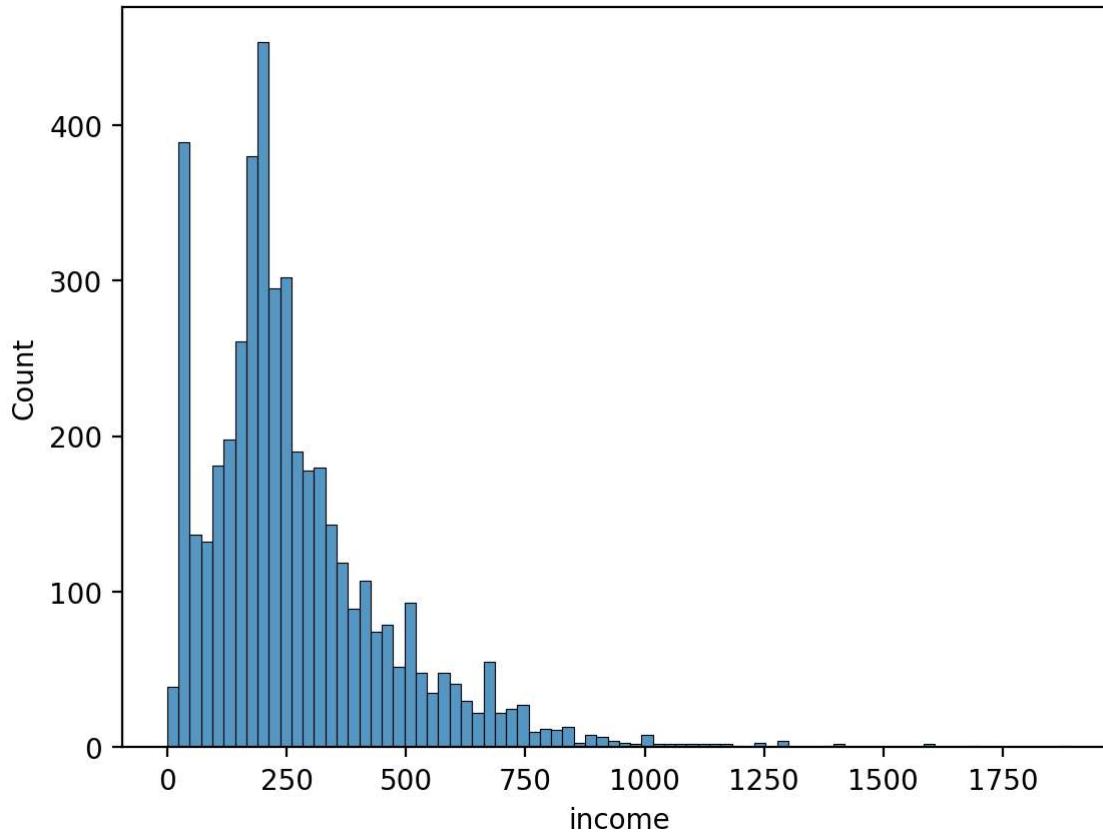
In [22]: ## 요약 통계량 구하기

```
welfare['income'].describe() # 요약 통계량 구하기
```

```
Out[22]: count    4534.000000
mean     268.455007
std      198.021206
min      0.000000
25%     150.000000
50%     220.000000
75%     345.750000
max     1892.000000
Name: income, dtype: float64
```

```
In [23]: ## 히스토그램 그려기: 'income' 빈도수
sns.histplot(data = welfare, x = 'income') # 히스토그램 만들기
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
... if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be
removed in a future version. Convert inf values to NaN before operating instead.
... with pd.option_context('mode.use_inf_as_na', True):
<Axes: xlabel='income', ylabel='Count'>
```



2. 전처리하기

```
In [24]: ## 이상치 확인  
welfare['income'].describe() # 이상치 확인
```

```
Out[24]: count    4534.000000  
mean     268.455007  
std      198.021206  
min      0.000000  
25%     150.000000  
50%     220.000000  
75%     345.750000  
max     1892.000000  
Name: income, dtype: float64
```

```
In [25]: ## 결측치 확인  
welfare['income'].isna().sum() # 결측치 확인
```

```
Out[25]: 9884
```

```
In [27]: ## incom = 9999 무응답 빈도 구하기  
welfare[welfare['income'] == 9999].value_counts() # 빈도 구하기
```

```
Out[27]: Series([], Name: count, dtype: int64)
```

```
In [28]: ## 이상치 결측 처리  
welfare['income'] = np.where(welfare['income'] == 9999, np.nan, welfare['income'])  
  
## 결측치 확인  
welfare['income'].isna().sum()
```

```
Out[28]: 9884
```

성별에 따른 월급 차이 분석하기

1. 성별 월급 평균표 만들기

```
In [29]: ## 성별 월급 평균표 만들기  
# income 결측치 제거  
# sex별 분리  
# income 평균 구하기  
sex_income = welfare.dropna(subset = ['income'])  
            .groupby('sex', as_index = False)  
            .agg(mean_income = ('income', 'mean'))  
sex_income
```

```
Out[29]:      sex  mean_income  
0  female    186.293096  
1   male     349.037571
```

2. 그래프 만들기

```
In [30]: # 막대 그래프 만들기
```

```
sns.barplot(data = sex_income, x = 'sex', y = 'mean_income')
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
```

```
.. if pd.api.types.is_categorical_dtype(vector):
```

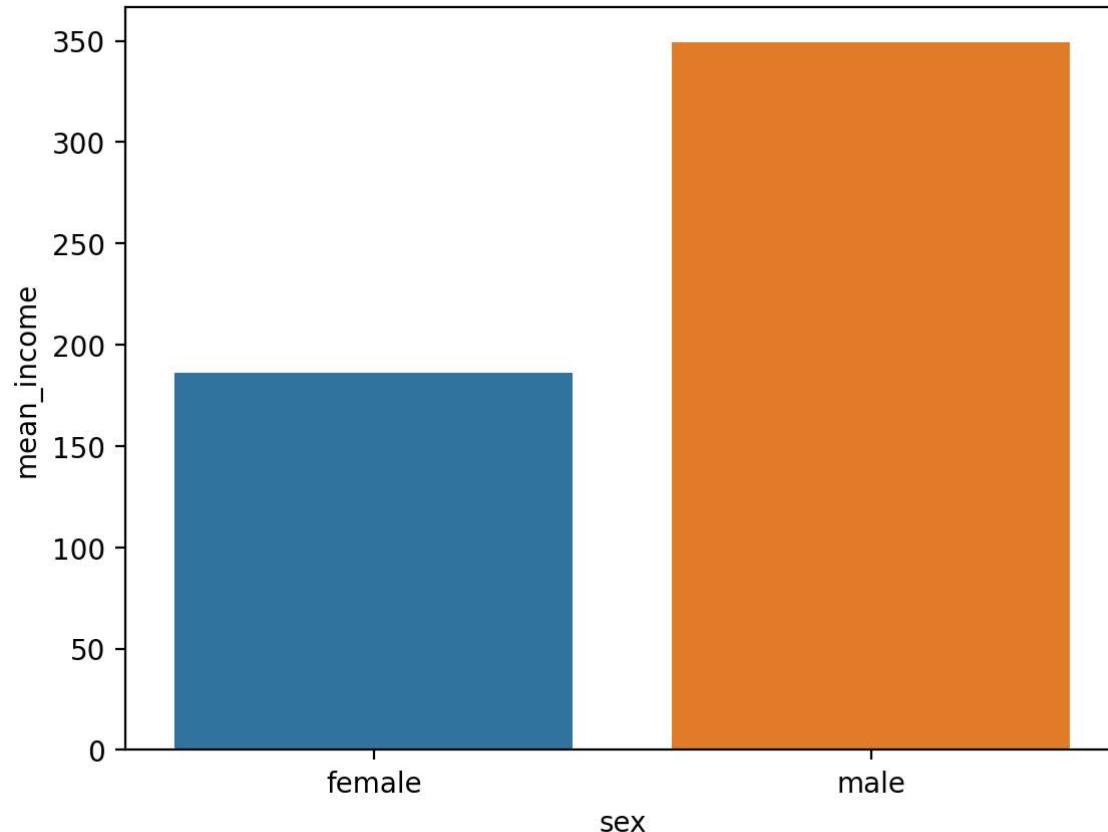
```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
```

```
.. if pd.api.types.is_categorical_dtype(vector):
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
```

```
.. if pd.api.types.is_categorical_dtype(vector):
```

```
Out[30]: <Axes: xlabel='sex', ylabel='mean_income'>
```



[선택적 실행] 여기에서부터 시작하고자 할 때 실행

In [152]:

```
# 데이터 불러오기
welfare = pd.read_spss('Koweps_hpwc14_2019_beta2.sav')

## 변수명 바꾸기
welfare = welfare.rename(columns = {'h14_g3' : 'sex', # 성별
                                    'h14_g4' : 'birth', # 태어난 연도
                                    'h14_g10' : 'marriage_type', # 혼인 상태
                                    'h14_g11' : 'religion', # 종교
                                    'p1402_8aq1' : 'income', # 월급
                                    'h14_ec09' : 'code_job', # 직업 코드
                                    'h14_reg7' : 'code_region'}) # 지역 코드

# 성별 항목 이름 부여
welfare['sex'] = np.where(welfare['sex'] == 1, 'male', 'female')
```

In []:

09-3 나이와 월급의 관계 - 몇 살 때 월급을 가장 많이 받을까?

나이 변수 검토 및 전처리하기

1. 변수 검토하기

In [31]:

```
## 변수 타입 확인
welfare['birth'].dtypes # 변수 타입 출력
```

Out[31]:

```
dtype('float64')
```

In [32]:

```
## 요약 통계량 확인
welfare['birth'].describe() # 요약 통계량 구하기
```

Out[32]:

```
count    14418.000000
mean     1969.280205
std      24.402250
min      1907.000000
25%     1948.000000
50%     1968.000000
75%     1990.000000
max     2018.000000
Name: birth, dtype: float64
```

```
In [33]: ## 히스토그램 그리기
```

```
sns.histplot(data = welfare, x = 'birth') # 히스토그램 만들기
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
```

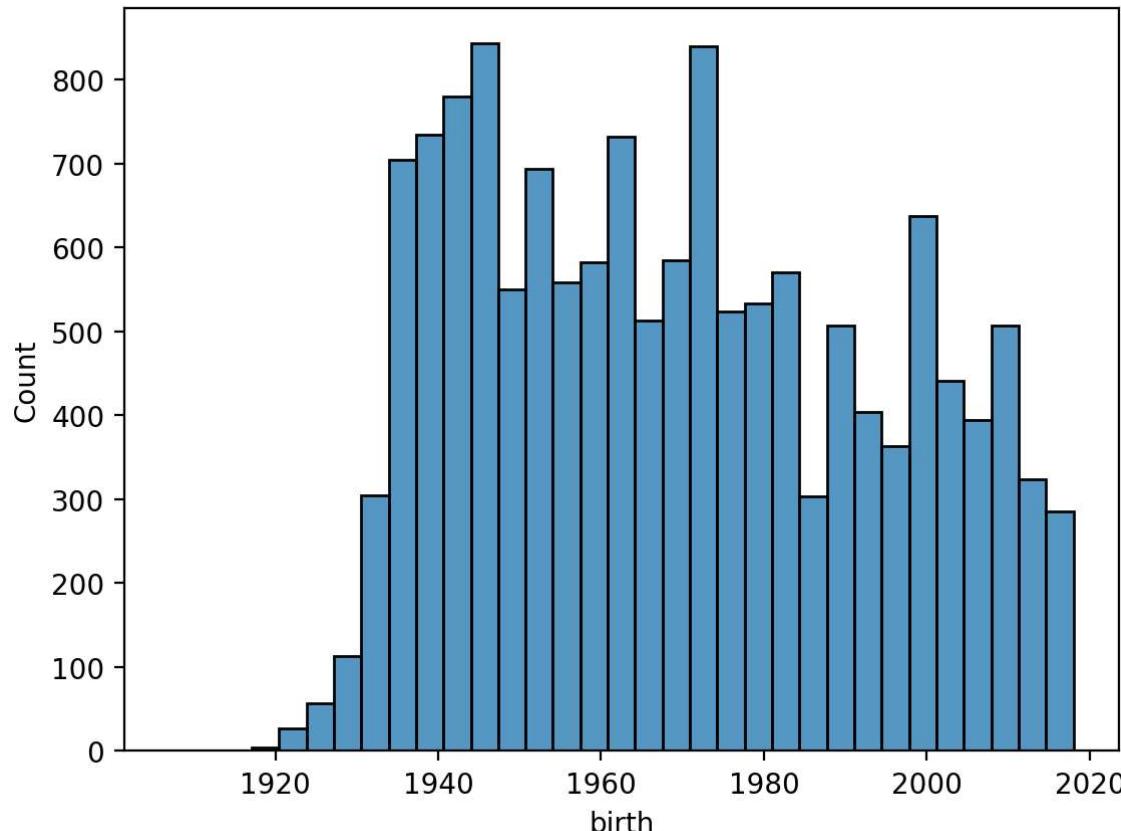
```
    if pd.api.types.is_categorical_dtype(vector):
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
```

```
    with pd.option_context('mode.use_inf_as_na', True):
```

```
<Axes: xlabel='birth', ylabel='Count'>
```

```
Out[33]:
```



2. 전처리하기

```
In [34]: ## 이상치 확인
```

```
welfare['birth'].describe() # 이상치 확인
```

```
Out[34]: count    14418.000000
          mean     1969.280205
          std      24.402250
          min     1907.000000
          25%    1948.000000
          50%    1968.000000
          75%    1990.000000
          max     2018.000000
          Name: birth, dtype: float64
```

```
In [35]: ## birth = 9999 빈도 구하기
welfare[welfare['birth'] == 9999].value_counts() # 빈도 구하기
```

```
Out[35]: Series([], Name: count, dtype: int64)
```

```
In [36]: ## 결측치 확인
welfare['birth'].isna().sum() # 결측치 확인
```

```
Out[36]: 0
```

```
In [37]: ## 이상치 결측 처리
welfare['birth'] = np.where(welfare['birth'] == 9999, np.nan, welfare['birth'])

## 결측치 확인
welfare['birth'].isna().sum()
```

```
Out[37]: 0
```

3. 파생변수 만들기 - 나이 'age'

> 나이는 2019년도를 기준으로 'birth'로 환산하여 구함

```
In [40]: ## 나이 변수 'age' 만들기
welfare = welfare.assign(age = 2019 - welfare['birth'] + 1) # 나이 변수 만들기
welfare['age'].describe() # 요약 통계량 구하기
```

```
Out[40]: count    14418.000000
          mean     50.719795
          std      24.402250
          min      2.000000
          25%     30.000000
          50%     52.000000
          75%     72.000000
          max     113.000000
          Name: age, dtype: float64
```

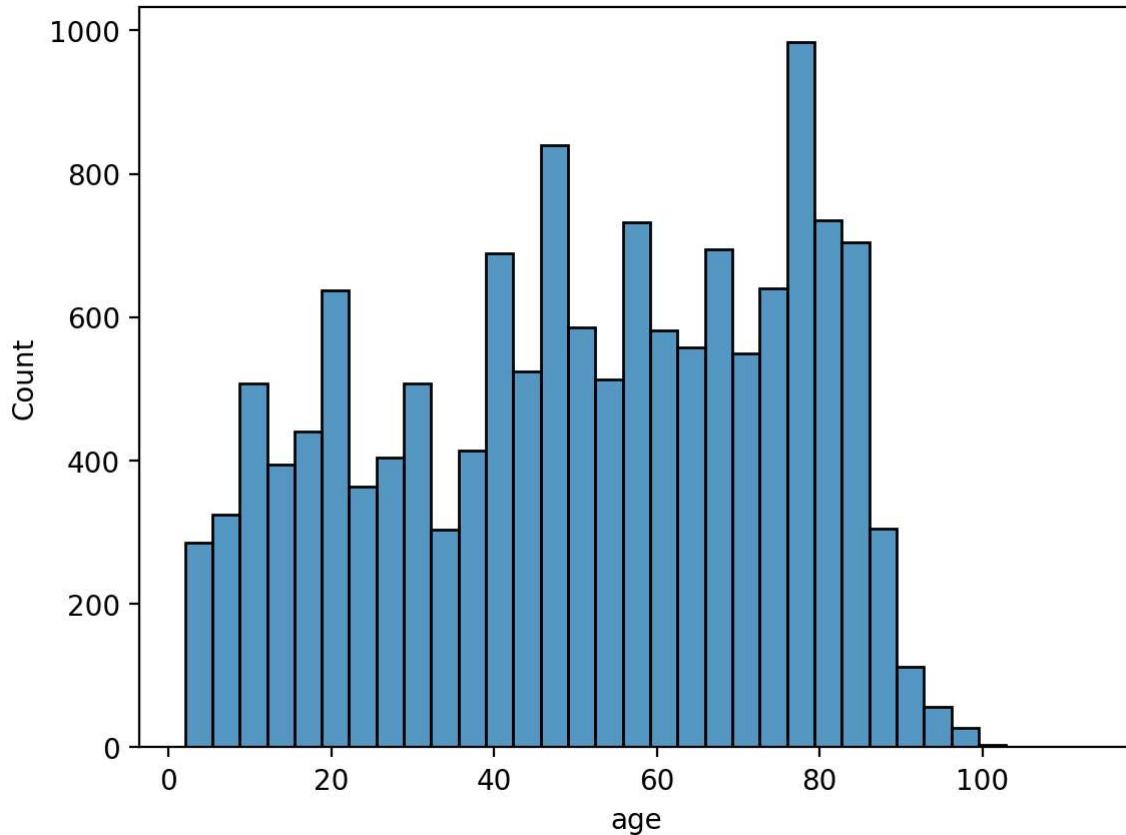
```
In [41]: ## 나이 빈도 구하기
welfare['age'].value_counts() # 빈도 구하기
```

```
Out[41]: age
78.0    317
81.0    258
80.0    255
73.0    246
77.0    233
...
96.0     4
101.0    2
100.0    2
103.0    1
113.0    1
Name: count, Length: 102, dtype: int64
```

```
In [42]: sns.histplot(data = welfare, x = 'age') # 히스토그램 만들기
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be
removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

```
Out[42]: <Axes: xlabel='age', ylabel='Count'>
```



나이와 월급의 관계 분석하기

1. 나이에 따른 월급 평균표 만들기

```
In [43]: ## 나이별 월급 평균표 만들기
# income 결측치 제거
# age별 분리
# income 평균 구하기
age_income = welfare.dropna(subset = ['income']) \
    .groupby('age') \
    .agg(mean_income = ('income', 'mean'))
age_income.head()
```

Out[43]: mean_income

age	
19.0	162.000000
20.0	121.333333
21.0	136.400000
22.0	123.666667
23.0	179.676471

2. 그래프 만들기

In [44]: # 선 그래프 만들기

```
sns.lineplot(data = age_income, x = 'age', y = 'mean_income')
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
```

```
.. if pd.api.types.is_categorical_dtype(vector):
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
```

```
.. if pd.api.types.is_categorical_dtype(vector):
```

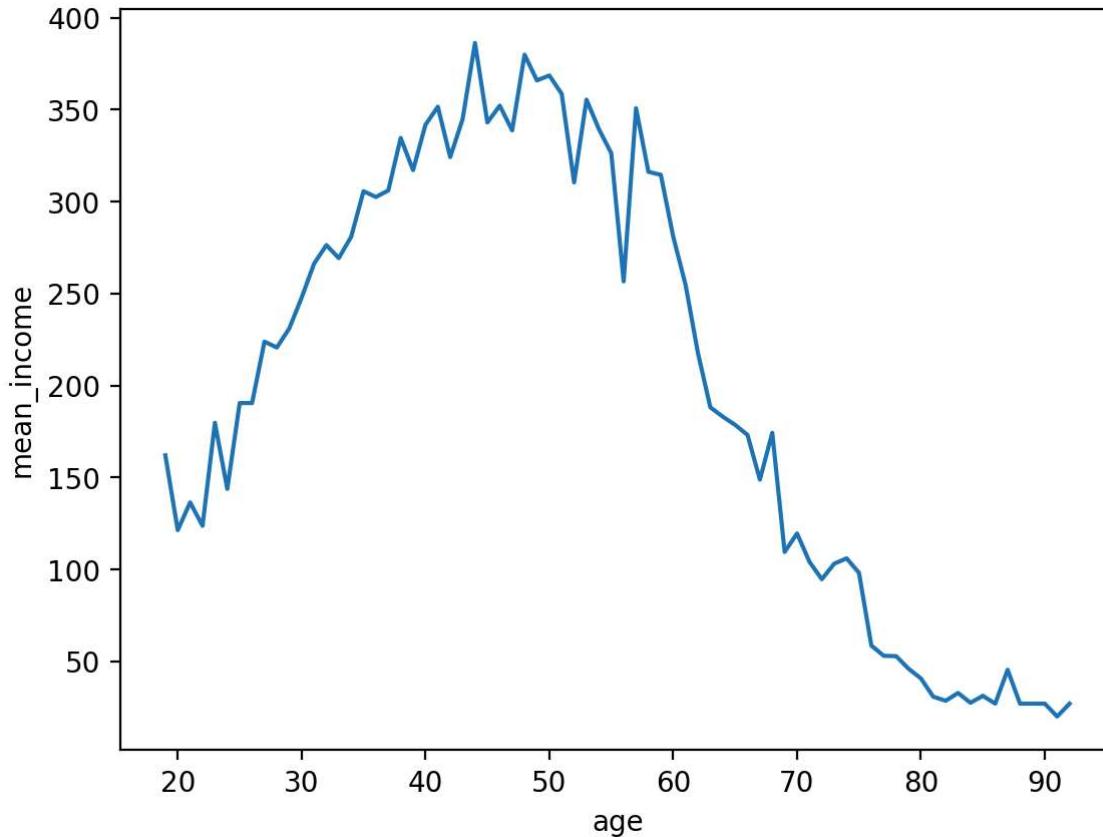
```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
```

```
.. with pd.option_context('mode.use_inf_as_na', True):
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
```

```
.. with pd.option_context('mode.use_inf_as_na', True):
```

Out[44]: <Axes: xlabel='age', ylabel='mean_income'>



In []:

09-4 연령대에 따른 월급 차이 - 어떤 연령대의 월급이 가장 많을까?

연령대 변수 검토 및 전처리하기

파생변수 만들기 - 연령대 'ageg'

> `age < 30 : 'young', 30 <= age < 60 : 'middle', age >= 60 : 'old'`

In [47]: `## 나이 변수 살펴보기
welfare['age'].head()`

```
Out[47]: 0    75.0
1    72.0
2    78.0
3    58.0
4    57.0
Name: age, dtype: float64
```

```
In [48]: ## 연령대 변수 만들기
welfare = welfare.assign(ageg = np.where(welfare['age'] < 30, 'young',
                                         np.where(welfare['age'] <= 59, 'middle', 'old')))

# 빈도 구하기
welfare['ageg'].value_counts()
```

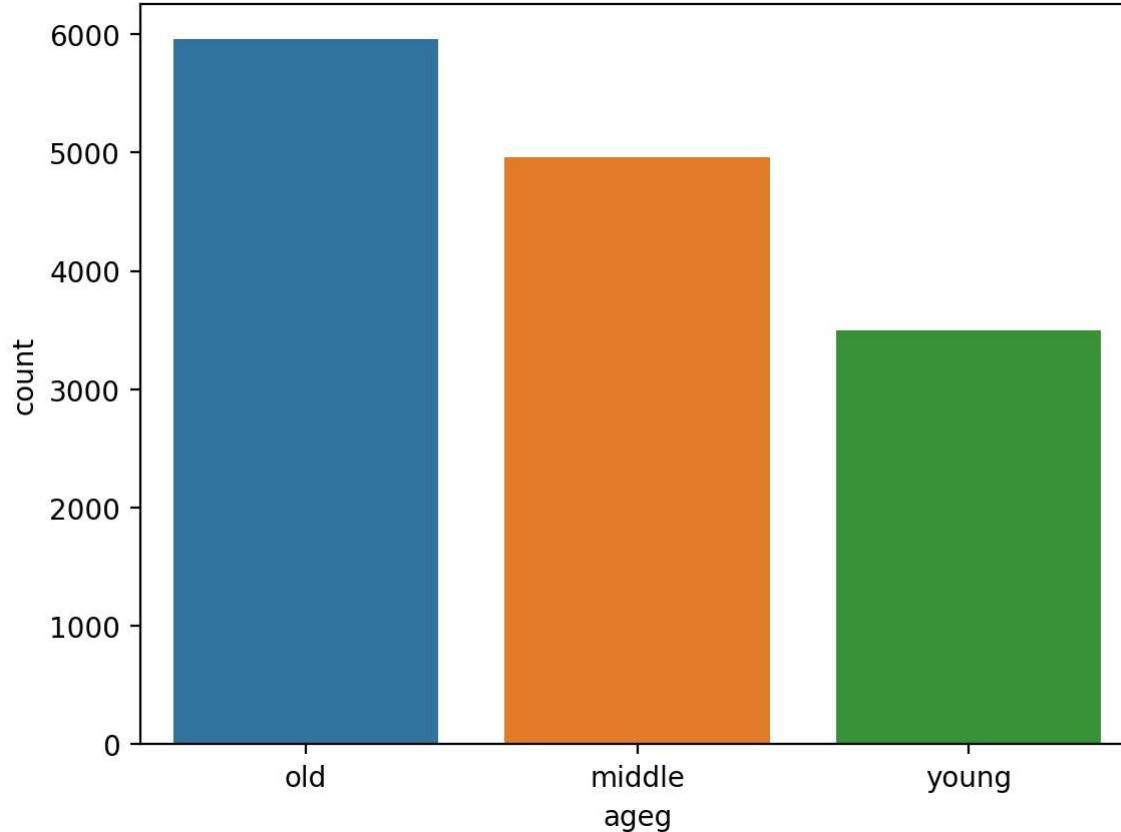
```
Out[48]: ageg
old      5955
middle   4963
young    3500
Name: count, dtype: int64
```

빈도 막대 그래프 만들기 - 연령대 'ageg'

```
In [49]: # 빈도 막대 그래프 만들기
sns.countplot(data = welfare, x = 'ageg')
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
.. if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
.. if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
.. if pd.api.types.is_categorical_dtype(vector):
```

```
Out[49]: <Axes: xlabel='ageg', ylabel='count'>
```



[실습-4] 연령대별 인원 수 비율 분석하기

: 새로 열 추출 생성한 newwel 데이터 프레임 사용

0. 'age' 변수 추가

> 나이는 2019년도를 기준으로 'birth'로 환산하여 구함

1. 연령대 파생변수 'age10' 추가

>> '00' : 0-9, '10' : 10-19, '20' : 10-29, '30' : 30-39, '40' : 40-49,

>> '50' : 50-59, '60' : 60-69, '70' : 70-79, '80' : 80-89, '90' : 90-99, '100' : 100-

2. 연령대별 평균 임금 데이터 프레임 구성

>> 'age10' 기준 연령대별 인원 수 데이터 프레임 구성

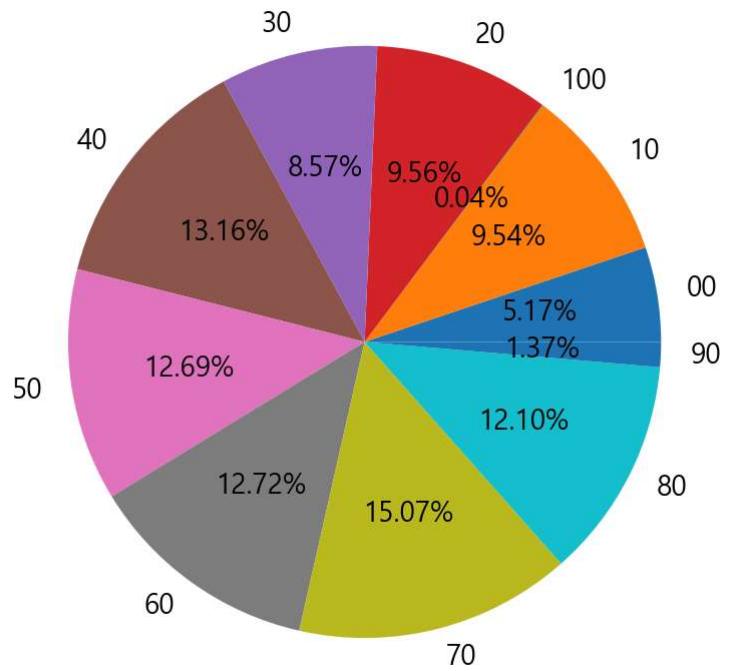
3. 파이 그래프로 표현하기

>> 연령대별 인원 수 비율 파이 그래프 그리기

In [153...]

```
## 0. 나이 변수 'age' 만들기  
  
## >> 연령대 파생변수 'age10' 추가  
  
## 1. age10 별 빈도 수 구하기  
  
## 2. 파이그래프 그리기 : 'age10'별 빈도수 비율 파이그래프
```

```
Out[153]: ([<matplotlib.patches.Wedge at 0x291acefa640>,
 . . . <matplotlib.patches.Wedge at 0x291acefa550>,
 . . . <matplotlib.patches.Wedge at 0x291aceeee370>,
 . . . <matplotlib.patches.Wedge at 0x291aceeee00>,
 . . . <matplotlib.patches.Wedge at 0x291c01af0d0>,
 . . . <matplotlib.patches.Wedge at 0x291c01af760>,
 . . . <matplotlib.patches.Wedge at 0x291c01afdf0>,
 . . . <matplotlib.patches.Wedge at 0x291c01a24c0>,
 . . . <matplotlib.patches.Wedge at 0x291c01a2b50>,
 . . . <matplotlib.patches.Wedge at 0x291c018a220>,
 . . . <matplotlib.patches.Wedge at 0x291acefa610>],
 [Text(1.0854998125739637, 0.1780172938281284, '00'),
 . . . Text(0.8921113050466446, 0.6435350957080528, '10'),
 . . . Text(0.6610964987653354, 0.8791765575356381, '100'),
 . . . Text(0.3701345051584889, 1.0358573492962633, '20'),
 . . . Text(-0.24690900358045392, 1.0719309417825889, '30'),
 . . . Text(-0.8678143369759413, 0.675942509788375, '40'),
 . . . Text(-1.087591117552792, -0.16475909996194268, '50'),
 . . . Text(-0.6410054437457209, -0.8939306578747321, '60'),
 . . . Text(0.27322461798300507, -1.0655272441979329, '70'),
 . . . Text(0.9826730137468299, -0.49432150272239117, '80'),
 . . . Text(1.098986749652936, -0.047203009303173823, '90')],
 [Text(0.5920908068585256, 0.09710034208807002, '5.17%'),
 . . . Text(0.48660616638907883, 0.3510191431134833, '9.54%'),
 . . . Text(0.36059809023563744, 0.4795508495648934, '0.04%'),
 . . . Text(0.20189154826826664, 0.5650130996161435, '9.56%'),
 . . . Text(-0.13467763831661123, 0.5846896046086847, '8.57%'),
 . . . Text(-0.4733532747141498, 0.3686959144300227, '13.16%'),
 . . . Text(-0.5932315186651591, -0.08986859997924145, '12.69%'),
 . . . Text(-0.3496393329522114, -0.48759854065894476, '12.72%'),
 . . . Text(0.14903160980891184, -0.5811966786534178, '15.07%'),
 . . . Text(0.5360034620437253, -0.2696299105758497, '12.10%'),
 . . . Text(0.5994473179925104, -0.025747095983549354, '1.37%')])
```



연령대에 따른 월급 차이 분석하기

1. 연령대별 월급 평균표 만들기

```
In [51]: ## 연령대별 월급 평균표 만들기
# income 결측치 제거
# ageg별 분리
# income 평균 구하기
ageg_income = welfare.dropna(subset = ['income']) %
    .groupby('ageg', as_index = False) %
    .agg(mean_income = ('income', 'mean'))
ageg_income
```

```
Out[51]:    ageg  mean_income
```

	ageg	mean_income
0	middle	329.157157
1	old	140.129003
2	young	195.663424

2. 그래프 만들기

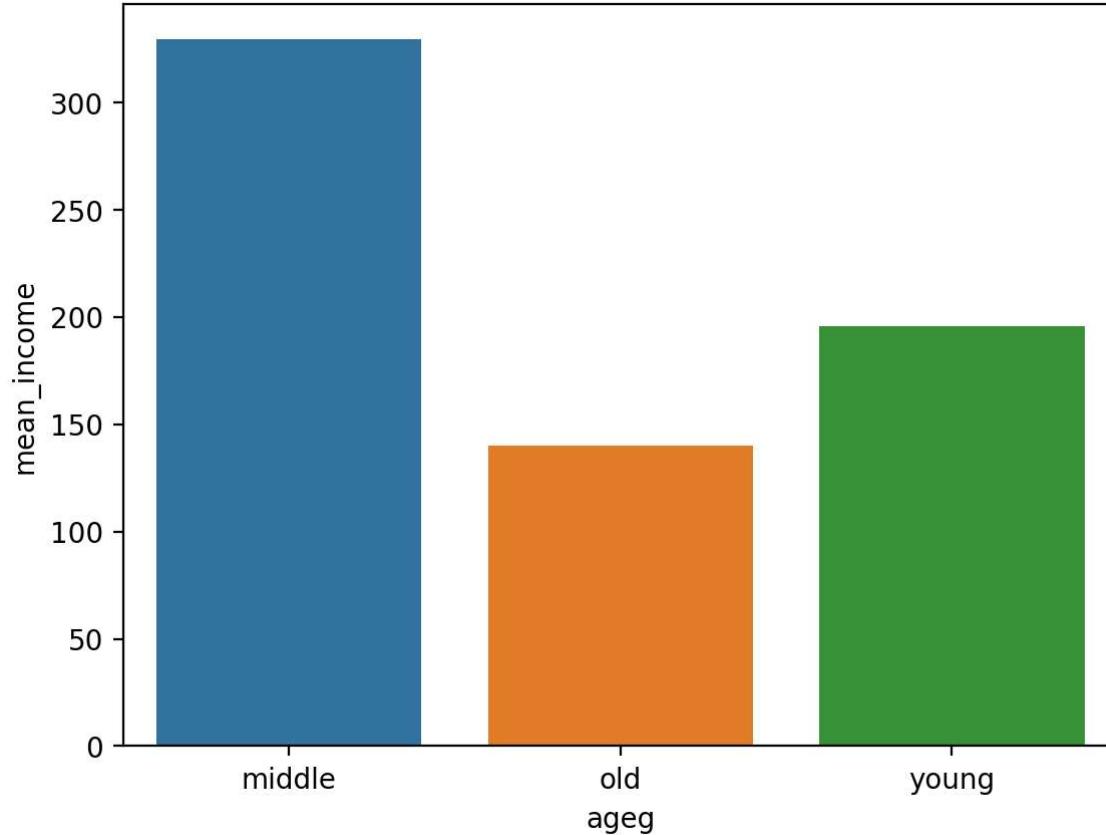
```
In [52]:
```

```
# 막대 그래프 만들기
sns.barplot(data = ageg_income, x = 'ageg', y = 'mean_income')
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
.. if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
.. if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
.. if pd.api.types.is_categorical_dtype(vector):
```

```
Out[52]:
```

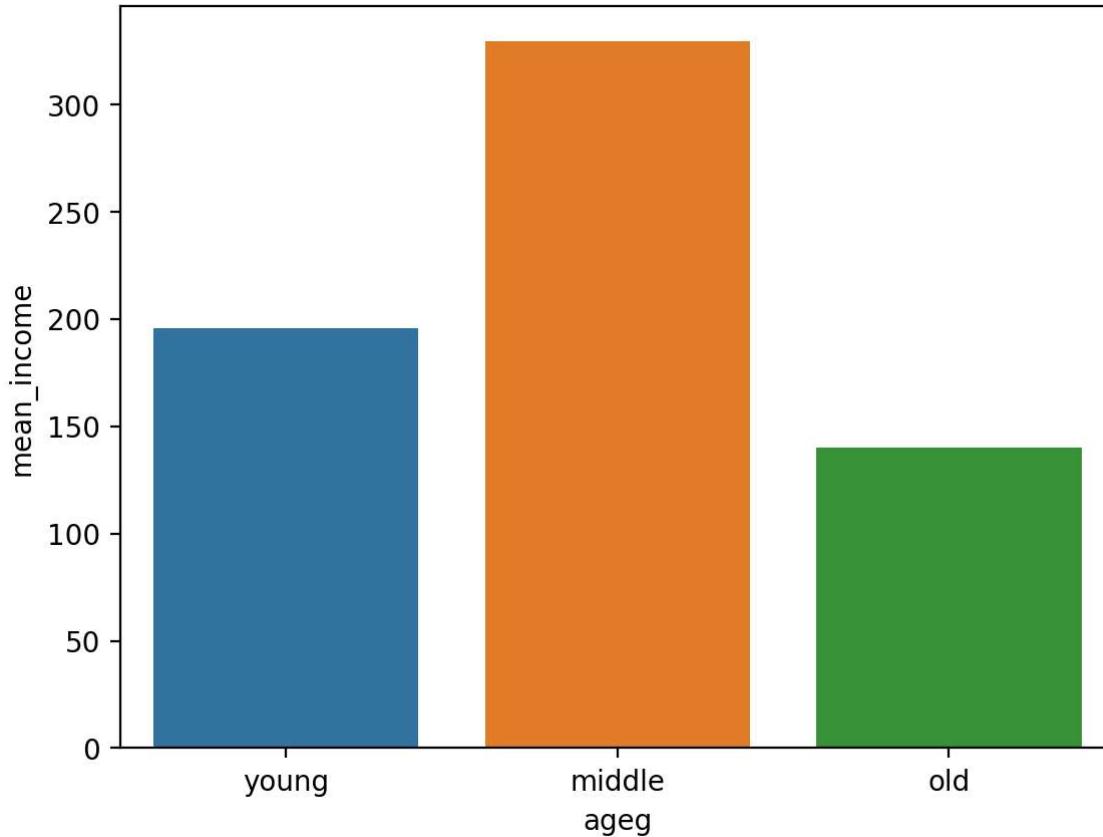
```
<Axes: xlabel='ageg', ylabel='mean_income'>
```



```
In [53]: # 막대 x-축 정렬하기
sns.barplot(data = ageg_income, x = 'ageg', y = 'mean_income',
             order = ['young', 'middle', 'old'])
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
```

```
Out[53]: <Axes: xlabel='ageg', ylabel='mean_income'>
```



[선택적 실행] 여기에서부터 시작하고자 할 때 실행

```
In [238...]: # 데이터 불러오기
welfare = pd.read_spss('Koweps_hpwc14_2019_beta2.sav')

## 변수명 바꾸기
welfare = welfare.rename(columns = {'h14_g3' : 'sex',          # 성별
                                    'h14_g4' : 'birth',        # 태어난 연도
                                    'h14_g10' : 'marriage_type', # 혼인 상태
                                    'h14_g11' : 'religion',    # 종교
                                    'p1402_8aq1' : 'income',   # 월급
                                    'h14_eco9' : 'code_job',   # 직업 코드
                                    'h14_reg7' : 'code_region'}) # 지역 코드

# 성별 항목 이름 부여
welfare['sex'] = np.where(welfare['sex'] == 1, 'male', 'female')
```

```

## 나이 변수 'age' 만들기
welfare = welfare.assign(age = 2019 - welfare['birth'] + 1) # 나이 변수 만들기

## 연령대 변수 'ageg' 만들기
welfare = welfare.assign(ageg = np.where(welfare['age'] < 30, 'young',
                                         np.where(welfare['age'] <= 59, 'middle', 'old')))

```

In []:

09-5 연령대 및 성별 월급 차이 - 성별 월급 차이는 연령대별로 다를까?

연령대 및 성별 월급 차이 분석하기

1. 연령대 및 성별 월급 평균표 만들기

In [54]: ## 연령대 및 성별 평균표 만들기

```

# income 결측치 제거
# ageg 및 sex별 분리
# income 평균 구하기
sex_income = welfare.dropna(subset = ['income']) \
    .groupby(['ageg', 'sex'], as_index = False) \
    .agg(mean_income = ('income', 'mean'))
sex_income

```

Out[54]:

	ageg	sex	mean_income
0	middle	female	230.481735
1	middle	male	409.541228
2	old	female	90.228896
3	old	male	204.570231
4	young	female	189.822222
5	young	male	204.909548

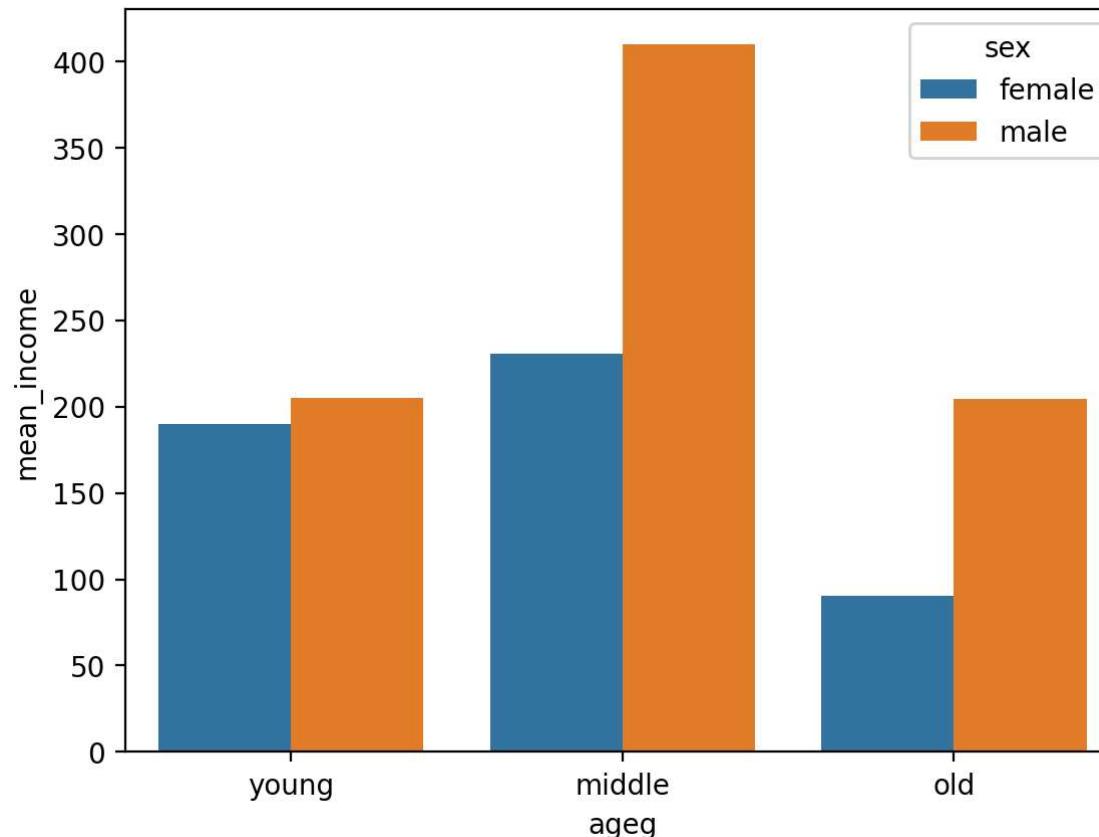
2. 그래프 만들기

```
In [55]: # 막대 그래프 만들기
```

```
sns.barplot(data = sex_income, x = 'ageg', y = 'mean_income', hue = 'sex',
             order = ['young', 'middle', 'old'])
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
.. if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
.. if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
.. if pd.api.types.is_categorical_dtype(vector):
```

```
Out[55]: <Axes: xlabel='ageg', ylabel='mean_income'>
```



나이 및 성별 월급 차이 분석하기

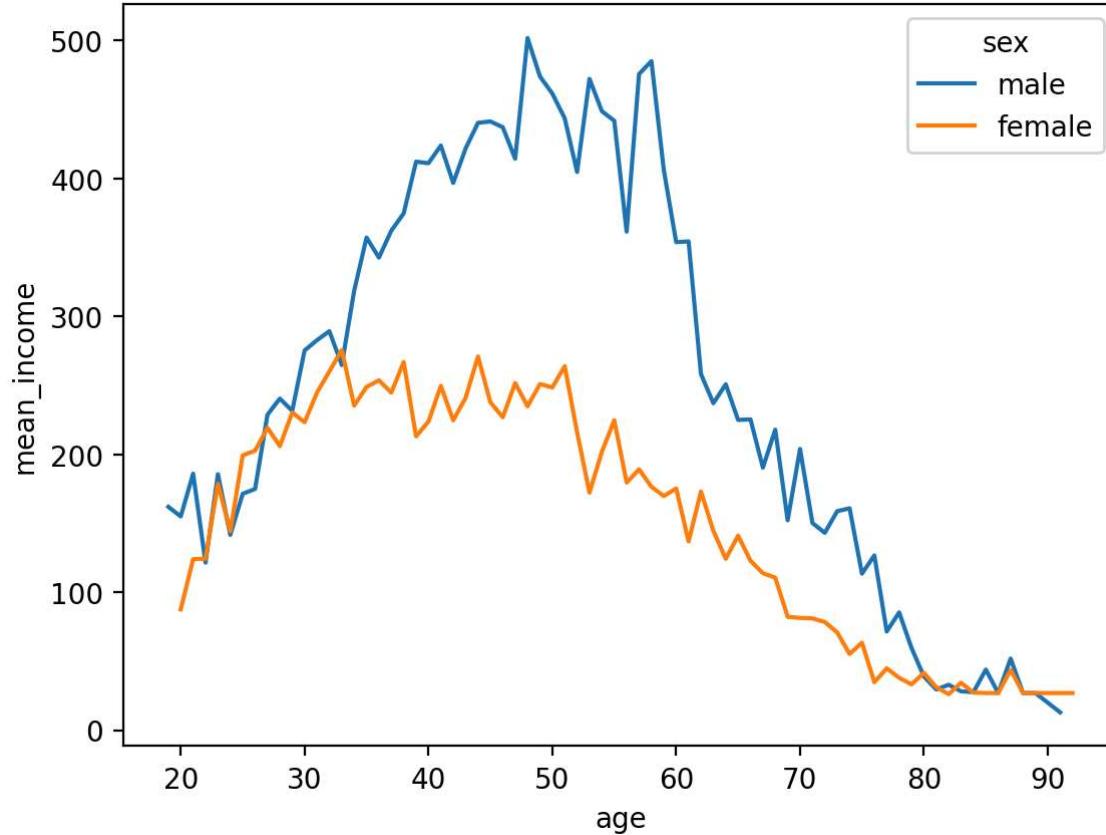
```
In [56]: ## 나이 및 성별 월급 평균표 만들기  
# income 결측치 제거  
# age 및 sex별 분리  
# income 평균 구하기  
sex_age = welfare.dropna(subset = ['income'])  
    .groupby(['age', 'sex'], as_index = False)  
    .agg(mean_income = ('income', 'mean'))  
sex_age.head()
```

```
Out[56]:   age    sex  mean_income  
0  19.0  male    162.000000  
1  20.0 female    87.666667  
2  20.0  male    155.000000  
3  21.0 female    124.000000  
4  21.0  male    186.000000
```

```
In [57]: # 선 그래프 만들기  
sns.lineplot(data = sex_age, x = 'age', y = 'mean_income', hue = 'sex')
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
.. if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
.. if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
.. if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
.. if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
.. if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
.. with pd.option_context('mode.use_inf_as_na', True):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
.. with pd.option_context('mode.use_inf_as_na', True):
<Axes: xlabel='age', ylabel='mean_income'>
```

Out[57]:



In []:

09-6 직업별 월급 차이 - 어떤 직업이 월급을 가장 많이 받을까?

직업 변수 검토 및 전처리하기

1. 변수 검토하기

```
In [58]: ## 변수 타입 확인  
welfare['code_job'].dtypes # 변수 타입 출력
```

Out[58]: dtype('float64')

```
In [59]: ## 빈도 확인하기  
welfare['code_job'].value_counts() # 빈도 구하기
```

```
Out[59]: code_job  
611.0    962  
941.0    391  
521.0    354  
312.0    275  
873.0    236  
...  
112.0     2  
784.0     2  
423.0     1  
861.0     1  
872.0     1  
Name: count, Length: 150, dtype: int64
```

2. 전처리하기

한국복지패널 제공 관련 **CodeBook** 파일에서 직종코드 로딩

> 'Koweps_Codebook_2019.xlsx' 파일의 '직종코드' 시트를 읽어들여서 데이터 프레임 구성 후 출력

```
In [60]: ## '직종코드' 시트를 읽어들여서 데이터 프레임 구성  
list_job = pd.read_excel('Koweps_Codebook_2019.xlsx', sheet_name = '직종코드')  
list_job.head()
```

	code_job	job
0	111	의회 의원·고위 공무원 및 공공단체 임원
1	112	기업 고위 임원
2	121	행정 및 경영 지원 관리자
3	122	마케팅 및 광고·홍보 관리자
4	131	연구·교육 및 법률 관련 관리자

```
In [65]: ## 행, 열 개수 확인  
list_job.shape # 행, 열 개수 출력
```

```
Out[65]: (156, 2)
```

```
In [66]: ## welfare['code_job'] 결측치 확인
```

```
welfare['code_job'].isna().sum() # 결측치 확인
```

```
Out[66]: 7540
```

```
In [70]: ## welfare에 list_job 결합하기
```

```
welfare = welfare.merge(list_job, how = 'left', on = 'code_job')  
welfare.head()
```

	h14_id	h14_ind	h14_sn	h14_merkey	h_new	h14_cobf	p14_wsc	p14_wsl	p14_wgc	p14_wgl	...	wc14_5aq5	wc14_5aq6	h14_pers_income1
0	2.0	1.0	1.0	20101.0	0.0	NaN	0.291589	0.291589	1307.764781	1307.764781	...	NaN	NaN	NaN
1	3.0	1.0	1.0	30101.0	0.0	NaN	0.419753	0.419753	1882.570960	1882.570960	...	NaN	NaN	NaN
2	4.0	1.0	1.0	40101.0	0.0	NaN	0.265263	0.265980	1189.691668	1192.908537	...	NaN	NaN	NaN
3	6.0	1.0	1.0	60101.0	0.0	NaN	0.494906	0.495941	2219.630833	2224.273816	...	4.0	4.0	2304.0
4	6.0	1.0	1.0	60101.0	0.0	NaN	1.017935	1.017935	4565.389177	4565.389177	...	5.0	2.0	NaN

5 rows × 833 columns

!!! 아래 셀은 위에서 생성한 컬럼을 삭제할 때만 사용 !!!

```
In [68]: ## 열 제거 [필요 시 실행]
```

```
welfare = welfare.drop(['job_x', 'job_y'], axis = 'columns')  
welfare.head()
```

Out[68]:

	h14_id	h14_ind	h14_sn	h14_merkey	h_new	h14_cobf	p14_wsc	p14_wsl	p14_wgc	p14_wgl	...	wc14_5aq4	wc14_5aq5	wc14_5aq6	h14_p
0	2.0	1.0	1.0	20101.0	0.0	NaN	0.291589	0.291589	1307.764781	1307.764781	...	NaN	NaN	NaN	NaN
1	3.0	1.0	1.0	30101.0	0.0	NaN	0.419753	0.419753	1882.570960	1882.570960	...	NaN	NaN	NaN	NaN
2	4.0	1.0	1.0	40101.0	0.0	NaN	0.265263	0.265980	1189.691668	1192.908537	...	NaN	NaN	NaN	NaN
3	6.0	1.0	1.0	60101.0	0.0	NaN	0.494906	0.495941	2219.630833	2224.273816	...	2.0	4.0	4.0	4.0
4	6.0	1.0	1.0	60101.0	0.0	NaN	1.017935	1.017935	4565.389177	4565.389177	...	1.0	5.0	2.0	2.0

5 rows × 832 columns

In [71]: # code_job 결측치 제거하고 code_job, job 출력
welfare.dropna(subset = ['code_job'])[['code_job', 'job']].head()

Out[71]:

	code_job	job
2	762.0	전기공
3	855.0	금속기계 부품 조립원
7	941.0	청소원 및 환경미화원
8	999.0	기타 서비스 관련 단순 종사자
14	312.0	경영 관련 사무원

In [72]: # code_job 결측치 제거하고 행의 수 출력
len(welfare.dropna(subset = ['code_job']))

Out[72]: 6878

직업별 월급 차이 분석하기

1. 직업별 월급 평균표 만들기

In [73]: ## 직업별 월급 평균표 만들기

job, income 결측치 제거
job별 분리

```
# income 평균 구하기
job_income = welfare.dropna(subset = ['job', 'income']) %>
  .groupby('job', as_index = False) %>
  .agg(mean_income = ('income', 'mean'))
job_income.head()
```

Out[73]:

	job	mean_income
0	가사 및 육아 도우미	92.455882
1	간호사	265.219178
2	감정·기술영업및증개관련종사자	391.000000
3	건물 관리원 및 검표원	168.375000
4	건설 및 광업 단순 종사자	261.975000

3. 그래프 만들기

(1) 월급이 많은 직업

In [74]: ## 상위 10위 추출
top10 = job_income.sort_values('mean_income', ascending = False).head(10)
top10

Out[74]:

	job	mean_income
98	의료 진료 전문가	781.000000
60	법률 전문가	776.333333
140	행정 및 경영 지원 관리자	771.833333
63	보험 및 금융 관리자	734.750000
110	재활용 처리 및 소각로 조작원	688.000000
131	컴퓨터 하드웨어 및 통신공학 전문가	679.444444
24	기계·로봇공학 기술자 및 시험원	669.166667
6	건설·전기 및 생산 관련 관리자	603.083333
120	제관원 및 판금원	597.000000
100	의회 의원·고위 공무원 및 공공단체 임원	580.500000

In [75]:

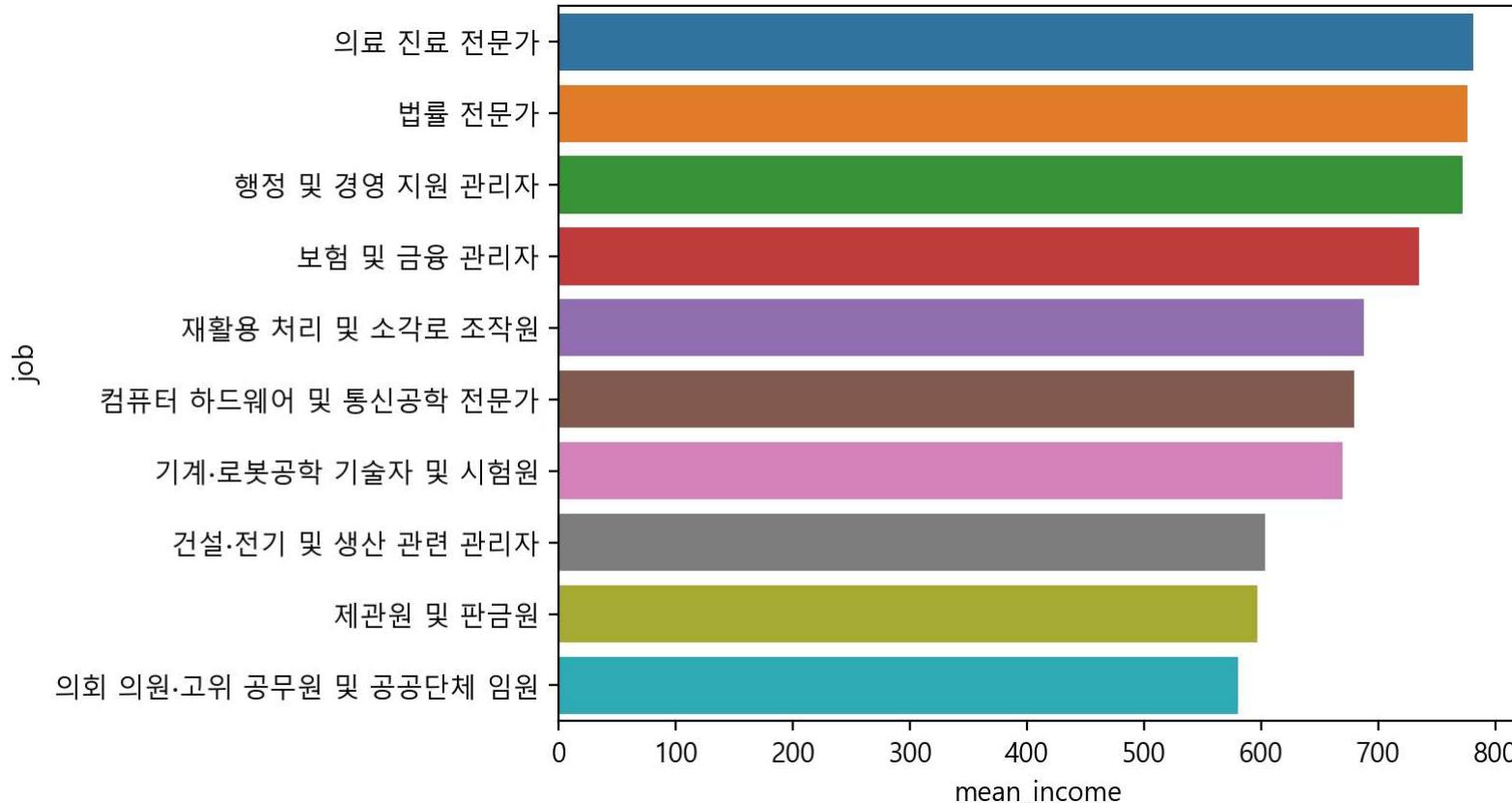
```
# 맑은 고딕 폰트 설정
import matplotlib.pyplot as plt
plt.rcParams.update({'font.family' : 'Malgun Gothic'})
```

In [76]:

```
# 막대 그래프 만들기
sns.barplot(data = top10, y = 'job', x = 'mean_income')
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
... if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
... if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
... if pd.api.types.is_categorical_dtype(vector):
<Axes: xlabel='mean_income', ylabel='job'>
```

Out[76]:



(2) 월급이 적은 직업

```
In [77]: ## 하위 10위 추출  
bottom10 = job_income.sort_values('mean_income').head(10)  
bottom10
```

Out[77]:

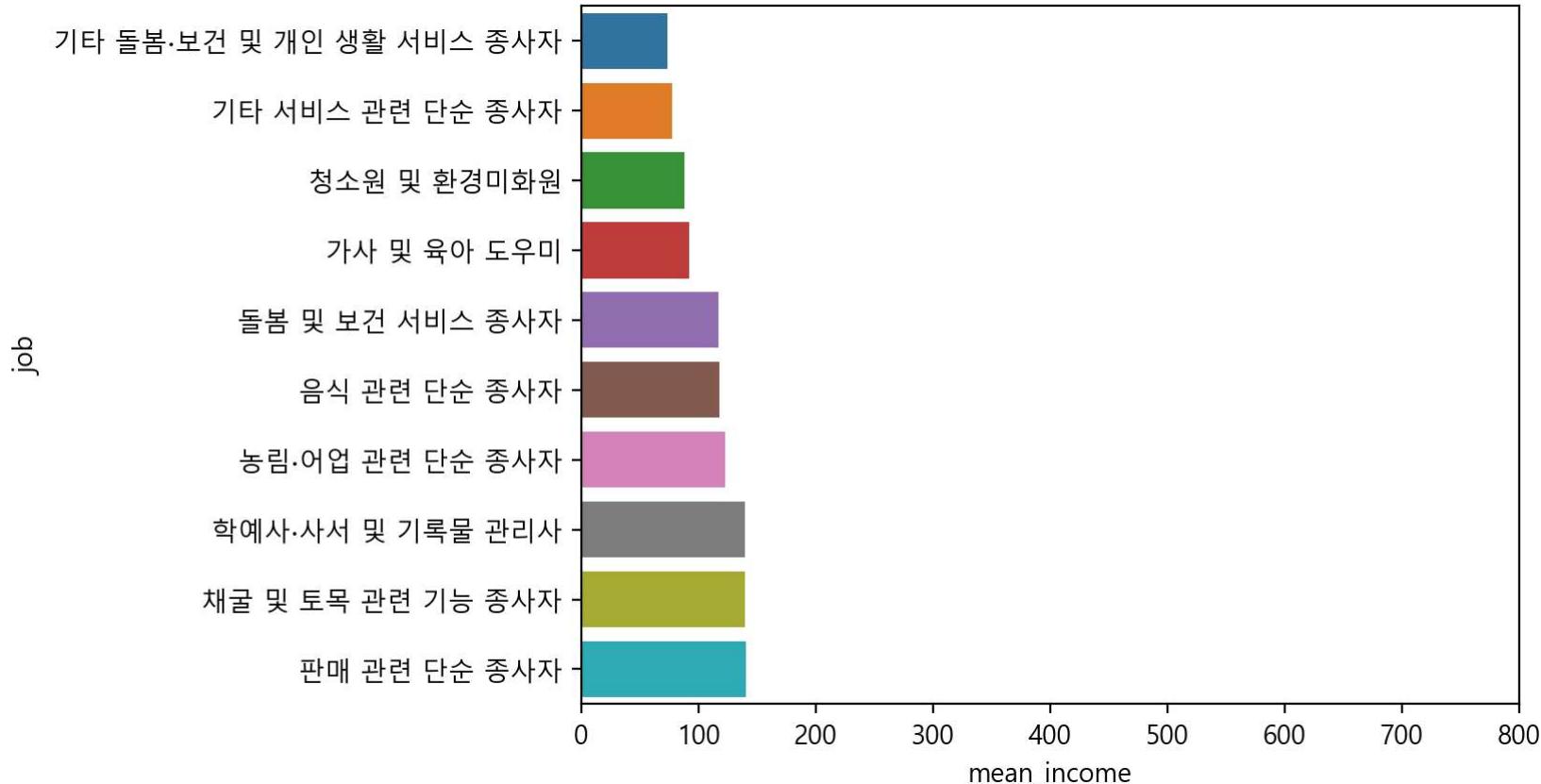
	job	mean_income
33	기타 돌봄·보건 및 개인 생활 서비스 종사자	73.964286
34	기타 서비스 관련 단순 종사자	77.789474
128	청소원 및 환경미화원	88.461756
0	가사 및 육아 도우미	92.455882
43	돌봄 및 보건 서비스 종사자	117.162338
97	음식 관련 단순 종사자	118.187500
39	농림·어업 관련 단순 종사자	122.625000
139	학예사·사서 및 기록물 관리사	140.000000
126	채굴 및 토목 관련 기능 종사자	140.000000
135	판매 관련 단순 종사자	140.909091

In [78]:

```
# 막대 그래프 만들기
sns.barplot(data = bottom10, y = 'job', x = 'mean_income') # 
    .set(xlim = [0, 800])
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
[(0.0, 800.0)]
```

Out[78]:



[선택적 실행] 여기에서부터 시작하고자 할 때 실행

```
In [260...]: # 데이터 불러오기
welfare = pd.read_spss('Koweps_hpwc14_2019_beta2.sav')

## 변수명 바꾸기
welfare = welfare.rename(columns = {'h14_g3' : 'sex',          # 성별
                                    'h14_g4' : 'birth',        # 태어난 연도
                                    'h14_g10' : 'marriage_type', # 혼인 상태
                                    'h14_g11' : 'religion',    # 종교
                                    'p1402_8aq1' : 'income',   # 월급
                                    'h14_eco9' : 'code_job',   # 직업 코드
                                    'h14_reg7' : 'code_region'}) # 지역 코드

# 성별 항목 이름 부여
welfare['sex'] = np.where(welfare['sex'] == 1, 'male', 'female')
```

```

## 나이 변수 'age' 만들기
welfare = welfare.assign(age = 2019 - welfare['birth'] + 1) # 나이 변수 만들기

## 연령대 변수 'ageg' 만들기
welfare = welfare.assign(ageg = np.where(welfare['age'] < 30, 'young',
                                         np.where(welfare['age'] <= 59, 'middle', 'old')))

## '직종코드' 시트를 읽어들여서 데이터 프레임 구성
list_job = pd.read_excel('Koweps_Codebook_2019.xlsx', sheet_name = '직종코드')

## welfare에 list_job 결합하기
welfare = welfare.merge(list_job, how = 'left', on = 'code_job')

```

In []:

09-7 성별 직업 빈도 - 성별로 어떤 직업이 가장 많을까?

성별 직업 빈도 분석하기

1. 성별 직업 빈도표 만들기

```

In [79]: ## 남성 직업 빈도 상위 10개 추출
# job 결측치 제거, # male 추출, # job별 분리, # job 빈도 구하기
# 내림차순 정렬, # 상위 10행 추출,
job_male = welfare.dropna(subset = ['job']) \
    .query("sex == 'male'" ) \
    .groupby('job', as_index = False) \
    .agg(n = ('job', 'count')) \
    .sort_values('n', ascending = False) \
    .head(10)
job_male

```

Out[79]:

	job	n
107	작물 재배 종사자	486
104	자동차 운전원	230
11	경영 관련 사무원	216
46	매장 판매 종사자	142
89	영업 종사자	113
127	청소원 및 환경미화원	109
4	건설 및 광업 단순 종사자	96
120	제조 관련 단순 종사자	80
3	건물 관리원 및 검표원	79
141	행정 사무원	74

In [80]:

```
## 여성 직업 빈도 상위 10개 추출
# job 결측치 제거, # female 추출, # job별 분리
# job 빈도 구하기, # 내림차순 정렬, # 상위 10행 추출
job_female = welfare.dropna(subset = ['job']) |
    .query("sex == 'female'") |
    .groupby('job', as_index = False) |
    .agg(n = ('job', 'count')) |
    .sort_values('n', ascending = False) |
    .head(10)
job_female
```

Out[80]:

	job	n
83	작물 재배 종사자	476
91	청소원 및 환경미화원	282
33	매장 판매 종사자	212
106	회계 및 경리 사무원	163
31	돌봄 및 보건 서비스 종사자	155
87	제조 관련 단순 종사자	148
73	음식 관련 단순 종사자	126
58	식음료 서비스 종사자	117
88	조리사	114
24	기타 서비스 관련 단순 종사자	97

2. 그래프 만들기

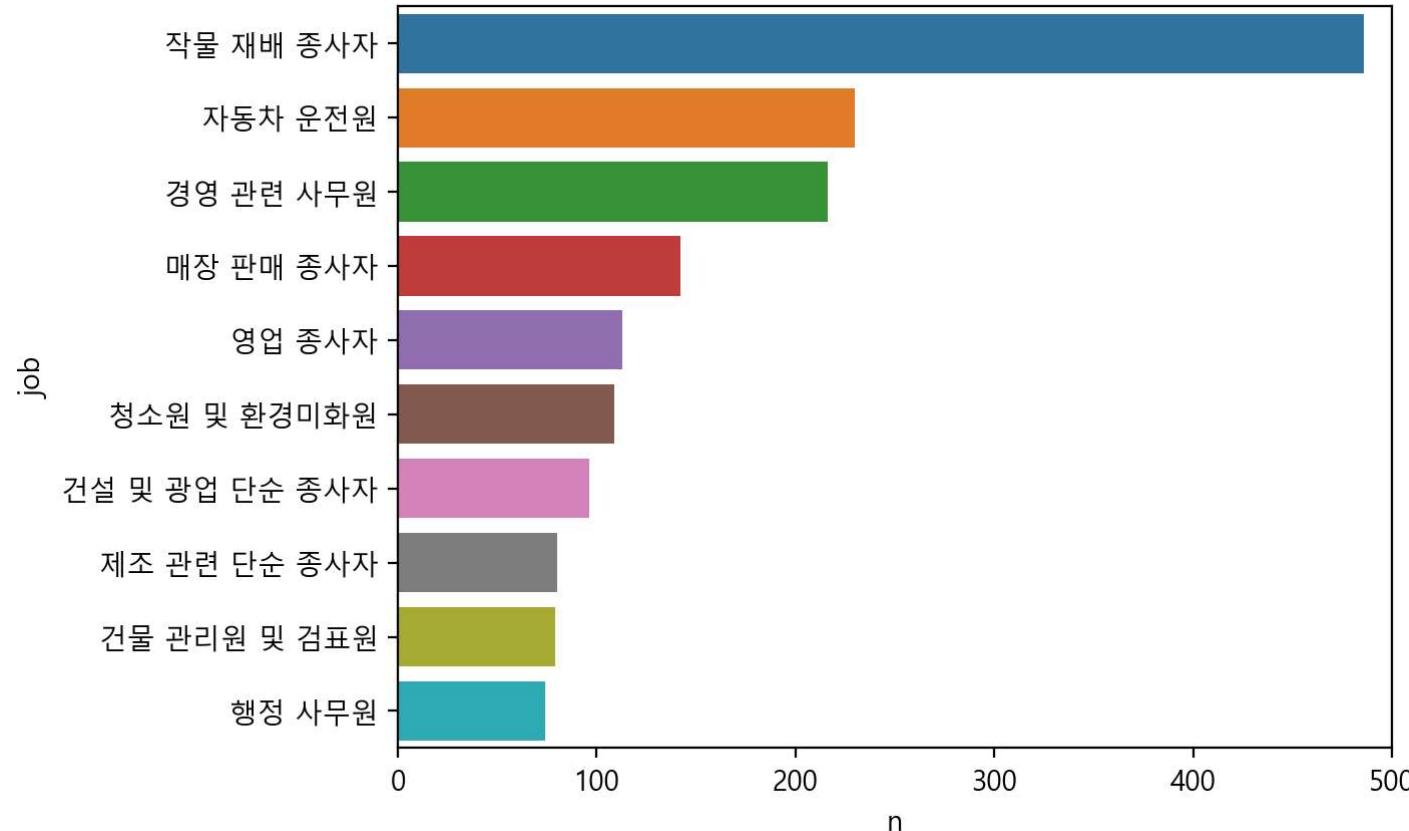
In [81]:

```
# 남성 직업 빈도 막대 그래프 만들기
sns.barplot(data = job_male, y = 'job', x = 'n').set(xlim = [0, 500])
```

C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
... if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
... if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
... if pd.api.types.is_categorical_dtype(vector):

Out[81]:

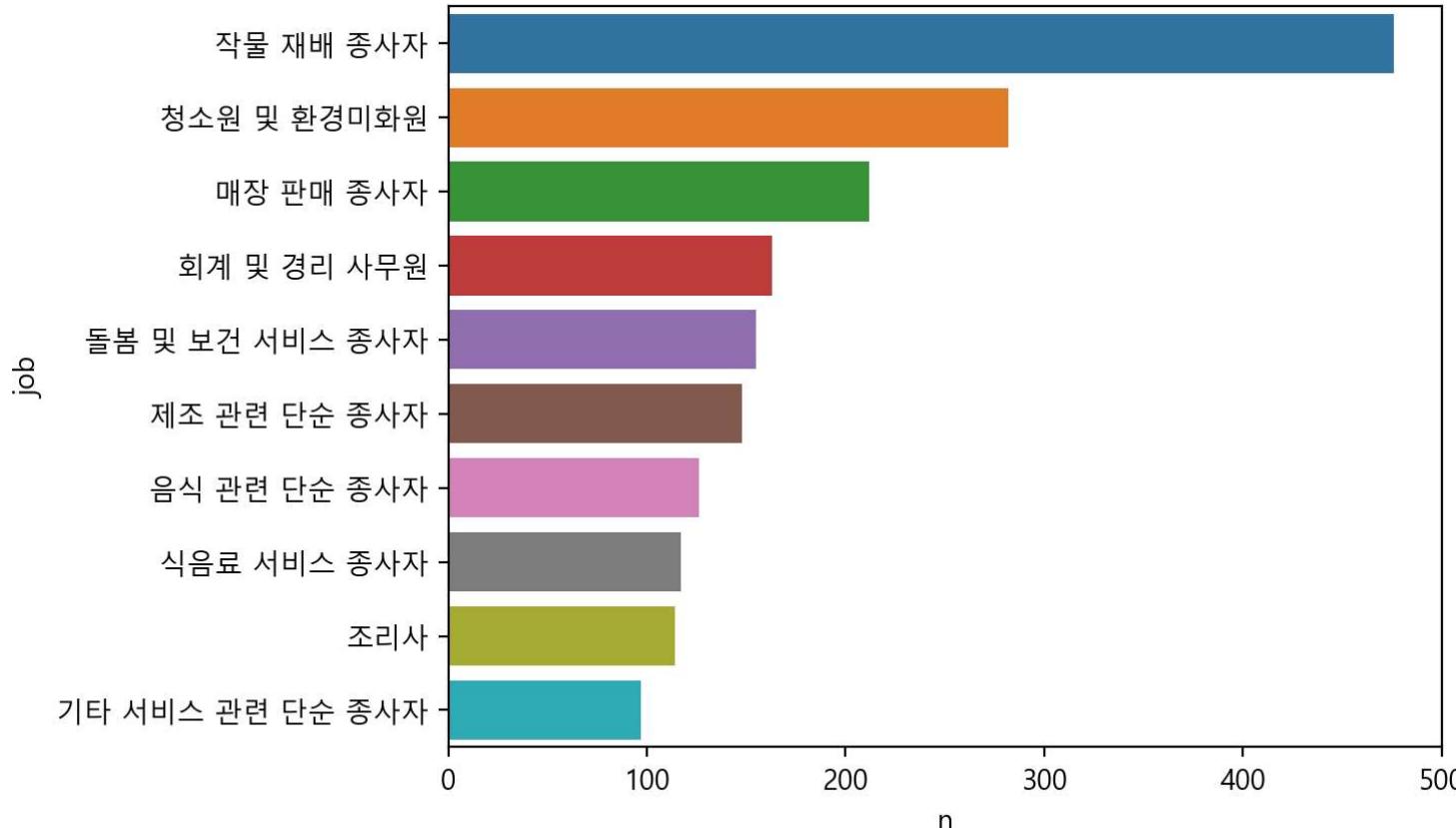
```
[(0.0, 500.0)]
```



```
In [82]: # 여성 직업 빈도 막대 그래프 만들기
sns.barplot(data = job_female, y = 'job', x = 'n').set(xlim = [0, 500])
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

```
Out[82]: [(0.0, 500.0)]
```



In []:

09-8 종교 유무에 따른 이혼율 - 종교가 있으면 이혼을 덜 할까?

종교 변수 검토 및 전처리하기

1. 변수 검토하기

```
In [83]: ## 변수 타입 확인  
welfare['religion'].dtypes # 변수 타입 출력
```

```
Out[83]: dtype('float64')
```

```
In [84]: ## 빈도 확인  
welfare['religion'].value_counts() # 빈도 구하기
```

```
Out[84]: religion  
2.0    7815  
1.0    6603  
Name: count, dtype: int64
```

2. 전처리하기

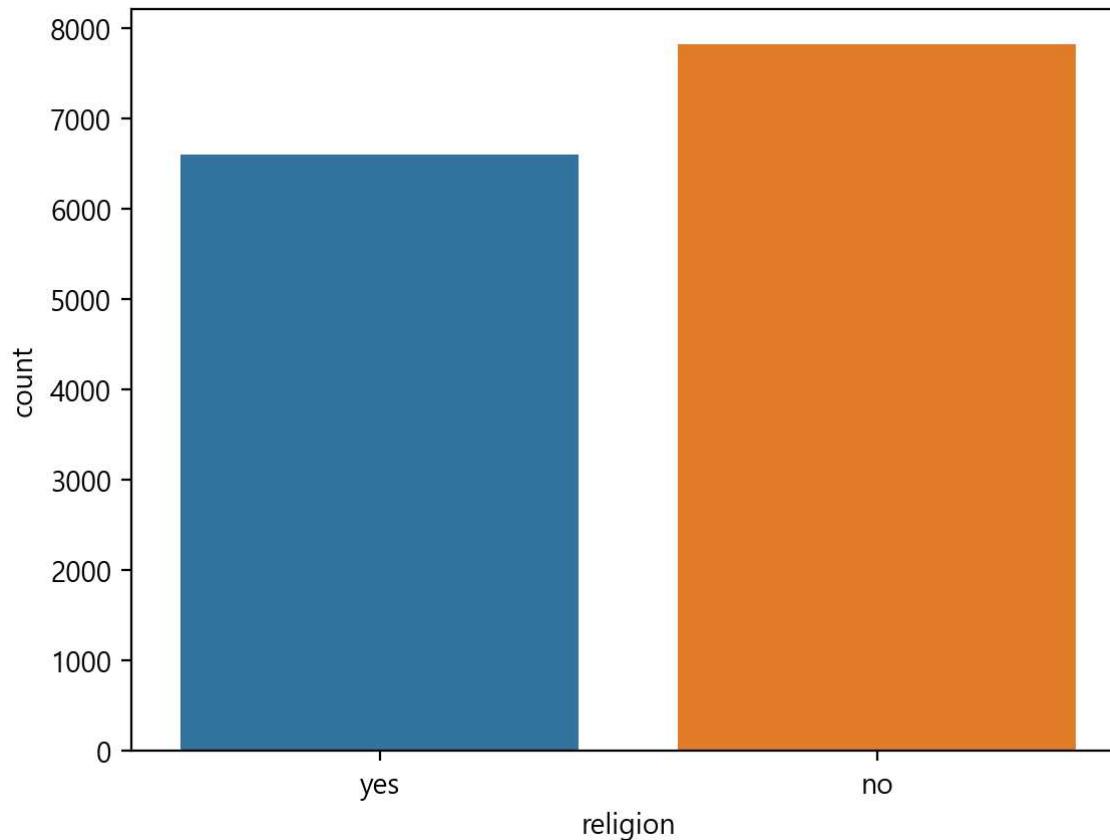
```
In [85]: ## 종교 유무 이름 부여  
welfare['religion'] = np.where(welfare['religion'] == 1, 'yes', 'no')  
  
# 빈도 구하기  
welfare['religion'].value_counts()
```

```
Out[85]: religion  
no    7815  
yes   6603  
Name: count, dtype: int64
```

```
In [86]: # 막대 그래프 만들기  
sns.countplot(data = welfare, x = 'religion')
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead  
... if pd.api.types.is_categorical_dtype(vector):  
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead  
... if pd.api.types.is_categorical_dtype(vector):  
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead  
... if pd.api.types.is_categorical_dtype(vector):
```

```
Out[86]: <Axes: xlabel='religion', ylabel='count'>
```



혼인 상태 변수 검토 및 전처리하기

1. 변수 검토하기

```
In [87]: ## 변수 타입 확인  
welfare['marriage_type'].dtypes # 변수 타입 출력
```

```
Out[87]: dtype('float64')
```

```
In [88]: ## 빈도 확인  
welfare['marriage_type'].value_counts() # 빈도 구하기
```

```
Out[88]: marriage_type
1.0    7190
5.0    2357
0.0    2121
2.0    1954
3.0     689
4.0      78
6.0      29
Name: count, dtype: int64
```

2. 파생변수 만들기 - 이혼 여부

```
In [89]: ## 이혼 여부 변수 만들기
welfare['marriage'] = np.where(welfare['marriage_type'] == 1, 'marriage',
                                np.where(welfare['marriage_type'] == 3, 'divorce', 'etc'))
```

```
In [90]: ## 이혼 여부별 빈도

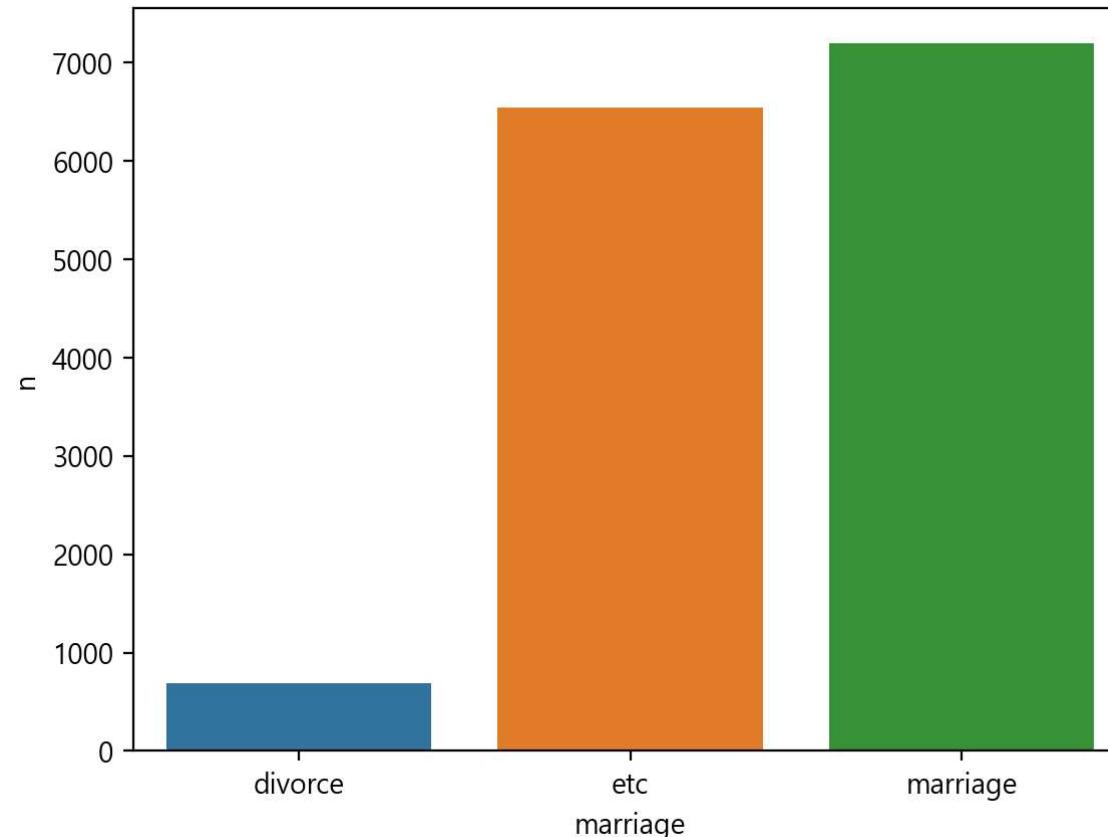
# marriage별 분리
# marriage별 빈도 구하기
n_divorce = welfare.groupby('marriage', as_index = False) # .agg(n = ('marriage', 'count'))
n_divorce
```

```
Out[90]:   marriage    n
0      divorce  689
1        etc  6539
2    marriage  7190
```

```
In [91]: # 막대 그래프 만들기
sns.barplot(data = n_divorce, x = 'marriage', y = 'n')
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

Out[91]: <Axes: xlabel='marriage', ylabel='n'>



종교 유무에 따른 이혼율 분석하기

1. 종교 유무에 따른 이혼율표 만들기

```
In [92]: # etc 제외  
# religion별 분리  
# marriage 추출  
# 비율 구하기  
rel_div = welfare.query('marriage != "etc"')  
    .groupby('religion', as_index = False)  
    ['marriage']  
    .value_counts(normalize = True) #normalize로 비율(proportion)을 구함  
rel_div
```

```
Out[92]:   religion  marriage  proportion  
0         no     marriage  0.905045  
1         no      divorce  0.094955  
2        yes     marriage  0.920469  
3        yes      divorce  0.079531
```

2. 그래프 만들기

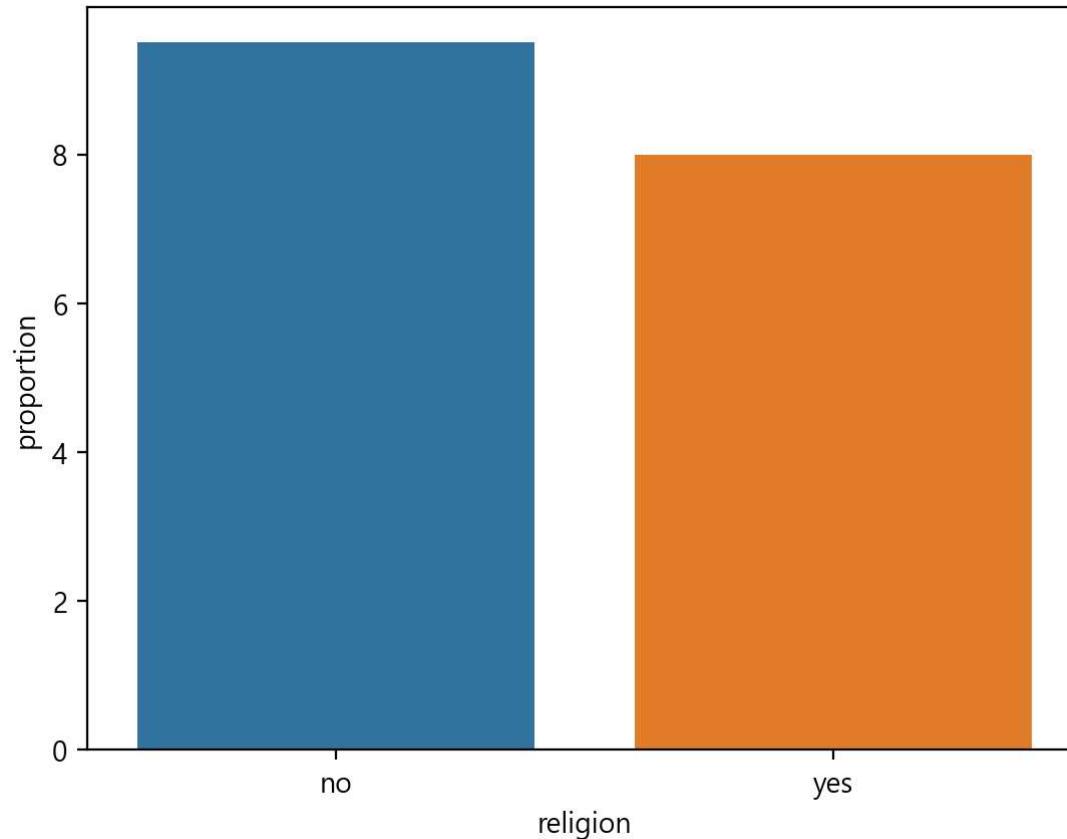
```
In [93]: # divorce 추출  
# 백분율로 바꾸기  
# 반올림  
rel_div = rel_div.query('marriage == "divorce"')  
    .assign(proportion = rel_div['proportion'] * 100)  
    .round(1) #소수점 이하 1자리까지 반올림  
rel_div
```

```
Out[93]:   religion  marriage  proportion  
1         no      divorce    9.5  
3        yes      divorce    8.0
```

```
In [94]: # 막대 그래프 만들기  
sns.barplot(data = rel_div, x = 'religion', y = 'proportion')
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

```
Out[94]: <Axes: xlabel='religion', ylabel='proportion'>
```



연령대 및 종교 유무에 따른 이혼율 분석하기

1. 연령대별 이혼율표 만들기

```
In [95]: # etc 제외  
# ageg별 분리  
# marriage 추출  
# 비율 구하기  
age_div = welfare.query('marriage != "etc"')  
    .groupby('ageg', as_index = False)  
    ['marriage']  
    .value_counts(normalize = True)  
age_div
```

```
Out[95]:   ageg  marriage  proportion  
0   middle   marriage    0.910302  
1   middle   divorce     0.089698  
2     old   marriage    0.914220  
3     old   divorce     0.085780  
4   young   marriage    0.950000  
5   young   divorce     0.050000
```

```
In [96]: ## 연령대 및 이혼 여부별 빈도  
  
# etc 제외  
# ageg별 분리  
# marriage 추출  
# 빈도 구하기  
welfare.query('marriage != "etc"')  
    .groupby('ageg', as_index = False)  
    ['marriage']  
    .value_counts()
```

```
Out[96]:
```

	ageg	marriage	count
0	middle	marriage	3552
1	middle	divorce	350
2	old	marriage	3581
3	old	divorce	336
4	young	marriage	57
5	young	divorce	3

2. 연령대별 이혼율 그래프 만들기

```
In [97]:
```

```
# 초년층 제외, 이혼 추출
# 백분율로 바꾸기
# 반올림
age_div = age_div.query('ageg != "young" & marriage == "divorce"') \
    .assign(proportion = age_div['proportion'] * 100) \
    .round(1)
age_div
```

```
Out[97]:
```

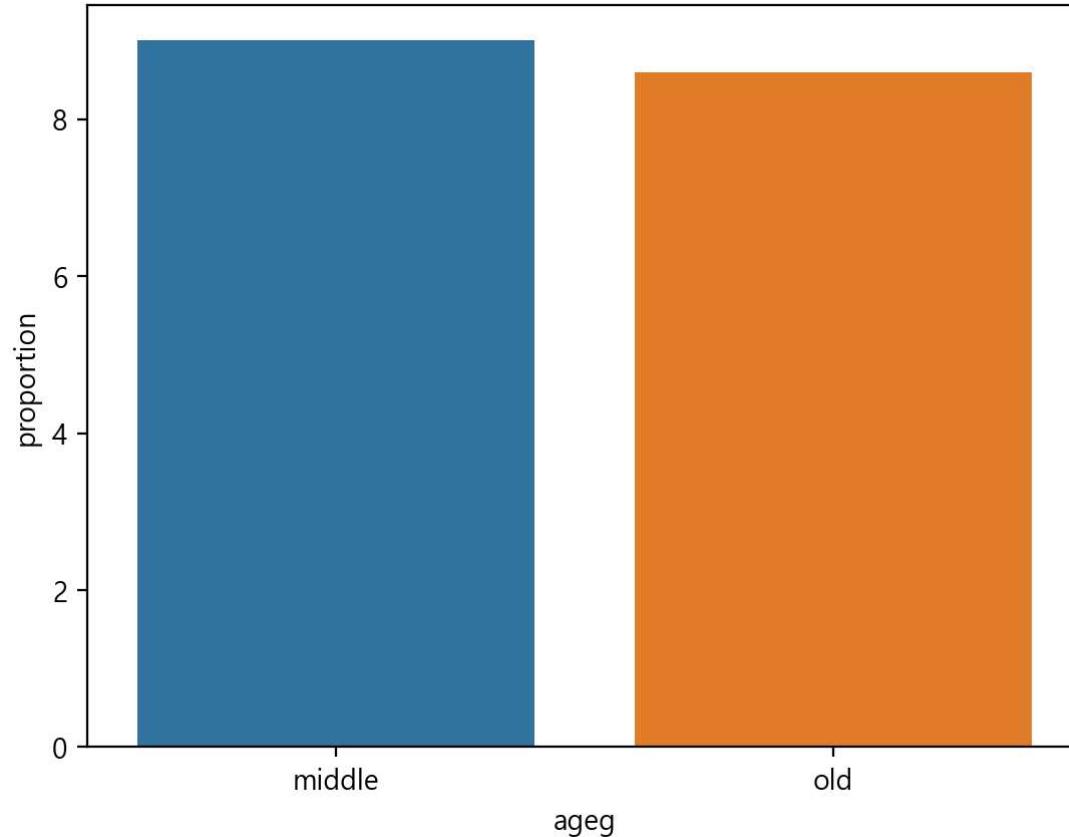
	ageg	marriage	proportion
1	middle	divorce	9.0
3	old	divorce	8.6

```
In [98]:
```

```
# 막대 그래프 만들기
sns.barplot(data = age_div, x = 'ageg', y = 'proportion')
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
.. if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
.. if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_\_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
.. if pd.api.types.is_categorical_dtype(vector):
```

Out[98]: <Axes: xlabel='ageg', ylabel='proportion'>



3. 연령대 및 종교 유무에 따른 이혼율표 만들기

```
In [99]: # etc 제외, 초년층 제외  
# ageg, religion별 분리  
# marriage 추출  
# 비율 구하기  
age_rel_div = welfare.query('marriage != "etc" & ageg != "young"')  
    .groupby(['ageg', 'religion'], as_index = False)  
    ['marriage']  
    .value_counts(normalize = True)
```

age_rel_div

Out[99]:

	ageg	religion	marriage	proportion
0	middle	no	marriage	0.904953
1	middle	no	divorce	0.095047
2	middle	yes	marriage	0.917520
3	middle	yes	divorce	0.082480
4	old	no	marriage	0.904382
5	old	no	divorce	0.095618
6	old	yes	marriage	0.922222
7	old	yes	divorce	0.077778

4. 연령대 및 종교 유무에 따른 이혼율 그래프 만들기

In [100...]

```
# divorce 추출
# 백분율로 바꾸기
# 반올림
age_rel_div = age_rel_div.query('marriage == "divorce"') \
    .assign(proportion = age_rel_div['proportion'] * 100) \
    .round(1)

age_rel_div
```

Out[100]:

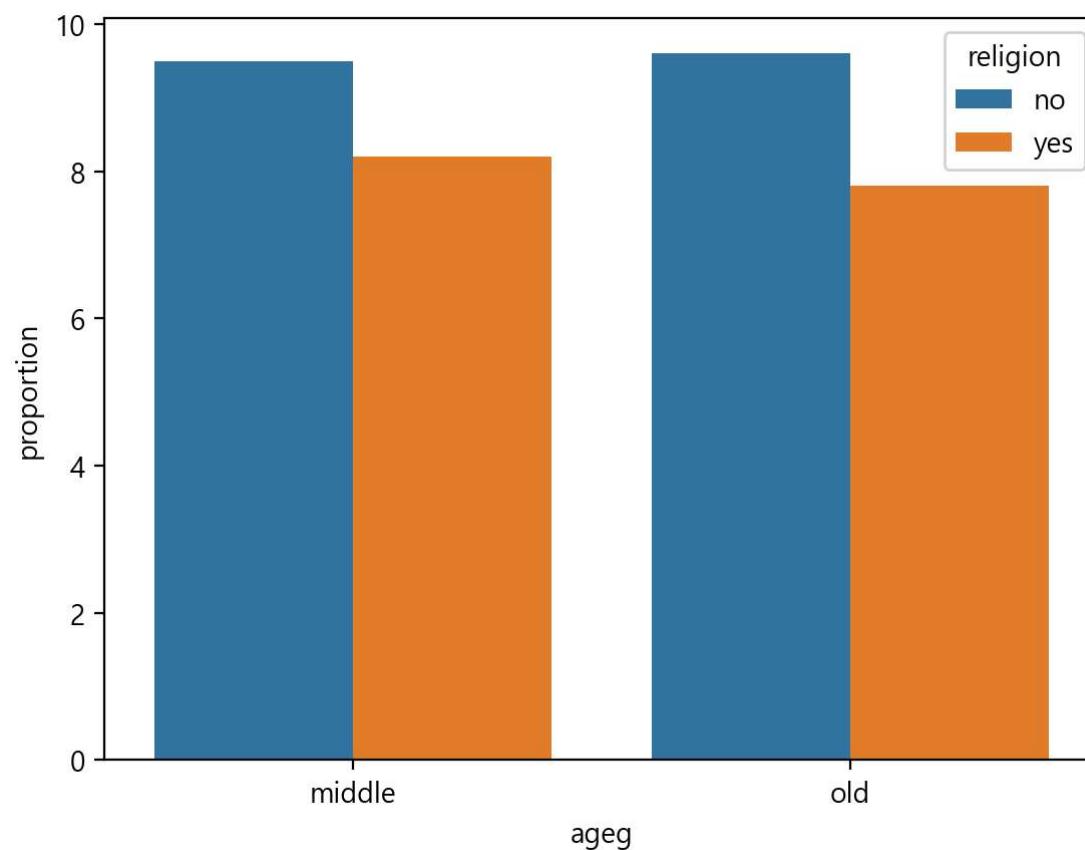
	ageg	religion	marriage	proportion
1	middle	no	divorce	9.5
3	middle	yes	divorce	8.2
5	old	no	divorce	9.6
7	old	yes	divorce	7.8

In [101...]

```
# 막대 그래프 만들기
sns.barplot(data = age_rel_div, x = 'ageg', y = 'proportion', hue = 'religion')
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
.. if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
.. if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
.. if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
.. if pd.api.types.is_categorical_dtype(vector):
```

Out[101]: <Axes: xlabel='ageg', ylabel='proportion'>



[선택적 실행] 여기에서부터 시작하고자 할 때 실행

In [132]:

```
# 데이터 불러오기
welfare = pd.read_spss('Koweps_hpwc14_2019_beta2.sav')

## 변수명 바꾸기
welfare = welfare.rename(columns = {'h14_g3' : 'sex', # 성별
                                    'h14_g4' : 'birth', # 태어난 연도
                                    'h14_g10' : 'marriage_type', # 혼인 상태
                                    'h14_g11' : 'religion', # 종교
                                    'p1402_8aq1' : 'income', # 월급
                                    'h14_ec09' : 'code_job', # 직업 코드
                                    'h14_reg7' : 'code_region'}) # 지역 코드

# 성별 항목 이름 부여
welfare['sex'] = np.where(welfare['sex'] == 1, 'male', 'female')

## 나이 변수 'age' 만들기
welfare = welfare.assign(age = 2019 - welfare['birth'] + 1) # 나이 변수 만들기

## 연령대 변수 'ageg' 만들기
welfare = welfare.assign(ageg = np.where(welfare['age'] < 30, 'young',
                                         np.where(welfare['age'] <= 59, 'middle', 'old')))

## '직종코드' 시트를 읽어들여서 데이터 프레임 구성
list_job = pd.read_excel('Koweps_Codebook_2019.xlsx', sheet_name = '직종코드')

## welfare에 list_job 결합하기
welfare = welfare.merge(list_job, how = 'left', on = 'code_job')

## 종교 유무 이름 부여
welfare['religion'] = np.where(welfare['religion'] == 1, 'yes', 'no')

## 이혼 여부 변수 만들기
welfare['marriage'] = np.where(welfare['marriage_type'] == 1, 'marriage',
                               np.where(welfare['marriage_type'] == 3, 'divorce', 'etc'))
```

In []:

09-9 지역별 연령대 비율 - 어느 지역에 노년층이 많을까?

지역 변수 검토 및 전처리하기

1. 변수 검토하기

```
In [136...]: ## 변수 타입 확인  
welfare['code_region'].dtypes # 변수 타입 출력  
  
Out[136]: dtype('float64')
```

```
In [137]: ## 빈도 확인  
welfare['code_region'].value_counts() # 빈도 구하기
```

```
Out[137]: code_region  
2.0    3246  
7.0    2466  
3.0    2448  
1.0    2002  
4.0    1728  
5.0    1391  
6.0    1137  
Name: count, dtype: int64
```

2. 전처리하기

Out[138]:

	code_region	region
0	1	서울
1	2	수도권(인천/경기)
2	3	부산/경남/울산
3	4	대구/경북
4	5	대전/충남
5	6	강원/충북
6	7	광주/전남/전북/제주도

In [139...]:

```
## 결측치 확인
welfare['code_region'].isna().sum()
```

Out[139]:

```
0
```

In [140...]:

```
## 병합으로 지역명 변수 추가
welfare = welfare.merge(list_region, how = 'left', on = 'code_region')
welfare[['code_region', 'region']].head()
```

Out[140]:

	code_region	region
0	1.0	서울
1	1.0	서울
2	1.0	서울
3	1.0	서울
4	1.0	서울

!!! 아래 셀은 위에서 생성한 컬럼을 삭제할 때만 사용 !!!

In [77]:

```
## 열 제거 [필요 시 실행]
welfare = welfare.drop(['region_x', 'region_y'], axis = 'columns')
welfare.head()
```

```
-----  
KeyError Traceback (most recent call last)  
~\AppData\Local\Temp\ipykernel_10092\3617149125.py in <module>  
    1 ## 열 제거 [필요 시 실행]  
----> 2 welfare = welfare.drop(['region_x', 'region_y'], axis = 'columns')  
    3 welfare.head()  
  
~\Anaconda3\lib\site-packages\pandas\core\frame.py in drop(self, labels, axis, index, columns, level, inplace, errors)  
    5345 ... weight=1.0, ..., 0.8  
    5346 ...  
-> 5347 ...     return super().drop(  
    5348 ...         labels=labels,  
    5349 ...         axis=axis,  
  
~\Anaconda3\lib\site-packages\pandas\core\generic.py in drop(self, labels, axis, index, columns, level, inplace, errors)  
    4709 ...     for axis, labels in axes.items():  
    4710 ...         if labels is not None:  
-> 4711 ...             obj = obj._drop_axis(labels, axis, level=level, errors=errors)  
    4712 ...  
    4713 ...         if inplace:  
  
~\Anaconda3\lib\site-packages\pandas\core\generic.py in _drop_axis(self, labels, axis, level, errors, only_slice)  
    4751 ...     new_axis = axis.drop(labels, level=level, errors=errors)  
    4752 ...     else:  
-> 4753 ...         new_axis = axis.drop(labels, errors=errors)  
    4754 ...     indexer = axis.get_indexer(new_axis)  
    4755 ...  
  
~\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in drop(self, labels, errors)  
    6990 ...     if mask.any():  
    6991 ...         if errors != "ignore":  
-> 6992 ...             raise KeyError(f"{labels[mask].tolist()} not found in axis")  
    6993 ...     indexer = indexer[~mask]  
    6994 ...     return self.delete(indexer)  
  
KeyError: "['region_x', 'region_y'] not found in axis"
```

In []:

지역별 연령대 비율 분석하기

1. 지역별 연령대 비율표 만들기

In [141...]

```
# region별 분리  
# ageg 추출  
# 비율 구하기  
region_ageg = welfare.groupby('region', as_index = False) [  
    'ageg'] [  
    .value_counts(normalize = True)  
region_ageg
```

Out[141]:

	region	ageg	proportion
0	강원/충북	old	0.459103
1	강원/충북	middle	0.308707
2	강원/충북	young	0.232190
3	광주/전남/전북/제주도	old	0.449311
4	광주/전남/전북/제주도	middle	0.317924
5	광주/전남/전북/제주도	young	0.232766
6	대구/경북	old	0.504051
7	대구/경북	middle	0.296296
8	대구/경북	young	0.199653
9	대전/충남	old	0.413372
10	대전/충남	middle	0.336449
11	대전/충남	young	0.250180
12	부산/경남/울산	old	0.437500
13	부산/경남/울산	middle	0.333742
14	부산/경남/울산	young	0.228758
15	서울	middle	0.385115
16	서울	old	0.376124
17	서울	young	0.238761
18	수도권(인천/경기)	middle	0.388170
19	수도권(인천/경기)	old	0.325015
20	수도권(인천/경기)	young	0.286815

2. 그래프 만들기

In [142...]

```
# 백분율로 바꾸기  
# 반올림  
region_ageg = region_ageg.assign(proportion = region_ageg['proportion'] * 100)  
            .round(1)  
region_ageg
```

Out[142] :

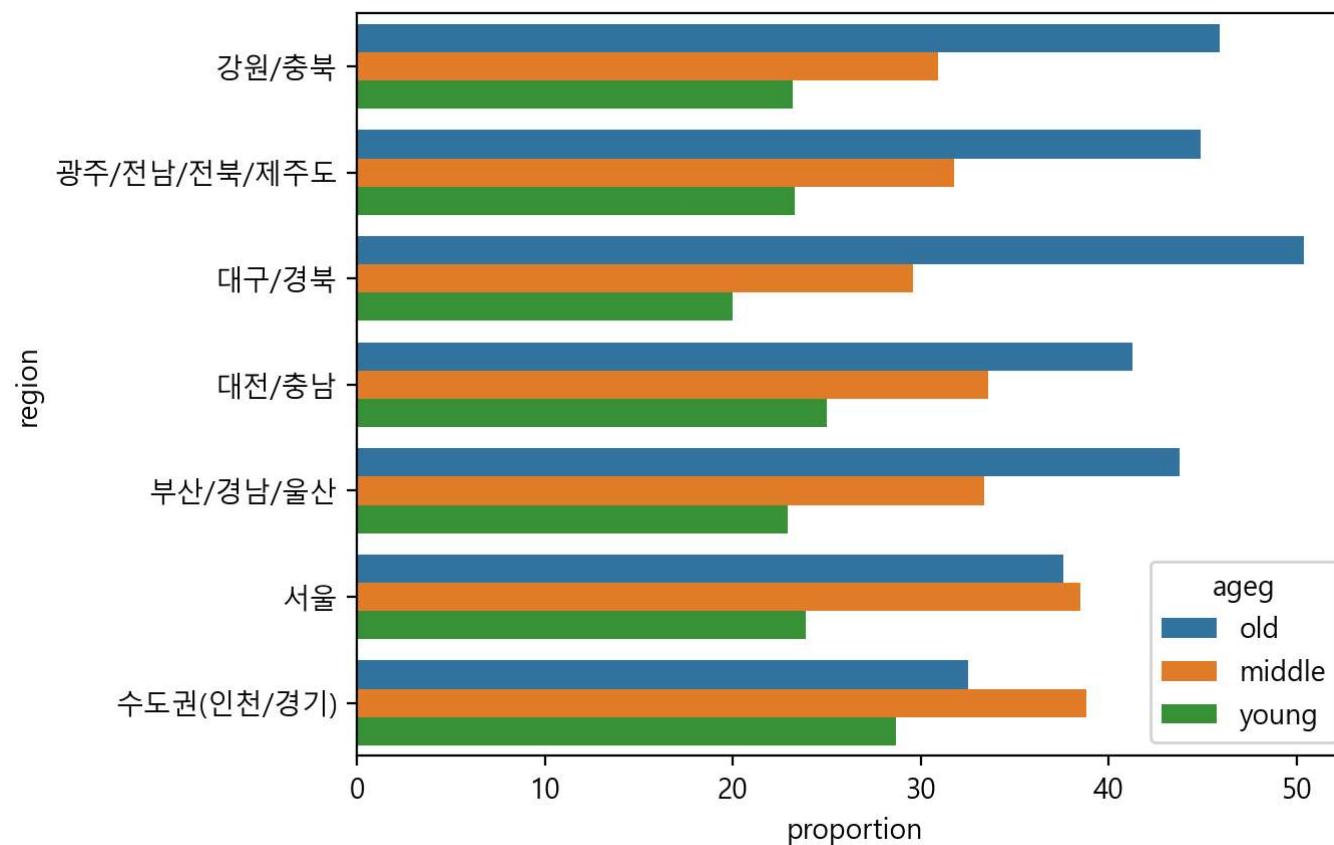
	region	ageg	proportion
0	강원/충북	old	45.9
1	강원/충북	middle	30.9
2	강원/충북	young	23.2
3	광주/전남/전북/제주도	old	44.9
4	광주/전남/전북/제주도	middle	31.8
5	광주/전남/전북/제주도	young	23.3
6	대구/경북	old	50.4
7	대구/경북	middle	29.6
8	대구/경북	young	20.0
9	대전/충남	old	41.3
10	대전/충남	middle	33.6
11	대전/충남	young	25.0
12	부산/경남/울산	old	43.8
13	부산/경남/울산	middle	33.4
14	부산/경남/울산	young	22.9
15	서울	middle	38.5
16	서울	old	37.6
17	서울	young	23.9
18	수도권(인천/경기)	middle	38.8
19	수도권(인천/경기)	old	32.5
20	수도권(인천/경기)	young	28.7

In [143...]

```
# 막대 그래프 만들기  
sns.barplot(data = region_ageg, y = 'region', x = 'proportion', hue = 'ageg')
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

Out[143]: <Axes: xlabel='proportion', ylabel='region'>



3. 누적 비율 막대 그래프 만들기

(1) 피벗하기

In [144...]

```
# 피벗
pivot_df = region_ageg[['region', 'ageg', 'proportion']].pivot(index = 'region',
                                                               columns = 'ageg',
                                                               values = 'proportion')
pivot_df
```

Out[144]:

	ageg	middle	old	young
region				
강원/충북	30.9	45.9	23.2	
광주/전남/전북/제주도	31.8	44.9	23.3	
대구/경북	29.6	50.4	20.0	
대전/충남	33.6	41.3	25.0	
부산/경남/울산	33.4	43.8	22.9	
서울	38.5	37.6	23.9	
수도권(인천/경기)	38.8	32.5	28.7	

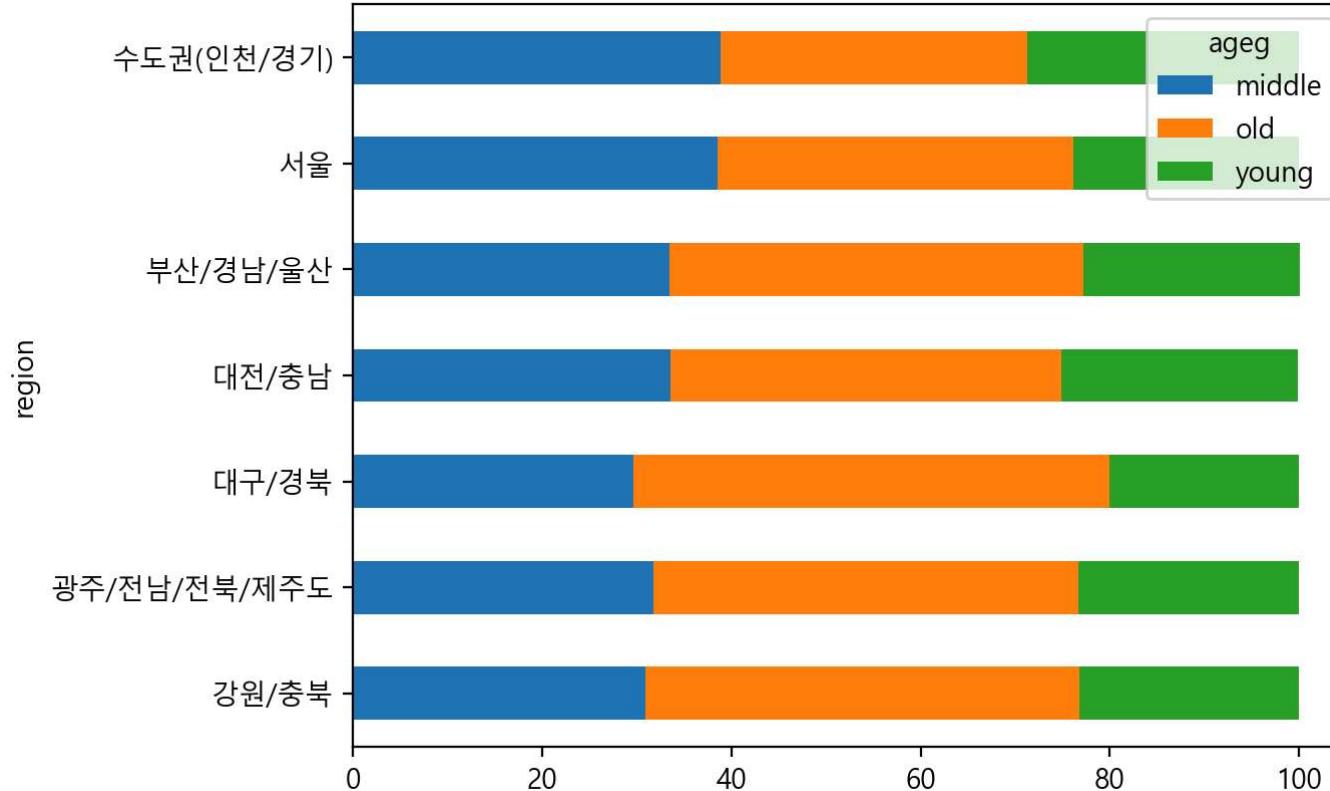
(2) 그래프 만들기

In [145...]

```
# 가로 점유 막대 그래프 만들기
pivot_df.plot.barh(stacked = True)
```

Out[145]:

```
<Axes: ylabel='region'>
```



(3) 막대 정렬하기

```
In [146...]: # 노년층 비율 기준 정렬, 변수 순서 바꾸기  
reorder_df = pivot_df.sort_values('old')[['young', 'middle', 'old']]  
reorder_df
```

Out[146]:

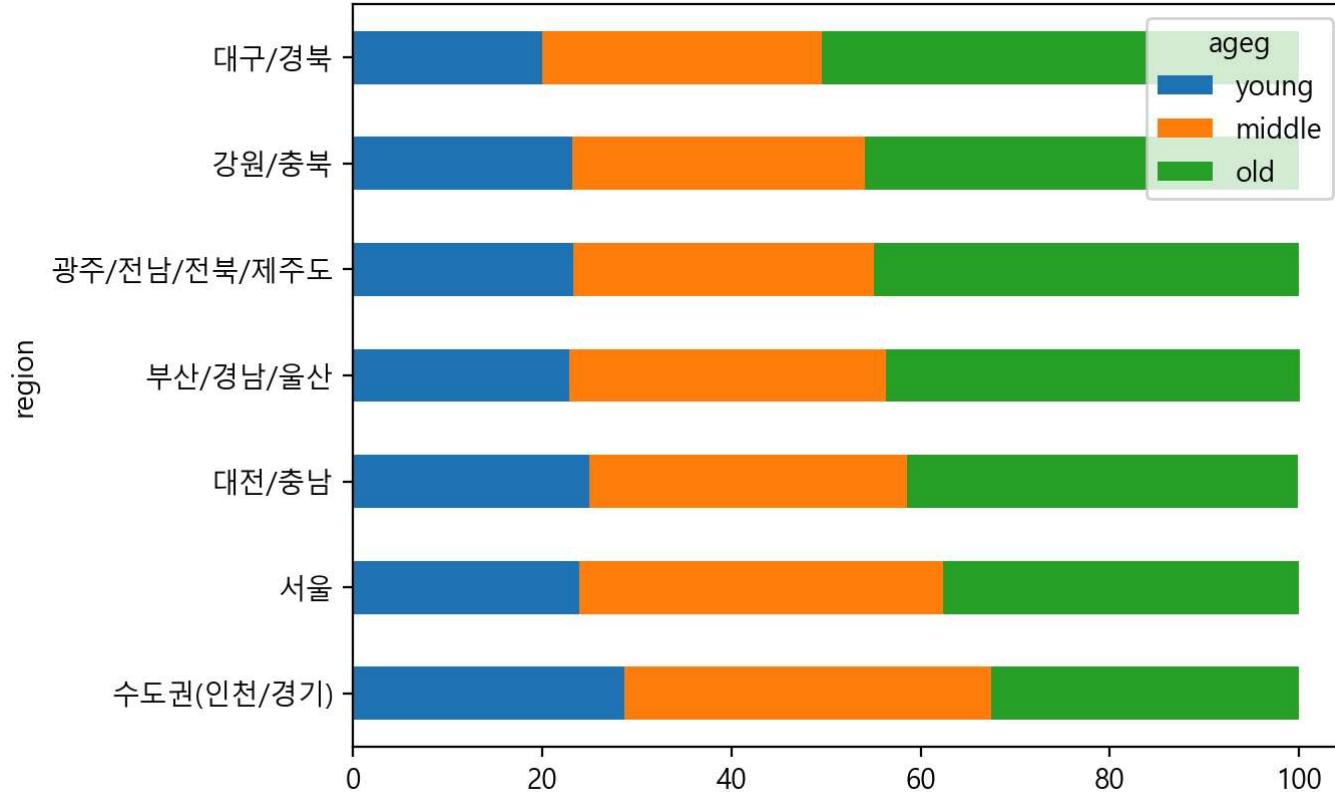
region	ageg	young	middle	old
수도권(인천/경기)	28.7	38.8	32.5	
서울	23.9	38.5	37.6	
대전/충남	25.0	33.6	41.3	
부산/경남/울산	22.9	33.4	43.8	
광주/전남/전북/제주도	23.3	31.8	44.9	
강원/충북	23.2	30.9	45.9	
대구/경북	20.0	29.6	50.4	

In [147...]

```
# 누적 가로 막대 그래프 만들기  
reorder_df.plot.barh(stacked = True)
```

Out[147]:

```
<Axes: ylabel='region'>
```



[실습-5] 연령별 수입의 산점도 그래프 그리기

: 새로 열 추출 생성한 newwel 데이터 프레임 사용

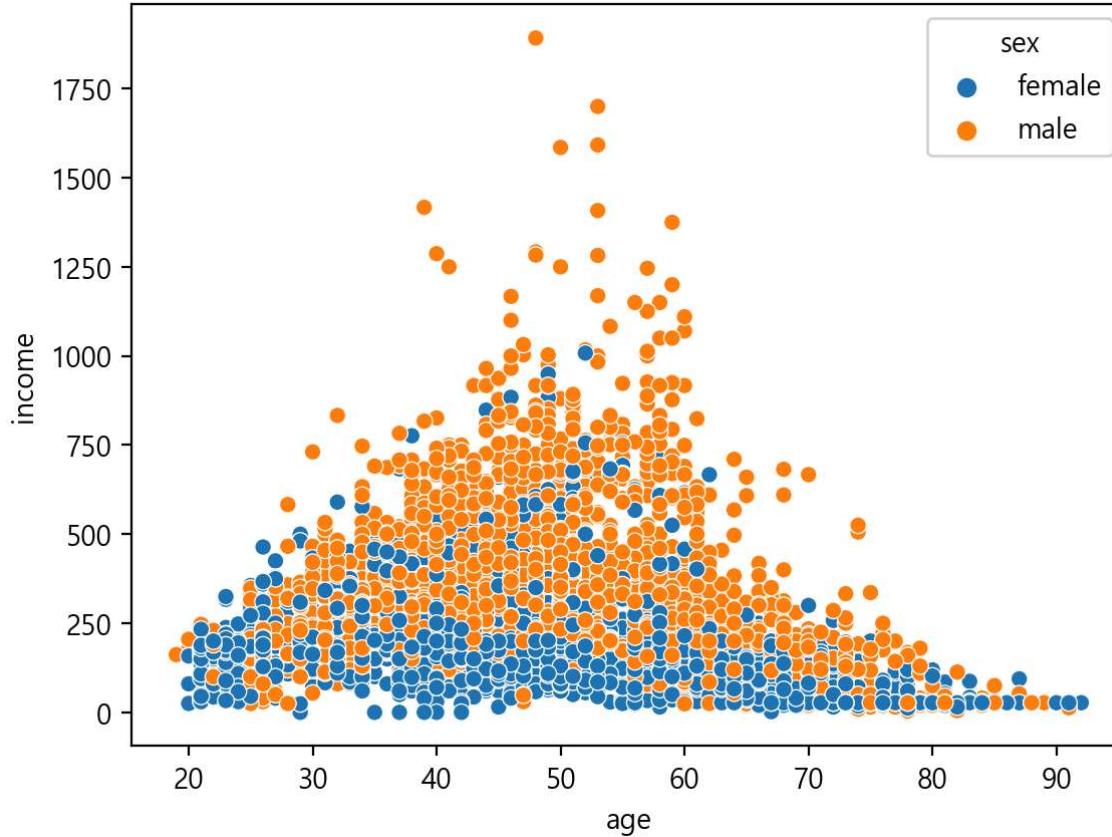
> 성별('sex')로 색조(hue)를 다르게 표현

In [215...]

```
#### 'age' 연령별 income의 산점도 그래프 그리기
#### > 색조(hue)는 'sex'로 구분
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be
removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
```

Out[215]: <Axes: xlabel='age', ylabel='income'>



[실습-6] 지역별 임금 비교 분석하기

: 새로 열 추출 생성한 newwel 데이터 프레임 사용

0. newwel 데이터 프레임 조정

> newwel에 병합으로 지역명 변수 추가

> 연령대 변수 'ageg' 만들기

1. 지역 및 연령대별 월급 평균 데이터 프레임 만들기

> 'income' 결측치 제거, index 생략

> 'reg_ageg_income' 이름으로 데이터 프레임 생성

2. 지역별 월급 평균 데이터 프레임 만들기

> 'income' 결측치 제거, **index** 생략

> 'reg_income' 이름으로 데이터 프레임 생성

3. 두 데이터 프레임을 하나로 병합

> 'reg_ageg_income'를 기준으로 결합

4. 막대 그래프로 표현하기

> 지역별 평균 월급이 큰 지역부터 표현되도록 데이터 프레임 재구성

> 'region'이 y-축이 되도록 그래프로 표현

In [115...]

```
## newwei에 병합으로 지역명 변수 추가  
# 지역 코드 목록 만들기
```

```
# 병합으로 지역명 변수 추가
```

Out[115]:

code_region region

0	1.0	서울
1	1.0	서울
2	1.0	서울
3	1.0	서울
4	1.0	서울

In [117...]

```
## 연령대 변수 'ageg' 만들기
```

Out[117]:

	sex	birth	marriage_type	religion	income	code_job	code_region	age	age10	region	ageg
0	2.0	1945.0	2.0	1.0	NaN	NaN	1.0	75.0	70	서울	old
1	1.0	1948.0	2.0	2.0	NaN	NaN	1.0	72.0	70	서울	old
2	1.0	1942.0	3.0	1.0	107.0	762.0	1.0	78.0	70	서울	old
3	1.0	1962.0	1.0	1.0	192.0	855.0	1.0	58.0	50	서울	middle
4	2.0	1963.0	1.0	1.0	NaN	NaN	1.0	57.0	50	서울	middle
...
14413	2.0	1967.0	1.0	1.0	NaN	NaN	5.0	53.0	50	대전/충남	middle
14414	2.0	1992.0	5.0	1.0	NaN	NaN	5.0	28.0	20	대전/충남	young
14415	1.0	1995.0	5.0	1.0	NaN	910.0	5.0	25.0	20	대전/충남	young
14416	2.0	1998.0	5.0	1.0	200.0	246.0	5.0	22.0	20	대전/충남	young
14417	1.0	2001.0	0.0	1.0	NaN	NaN	5.0	19.0	10	대전/충남	young

14418 rows × 11 columns

In [118...]

```
#### 1. 지역 및 연령대별 월급 평균 데이터 프레임 만들기
##### > 'income' 결측치 제거, index 생략
##### > 'reg_ageg_income' 이름으로 데이터 프레임 생성
```

Out[118]:

	region	ageg	mean_income
0	강원/충북	middle	322.644231
1	강원/충북	old	115.489796
2	강원/충북	young	197.000000
3	광주/전남/전북/제주도	middle	301.080000
4	광주/전남/전북/제주도	old	99.080925

In [119...]

```
#### 2. 지역별 월급 평균 데이터 프레임 만들기
##### > 'income' 결측치 제거, index 생략
##### > 'reg_income' 이름으로 데이터 프레임 생성
```

Out[119]:

	region	mean_region
0	강원/충북	251.492492
1	광주/전남/전북/제주도	238.434389
2	대구/경북	247.462069
3	대전/충남	299.073892
4	부산/경남/울산	270.336066

In [120...]

```
#### 3. 두 데이터 프레임을 하나로 병합  
##### > 'reg_ageg_income'를 기준으로 결합
```

Out[120]:

	region	ageg	mean_income	mean_region
0	강원/충북	middle	322.644231	251.492492
1	강원/충북	old	115.489796	251.492492
2	강원/충북	young	197.000000	251.492492
3	광주/전남/전북/제주도	middle	301.080000	238.434389
4	광주/전남/전북/제주도	old	99.080925	238.434389

In [121...]

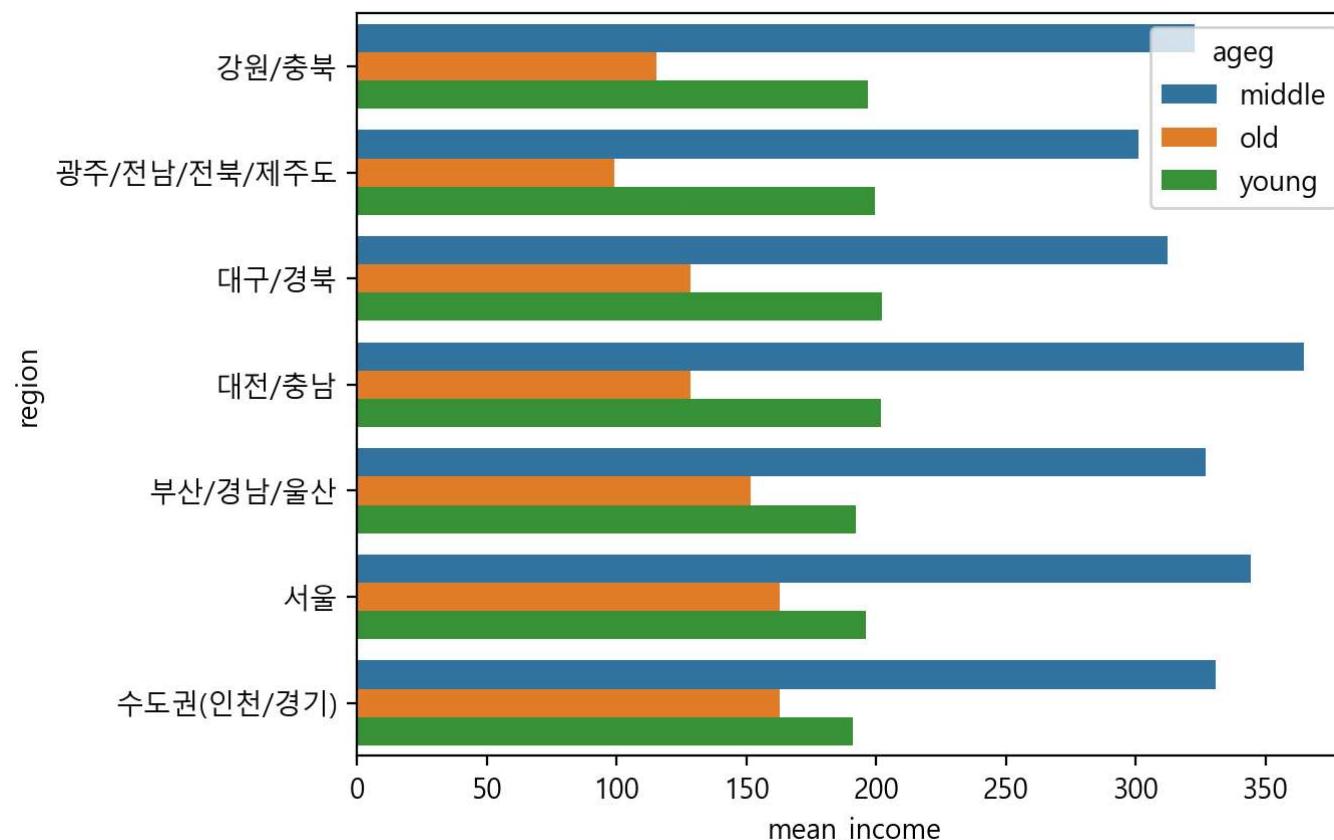
```
## 맑은 고딕 폰트 설정  
import matplotlib.pyplot as plt  
plt.rcParams.update({'font.family' : 'Malgun Gothic'})
```

In [122...]

```
#### 4. 막대 그래프로 표현하기
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

Out[122]: <Axes: xlabel='mean_income', ylabel='region'>



In [123...]

```
#### 4. 막대 그래프로 표현하기
##### > 평균 월급이 큰 지역부터 표현
```

```
# 노년층 비율 기준 정렬, 변수 순서 바꾸기
```

Out[123]:

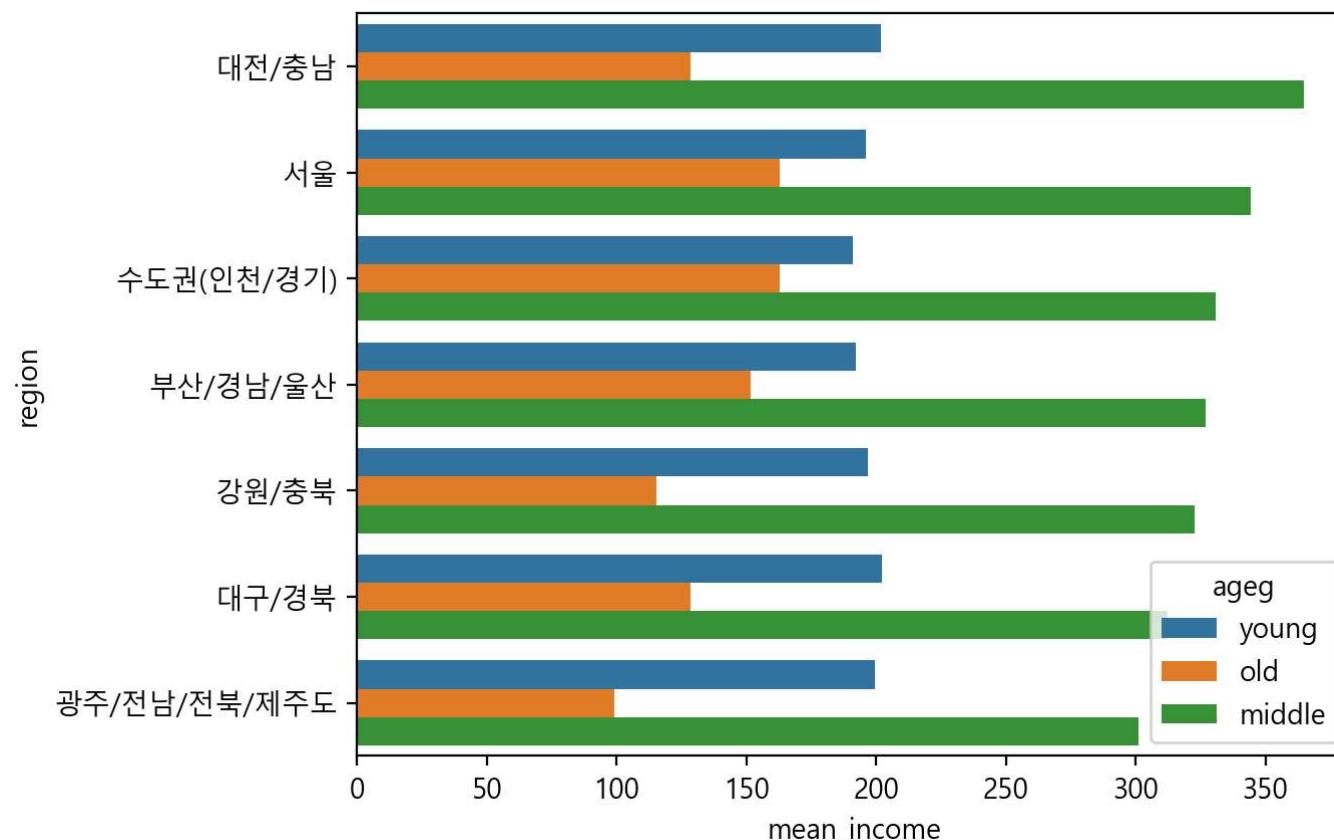
	region	ageg	mean_income	mean_region
11	대전/충남	young	202.066667	299.073892
10	대전/충남	old	128.378049	299.073892
9	대전/충남	middle	364.888889	299.073892
17	서울	young	195.891892	282.818299
16	서울	old	162.755814	282.818299
15	서울	middle	344.277890	282.818299
20	수도권(인천/경기)	young	191.255172	276.638352
19	수도권(인천/경기)	old	162.961977	276.638352
18	수도권(인천/경기)	middle	330.770807	276.638352
14	부산/경남/울산	young	192.269231	270.336066
13	부산/경남/울산	old	151.774011	270.336066
12	부산/경남/울산	middle	327.096436	270.336066
2	강원/충북	young	197.000000	251.492492
1	강원/충북	old	115.489796	251.492492
0	강원/충북	middle	322.644231	251.492492
8	대구/경북	young	202.418605	247.462069
7	대구/경북	old	128.578125	247.462069
6	대구/경북	middle	312.439394	247.462069
5	광주/전남/전북/제주도	young	199.723077	238.434389
4	광주/전남/전북/제주도	old	99.080925	238.434389
3	광주/전남/전북/제주도	middle	301.080000	238.434389

In [124...]

```
#### 4. 막대 그래프로 표현하기  
##### > 'region'이 y-축이 되도록 그래프로 표현# 막대 그래프 만들기
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

Out[124]: <Axes: xlabel='mean_income', ylabel='region'>



In []:

[실습-7] 히트맵 그림 - 상관 관계 정도 표현하기

: 새로 열 추출 생성한 newwel 데이터 프레임 사용

0. 성별(sex) 문자 값을 숫자 값으로 원래대로 변경

> 'male' : 1, 'female' : 2

1. 상관행렬용 데이터 추출

> ['sex', 'marriage_type', 'religion', 'income', 'code_job', 'code_region', 'age'] 만 추출

2. 상관행렬 만들기

3. 히트맵 그리기

```
In [222...]: # 성별 항목 이름 변경
```

```
In [223...]: ##### 1. 상관행렬용 데이터 추출
```

Out[223]:

	sex	marriage_type	religion	income	code_job	code_region	age
0	2	2.0	1.0	NaN	NaN	1.0	75.0
1	1	2.0	2.0	NaN	NaN	1.0	72.0
2	1	3.0	1.0	107.0	762.0	1.0	78.0
3	1	1.0	1.0	192.0	855.0	1.0	58.0
4	2	1.0	1.0	NaN	NaN	1.0	57.0
...
14413	2	1.0	1.0	NaN	NaN	5.0	53.0
14414	2	5.0	1.0	NaN	NaN	5.0	28.0
14415	1	5.0	1.0	NaN	910.0	5.0	25.0
14416	2	5.0	1.0	200.0	246.0	5.0	22.0
14417	1	0.0	1.0	NaN	NaN	5.0	19.0

14418 rows × 7 columns

In [224...]

2. 상관행렬 만들기

Out[224]:

	sex	marriage_type	religion	income	code_job	code_region	age
sex	1.00	-0.00	-0.10	-0.41	-0.08	0.01	0.09
marriage_type	-0.00	1.00	0.04	-0.16	-0.10	-0.03	-0.04
religion	-0.10	0.04	1.00	0.09	-0.04	0.07	-0.18
income	-0.41	-0.16	0.09	1.00	-0.35	-0.06	-0.22
code_job	-0.08	-0.10	-0.04	-0.35	1.00	0.07	0.35
code_region	0.01	-0.03	0.07	-0.06	0.07	1.00	0.07
age	0.09	-0.04	-0.18	-0.22	0.35	0.07	1.00

In [225...]

3. 히트맵 그리기

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\utils.py:80: UserWarning: Glyph 8722 (MINUS SIGN) missing from current font.  
fig.canvas.draw()
```

```
Out[225]: <Axes: >
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\IPython\core\events.py:89: UserWarning: Glyph 8722 (MINUS SIGN) missing from current  
font.
```

```
func(*args, **kwargs)
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\IPython\core\pylabtools.py:151: UserWarning: Glyph 8722 (MINUS SIGN) missing from cu  
rrent font.
```

```
fig.canvas.print_figure(bytes_io, **kw)
```

