
▣ Ch08 그래프 만들기

[08-1] 파이썬으로 만들 수 있는 그래프 살펴보기

- > 막대, 선, 상자, 산점도, 히스토그램, 등고선 그래프 등
- > 워드 크라우드, 지도 그래프, 네트워크 그래프, 모션 차트, 인터랙티브 그래프 등

<> Library를 통해 그릴 수 있는 그래프들

1. **Matplotlib**: Matplotlib은 Python의 가장 인기 있는 그래픽 라이브러리 중 하나로, 선 그래프, 막대 그래프, 히스토그램, 산점도, 등고선 그래프 등 다양한 종류의 그래프를 그릴 수 있습니다.
2. **Seaborn**: Seaborn은 Matplotlib을 기반으로 한 통계적 데이터 시각화 라이브러리로, 예쁘고 간편한 API를 제공하여 복잡한 시각화를 빠르게 생성할 수 있습니다. 주로 히트맵, 박스 플롯, 카운트 플롯 등에 사용됩니다.
3. **Plotly**: Plotly는 상호작용 그래프를 생성하는데 사용되는 라이브러리로 웹 기반 대화식 그래프를 만들 수 있습니다. 특히 대시보드 및 웹 애플리케이션 개발에 유용합니다.
4. **Bokeh**: Bokeh는 대화식 시각화를 위한 강력한 라이브러리로, 웹 기반 대시보드 및 애플리케이션을 만들기에 적합합니다.
5. **Pandas Plotting**: Pandas 자체에서도 데이터프레임에 포함된 데이터를 기반으로 그래프를 그릴 수 있는 기능을 제공합니다. DataFrame.plot() 메서드를 사용하여 다양한 종류의 그래프를 그릴 수 있습니다.
6. **Altair**: Altair는 선언적 문법을 사용하여 인터랙티브한 그래프를 만들기 위한 라이브러리입니다. 데이터를 시각화하기 위한 간단하고 직관적인 접근 방식을 제공합니다.
7. **ggplot**: ggplot은 R 언어에서 영감을 받아 만든 Python 패키지로, Grammar of Graphics 원칙에 따라 그래프를 생성합니다.
8. **Holoviews**: Holoviews는 복잡한 시각화 작업을 간단하게 만들어주는 라이브러리로, Matplotlib, Bokeh, Plotly 등 다양한 백엔드와 함께 사용할 수 있습니다.
9. **NetworkX**: NetworkX는 그래프 이론 및 네트워크 분석을 위한 라이브러리로, 그래프 구조를 시각화하는데 사용됩니다.
10. **D3.js**: D3.js는 JavaScript 기반의 데이터 시각화 라이브러리이지만, Python과 연동하여 사용할 수도 있으며, 웹 기반 대화식 시각화를 만드는데 매우 강력합니다.

In []:

[08-2] 산점도 - 변수 간 관계 표현하기

▣ 산점도 만들기

> 데이터의 두 변수 간의 관계를 시각화하기 위해 사용되는 간단하면서도 매우 유용한 그래프

```
In [1]: ## mpg 데이터 불러오기
import pandas as pd
mpg = pd.read_csv('mpg.csv')
mpg
```

```
Out[1]:
```

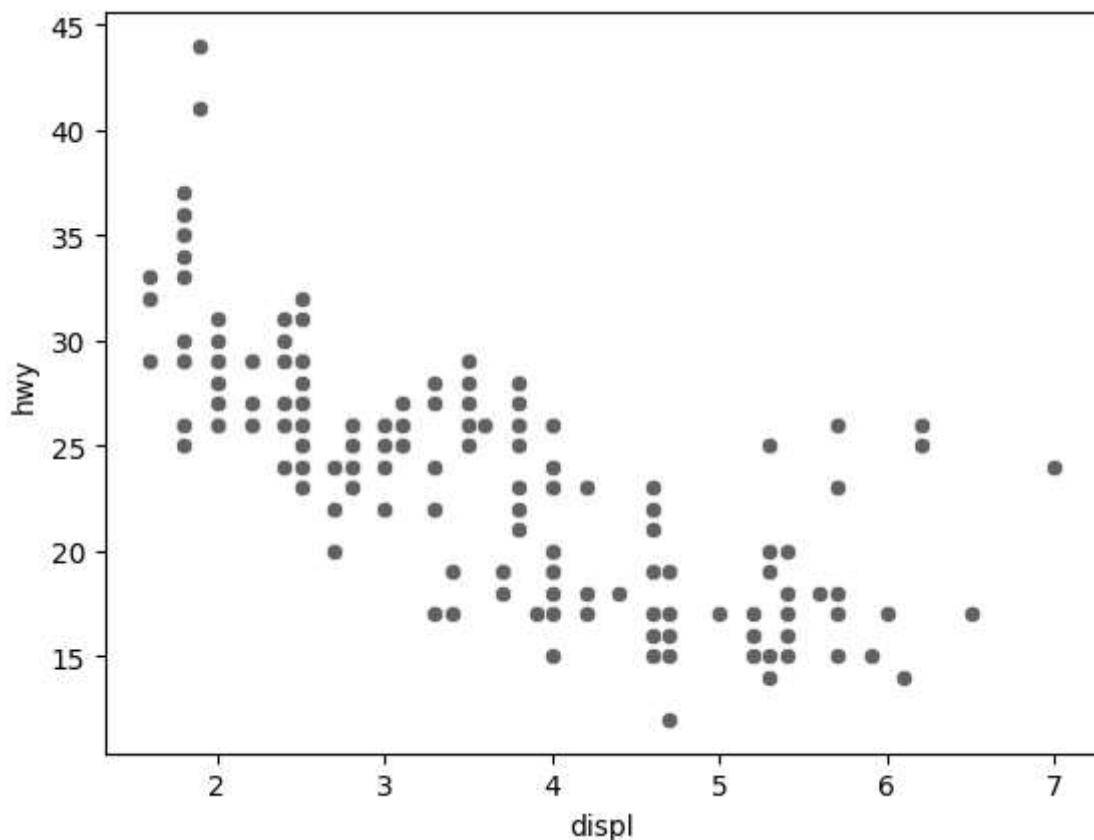
	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	category
0	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
1	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
2	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
3	audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
4	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
...
229	volkswagen	passat	2.0	2008	4	auto(s6)	f	19	28	p	midsize
230	volkswagen	passat	2.0	2008	4	manual(m6)	f	21	29	p	midsize
231	volkswagen	passat	2.8	1999	6	auto(l5)	f	16	26	p	midsize
232	volkswagen	passat	2.8	1999	6	manual(m5)	f	18	26	p	midsize
233	volkswagen	passat	3.6	2008	6	auto(s6)	f	17	26	p	midsize

234 rows × 11 columns

```
In [2]: ## scatterplot()로 산점도 그리기
# x축은 displ, y축은 hwy를 나타낸 산점도 만들기
import seaborn as sns
sns.scatterplot(data = mpg, x = 'displ', y = 'hwy')
```

```
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}""
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='displ', ylabel='hwy'>
```

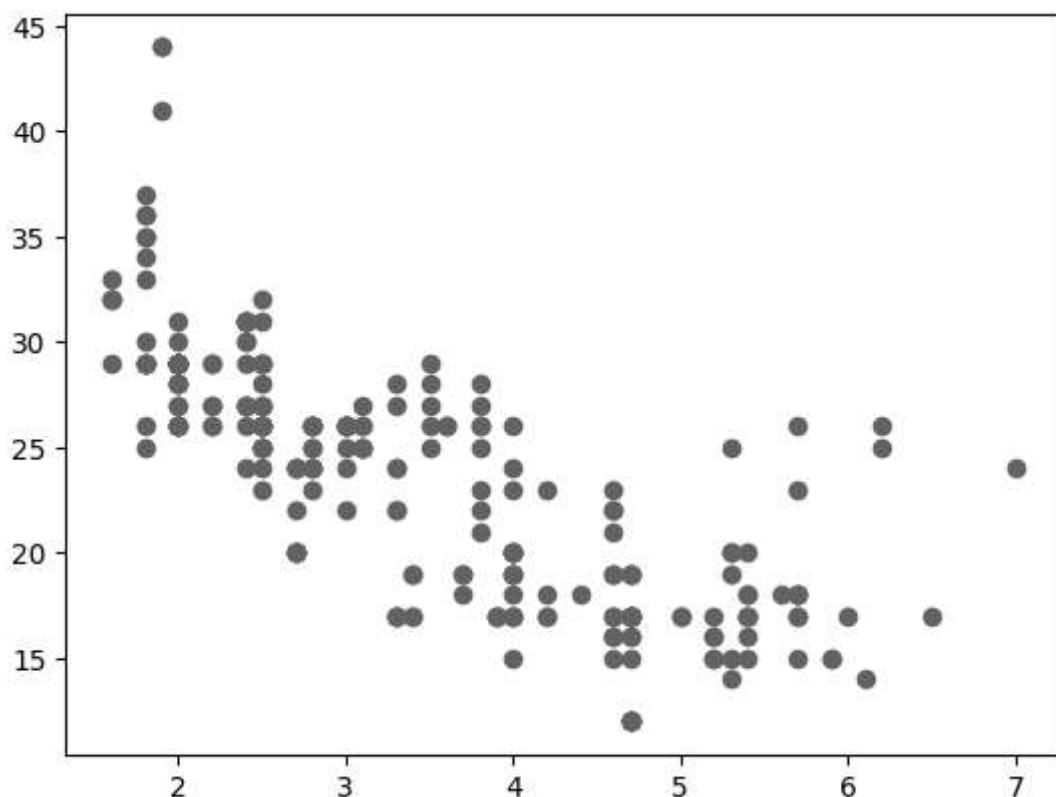
```
Out[2]:
```



> matplotlib 라이브러리로 그리기

```
In [12]: ## matplotlib 라이브러리로 그리기
import matplotlib.pyplot as plt

plt.scatter(mpg['displ'], mpg['hwy'])
plt.show()
```



'displ' 열 값 분포 확인

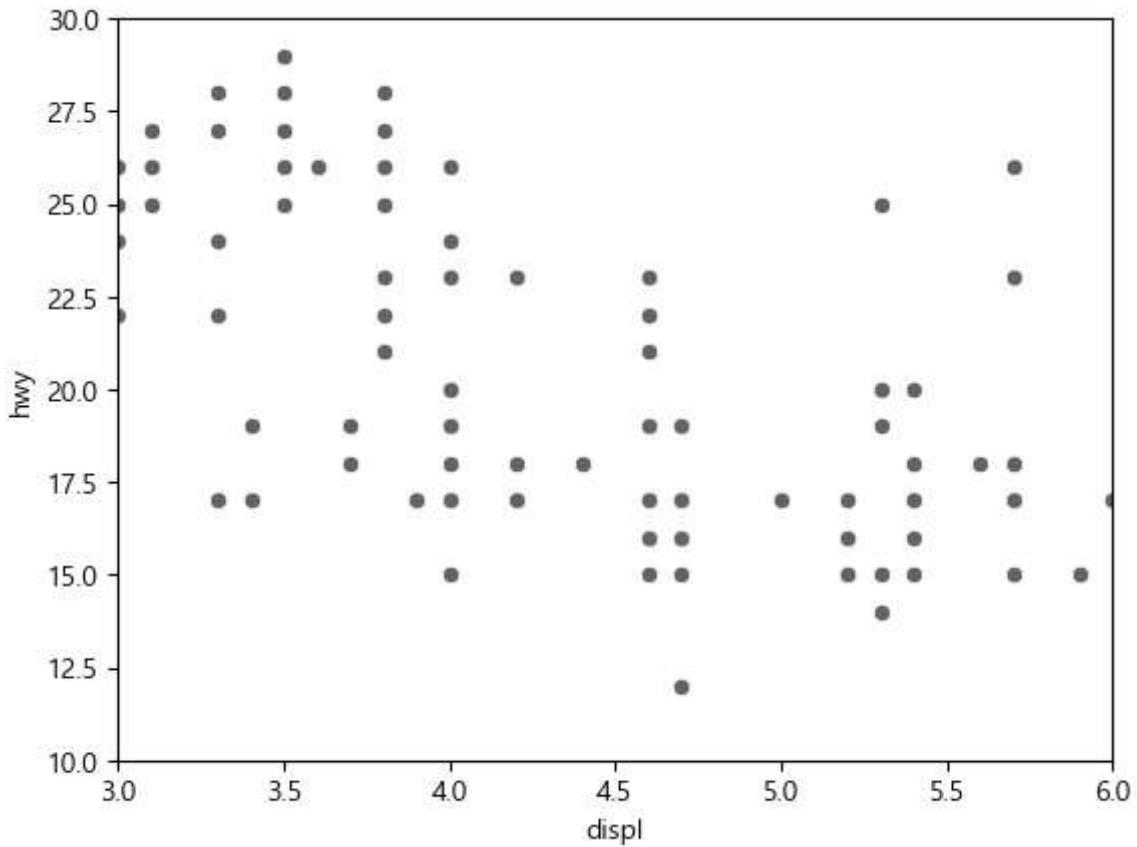
```
In [76]: ## 'displ' 열 값 분포 확인  
mpg['displ'].sort_values().value_counts(sort=False) #'displ' 열 값을 추출해서 정렬  
  
Out[76]: displ  
1.6 .... 5  
1.8 .... 14  
1.9 .... 3  
2.0 .... 21  
2.2 .... 6  
2.4 .... 13  
2.5 .... 20  
2.7 .... 8  
2.8 .... 10  
3.0 .... 8  
3.1 .... 6  
3.3 .... 9  
3.4 .... 4  
3.5 .... 5  
3.6 .... 2  
3.7 .... 3  
3.8 .... 8  
3.9 .... 3  
4.0 .... 15  
4.2 .... 4  
4.4 .... 1  
4.6 .... 11  
4.7 .... 17  
5.0 .... 2  
5.2 .... 5  
5.3 .... 6  
5.4 .... 8  
5.6 .... 1  
5.7 .... 8  
5.9 .... 2  
6.0 .... 1  
6.1 .... 1  
6.2 .... 2  
6.5 .... 1  
7.0 .... 1  
Name: count, dtype: int64
```

[] 축 범위 설정하기

```
In [77]: ## x축 범위 3~6으로 제한, y축 범위 10~30으로 제한  
sns.scatterplot(data = mpg, x = 'displ', y = 'hwy')  
    .set(xlim = [3, 6], ylim = [10, 30])
```

```
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 isin  
stance(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 isin  
stance(dtype, CategoricalDtype) instead  
    if pd.api.types.is_categorical_dtype(vector):  
[(3.0, 6.0), (10.0, 30.0)]
```

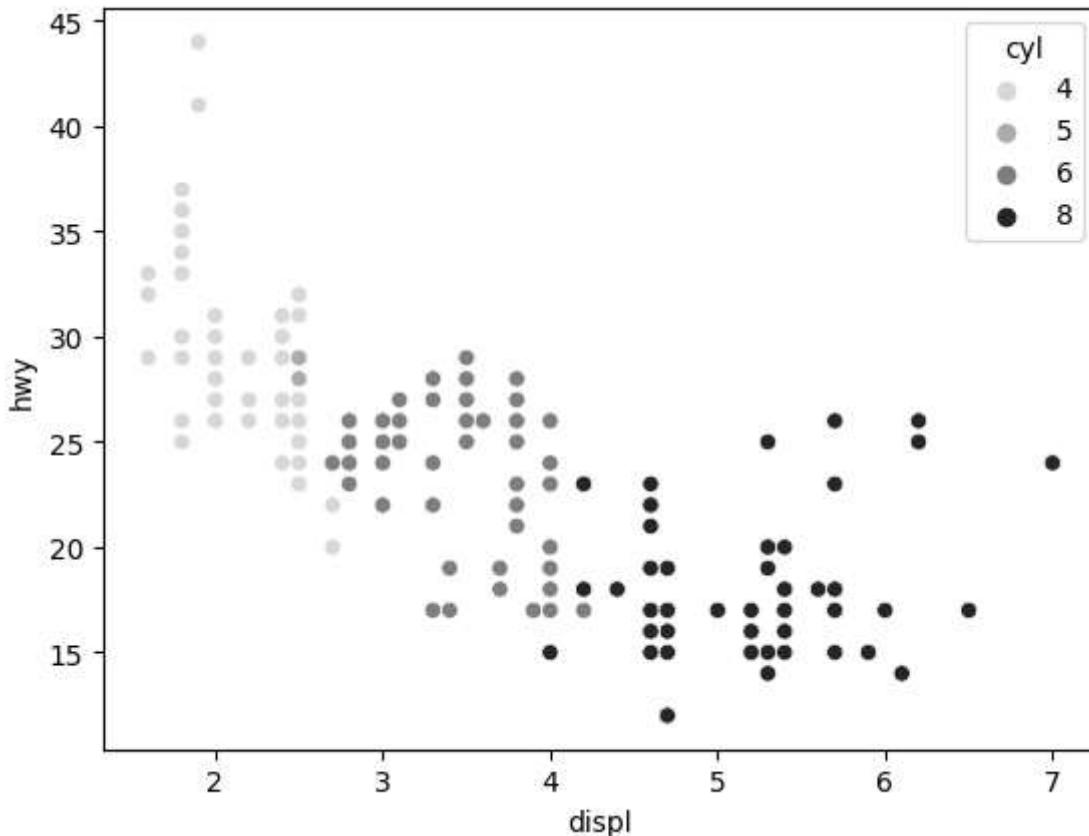
```
Out[77]:
```



▣ 종류별로 표식 색깔 바꾸기

```
In [8]: ## drv별로 표식 색깔 다르게 표현
sns.scatterplot(data = mpg, x = 'displ', y = 'hwy', hue = 'cyl') #색조(hue)는 'drv'

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='displ', ylabel='hwy'>
Out[8]:
```



(알아 두면 좋아요) 그래프 활용하기

그래프 이미지 복사하기

> Shift + (마우스 우측 클릭)

그래프 설정 바꾸기

```
In [11]: ## 그래프 설정 바꾸기: 개별
import matplotlib.pyplot as plt
plt.rcParams.update({'figure.dpi' : '150'})           # 해상도, 기본값 72
plt.rcParams.update({'figure.figsize' : [8, 6]})        # 그림 크기, 기본값 [6, 4]
plt.rcParams.update({'font.size' : '15'})                # 글자 크기, 기본값 10
plt.rcParams.update({'font.family' : 'Malgun Gothic'})  # 폰트, 기본값 sans-serif
```

```
In [111...]: ## 그래프 설정 바꾸기: 한번에
import matplotlib.pyplot as plt
plt.rcParams.update({'figure.dpi'      : '150',          # 해상도, 기본값 72
                     'figure.figsize' : [8, 6],          # 그림 크기, 기본값 [6, 4]
                     'font.size'       : '15',          # 글자 크기, 기본값 10
                     'font.family'     : 'Malgun Gothic'}) # 폰트, 기본값 sans-serif
```

설정 Default 값으로 되돌리기

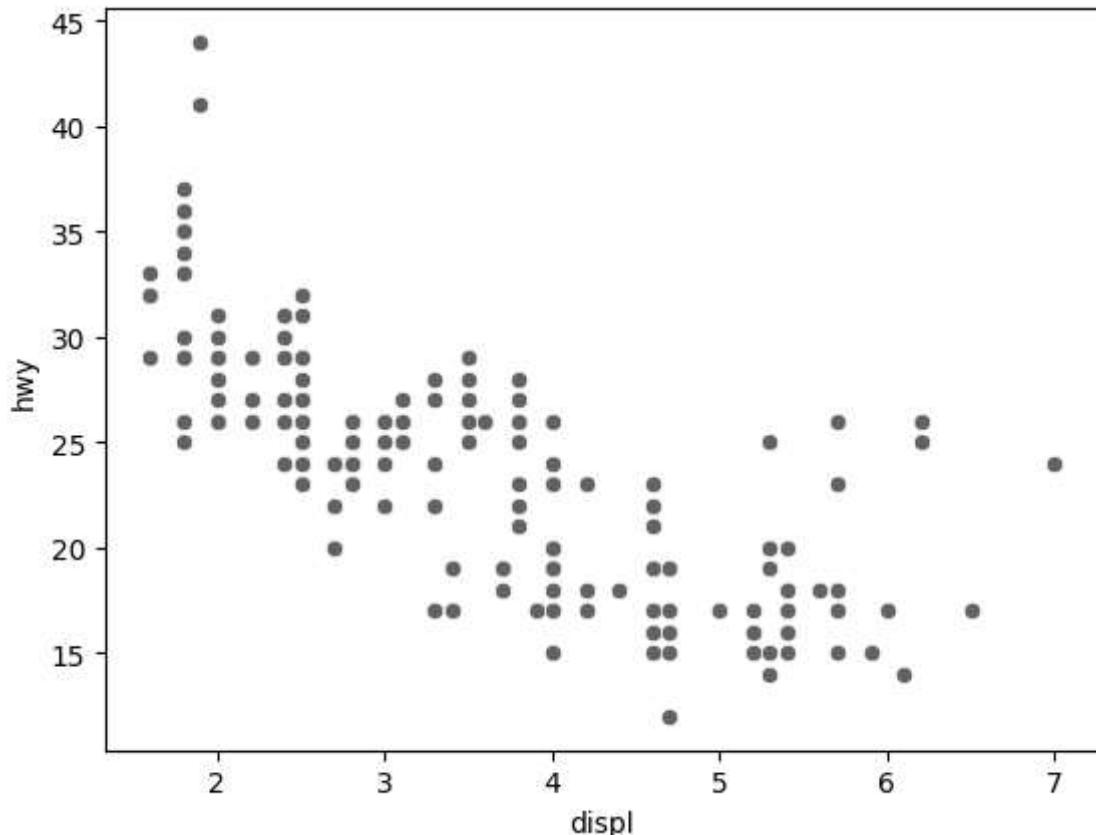
```
In [9]: # 모든 설정 되돌리기
plt.rcParams.update(plt.rcParamsDefault)
```

설명 메시지 숨기기

> 코드 맨 뒤에 ';'를 추가하면 '<Axes: xlabel='displ', ylabel='hwy'>' 메시지를 숨김

```
In [3]: ## 설명 메시지 숨기기  
sns.scatterplot(data = mpg, x = 'displ', y = 'hwy'); # 맨 뒤에 ';' 추가
```

```
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 isin  
stance(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 isin  
stance(dtype, CategoricalDtype) instead  
... if pd.api.types.is_categorical_dtype(vector):
```



```
In [ ]:
```

[08-3] 막대 그래프 - 집단 간 차이 표현하기

[] 평균 막대 그래프 만들기

1. 집단별 평균표 만들기

> **groupby()** : 집단별로 통계값을 구해줌

>> **groupby(집단구분_변수명).agg(변수명 = (대상_변수명, 적용할_함수명))**

> **agg()** : 대상 변수에 대해 단일 통계값을 구해줌

>> **agg(변수명 = (대상_변수명, 적용할_함수명))**

>> 적용 가능 내장함수 : 'sum', 'mean', 'median', 'min', 'max', 'count', 'std', 'var'

```
In [1]: ## mpg 데이터 불러오기
import pandas as pd
mpg = pd.read_csv('mpg.csv')
mpg
```

Out[1]:

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	category
0	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
1	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
2	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
3	audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
4	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
...
229	volkswagen	passat	2.0	2008	4	auto(s6)	f	19	28	p	midsize
230	volkswagen	passat	2.0	2008	4	manual(m6)	f	21	29	p	midsize
231	volkswagen	passat	2.8	1999	6	auto(l5)	f	16	26	p	midsize
232	volkswagen	passat	2.8	1999	6	manual(m5)	f	18	26	p	midsize
233	volkswagen	passat	3.6	2008	6	auto(s6)	f	17	26	p	midsize

234 rows × 11 columns

구동방식('drv')별 고속도로주행연비('hwy') 평균 구하기

```
In [6]: ## 구동방식('drv')별 고속도로주행연비('hwy') 평균 구하기
# 구분 열 'drv'가 index 열로 지정됨
df_mpg = mpg.groupby('drv') \
    .agg(mean_hwy = ('hwy', 'mean'))
df_mpg
```

Out[6]:

drv	mean_hwy
4	19.174757
f	28.160377
r	21.000000

```
In [7]: ## 구동방식('drv')별 고속도로주행연비('hwy') 평균 구하기
# 구분 열 'drv'를 index로 미 사용(자동 부여 index 사용)
df_mpg = mpg.groupby('drv', as_index = False) \
    .agg(mean_hwy = ('hwy', 'mean'))
df_mpg
```

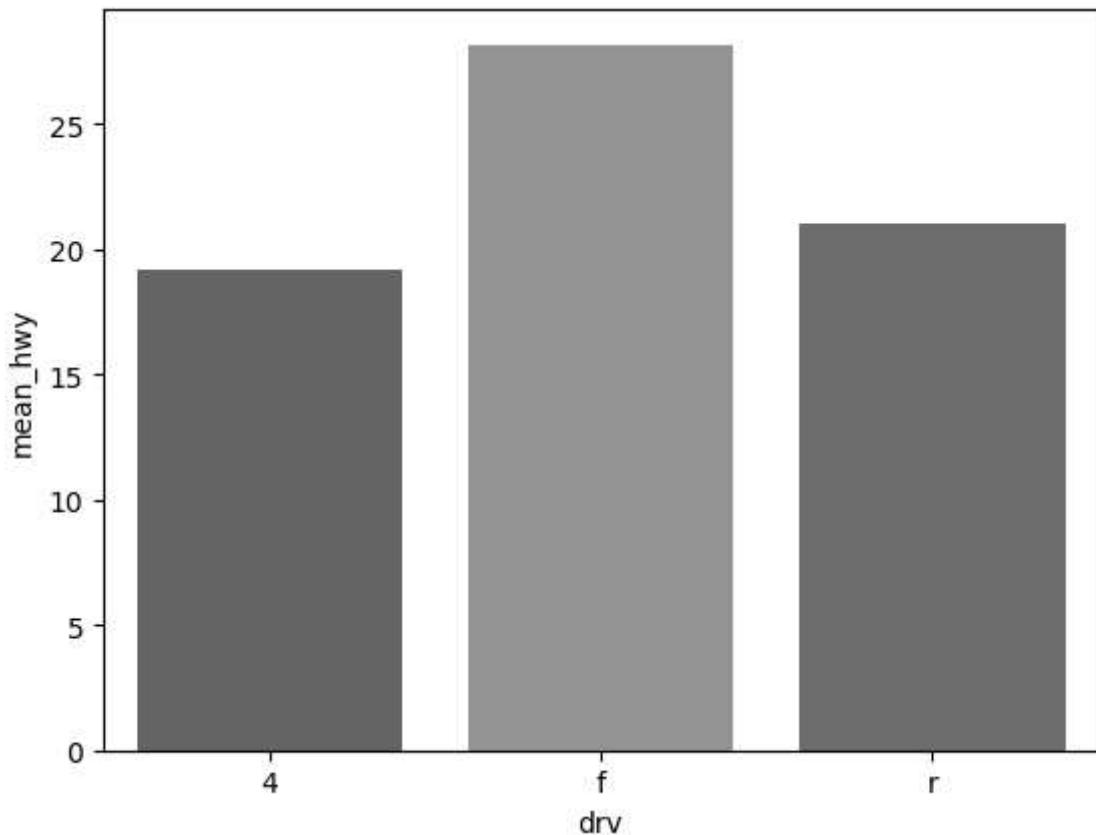
Out[7]:

drv	mean_hwy
0	4 19.174757
1	f 28.160377
2	r 21.000000

2. 그래프 만들기

```
In [8]: ## 막대 그래프 만들기
import seaborn as sns
sns.barplot(data = df_mpg, x = 'drv', y = 'mean_hwy')

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}""
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 isin
stance(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 isin
stance(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 isin
stance(dtype, CategoricalDtype) instead
... if pd.api.types.is_categorical_dtype(vector):
<Axes: xlabel='drv', ylabel='mean_hwy'>
```

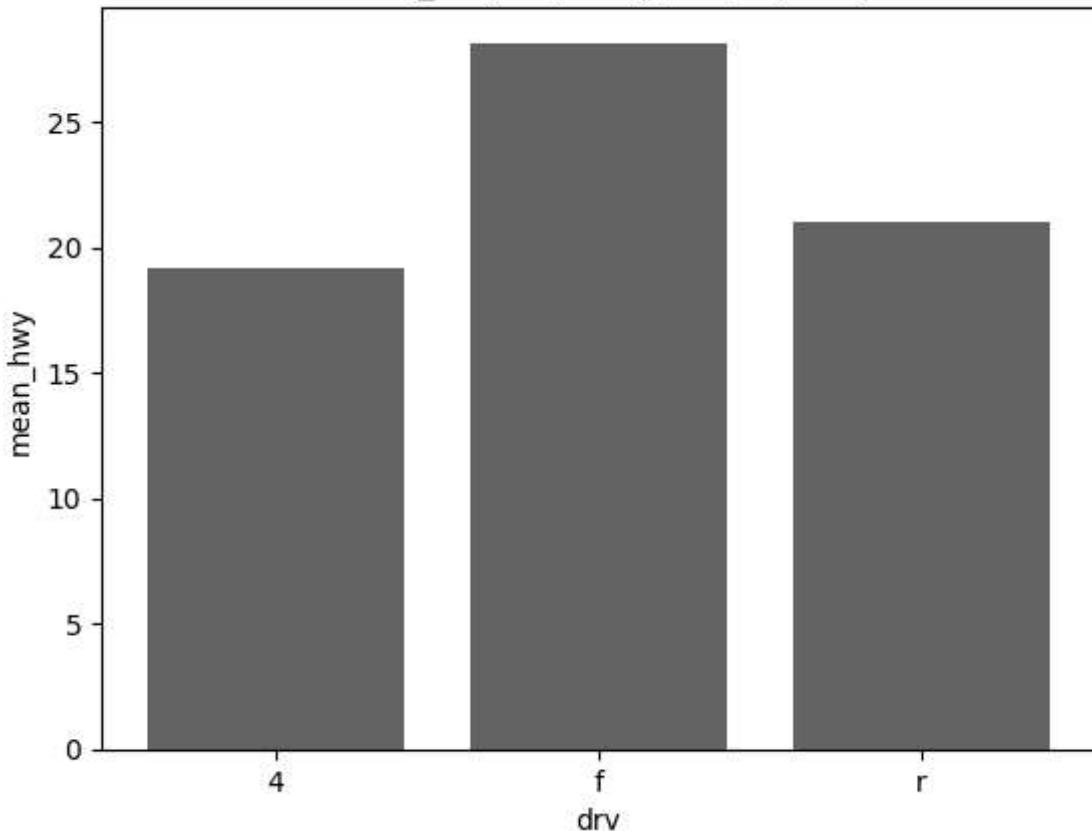


> matplotlib 라이브러리로 그리기

```
In [18]: ## matplotlib 라이브러리로 그리기
import matplotlib.pyplot as plt

plt.title("mean_hwy Report (group by drv)")
plt.xlabel('drv')
plt.ylabel('mean_hwy')
plt.bar(df_mpg['drv'], df_mpg['mean_hwy'])
plt.show()
```

mean_hwyl Report (group by drv)



3. 크기순으로 정렬하기

In [16]:

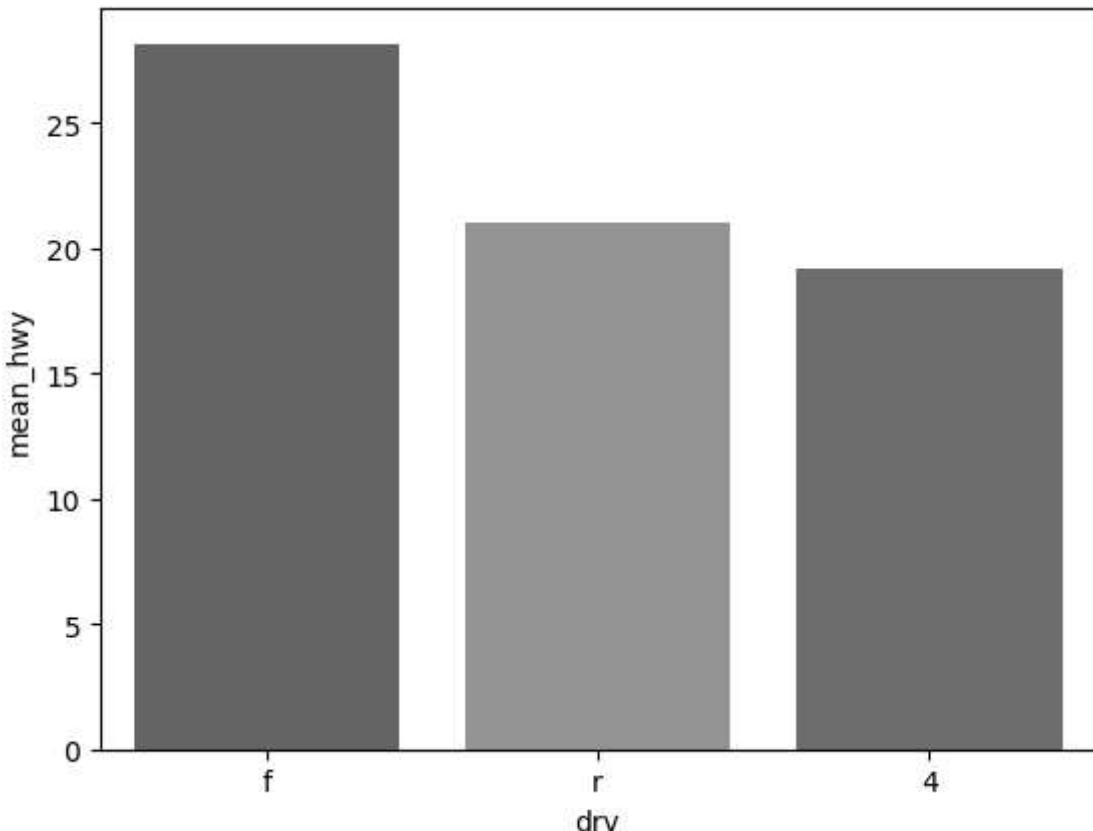
```
## 데이터 프레임 정렬하기
df_mpg = df_mpg.sort_values('mean_hwyl', ascending = False)

## 막대 그래프 만들기
sns.barplot(data = df_mpg, x = 'drv', y = 'mean_hwyl')
```

```
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 isin-
stance(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 isin-
stance(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 isin-
stance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
```

Out[16]:

```
<Axes: xlabel='drv', ylabel='mean_hwyl'>
```



[] barplot()으로 빈도 막대 그래프 만들기

>> agg(변수명 = (대상_변수명, 적용할_함수명))

>> 적용 가능 내장함수 : 'sum', 'mean', 'median', 'min', 'max', 'count', 'std', 'var'

1. 집단별 빈도표 만들기

```
In [17]: ## 집단별 빈도표 만들기
df_mpg = mpg.groupby('drv', as_index = False) \
    .agg(n = ('drv', 'count'))

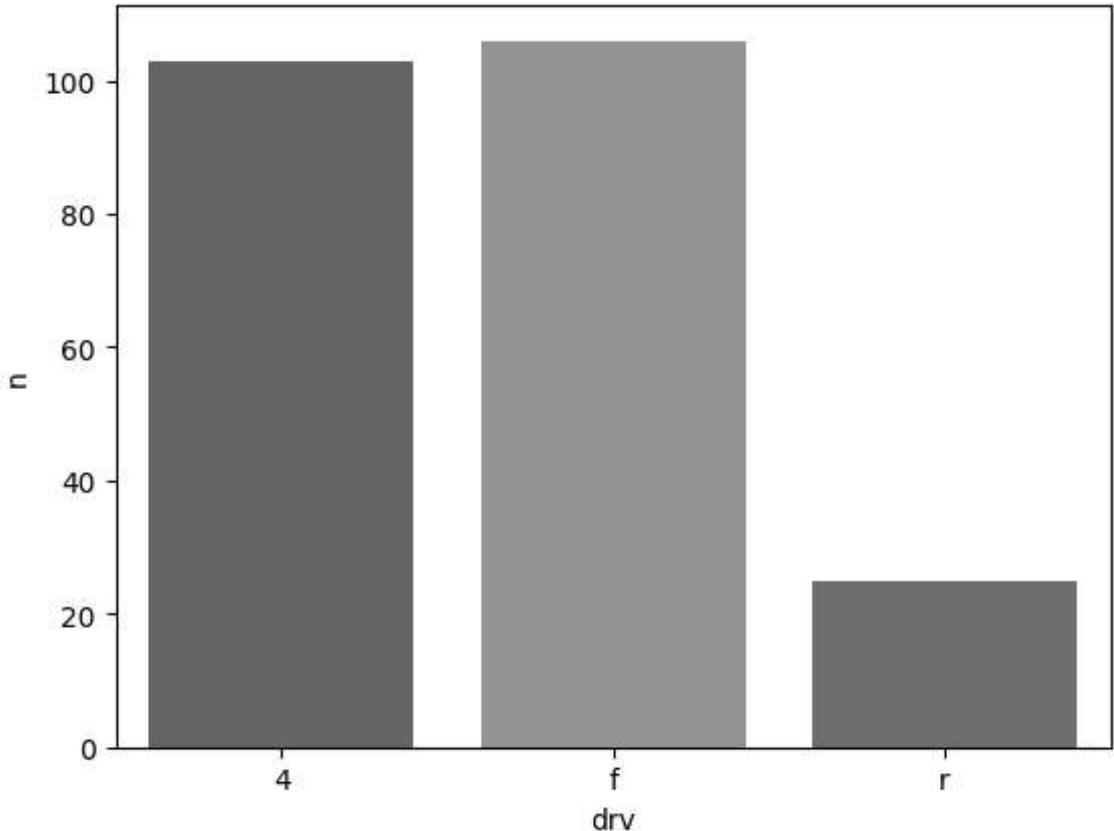
df_mpg
```

```
Out[17]:   drv   n
0     4  103
1     f  106
2     r   25
```

2. 그래프 만들기

```
In [19]: ## 막대 그래프 만들기
import seaborn as sns
sns.barplot(data = df_mpg, x = 'drv', y = 'n')
```

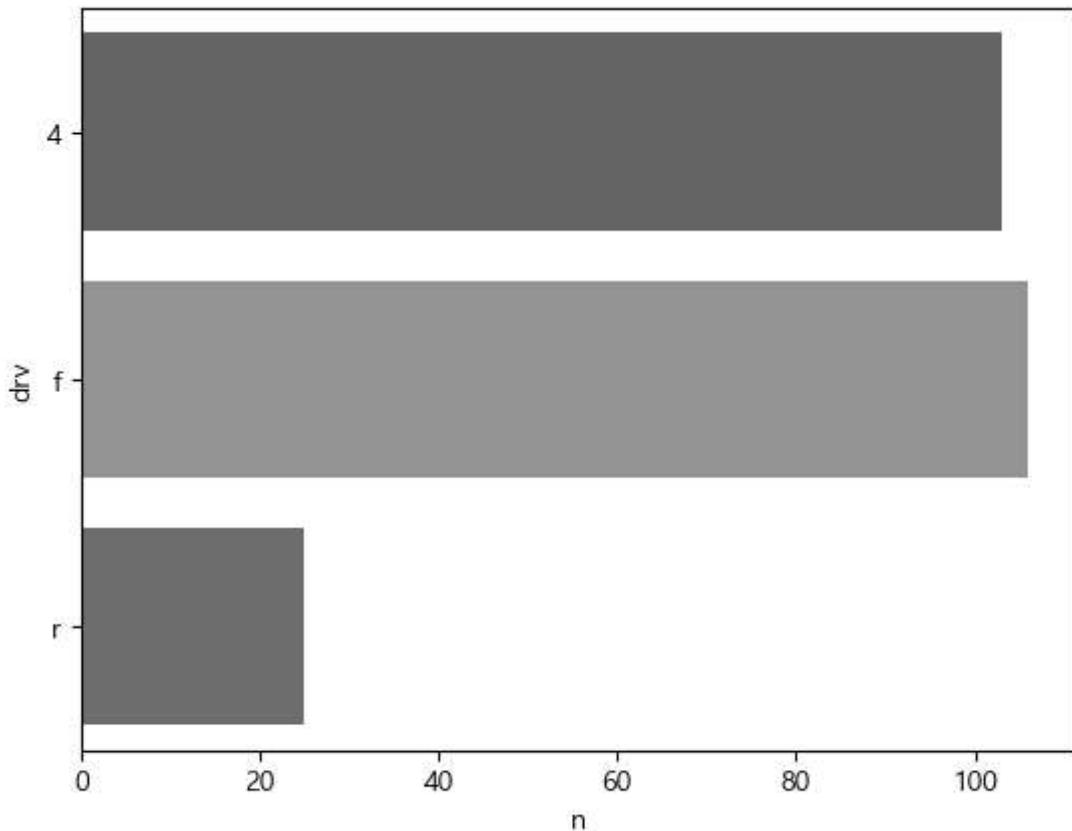
```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning:  
is_categorical_dtype is deprecated and will be removed in a future version. Use isin  
stance(dtype, CategoricalDtype) instead  
... if pd.api.types.is_categorical_dtype(vector):  
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning:  
is_categorical_dtype is deprecated and will be removed in a future version. Use isin  
stance(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 isin  
stance(dtype, CategoricalDtype) instead  
... if pd.api.types.is_categorical_dtype(vector):  
<Axes: xlabel='drv', ylabel='n'>  
Out[19]:
```



x, y 축 바꾸어 그리기

```
In [94]: ## 막대 그래프 만들기  
import seaborn as sns  
sns.barplot(data = df_mpg, y = 'drv', x = '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 isin  
stance(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 isin  
stance(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 isin  
stance(dtype, CategoricalDtype) instead  
... if pd.api.types.is_categorical_dtype(vector):  
<Axes: xlabel='n', ylabel='drv'>
```



[] sns.countplot()으로 빈도 막대 그래프 만들기

In [20]: ## 빈도 막대 그래프 만들기

```
import seaborn as sns  
sns.countplot(data = mpg, x = 'drv')
```

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 isin
stance(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 isin
stance(dtype, CategoricalDtype) instead

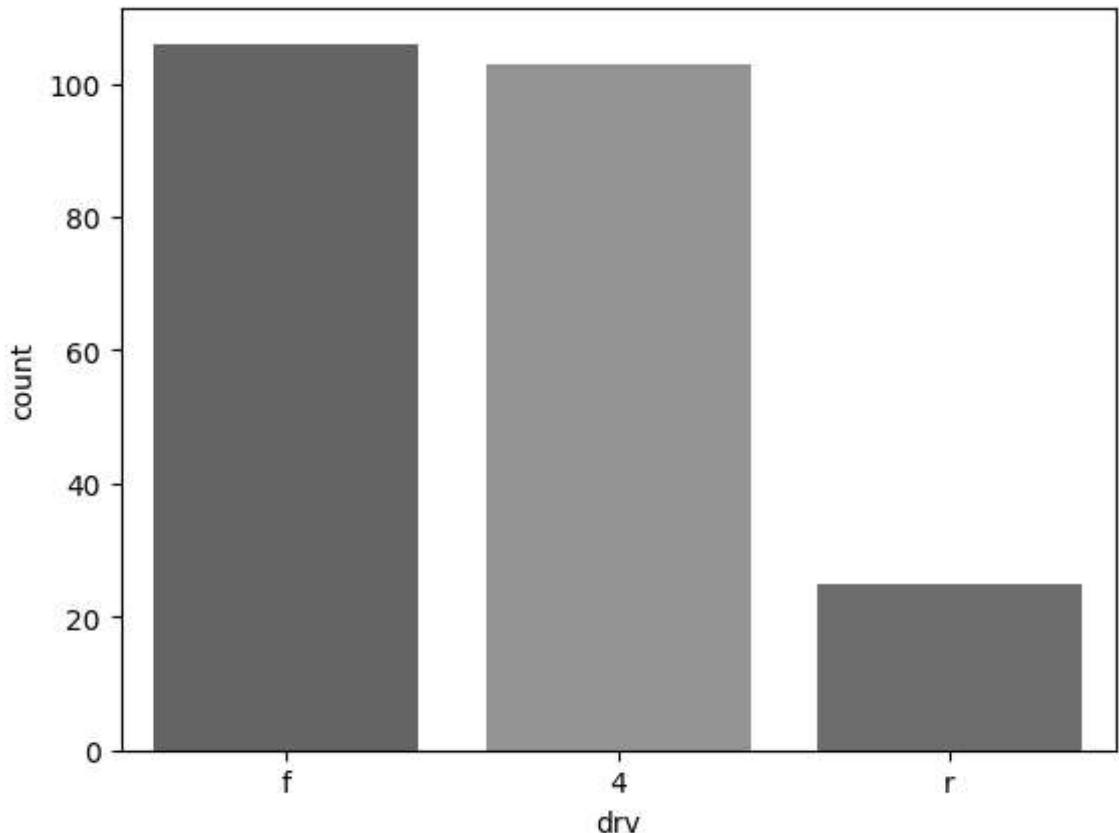
```
.. if pd.api.types.is_categorical_dtype(vector):
```

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 isin
stance(dtype, CategoricalDtype) instead

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

```
<Axes: xlabel='drv', ylabel='count'>
```

Out[20]:

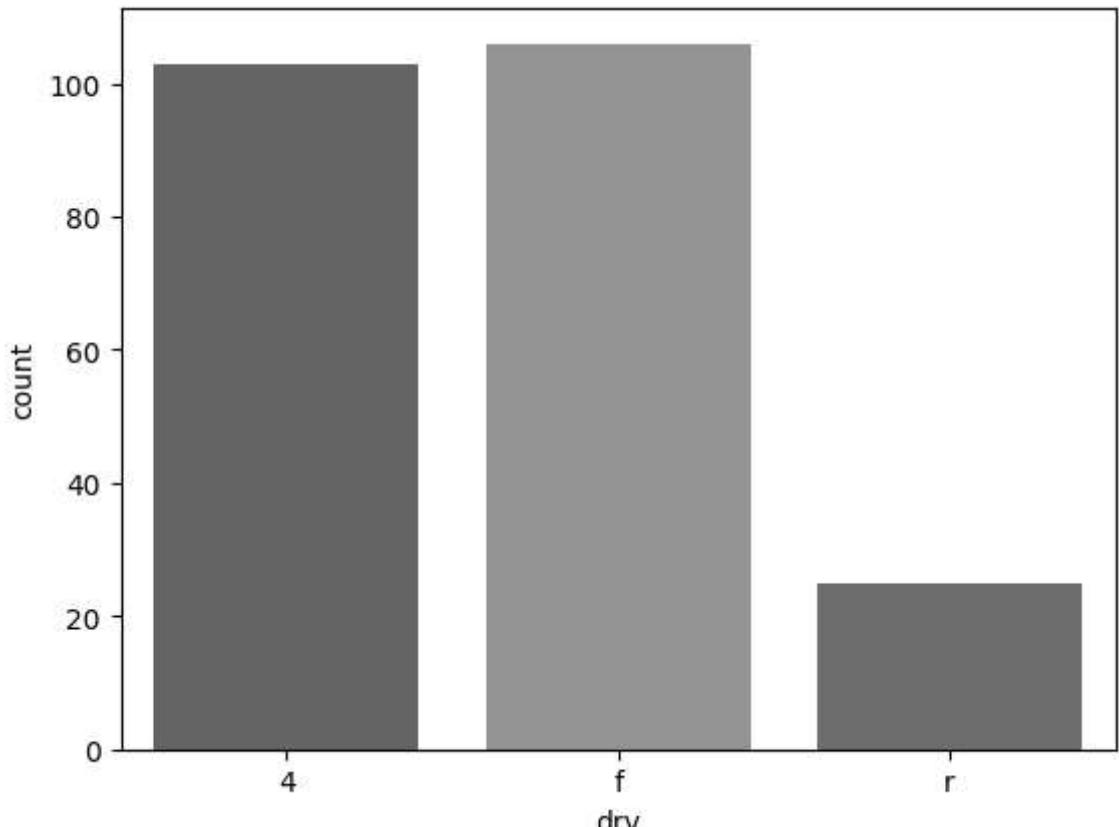


x축 순서 조정

```
In [29]: # 4, f, r 순으로 막대 정렬: 순서 강제 지정  
sns.countplot(data = mpg, x = 'drv', order = ['4', 'f', 'r'])
```

```
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 isin  
stance(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 isin  
stance(dtype, CategoricalDtype) instead  
    if pd.api.types.is_categorical_dtype(vector):  
<Axes: xlabel='drv', ylabel='count'>
```

Out[29]:

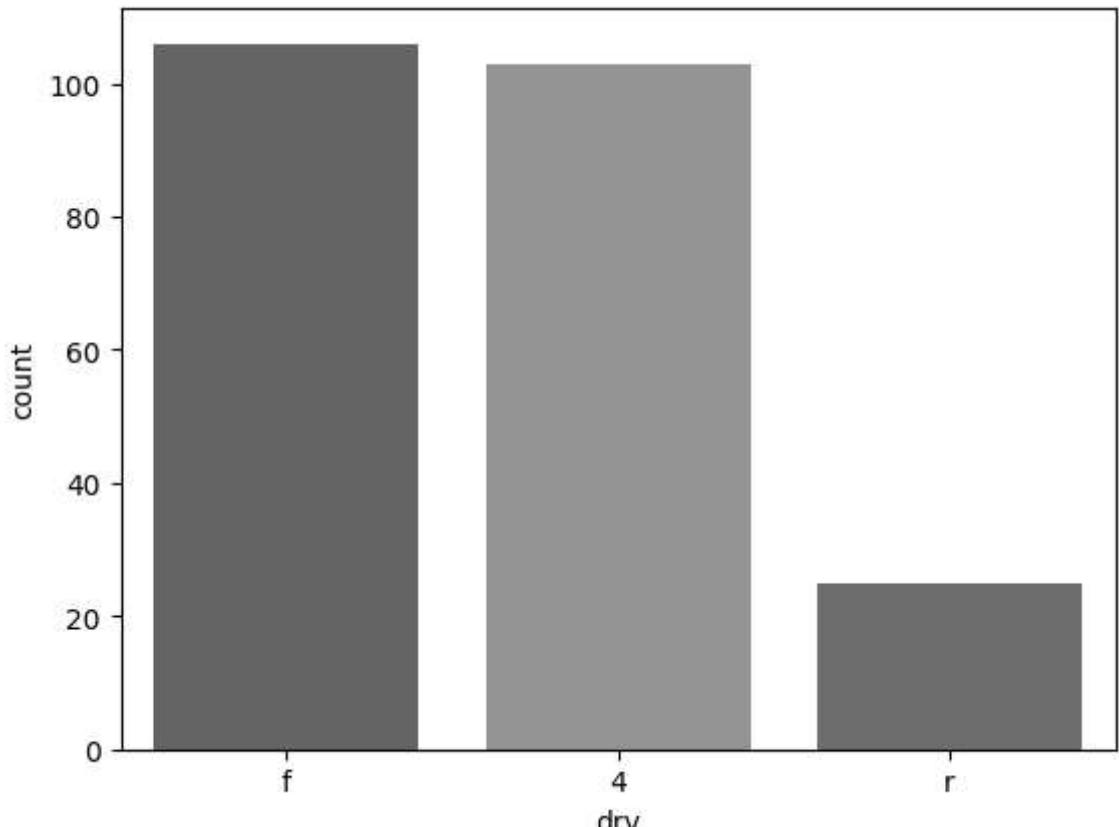


```
In [24]: ## 열 값 도메인 추출: unique()
drv_order = df_mpg['drv'].unique()
drv_order.sort() #List 정렬
drv_order
```

```
Out[24]: array(['4', 'f', 'r'], dtype=object)
```

```
In [30]: ## 4, f, r 순으로 막대 정렬: 순서 정렬 지정
drv_order = df_mpg['drv'].unique().sort() #열 값 도메인 추출 후 정렬
sns.countplot(data = mpg, x = 'drv', order = drv_order)
```

```
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 isin
stance(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 isin
stance(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 isin
stance(dtype, CategoricalDtype) instead
... if pd.api.types.is_categorical_dtype(vector):
<Axes: xlabel='drv', ylabel='count'>
```

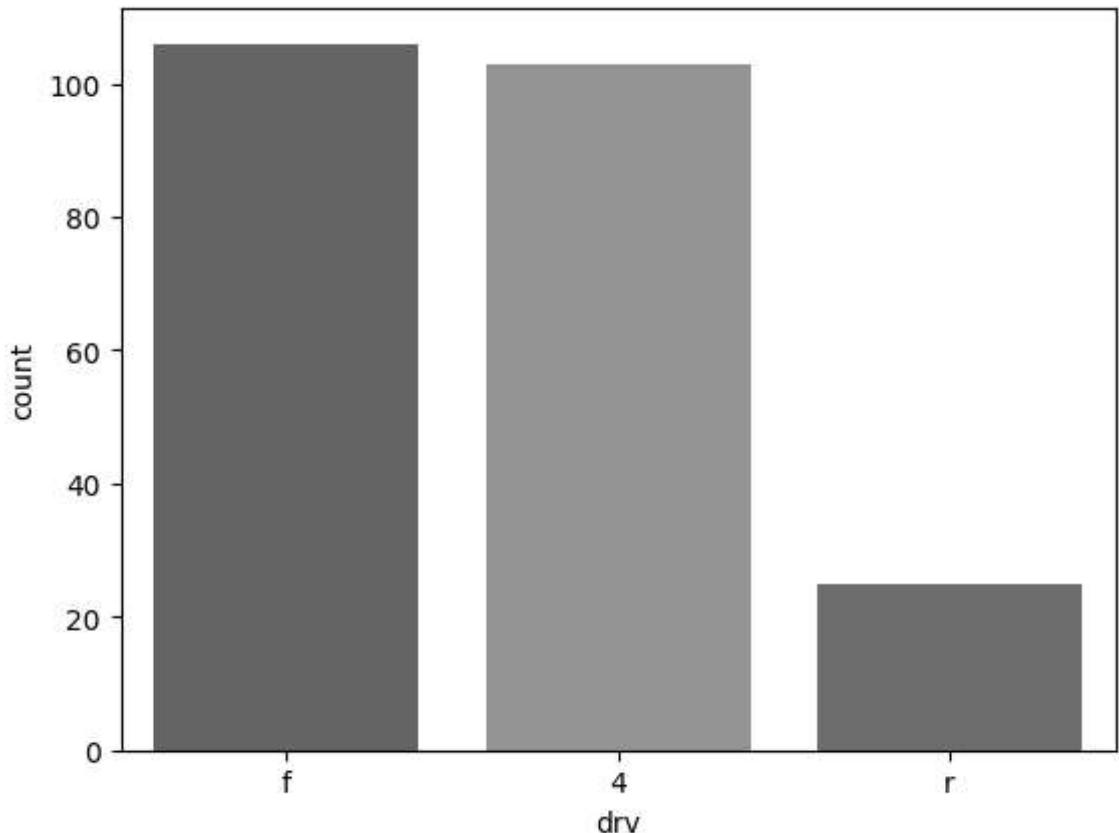


```
In [36]: ## drv의 값을 빈도가 높은 순으로 정렬
drv_order = mpg['drv'].value_counts().index
print(mpg['drv'].value_counts())
drv_order
```

```
drv
f    106
4    103
r    25
Name: count, dtype: int64
Out[36]: Index(['f', '4', 'r'], dtype='object', name='drv')
```

```
In [38]: ## drv 빈도 높은 순으로 막대 정렬
drv_order = mpg['drv'].value_counts().index
sns.countplot(data = mpg, x = 'drv', order = drv_order)
```

```
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 isin-
stance(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 isin-
stance(dtype, CategoricalDtype) instead
... if pd.api.types.is_categorical_dtype(vector):
<Axes: xlabel='drv', ylabel='count'>
Out[38]:
```

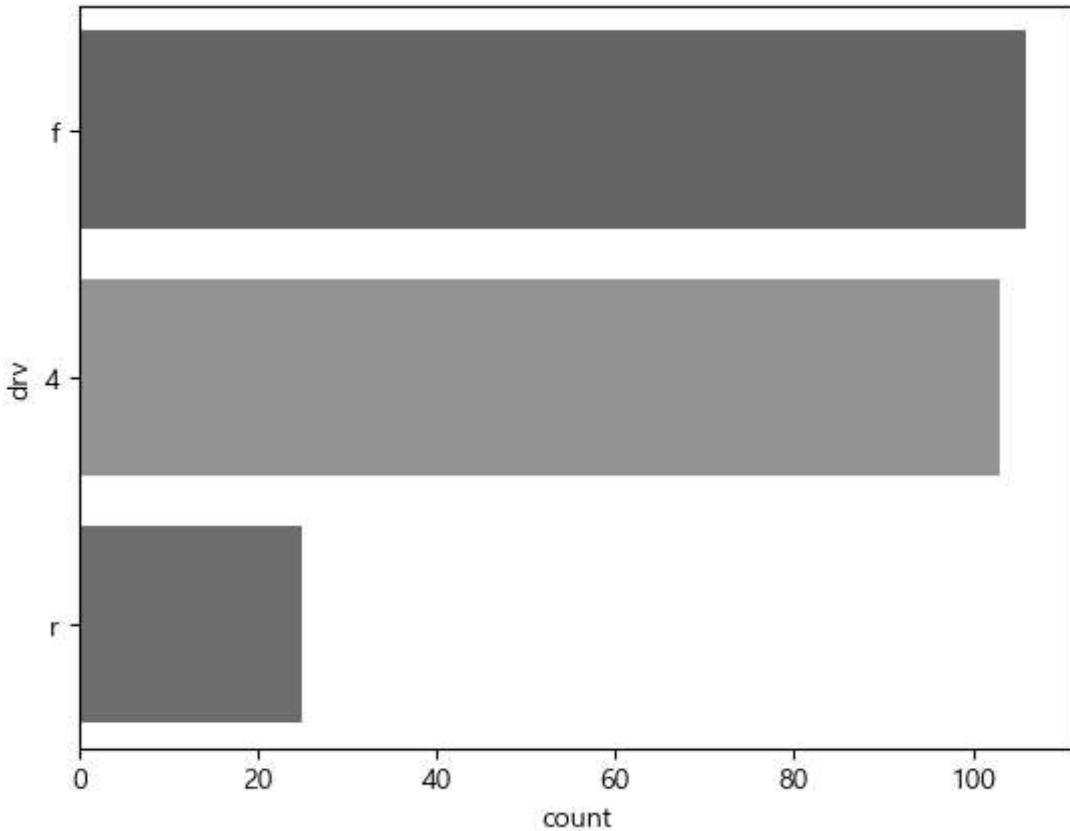


x, y 축 바꾸어 그리기

```
In [93]: ## 막대 그래프 만들기
import seaborn as sns
sns.countplot(data = mpg, y = 'drv', order = drv_order)
```

```
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='count', ylabel='drv'>
```

```
Out[93]:
```



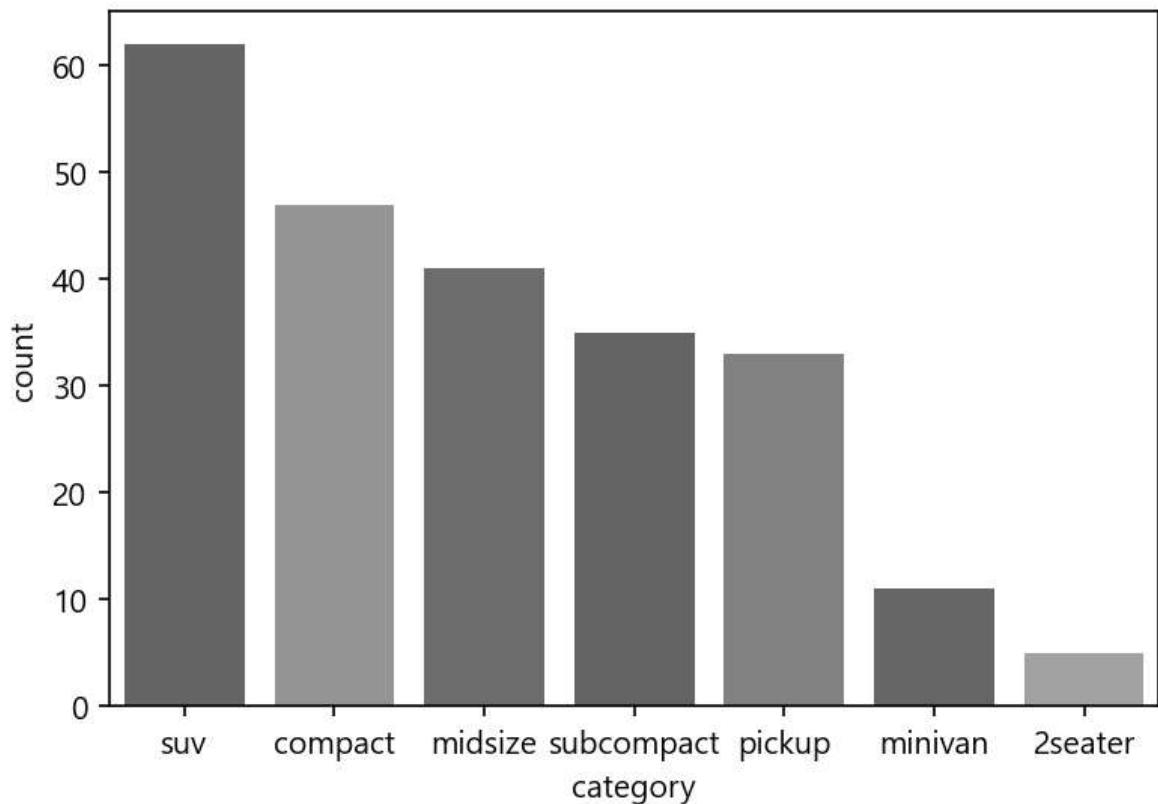
```
In [116]: ## 그래프 설정 바꾸기: 한번에
import matplotlib.pyplot as plt
plt.rcParams.update({'figure.dpi'      : '150',          # 해상도, 기본값 72
                     'figure.figsize': [6, 4],          # 그림 크기, 기본값 [6, 4]
                     'font.size'     : '10',           # 글자 크기, 기본값 10
                     'font.family'   : 'Malgun Gothic'}) # 폰트, 기본값 sans-serif
```



```
In [119]: ## 'category'별 빈도수 그래프 막대 그리기
sns.countplot(data = mpg, x = 'category', order = mpg['category'].value_counts().index)
```

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 isin
 stance(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 isin
 stance(dtype, CategoricalDtype) instead
 ... if pd.api.types.is_categorical_dtype(vector):
 <Axes: xlabel='category', ylabel='count'>

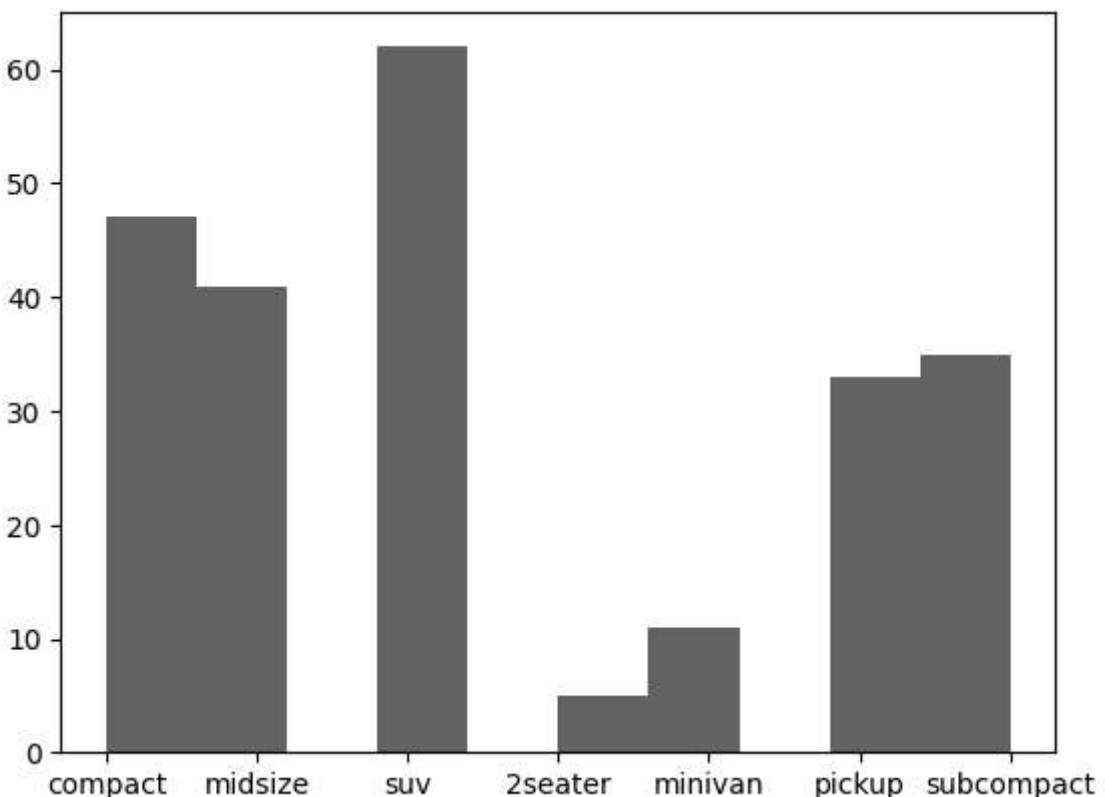
Out[119]:



> matplotlib 라이브러리로 그리기

```
In [20]: ## matplotlib 라이브러리로 그래프 그리기
import matplotlib.pyplot as plt

plt.hist(mpg['category'])
plt.show()
```

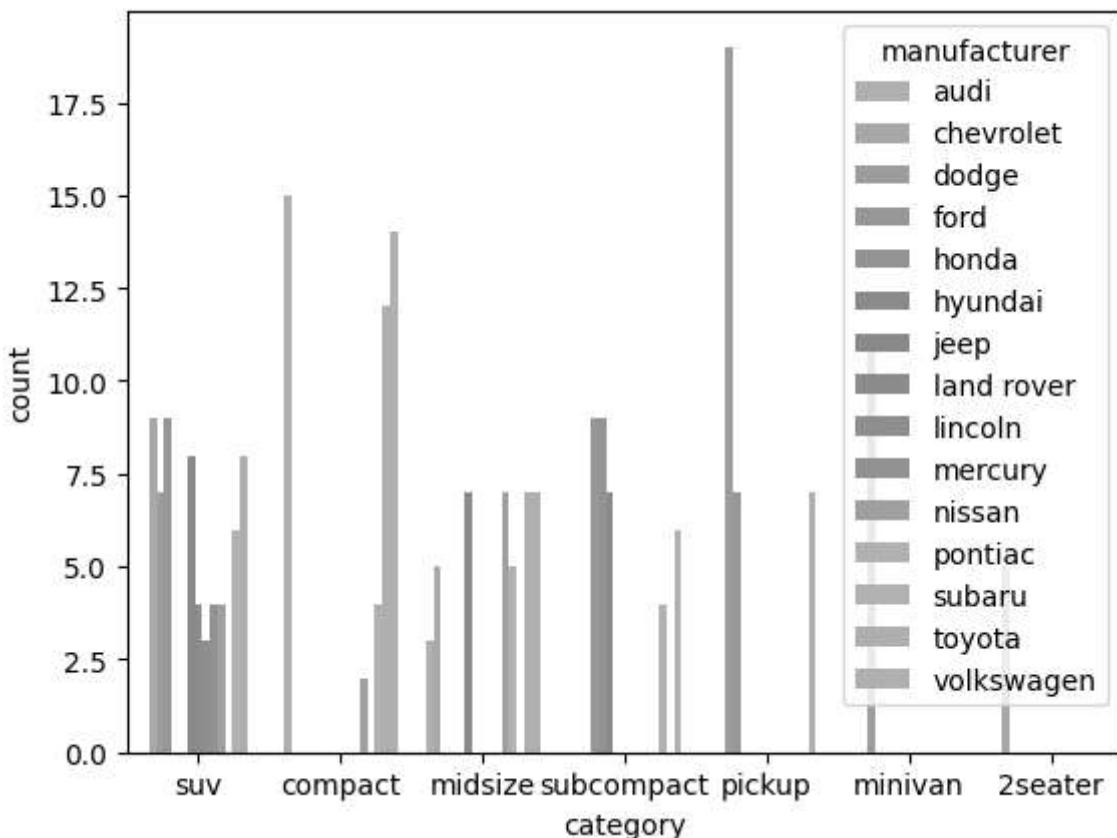


~별, ~별 빈도수 막대 그래프 그리기

```
In [28]: ## 'category'별 'manufacturer'별 빈도수 그래프 막대 그리기
sns.countplot(data = mpg, x = 'category', order = mpg['category'].value_counts().index)

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 isin
stance(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 isin
stance(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 isin
stance(dtype, CategoricalDtype) instead
... if pd.api.types.is_categorical_dtype(vector):
<Axes: xlabel='category', ylabel='count'>

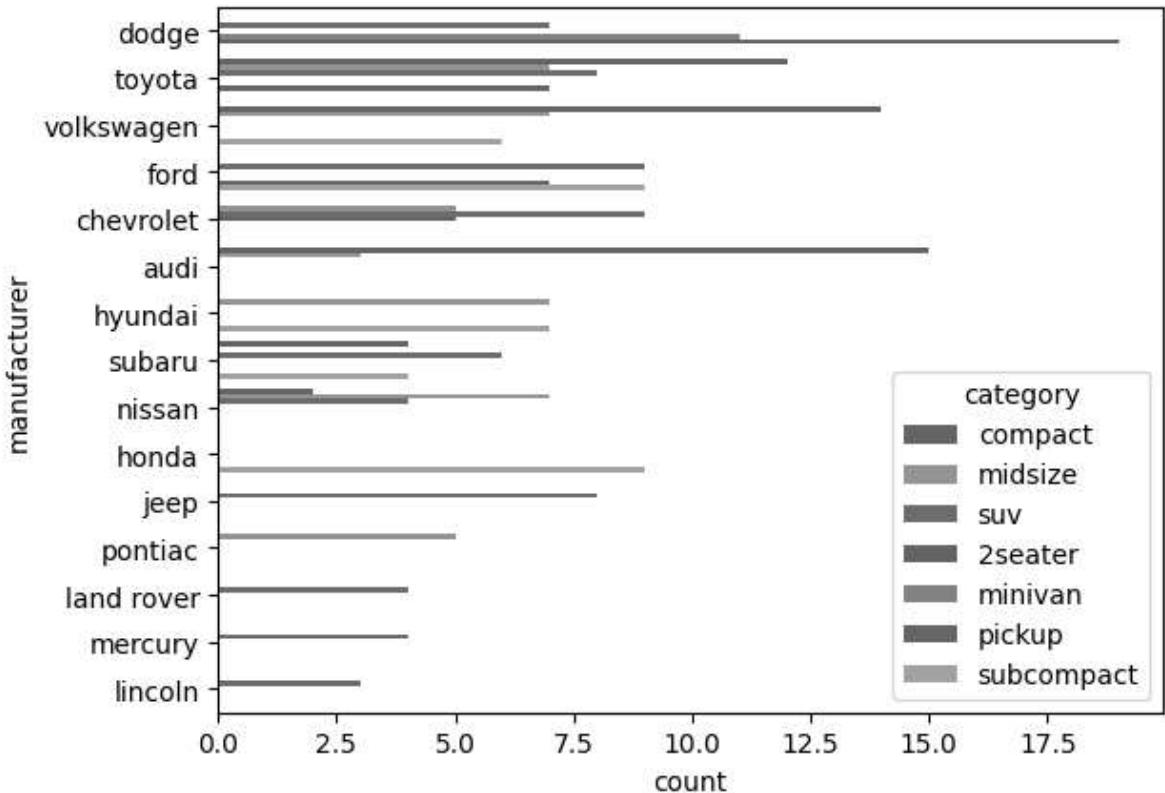
Out[28]:
```



```
In [27]: ## 'manufacturer'별 'category'별 빈도수 그래프 막대 그리기
sns.countplot(data = mpg, y = 'manufacturer', order = mpg['manufacturer'].value_counts().index)

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 isin
stance(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 isin
stance(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 isin
stance(dtype, CategoricalDtype) instead
... if pd.api.types.is_categorical_dtype(vector):
<Axes: xlabel='count', ylabel='manufacturer'>

Out[27]:
```



In []:

[08-3] 파이 그래프 - 집단 간 점유율 표현하기

In []: ## mpg 데이터 불러오기

```
import pandas as pd
mpg = pd.read_csv('mpg.csv')
mpg
```

Out[]:

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	category
0	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
1	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
2	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
3	audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
4	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
...
229	volkswagen	passat	2.0	2008	4	auto(s6)	f	19	28	p	midsize
230	volkswagen	passat	2.0	2008	4	manual(m6)	f	21	29	p	midsize
231	volkswagen	passat	2.8	1999	6	auto(l5)	f	16	26	p	midsize
232	volkswagen	passat	2.8	1999	6	manual(m5)	f	18	26	p	midsize
233	volkswagen	passat	3.6	2008	6	auto(s6)	f	17	26	p	midsize

234 rows × 11 columns

구동방식('drv')별 빈도수('count_drv') 구하기

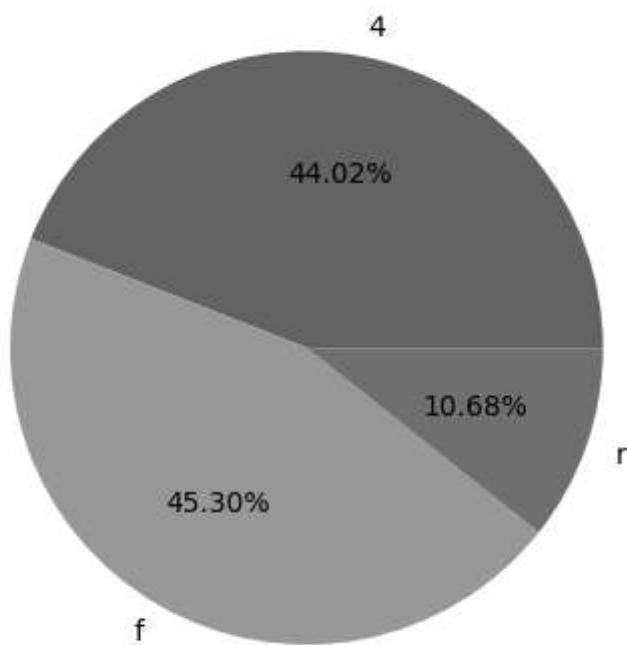
```
In [21]: ## 구동방식('drv')별 고속도로주행연비('hwy') 평균 구하기  
# 구분 열 'drv'가 index 열로 지정됨  
df_mpg = mpg.groupby('drv', as_index = False) #  
        .agg(count_drv = ('drv', 'count'))  
df_mpg
```

```
Out[21]:    drv  count_drv
```

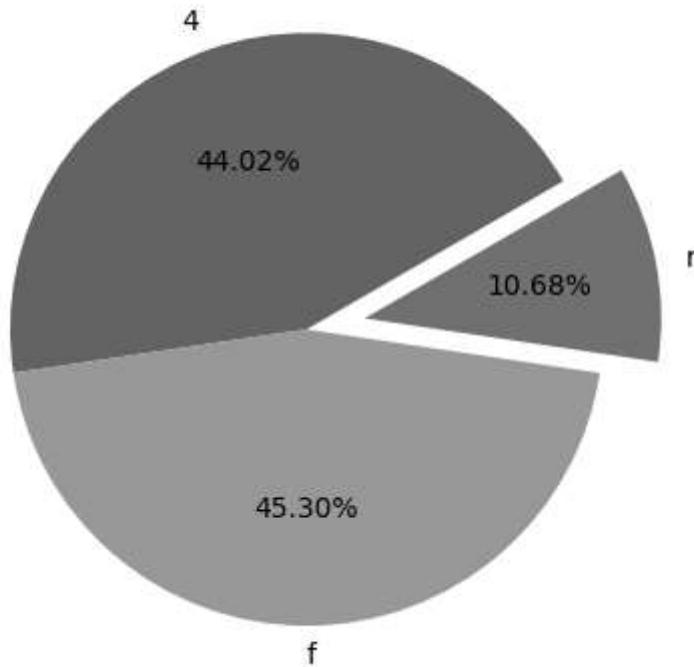
	drv	count_drv
0	4	103
1	f	106
2	r	25

빈도수 파이 그래프 그리기

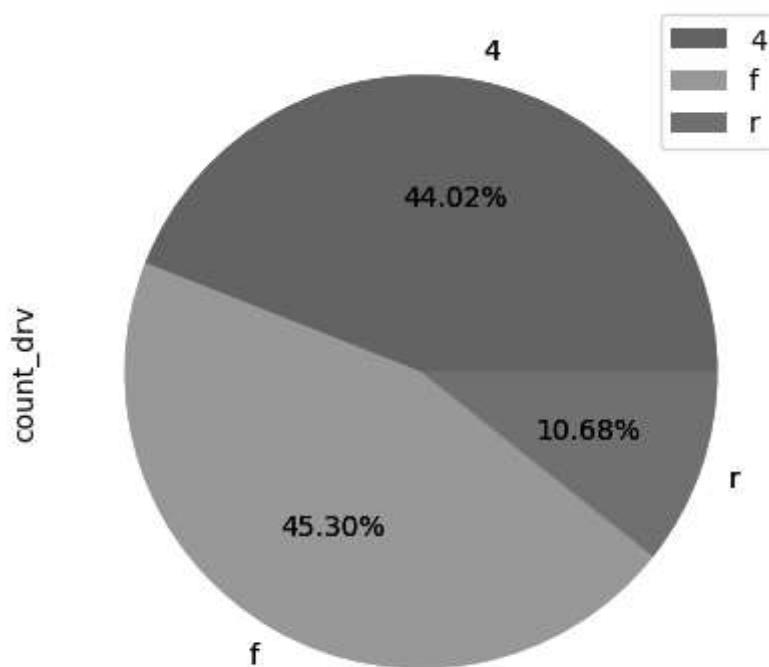
```
In [23]: ##### 파이그래프 그리기  
##### > 'drv'별 'count_drv'에 대한 비율 파이그래프 그리기  
import matplotlib.pyplot as plt  
plt.pie(df_mpg['count_drv'], labels = df_mpg['drv'], autopct = '%0.2f%%')  
plt.show()
```



```
In [24]: ## [matplotlib로 그래프 표현]  
expl_list = [0, 0, 0.2] #중심으로 이격 표현  
plt.pie(df_mpg['count_drv'], labels = df_mpg['drv'], autopct = '%0.2f%%', explode =  
        plt.show()
```



```
In [25]: ## 또 다른 방식  
df_mpg.plot(kind='pie', y='count_drv', labels = df_mpg['drv'], autopct = '%0.2f%%')  
df_mpg['count_drv'].plot(kind='pie', labels = df_mpg['drv'], autopct = '%0.2f%%')  
plt.show()
```



```
In [ ]:
```

[08-4] 선 그래프 - 시간에 따라 달라지는 데이터 표현하기

선 그래프 그리기

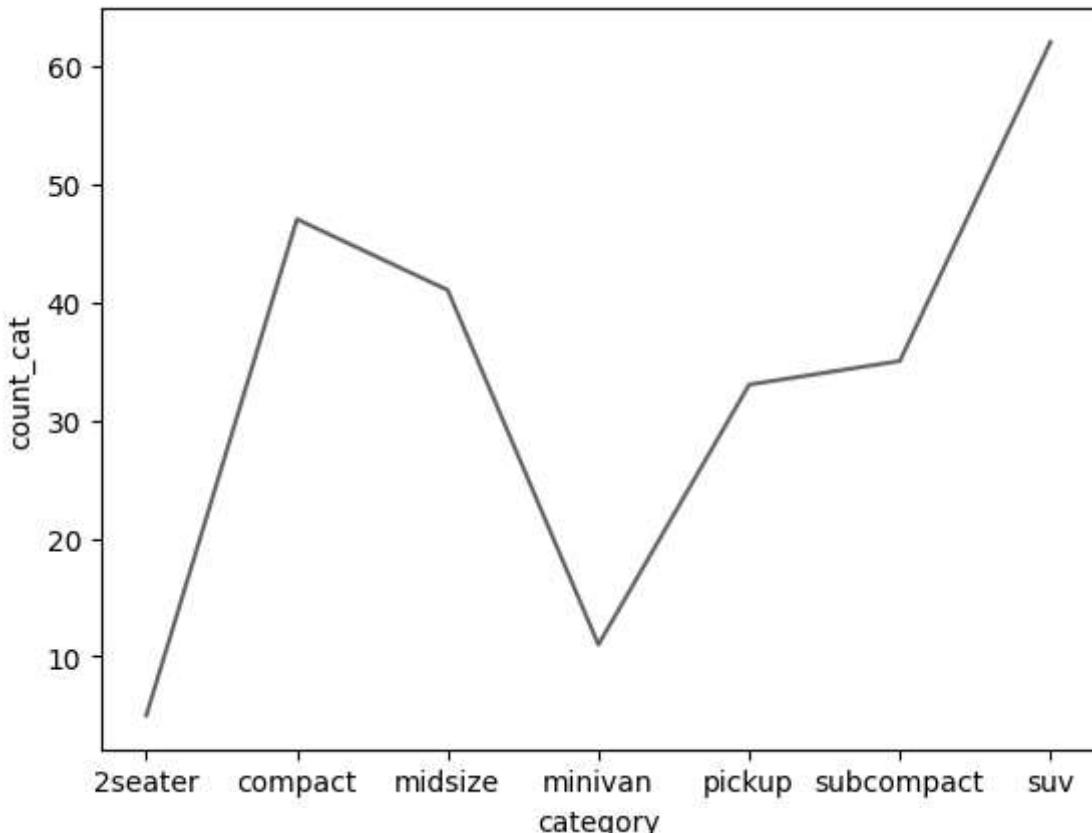
```
In [30]: ## 'category'별 빈도수 구하기  
df_mpg = mpg.groupby('category', as_index = False) %>%  
    .agg(count_cat = ('category', 'count'))  
df_mpg
```

```
Out[30]:   category  count_cat  
0      2seater        5  
1     compact       47  
2    midsize       41  
3   minivan        11  
4    pickup        33  
5  subcompact       35  
6       suv        62
```

```
In [31]: ## 'category'의 빈도수에 대한 선 그래프 구하기  
import seaborn as sns  
  
sns.lineplot(data = df_mpg, x = 'category', y = 'count_cat')
```

```
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 isin  
stance(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 isin  
stance(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 isin  
stance(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='category', ylabel='count_cat'>
```

```
Out[31]:
```

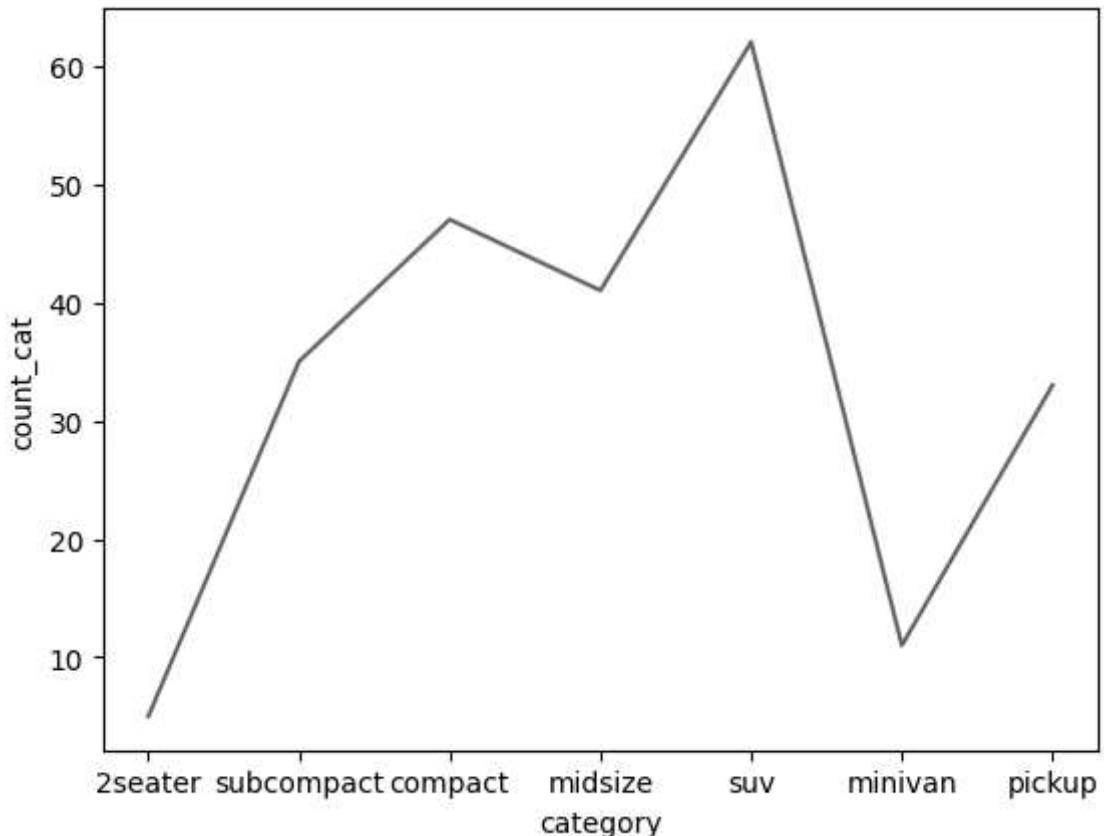


```
In [32]: ## 'category'의 빈도수에 대한 선 그래프 구하기
import seaborn as sns
```

```
cat_order = [0, 5, 1, 2, 6, 3, 4]
df_mpg = df_mpg.reindex(cat_order) ## 행 순서 index 순서로 재정렬 하기
sns.lineplot(data = df_mpg, x = 'category', y = 'count_cat')
```

```
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 isin
stance(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 isin
stance(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 isin
stance(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[32]: <Axes: xlabel='category', ylabel='count_cat'>
```



▣ 시계열 그래프 만들기

```
In [33]: ## economics 데이터 불러오기
economics = pd.read_csv('economics.csv')
economics.head()
```

```
Out[33]:
```

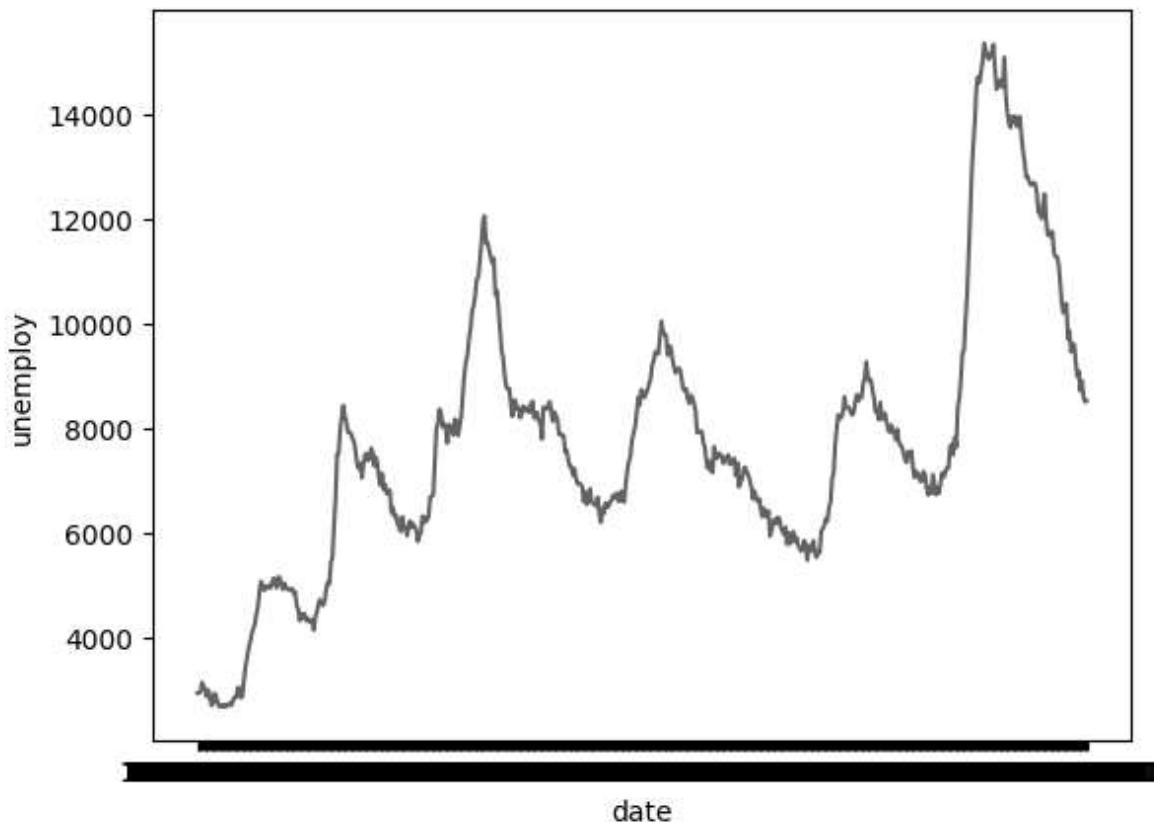
	date	pce	pop	psavert	uempmed	unemploy
0	1967-07-01	506.7	198712.0	12.6	4.5	2944
1	1967-08-01	509.8	198911.0	12.6	4.7	2945
2	1967-09-01	515.6	199113.0	11.9	4.6	2958
3	1967-10-01	512.2	199311.0	12.9	4.9	3143
4	1967-11-01	517.4	199498.0	12.8	4.7	3066

```
In [34]: ## lineplot()로 시계열 그래프 만들기
import seaborn as sns

sns.lineplot(data = economics, x = 'date', y = 'unemploy')
```

```
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 isin  
stance(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 isin  
stance(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 isin  
stance(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='date', ylabel='unemploy'>
```

Out[34]:



x축에 연도 표시하기

(1) 날짜 시간 타입 변수 만들기

```
In [35]: ## 'date' 변수 타입 확인 >> Object(문자) 타입  
economics.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 574 entries, 0 to 573
Data columns (total 6 columns):
 #   Column   Non-Null Count  Dtype  
--- 
 0   date     574 non-null    object 
 1   pce      574 non-null    float64 
 2   pop      574 non-null    float64 
 3   psavert  574 non-null    float64 
 4   uempmed  574 non-null    float64 
 5   unemploy 574 non-null    int64  
dtypes: float64(4), int64(1), object(1)
memory usage: 27.0+ KB
```

```
In [36]: ## 날짜 자료형에서 년, 월, 일 추출하기
economics['date'].dt.year
```

```
-----
-- 
AttributeError: Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_2604\53950759.py in <module>
    1 ## 날짜 자료형에서 년, 월, 일 추출하기
----> 2 economics['date'].dt.year

~\Anaconda3\lib\site-packages\pandas\core\generic.py in __getattr__(self, name)
    6200         ):
    6201             return self[name]
-> 6202         return object.__getattribute__(self, name)
    6203
    6204     @final

~\Anaconda3\lib\site-packages\pandas\core\accessor.py in __get__(self, obj, cls)
    222         # we're accessing the attribute of the class, i.e., Data
set.geo
    223             return self._accessor
--> 224         accessor_obj = self._accessor(obj)
    225         # Replace the property with the accessor object. Inspired b
y:
    226             # https://www.pydanny.com/cached-property.html

~\Anaconda3\lib\site-packages\pandas\core\indexes\accessors.py in __new__(c
ls, data)
    606             return PeriodProperties(data, orig)
    607
--> 608         raise AttributeError("Can only use .dt accessor with dateti
melike values")
AttributeError: Can only use .dt accessor with datetimelike values
```

```
In [37]: ## 날짜 시간 타입 변수 만들기
economics['date2'] = pd.to_datetime(economics['date']) #날짜 시간 타입으로 바꾸어 준다

# 변수 타입 확인
economics.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 574 entries, 0 to 573
Data columns (total 7 columns):
 #   Column   Non-Null Count  Dtype   ... 
---  -- 
 0   date     574 non-null    object  ...
 1   pce      574 non-null    float64 ...
 2   pop      574 non-null    float64 ...
 3   psavert  574 non-null    float64 ...
 4   uempmed  574 non-null    float64 ...
 5   unemploy 574 non-null    int64   ...
 6   date2    574 non-null    datetime64[ns]
dtypes: datetime64[ns](1), float64(4), int64(1), object(1)
memory usage: 31.5+ KB
```

```
In [38]: ## 날짜 자료형에서 년, 월, 일 추출하기
year = economics['date2'].dt.year
month = economics['date2'].dt.month
day = economics['date2'].dt.day
print(year, month, day)
```

```
0      1967
1      1967
2      1967
3      1967
4      1967
...
569    2014
570    2015
571    2015
572    2015
573    2015
Name: date2, Length: 574, dtype: int32 0      7
1      8
2      9
3      10
4      11
...
569    12
570    1
571    2
572    3
573    4
Name: date2, Length: 574, dtype: int32 0      1
1      1
2      1
3      1
4      1
...
569    1
570    1
571    1
572    1
573    1
Name: date2, Length: 574, dtype: int32
```

```
In [39]: ## 'date2'에서 년도를 추출하여 'year' 열 변수 만들기
economics['year'] = economics['date2'].dt.year
economics.head()
```

Out[39]:

	date	pce	pop	psavert	uempmmed	unemploy	date2	year
0	1967-07-01	506.7	198712.0	12.6	4.5	2944	1967-07-01	1967
1	1967-08-01	509.8	198911.0	12.6	4.7	2945	1967-08-01	1967
2	1967-09-01	515.6	199113.0	11.9	4.6	2958	1967-09-01	1967
3	1967-10-01	512.2	199311.0	12.9	4.9	3143	1967-10-01	1967
4	1967-11-01	517.4	199498.0	12.8	4.7	3066	1967-11-01	1967

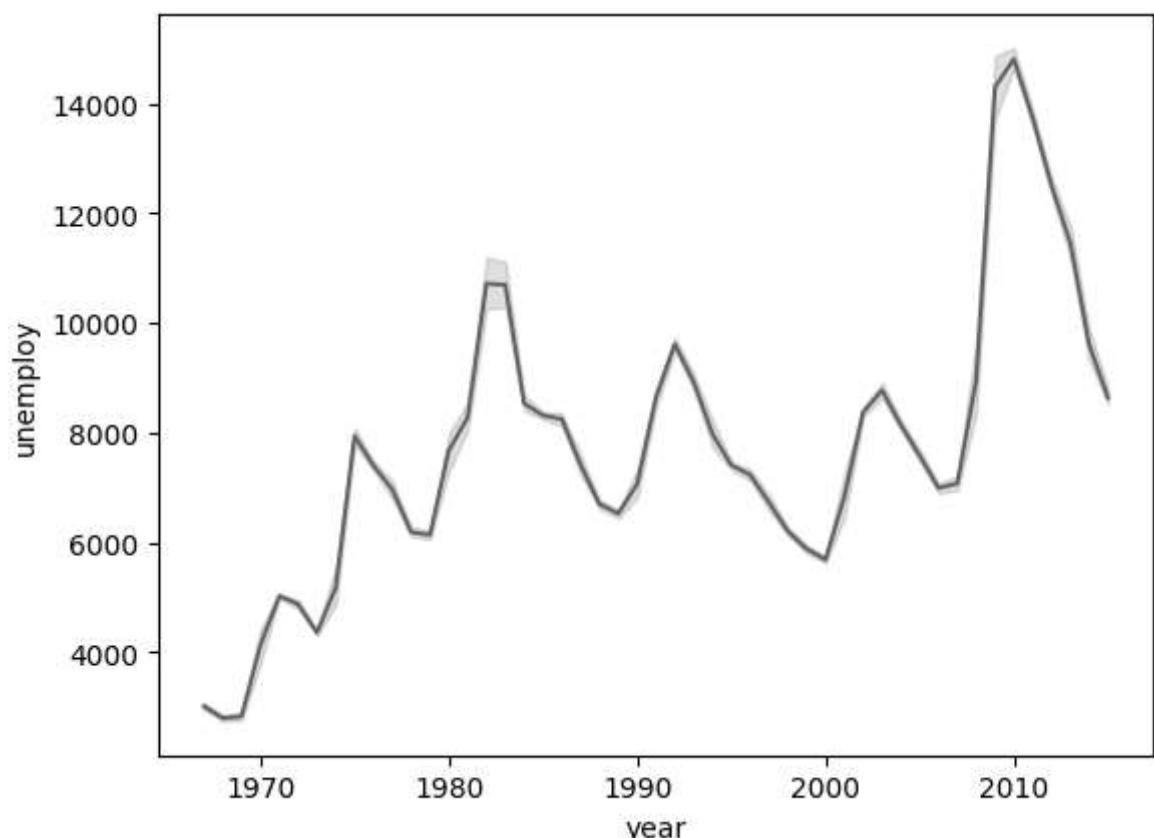
(2) x축에 연도 표시하기

In [40]:

```
## x축에 연도 표시
sns.lineplot(data = economics, x = 'year', y = 'unemploy')
```

```
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 isin
stance(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 isin
stance(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[40]:

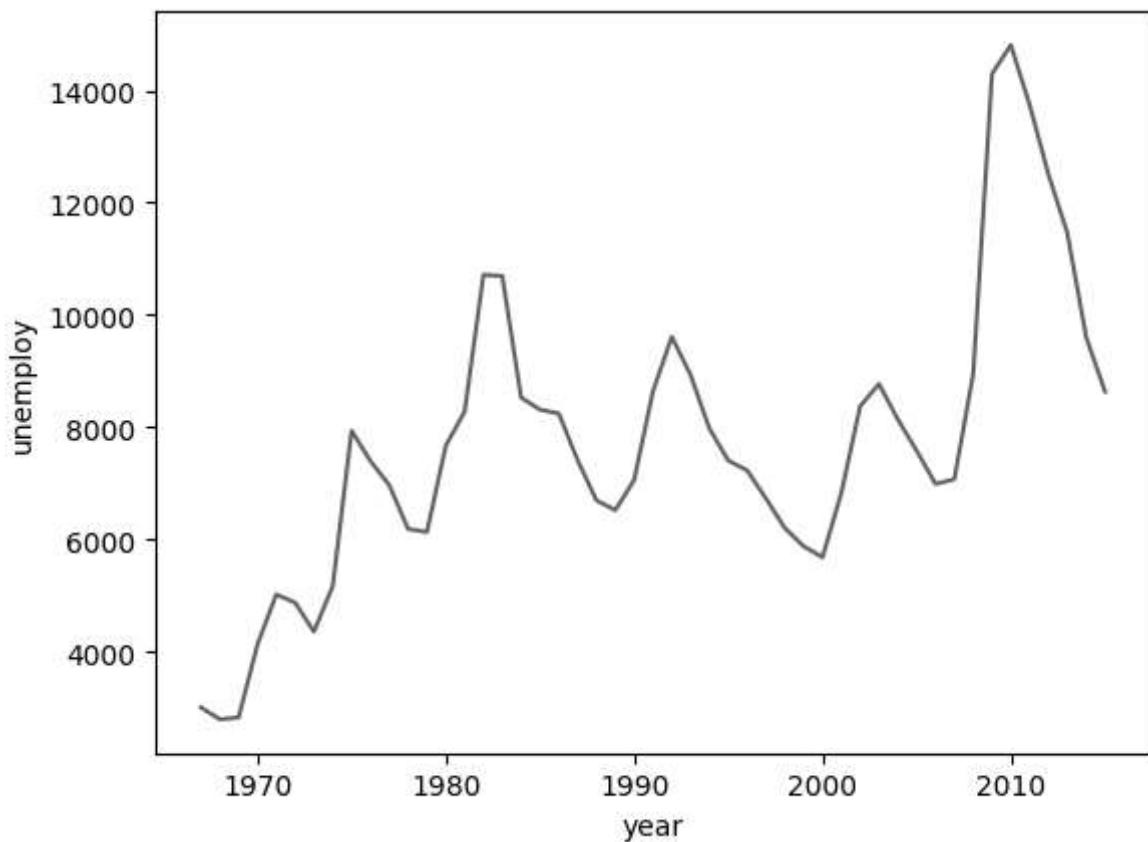


```
In [41]: ## 신뢰구간 제거
sns.lineplot(data = economics, x = 'year', y = 'unemploy', ci = None)

C:\Users\ADMIN\AppData\Local\Temp\ipykernel_2604\3892014309.py:2: FutureWarning:
The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

    sns.lineplot(data = economics, x = 'year', y = 'unemploy', ci = None)
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 isin
stance(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 isin
stance(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='year', ylabel='unemploy'>
```

Out[41]:



(알아 두면 좋아요) seaborn 더 알아보기

한글 표현하기

```
In [43]: ## 맑은 고딕 폰트 설정
import matplotlib.pyplot as plt
plt.rcParams.update({'font.family' : 'Malgun Gothic'})
```

```
In [88]: ## [Data Frame] 관련 함수 : 컬럼명 변경  
economics = economics.rename(columns = {'year' : '년도'}) # 컬럼명 변경  
economics.head()
```

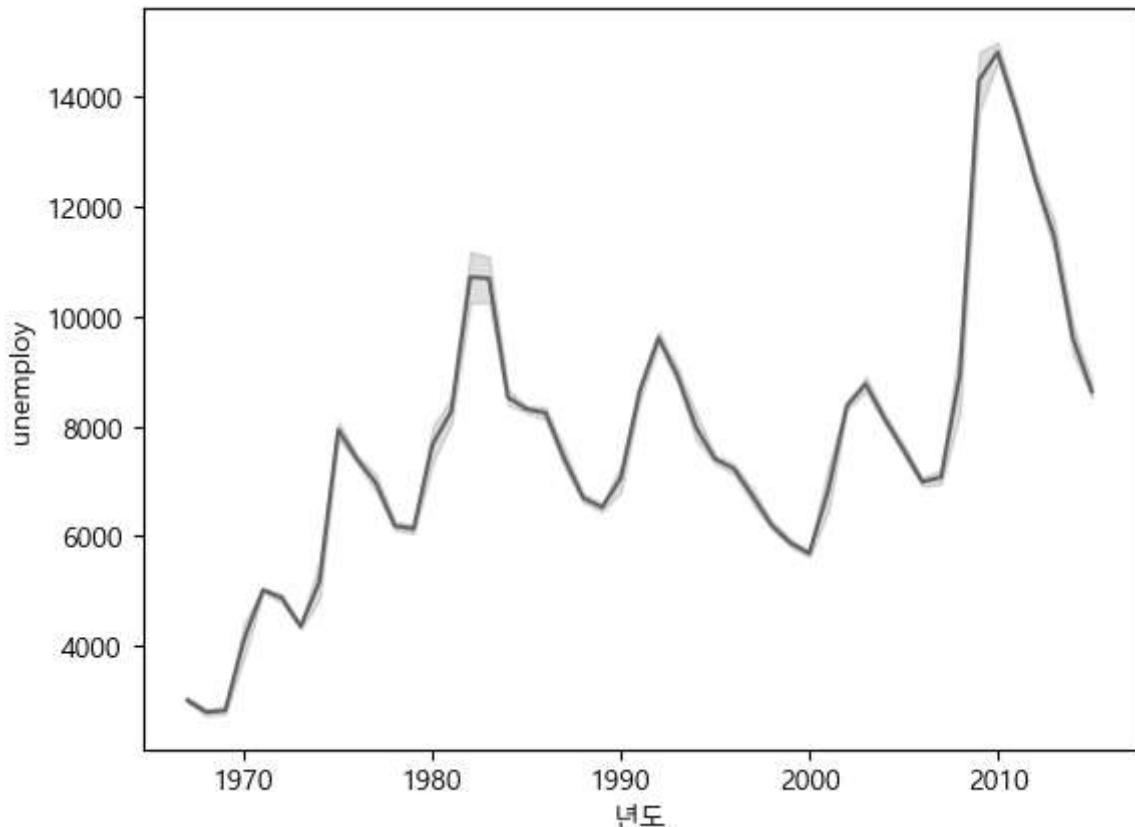
```
Out[88]:
```

	date	pce	pop	psavert	uempmed	unemploy	date2	년도
0	1967-07-01	506.7	198712.0	12.6	4.5	2944	1967-07-01	1967
1	1967-08-01	509.8	198911.0	12.6	4.7	2945	1967-08-01	1967
2	1967-09-01	515.6	199113.0	11.9	4.6	2958	1967-09-01	1967
3	1967-10-01	512.2	199311.0	12.9	4.9	3143	1967-10-01	1967
4	1967-11-01	517.4	199498.0	12.8	4.7	3066	1967-11-01	1967

```
In [89]: ## x축에 연도 표시  
sns.lineplot(data = economics, x = '년도', y = 'unemploy')
```

```
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 isin  
stance(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 isin  
stance(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='년도', ylabel='unemploy'>
```

```
Out[89]:
```



> matplotlib 라이브러리로 그리기

[] 다중 그래프 그리기

```
In [44]: ## 다중 그래프 그리기
import matplotlib.pyplot as plt

# 데이터 생성
x = [1, 2, 3, 4, 5]
y1 = [2, 4, 6, 8, 10]
y2 = [1, 3, 5, 7, 9]
y3 = [3, 6, 3, 9, 5]

# 그래프 그리기
plt.figure(figsize=(8, 6)) # 그림 크기 설정 (선택사항)

# 첫 번째 꺥은선 그래프
plt.plot(x, y1, label='선 그래프 1', marker='o', linestyle='-', color='b')

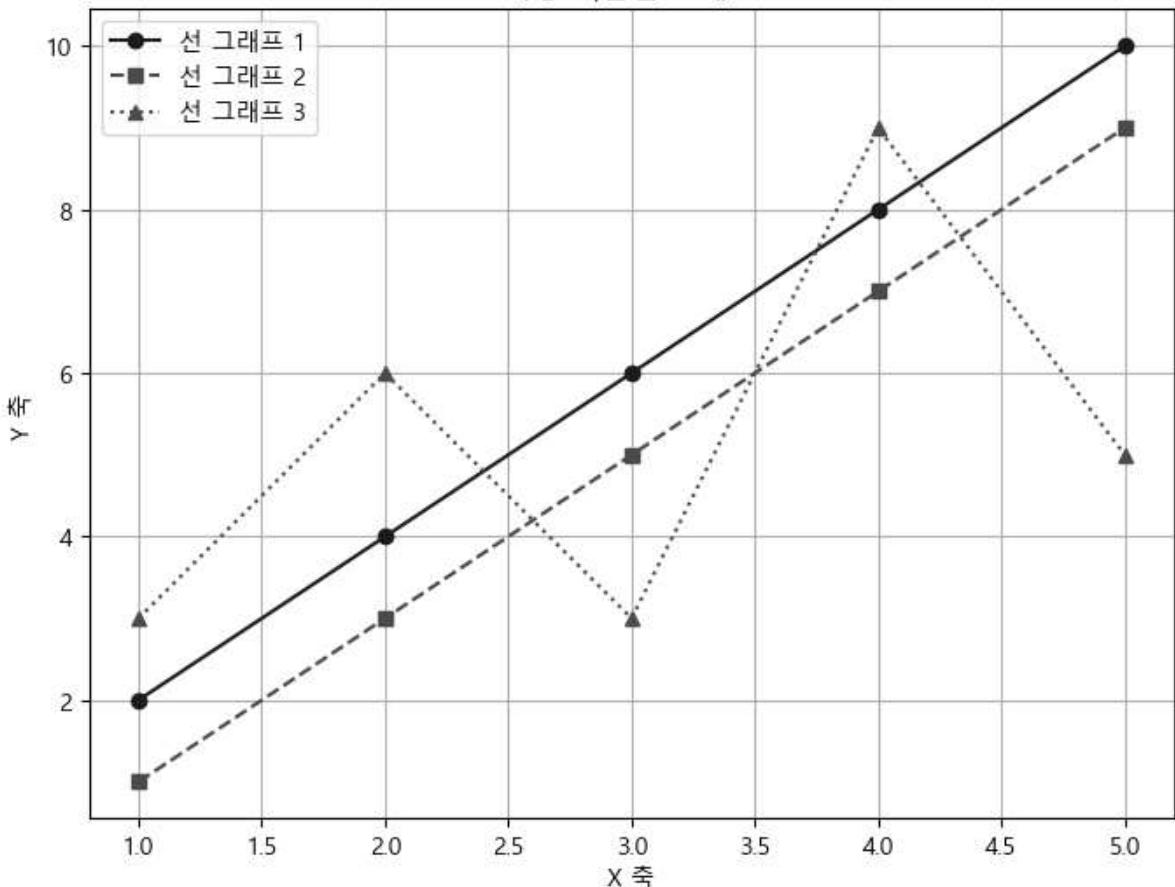
# 두 번째 꺥은선 그래프
plt.plot(x, y2, label='선 그래프 2', marker='s', linestyle='--', color='g')

# 세 번째 꺥은선 그래프
plt.plot(x, y3, label='선 그래프 3', marker='^', linestyle=':', color='r')

# 그래프 제목 및 레이블 설정
plt.title('다중 꺥은선 그래프')
plt.xlabel('X 축')
plt.ylabel('Y 축')
plt.legend() # 범례 표시

# 그래프 표시
plt.grid(True) # 그리드 표시 (선택사항)
plt.show()
```

다중 꺽은선 그래프



```
In [45]: ##다중 그래프 그리기
import matplotlib.pyplot as plt

# 그래프 그리기
plt.figure(figsize=(8, 6)) # 그림 크기 설정 (선택사항)

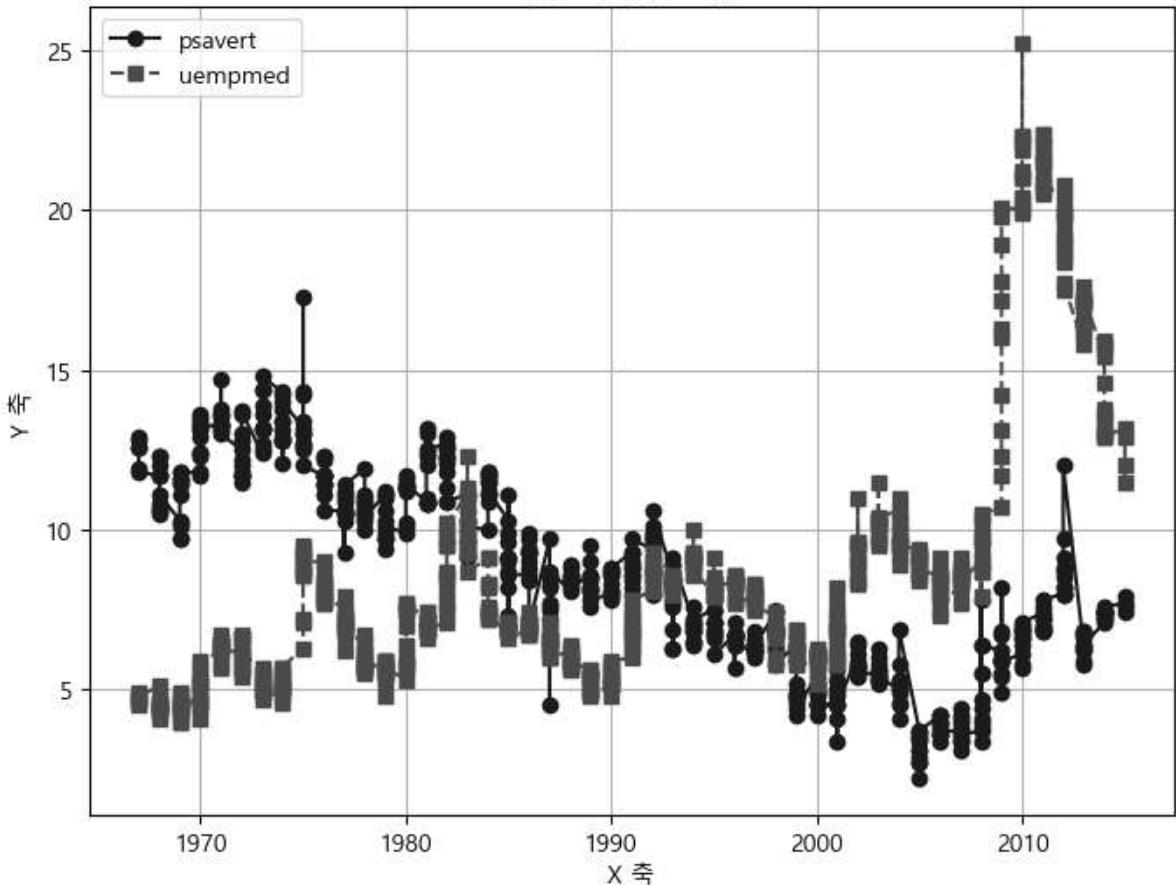
# 첫 번째 꺽은선 그래프
plt.plot(economics['year'], economics['psavert'], label='psavert', marker='o', lines

# 두 번째 꺽은선 그래프
plt.plot(economics['year'], economics['uempmed'], label='uempmed', marker='s', lines

# 그래프 제목 및 레이블 설정
plt.title('다중 꺽은선 그래프')
plt.xlabel('X 축')
plt.ylabel('Y 축')
plt.legend() # 범례 표시

# 그래프 표시
plt.grid(True) # 그리드 표시 (선택사항)
plt.show()
```

다중 겹은선 그래프



In []:

[08-5] 상자 그림 - 집단 간 분포 차이 표현하기

[] boxplot()으로 상자 그림 만들기

<boxplot 해석>

- > Outer 극단치
- > Maximum 극단치 경계 최대값
- > Upper Quartile 상위 25%(Q3 : 3사분위)
- > Median 중위(Q2 : 2사분위)
- > LowerR Quartile 하위 25%(Q1 : 1사분위)
- > Minimum 극단치 경계 최솟값

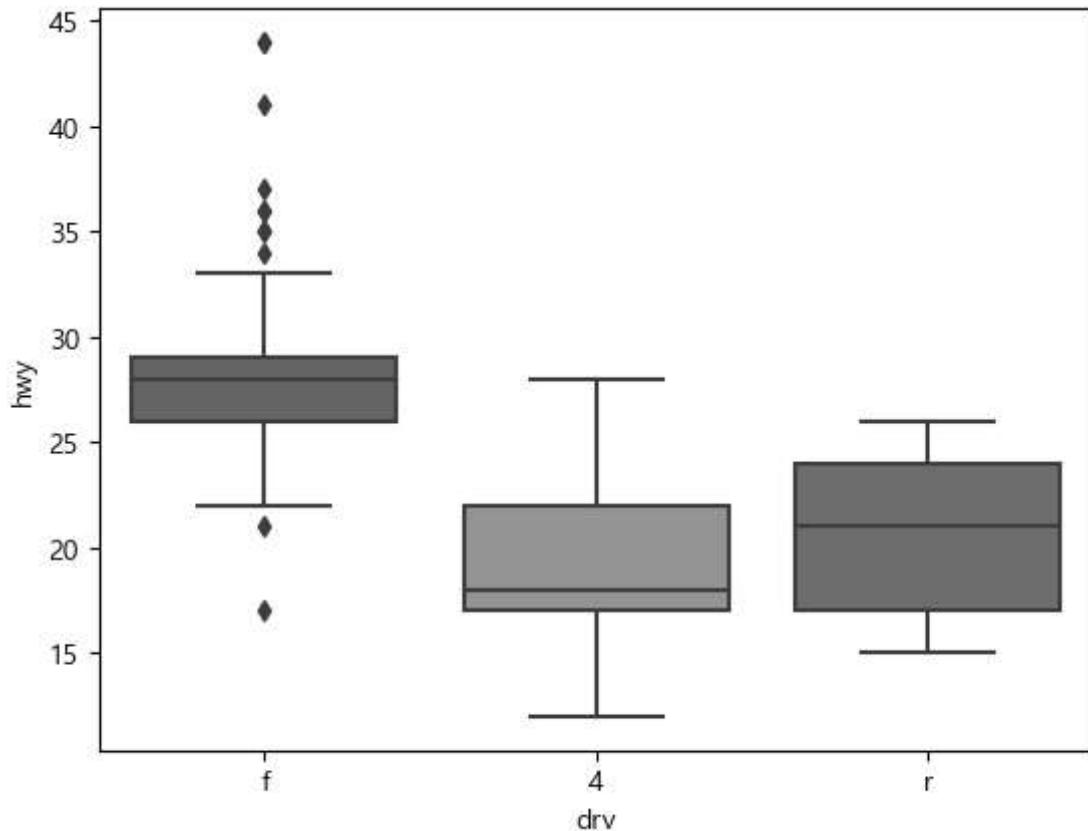
<https://img1.daumcdn.net/thumb/R1280x0/?scode=mtistory2&fname=https%3A%2F%2Ft1.daumcdn.net%2Fcfile%2Ftistory%2F99D3C43359>



In [46]: ## boxplot()으로 상자 그림 만들기

```
import seaborn as sns  
sns.boxplot(data = mpg, x = 'drv', y = 'hwy')
```

```
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 isin  
stance(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 isin  
stance(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 isin  
stance(dtype, CategoricalDtype) instead  
... if pd.api.types.is_categorical_dtype(vector):  
<Axes: xlabel='drv', ylabel='hwy'>  
Out[46]:
```



```
In [ ]:
```

boxplot() 구하기

(1) 1사분위수, 3사분위수 구하기

```
In [67]: ## 1사분위수 구하기  
pct25 = mpg['hwy'].quantile(.25)  
pct25
```

```
Out[67]: 18.0
```

```
In [68]: ## 3사분위수 구하기  
pct75 = mpg['hwy'].quantile(.75)  
pct75
```

```
Out[68]: 27.0
```

(2) IQR 구하기

> IQR(사분위 범위)는 1사분위수와 3사분위수의 거리

```
In [70]: ## IQR 구하기  
iqr = pct75 - pct25  
iqr
```

```
Out[70]: 9.0
```

(3) 하한, 상한 구하기

> 하한 : $Q1 - 1.5 \times IQR$

> 상한 : $Q3 + 1.5 \times IQR$

```
In [71]: pct25 - 1.5 * iqr # 하한
```

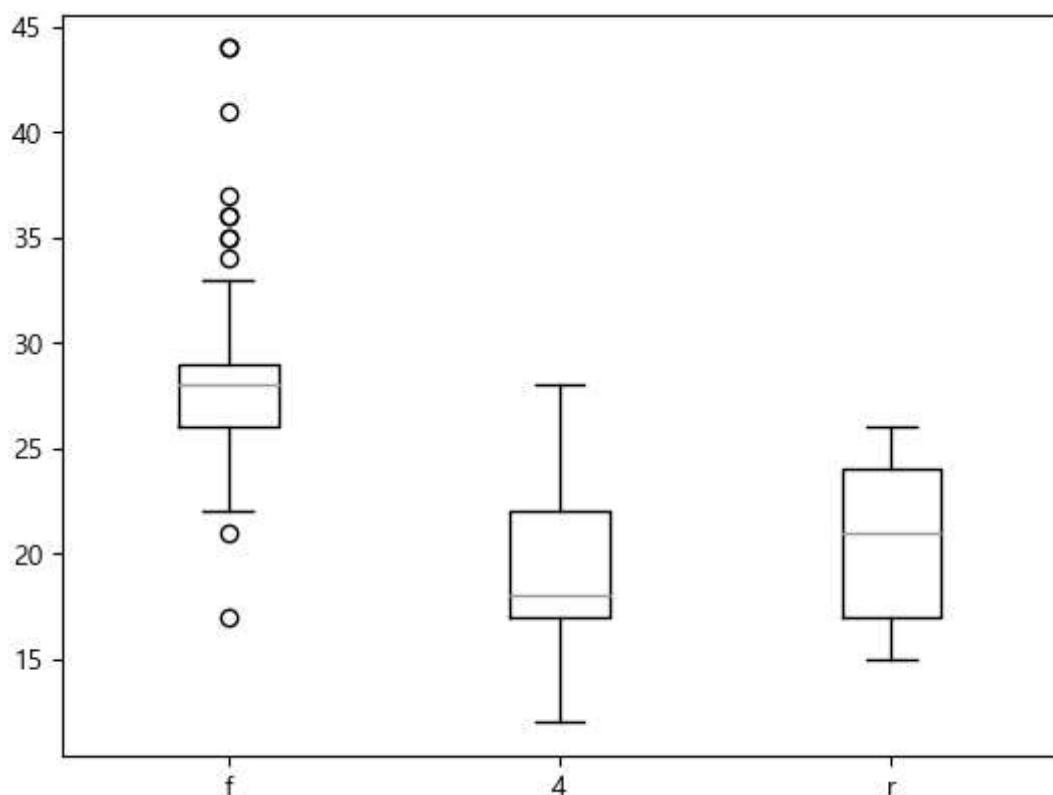
```
Out[71]: 4.5
```

```
In [72]: pct75 + 1.5 * iqr # 상한
```

```
Out[72]: 40.5
```

> matplotlib 라이브러리로 그리기

```
In [61]: ## matplotlib 라이브러리로 그래프 그리기  
import matplotlib.pyplot as plt  
  
hwy_f = mpg[mpg['drv'] == 'f']['hwy']  
hwy_4 = mpg[mpg['drv'] == '4']['hwy']  
hwy_r = mpg[mpg['drv'] == 'r']['hwy']  
hwy_drv = [hwy_f, hwy_4, hwy_r]  
  
plt.boxplot(hwy_drv, labels = ['f', '4', 'r'])  
plt.show()
```



In []:

[08-6] 점유 비율 그림 - 집단 내 점유 비율 표현하기

[] plot.barh()으로 그리기

(1) 점유 비율 데이터 프레임 구성하기

```
In [151...]  
## 'manufacturer'별 'category' 점유 비율(proportion) 구하기  
df_cat = mpg.groupby('manufacturer', as_index = False)[  
          ['category']].value_counts(normalize = True)  
df_cat
```

Out[151]:

	manufacturer	category	proportion
0	audi	compact	0.833333
1	audi	midsize	0.166667
2	chevrolet	suv	0.473684
3	chevrolet	midsize	0.263158
4	chevrolet	2seater	0.263158
5	dodge	pickup	0.513514
6	dodge	minivan	0.297297
7	dodge	suv	0.189189
8	ford	subcompact	0.360000
9	ford	suv	0.360000
10	ford	pickup	0.280000
11	honda	subcompact	1.000000
12	hyundai	midsize	0.500000
13	hyundai	subcompact	0.500000
14	jeep	suv	1.000000
15	land rover	suv	1.000000
16	lincoln	suv	1.000000
17	mercury	suv	1.000000
18	nissan	midsize	0.538462
19	nissan	suv	0.307692
20	nissan	compact	0.153846
21	pontiac	midsize	1.000000
22	subaru	suv	0.428571
23	subaru	compact	0.285714
24	subaru	subcompact	0.285714
25	toyota	compact	0.352941
26	toyota	suv	0.235294
27	toyota	pickup	0.205882
28	toyota	midsize	0.205882
29	volkswagen	compact	0.518519
30	volkswagen	midsize	0.259259
31	volkswagen	subcompact	0.222222

In [153...]

```
## 점유 비율(proportion)을 백분위로 환산하기
df_cat = df_cat.assign(proportion = df_cat['proportion'] * 100).round(1)
df_cat
```

Out[153]:

	manufacturer	category	proportion
0	audi	compact	83.3
1	audi	midsize	16.7
2	chevrolet	suv	47.4
3	chevrolet	midsize	26.3
4	chevrolet	2seater	26.3
5	dodge	pickup	51.4
6	dodge	minivan	29.7
7	dodge	suv	18.9
8	ford	subcompact	36.0
9	ford	suv	36.0
10	ford	pickup	28.0
11	honda	subcompact	100.0
12	hyundai	midsize	50.0
13	hyundai	subcompact	50.0
14	jeep	suv	100.0
15	land rover	suv	100.0
16	lincoln	suv	100.0
17	mercury	suv	100.0
18	nissan	midsize	53.8
19	nissan	suv	30.8
20	nissan	compact	15.4
21	pontiac	midsize	100.0
22	subaru	suv	42.9
23	subaru	compact	28.6
24	subaru	subcompact	28.6
25	toyota	compact	35.3
26	toyota	suv	23.5
27	toyota	pickup	20.6
28	toyota	midsize	20.6
29	volkswagen	compact	51.9
30	volkswagen	midsize	25.9
31	volkswagen	subcompact	22.2

(2) 피봇 데이터 프레임 만들기

In [149...]

```
### 분석 데이터 피봇
pivot_df = df_cat[['manufacturer', 'category', 'proportion']].pivot(index = 'manufac
pivot_df
```

Out[149]:

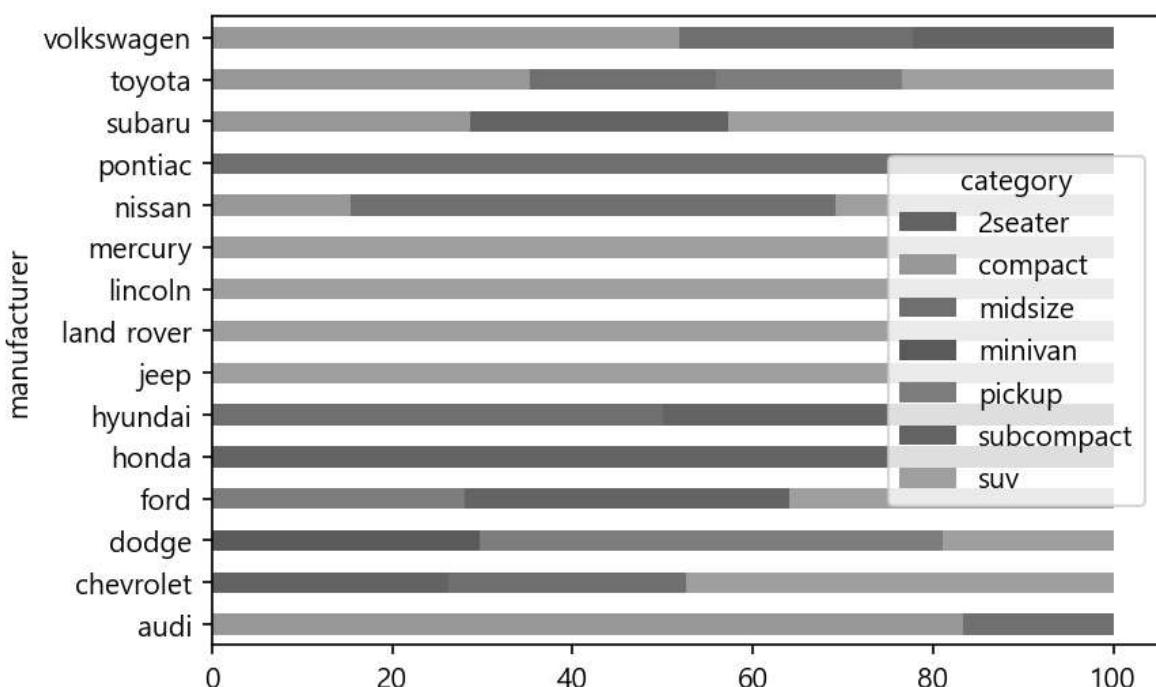
	category	2seater	compact	midsize	minivan	pickup	subcompact	suv
manufacturer								
audi		NaN	83.3	16.7	NaN	NaN	NaN	NaN
chevrolet		26.3	NaN	26.3	NaN	NaN	NaN	47.4
dodge		NaN	NaN	NaN	29.7	51.4	NaN	18.9
ford		NaN	NaN	NaN	NaN	28.0	36.0	36.0
honda		NaN	NaN	NaN	NaN	NaN	100.0	NaN
hyundai		NaN	NaN	50.0	NaN	NaN	50.0	NaN
jeep		NaN	NaN	NaN	NaN	NaN	NaN	100.0
land rover		NaN	NaN	NaN	NaN	NaN	NaN	100.0
lincoln		NaN	NaN	NaN	NaN	NaN	NaN	100.0
mercury		NaN	NaN	NaN	NaN	NaN	NaN	100.0
nissan		NaN	15.4	53.8	NaN	NaN	NaN	30.8
pontiac		NaN	NaN	100.0	NaN	NaN	NaN	NaN
subaru		NaN	28.6	NaN	NaN	NaN	28.6	42.9
toyota		NaN	35.3	20.6	NaN	20.6	NaN	23.5
volkswagen		NaN	51.9	25.9	NaN	NaN	22.2	NaN

(3) 피봇 데이터 프레임으로 그래프 그리기

In [150...]:

```
### 비율 그래프
pivot_df.plot.barh(stacked = True)
```

Out[150]:



In []:

[08-7] 히트맵 그림 - 상관 관계 정도 표현하기

[] heatmap()으로 그리기

(1) 상관 랭렬용 데이터 추출하기

In [156...]

```
## mpg 데이터 불러오기
import pandas as pd
mpg = pd.read_csv('mpg.csv')
mpg
```

Out[156]:

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	category
0	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
1	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
2	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
3	audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
4	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
...
229	volkswagen	passat	2.0	2008	4	auto(s6)	f	19	28	p	midsize
230	volkswagen	passat	2.0	2008	4	manual(m6)	f	21	29	p	midsize
231	volkswagen	passat	2.8	1999	6	auto(l5)	f	16	26	p	midsize
232	volkswagen	passat	2.8	1999	6	manual(m5)	f	18	26	p	midsize
233	volkswagen	passat	3.6	2008	6	auto(s6)	f	17	26	p	midsize

234 rows × 11 columns

In [163...]

```
## 상관 행렬용 데이터 추출
df_mpg = mpg[['cyl', 'cty', 'hwy']]
```

Out[163]:

	cyl	cty	hwy
0	4	18	29
1	4	21	29
2	4	20	31
3	4	21	30
4	6	16	26
...
229	4	19	28
230	4	21	29
231	6	16	26
232	6	18	26
233	6	17	26

234 rows × 3 columns

In [164...]

```
## 연비 평균 열('cty_hwy') 추가  
df_mpg['cty_hwy'] = (df_mpg['cty'] + df_mpg['hwy']) / 2  
df_mpg
```

C:\Users\ADMIN\AppData\Local\Temp\ipykernel_21028\3298684864.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
... df_mpg['cty_hwy'] = (df_mpg['cty'] + df_mpg['hwy']) / 2

Out[164]:

	cyl	cty	hwy	cty_hwy
0	4	18	29	23.5
1	4	21	29	25.0
2	4	20	31	25.5
3	4	21	30	25.5
4	6	16	26	21.0
...
229	4	19	28	23.5
230	4	21	29	25.0
231	6	16	26	21.0
232	6	18	26	22.0
233	6	17	26	21.5

234 rows × 4 columns

(2) 상관행렬 만들기

In [165]:

```
## 상관행렬 만들기
df_corr = df_mpg.corr() # 상관행렬 만들기
df_corr = round(df_corr, 2) # 소수점 2자리까지 반올림
df_corr
```

Out[165]:

	cyl	cty	hwy	cty_hwy
cyl	1.00	-0.81	-0.76	-0.79
cty	-0.81	1.00	0.96	0.98
hwy	-0.76	0.96	1.00	0.99
cty_hwy	-0.79	0.98	0.99	1.00

(3) 상관 그래프 그리기

In [166]:

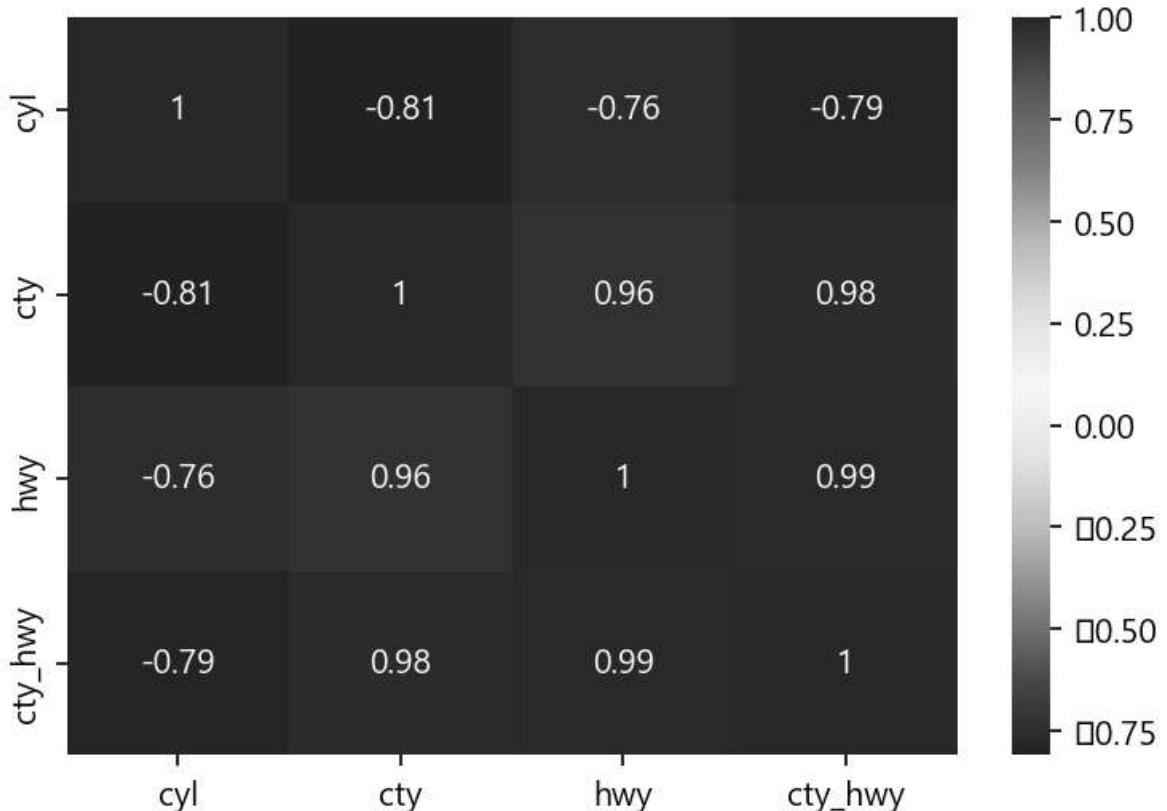
```
## 히트맵 만들기
import seaborn as sns
sns.heatmap(df_corr,
            annot = True, # 상관계수 표시
            cmap = 'RdBu') # 컬러맵
```

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

fig.canvas.draw()

Out[166]:

```
C:\Users\ADMIN\anaconda3\lib\site-packages\Python\core\events.py:89: UserWarning: Glyph 8722 (MINUS SIGN) missing from current font.
func(*args, **kwargs)
C:\Users\ADMIN\anaconda3\lib\site-packages\Python\core\pylabtools.py:151: UserWarning: Glyph 8722 (MINUS SIGN) missing from current font.
fig.canvas.print_figure(bytes_io, **kw)
```



(4) 하단만 표시하기

(4-1) 마스크 행렬 만들기

> 마스크 행렬은 상단 제거용 행렬

```
In [167]: ## 마스크 행렬 만들기
import numpy as np
mask = np.zeros_like(df_corr) # df_corr 크기로 0이 채워진 행렬 생성
mask
```

```
Out[167]: array([[0., 0., 0., 0.],
   ... [0., 0., 0., 0.],
   ... [0., 0., 0., 0.],
   ... [0., 0., 0., 0.]])
```

```
In [168]: ## 오른쪽 대각 행렬 값을 1로 바꾸기
mask[np.triu_indices_from(mask)] = 1
mask
```

```
Out[168]: array([[1., 1., 1., 1.],
   ... [0., 1., 1., 1.],
   ... [0., 0., 1., 1.],
   ... [0., 0., 0., 1.]])
```

(4-2) 상관관계=1인 구간 제거하기

```
In [169]: ## 상관계 = 1인 구간 제거하기
mask_new = mask[1:, :-1] # mask의 첫 행과 마지막 열 제거
corr_new = df_corr.iloc[1:, :-1] # 데이터에서 첫 행과 마지막 열 제거
corr_new
```

```
Out[169]:
```

	cyl	cty	hwy
cty	-0.81	1.00	0.96
hwy	-0.76	0.96	1.00
cty_hwy	-0.79	0.98	0.99

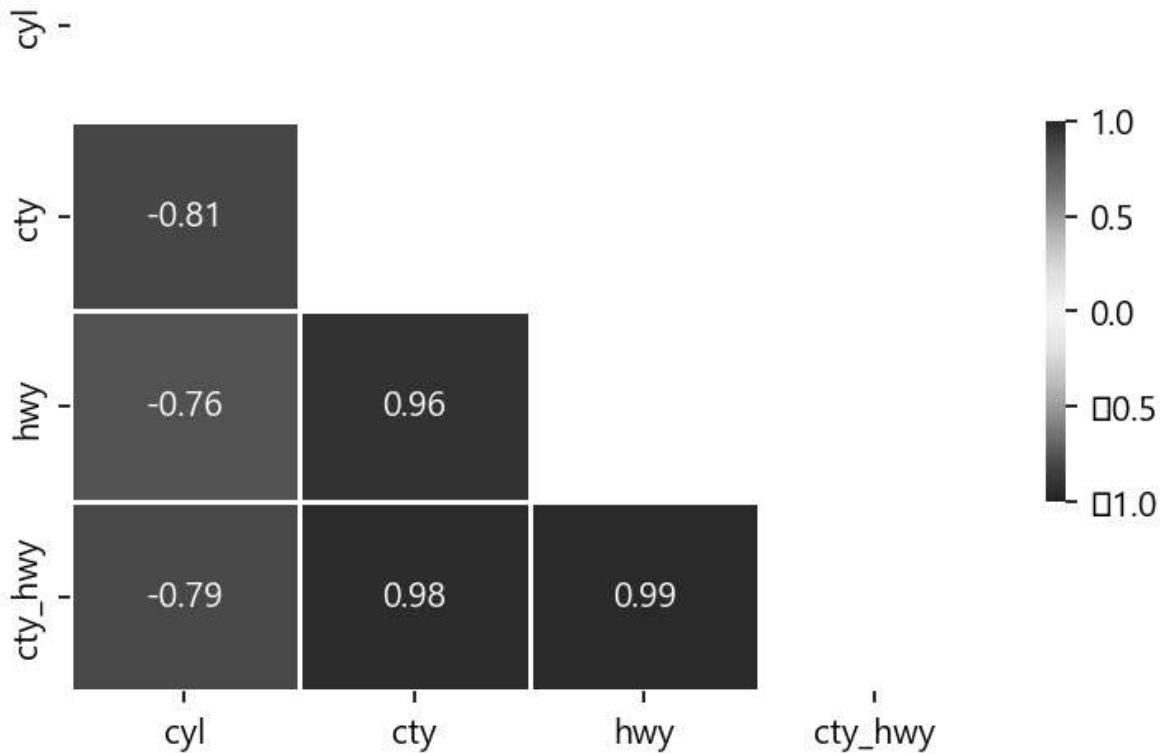
(4-3) Mask 행렬이 적용된 히트맵 그리기

```
In [170]: ## Mask가 적용된 히트맵 만들기
sns.heatmap(df_corr,
            annot = True, # 상관계수 표시
            mask = mask,
            linewidths = 0.5, # 경계 구분선 추가
            vmax = 1, # 진한 파란색으로 표현할 최대값
            vmin = -1, # 진한 붉은색으로 표현할 최대값
            cbar_kws = {'shrink': 0.5}, # 범례 크기 반으로
            cmap = 'RdBu') # 컬러맵
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\utils.py:80: UserWarning: Glyph 8
722 (\u2212) missing from current font.
... fig.canvas.draw()
```

```
Out[170]: <Axes: >
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\Python\core\events.py:89: UserWarning: G
lyph 8722 (\u2212) missing from current font.
... func(*args, **kwargs)
C:\Users\ADMIN\anaconda3\lib\site-packages\Python\core\pylabtools.py:151: UserWarni
ng: Glyph 8722 (\u2212) missing from current font.
... fig.canvas.print_figure(bytes_io, **kw)
```



In []:

[08-8] 네트워크 그래프 그리기

> pip install networkx 필요

Node와 **Edge**를 추가하고 그리기

In []: !pip install networkx

```
#> 노드와 에지 그래프 그리기
import networkx as nx
import matplotlib.pyplot as plt

# 빈 그래프 생성
G = nx.Graph()

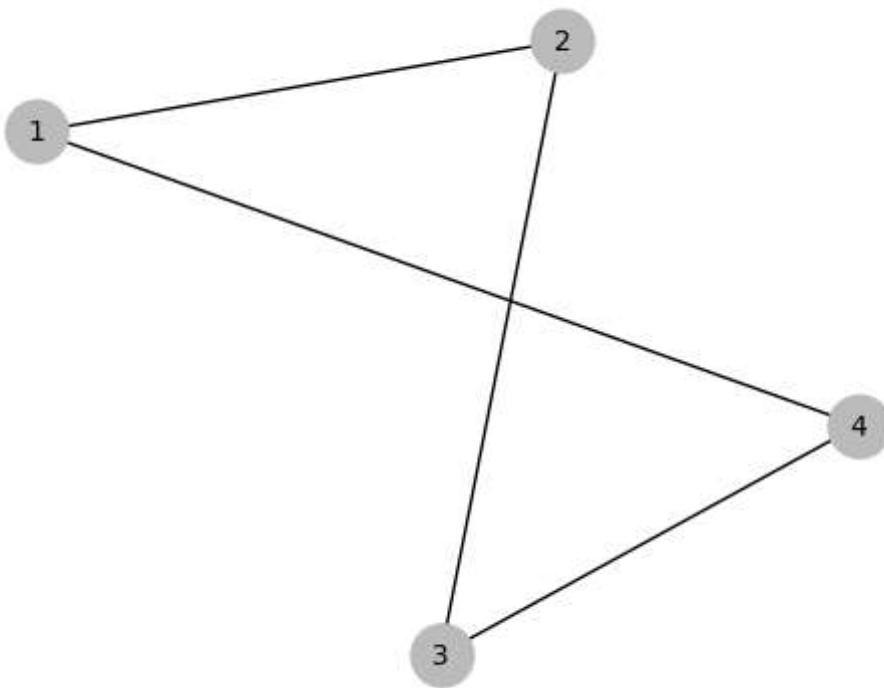
# 노드 추가
G.add_node(1)
G.add_node(2)
G.add_node(3)
G.add_node(4)

# 간선(엣지) 추가
G.add_edge(1, 2)
G.add_edge(2, 3)
G.add_edge(3, 4)
G.add_edge(4, 1)

# 그림과 축을 생성합니다.
fig, ax = plt.subplots()
```

```
# 그래프 그리기
pos = nx.spring_layout(G) # 그래프 레이아웃 정의
nx.draw(G, pos, with_labels=True, node_size=500, node_color="skyblue", font_size=10)
plt.title("네트워크 그래프 예제")
plt.show()
```

네트워크 그래프 예제



In [32]:

```
## 한글 폰트 지정하기
import matplotlib.pyplot as plt
plt.rcParams.update({'font.family' : 'Malgun Gothic'}) # 폰트, 기본값 sans-serif
```

In [59]:

```
## 방향성 그래프 그리기
import networkx as nx
import matplotlib.pyplot as plt

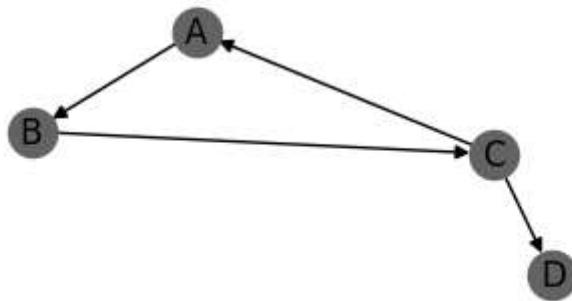
nlist = ['A', 'B', 'C', 'D', 'E']
elist = [('A', 'B'), ('B', 'C'), ('C', 'D'), ('C', 'A')]
G = nx.DiGraph()
G.add_nodes_from(nlist)
G.add_edges_from(elist)

# 그림과 축을 생성합니다.
fig, ax = plt.subplots()

# 네트워크 그래프 그리기
pos = nx.spring_layout(G) # 그래프 레이아웃 정의
nx.draw(G, pos, with_labels=True, ax=ax)
plt.title("네트워크(방향성) 그래프 예제")
plt.show()
```

네트워크(방향성) 그래프 예제

E



```
In [61]: ## 방향성 그래프 그리기 : Node size 조절하기
import networkx as nx
import matplotlib.pyplot as plt

nlist = ['A', 'B', 'C', 'D', 'E']
elist = [('A', 'B'), ('B', 'C'), ('C', 'D'), ('C', 'A')]
G = nx.DiGraph()
G.add_nodes_from(nlist)
G.add_edges_from(elist)

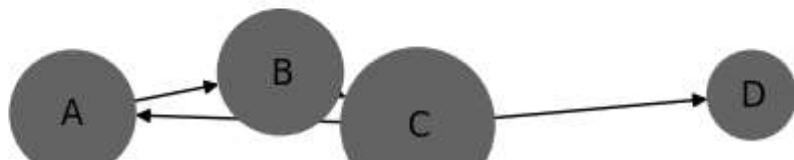
#Node size 구하기
d = dict(G.degree)
nsize = [v * 1000 for v in d.values()]

# 그림과 축을 생성합니다.
fig, ax = plt.subplots()

# 네트워크 그래프 그리기
pos = nx.spring_layout(G) # 그래프 레이아웃 정의
nx.draw(G, pos, with_labels=True, node_size = nsize, ax=ax)
plt.title("네트워크(방향성) 그래프 예제")
plt.show()
```

네트워크(방향성) 그래프 예제

E



정리하기

```
In [ ]: ### 1. 산점도  
sns.scatterplot(data = mpg, x = 'displ', y = 'hwy')  
  
# 축 제한  
sns.scatterplot(data = mpg, x = 'displ', y = 'hwy')  
    .set(xlim = [3, 6], ylim = [10, 30])  
  
# 종류별로 표식 색깔 바꾸기  
sns.scatterplot(data = mpg, x = 'displ', y = 'hwy', hue = 'drv')  
  
### 2. 막대 그래프  
  
## 평균 막대 그래프  
  
# 1단계. 평균표 만들기  
df_mpg = mpg.groupby('drv', as_index = False)  
    .agg(mean_hwy = ('hwy', 'mean'))  
  
# 2단계. 그래프 만들기  
sns.barplot(data = df_mpg, x = 'drv', y = 'mean_hwy')  
  
## 빈도 막대 그래프  
sns.countplot(data = mpg, x = 'drv')  
  
### 3. 선 그래프  
sns.lineplot(data = economics, x = 'date', y = 'unemploy')
```