

# 빅데이터 처리

-Big Data Processing



10주차

- 10월 31일 , 16:15 ~ (1시간 or 1시간 30분)
- 강사 : (주) 포스로직 , 송종현 대표
- 주제 : 성공 적인 커리어를 쌓기 위한 준비
- 7호관 522호

## ■ 프로젝트

- 주제 변경 될 경우 새로운 ppt 첨부 후 메일 (기존 ppt 삭제 x)

## ■ 계획

- 9주차 : 머신 러닝 분석
- 10주차 : 데이터 시각화
- 11주차 : 데이터 시각화
- 12주차 : 지리 정보 분석
- 13주차 : 텍스트 분석
- 14주차 : 시계열 데이터 분석
- 15주차

```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score

# 피마 인디언 당뇨병 데이터셋을 불러옵니다.
df = pd.read_csv('/content/pima-indians-diabetes3.csv')

# 세부 정보를 X로 지정합니다.
X = df.iloc[:,0:8]
# 당뇨병 여부를 Y로 지정합니다.
y = df.iloc[:,8]

al = RandomForestClassifier(n_estimators=10)
al.fit(X, y) => 필요 없음

cscore=cross_val_score(al,X,y,cv=5) # 교차 검증 k=5
print('accuracy',cscore.mean())
X.shape
```

## seaborn

- lineplot
- barplot
- scatterplot
- countplot
- violinplot, swarmplot
- boxplot

## matplotlib

- Pie chart
- Area chart

## Pandas

stacked plot

# 데이터 시각화 (Data Visualization)

인하공전 컴퓨터 정보 과

- 데이터 분석 결과를 쉽게 이해할 수 있도록 시각적으로 표현하고 전달하는 과정

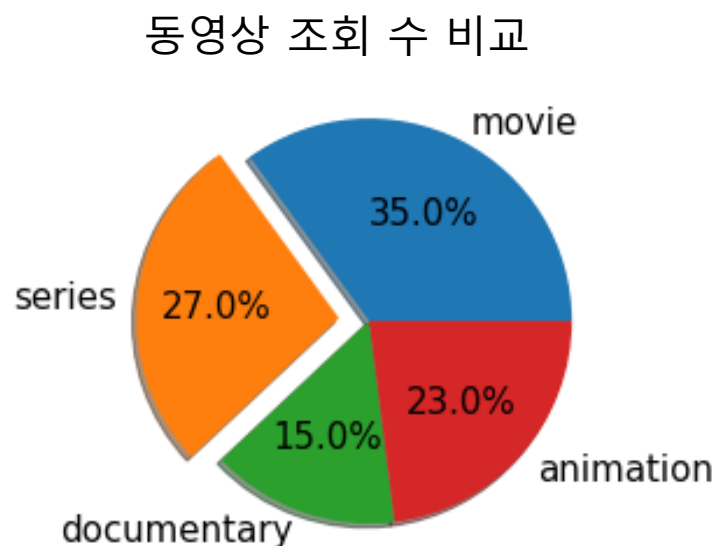
- 중요성

- 직관적인 이해를 제공
- 데이터의 구조와 패턴을 파악 하기 용이하다.
- 다양한 관점에서 데이터에 관한 통찰력을 제공한다.

---

- 동영상 조회 수를 비교 했을 때 series물의 비율은 27%였다.  
그 외에 영화는 35%, animation은 23%, documentary는 15%였다.

동영상 조회	비율
movie	35%
series	27%
animation	23%
documentary	15%





파이썬 표준 시각화 도구. pandas와 연동

<https://matplotlib.org/>



matplotlib 기반으로 한 시각화 도구, matplotlib 보다 다양한 함수 제공.  
pandas와 연동 <https://seaborn.pydata.org/>

[Cheat Sheet: Seaborn Charts | Kaggle](#)



비즈니스 인텔리전스 (Business Intelligence) 대시보드로의 역할을 하기 위해  
개발 된 도구. 의사 결정 자들이 빠르고 정확한 의사 결정을 할 수 있도록  
도와주는 도구의 모음. 인터랙션 그래프 지원.



Matplotlib의 기능 일부를 내장 하고 있음.

Import seaborn as sns

- lineplot
- barplot
- scatterplot
- countplot
- violinplot, swarmplot
- boxplot

import matplotlib.pyplot as plt

- Pie chart
- Area chart

Import pandas as pd

Pandas

- stacked plot



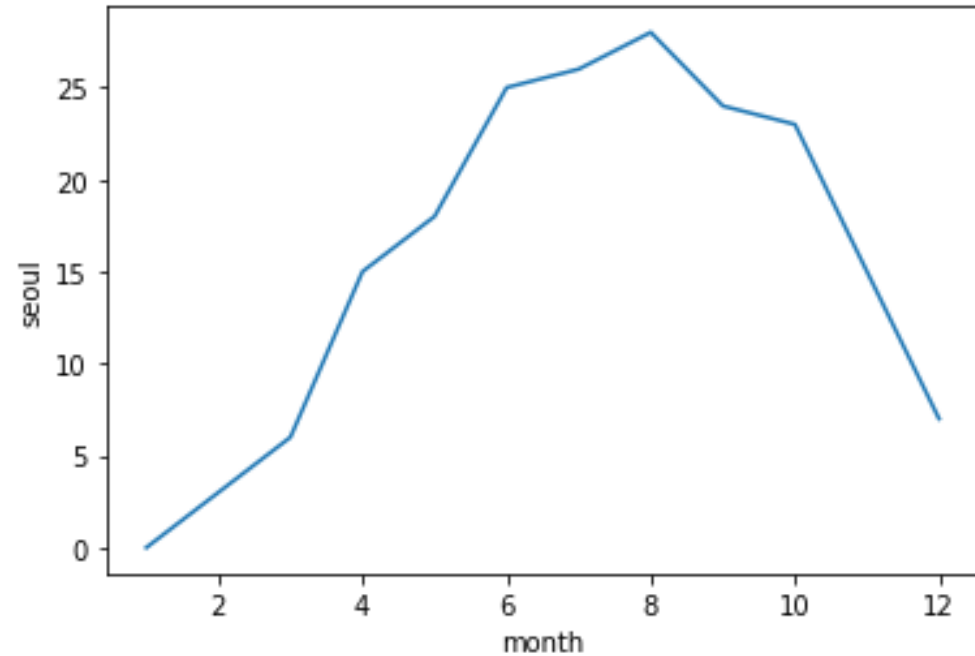
# Seaborn – lineplot

- 데이터의 변화를 line으로 표현

```
import pandas as pd
import seaborn as sns
```

```
df=pd.read_csv('/content/temperature.csv')
print(df)
sns.lineplot(data=df,x='month',y='seoul')
```

	month	seoul	busan
0	1	0	5
1	2	3	8
2	3	6	9
3	4	15	20
4	5	18	23
5	6	25	27
6	7	26	28
7	8	28	34
8	9	24	26
9	10	23	25
10	11	15	20
11	12	7	10

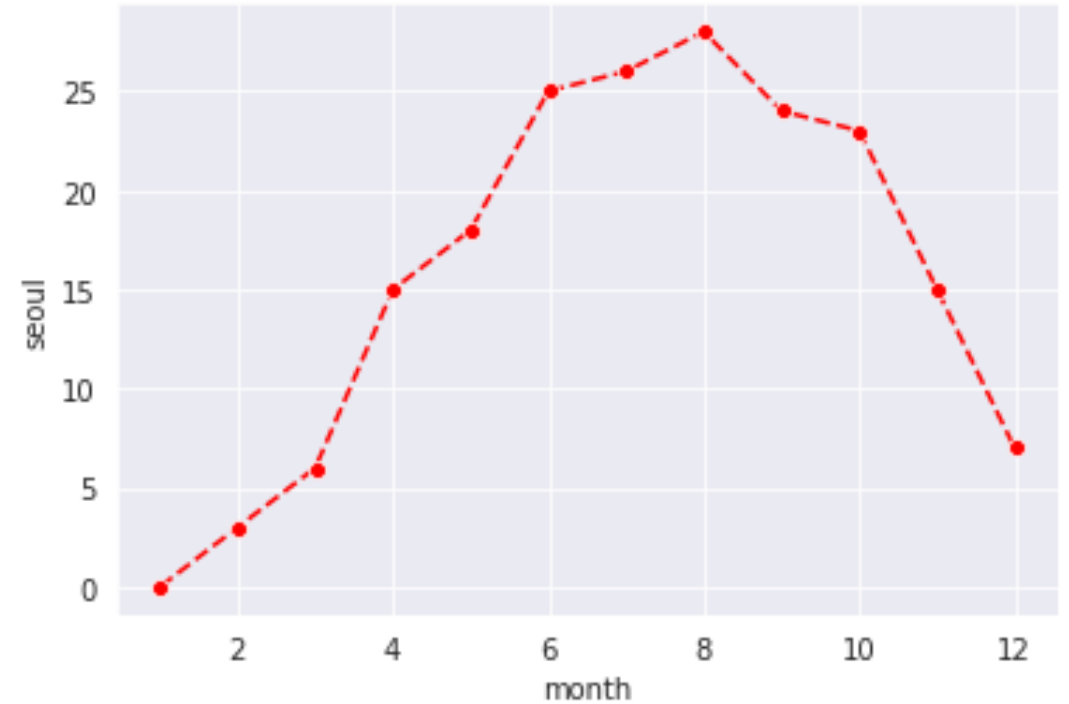


# Seaborn – lineplot

인하공전 컴퓨터 정보 과

```
import pandas as pd  
import seaborn as sns
```

```
df=pd.read_csv('/content/temperature.csv')  
sns.set_style('darkgrid') # option: whitegrid, white, dark  
sns.lineplot(data=df,x='month',y='seoul',marker='o',color  
='r',linestyle='--')
```



# Matplotlib cheatsheet

인하공전 컴퓨터 정보 과

API

## Lines

API

linestyle or ls

—— "—" :— "—." (0, (0.01, 2))

capstyle or dash\_capstyle

— "butt" — "round" — "projecting"

API

## Markers

API

API  
asform

API

o O s p X \* p D < > ^ v  
'.' 'o' 's' 'p' 'X' '\*' 'p' 'D' '<' '>' '^' 'v'  
Y Y Y Y + X | — < > ^ v  
'1' '2' '3' '4' '+' 'x' '|' '—' 4 5 6 7  
♠ ♣ ♥ ♦ → ← ↑ ↓ ○ ○ ○ ○  
'\$♠\$' '\$♣\$' '\$♥\$' '\$♦\$' '\$→\$' '\$←\$' '\$↑\$' '\$↓\$' '\$○\$' '\$○\$' '\$○\$' '\$○\$'  
markevery  
○ ○ ○ ○ ○ 10  
○ — ○ [0, -1]  
— ○ ○ ○ ○ (25, 5)  
○ — ○ [0, 25, -1]

from matplotlib

ax.[xy]ax

ticker.Nu

ticker.Fi  
zero

ticker.Fu  
[0.00] [0

ticker.Fo  
>0< ;

ticker.Sc  
0

ticker.St  
0.0

ticker.Pe  
0% ;

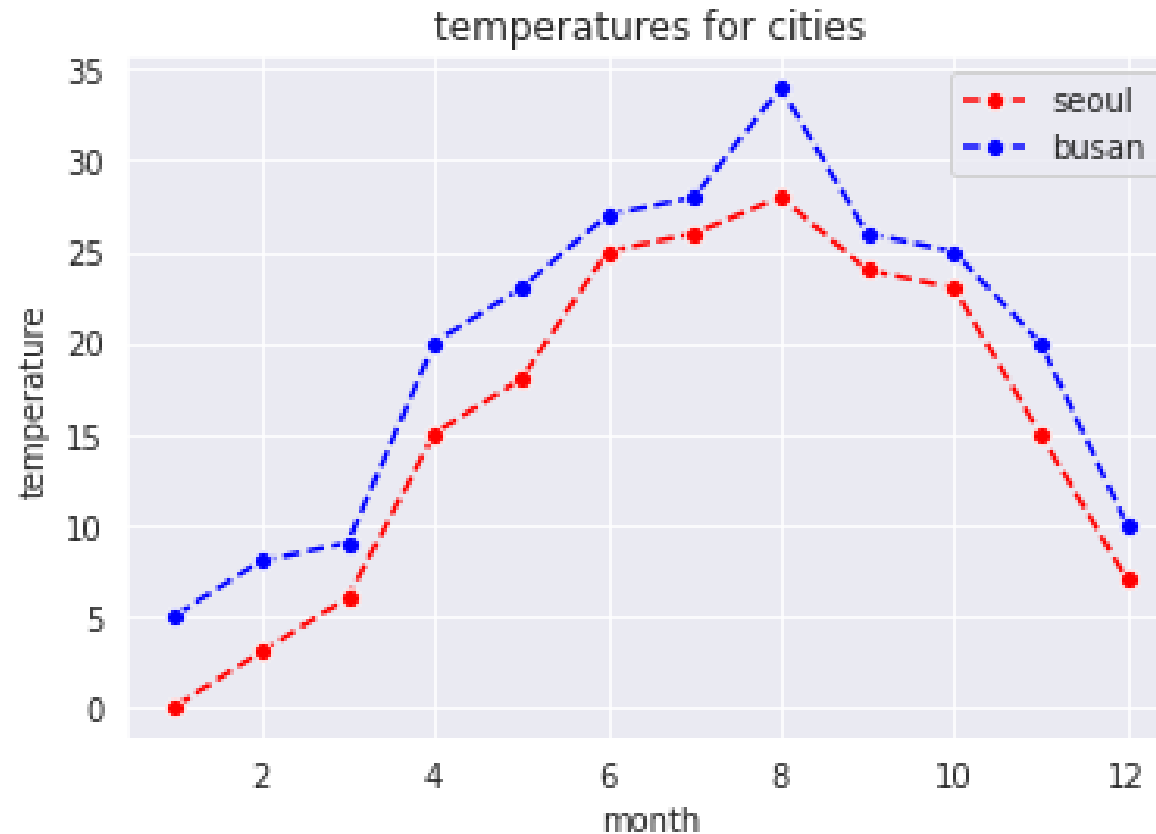
## Ornam

ax.leg  
handles,

# Seaborn – lineplot

```
sns.set_style('darkgrid') # option: whitegrid, white, dark
ax=sns.lineplot(data=df,x='month',y='seoul',marker='o',color='r',linestyle='--',label='seoul')
sns.lineplot(data=df,x='month',y='busan',marker='o',color='b',linestyle='--',label='busan')
ax.set(xlabel='month', ylabel='temperature',title='temperatures for cities')
```

	month	seoul	busan
0	1	0	5
1	2	3	8
2	3	6	9
3	4	15	20
4	5	18	23
5	6	25	27
6	7	26	28
7	8	28	34
8	9	24	26
9	10	23	25
10	11	15	20
11	12	7	10



# Tip 데이터

인하공전 컴퓨터 정보 과

```
import pandas as pd  
import seaborn as sns
```

```
tips=sns.load_dataset("tips")  
tips
```

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns

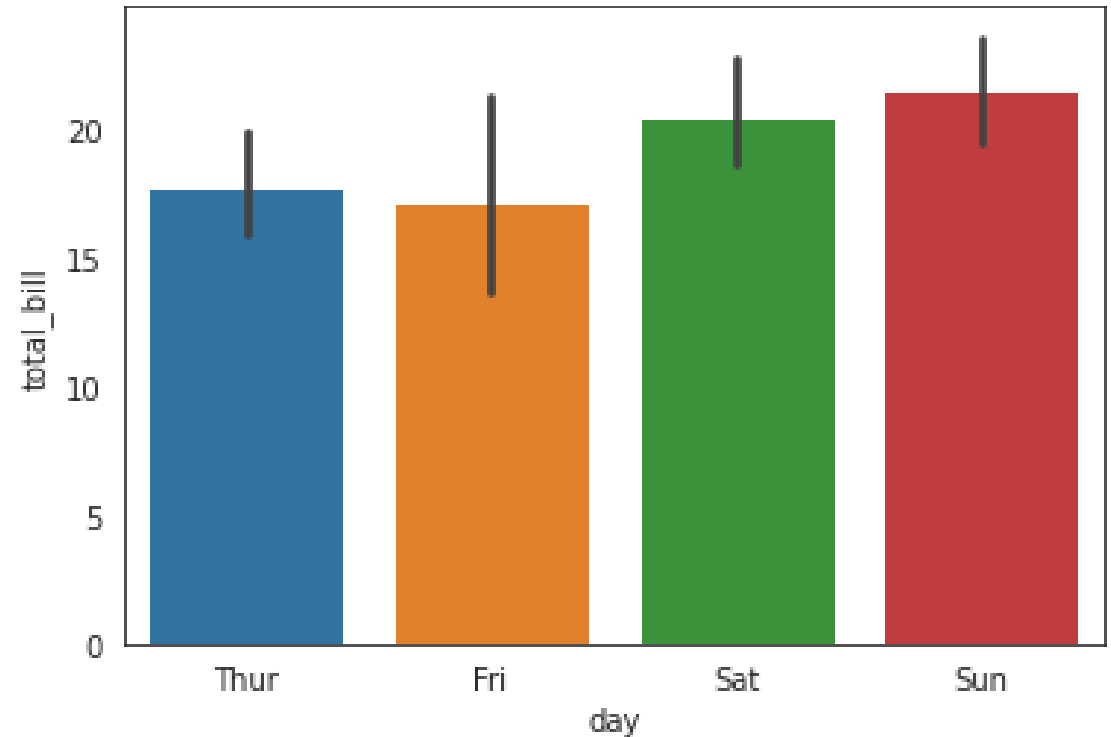
# Seaborn – barplot

- 데이터의 개수를 표현하거나 연속 값을 경우는 해당 값들의 평균값을 나타내줌

```
tips=sns.load_dataset("tips")
```

```
sns.barplot(data=tips,x='day',y='total_bill')
```

요일 별 계산 금액

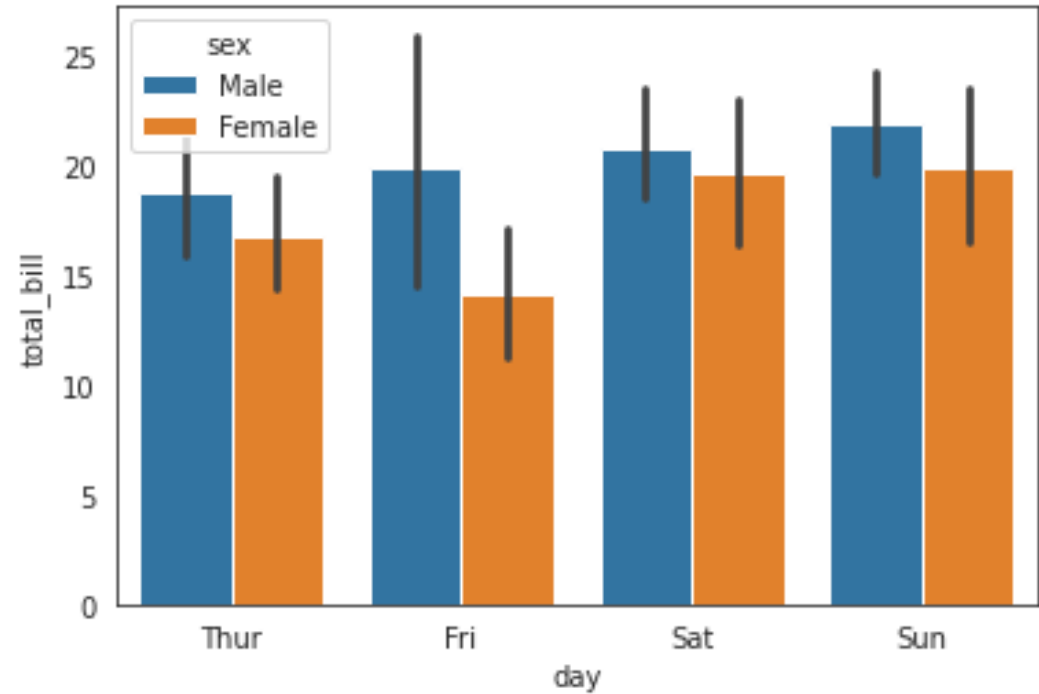


# Seaborn – barplot

```
tips=sns.load_dataset("tips")
```

```
sns.barplot(data=tips,x='day',y='total_bill',hue='sex')
```

Hue : 하위 분류



요일 별 계산 금액, 하위 분류: 성별

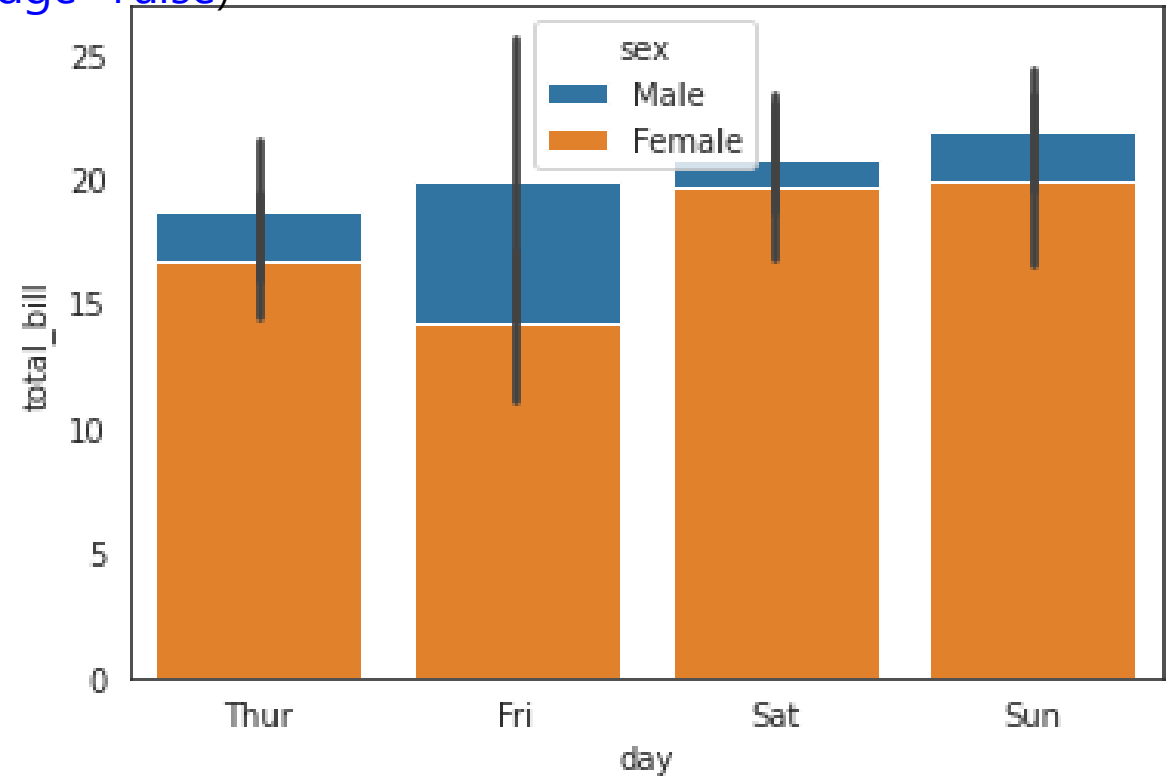
# Seaborn – barplot

```
tips=sns.load_dataset("tips")
```

```
sns.barplot(data=tips,x='day',y='total_bill',hue='sex',dodge=False)
```

dodge=False : 겹치게

요일 별 계산 금액, 하위 분류: 성별



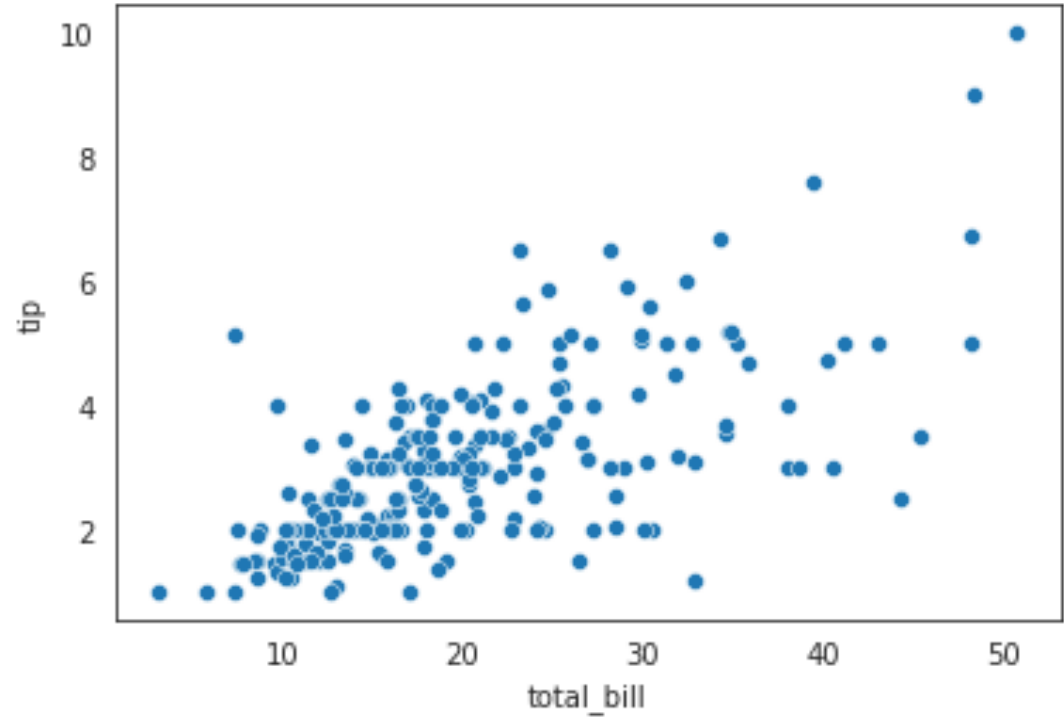


# Seaborn – scatterplot

- 데이터의 분포를 2차원 평면에 표현

```
tips=sns.load_dataset("tips")
```

```
sns.scatterplot(data=tips,x='total_bill',y='tip')
```



전체 금액과 팁과의 관계 분포

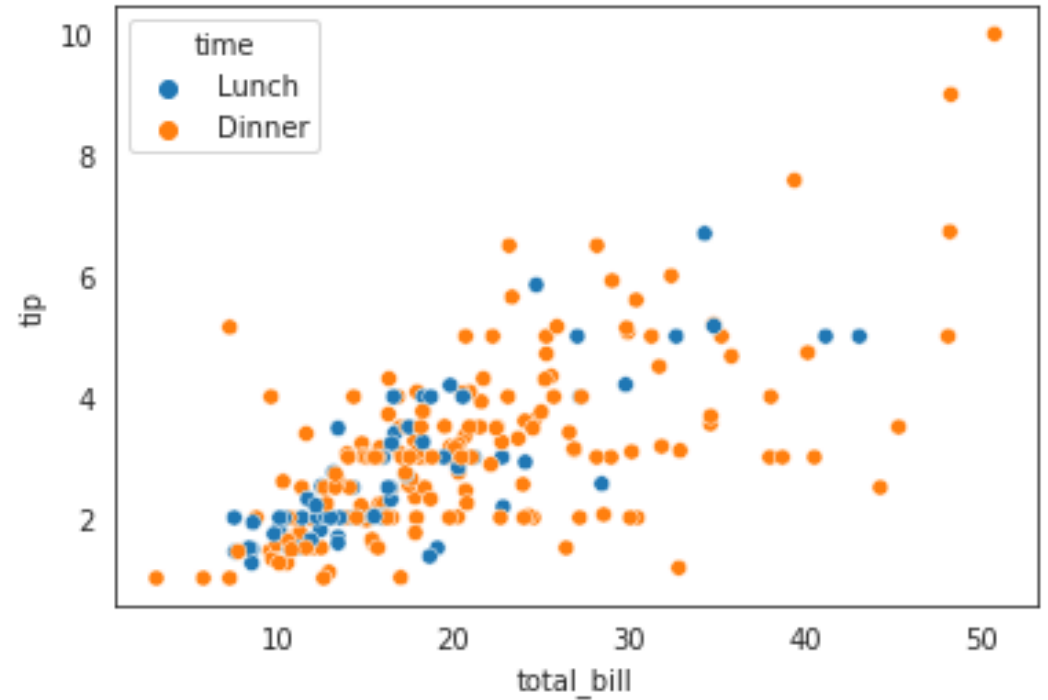
# Seaborn – scatterplot

- 데이터의 분포를 2차원 평면에 표현

```
tips=sns.load_dataset("tips")
```

```
sns.scatterplot(data=tips,x='total_bill',y='tip', hue='time')
```

Hue : 하위 분류



전체 금액과 팁과의 관계 분포, 하위 분류 time

# Seaborn – scatterplot

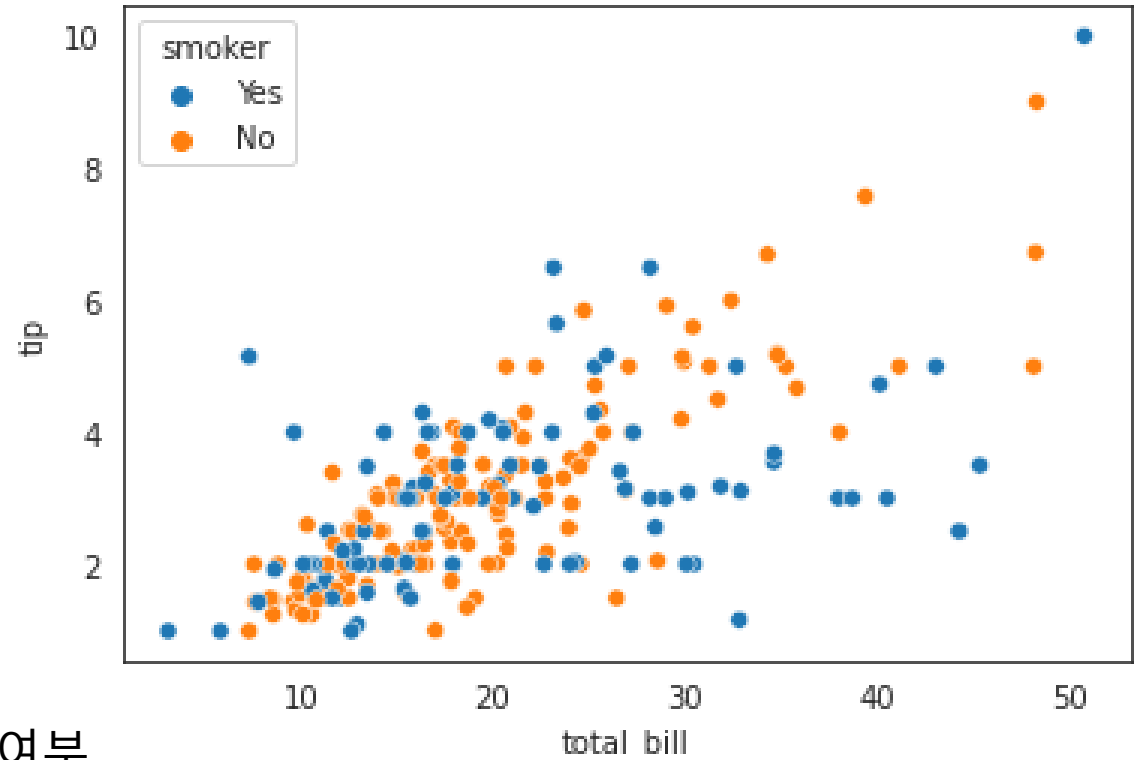
- 데이터의 분포를 2차원 평면에 표현

```
tips=sns.load_dataset("tips")
```

```
sns.scatterplot(data=tips,x='total_bill',y='tip',  
hue='smoker')
```

hue : 하위 분류

전체 금액과 팁과의 관계 분포, 하위 분류 흡연 여부

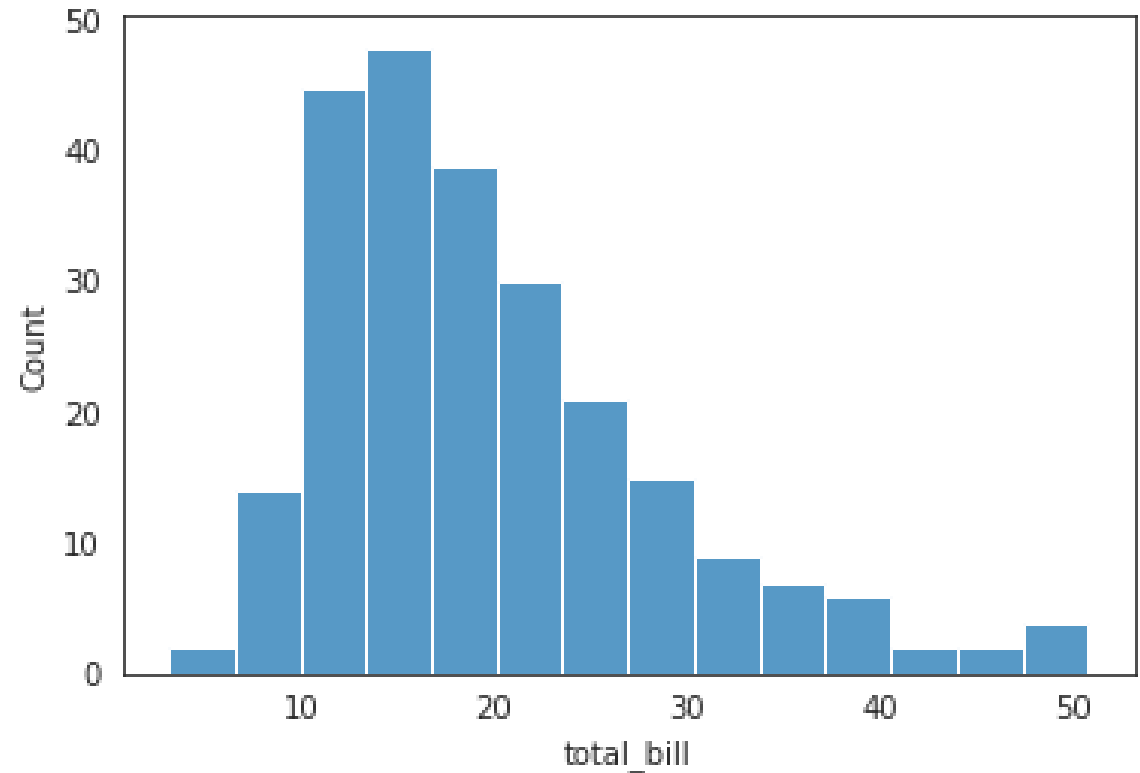


# Seaborn – histplot

- 히스토그램은 데이터의 분포를 표현하는 그래프.
- 구간별 해당 count를 표현

```
tips=sns.load_dataset("tips")
```

```
sns.histplot(data=tips,x='total_bill')
```

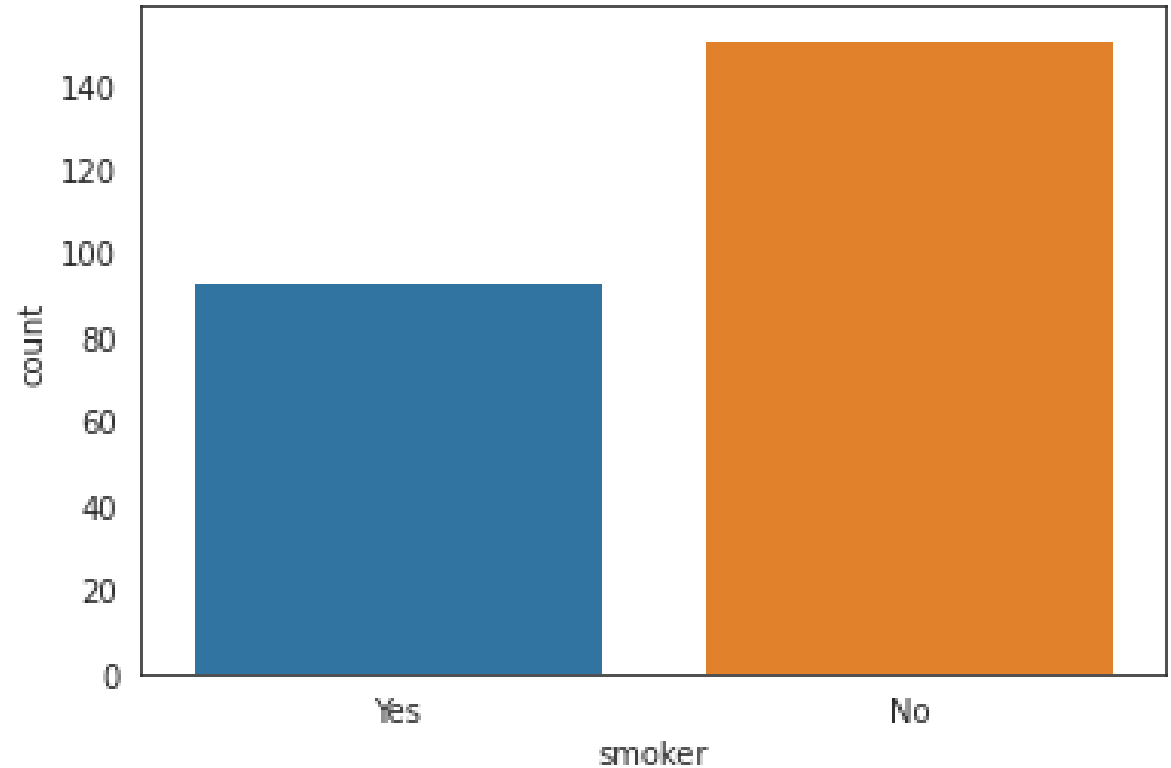


# Seaborn – countplot

- 범주형 데이터의 항목별 개수를 그래프로 표현

```
tips=sns.load_dataset("tips")
```

```
sns.countplot(data=tips,x='smoker')
```



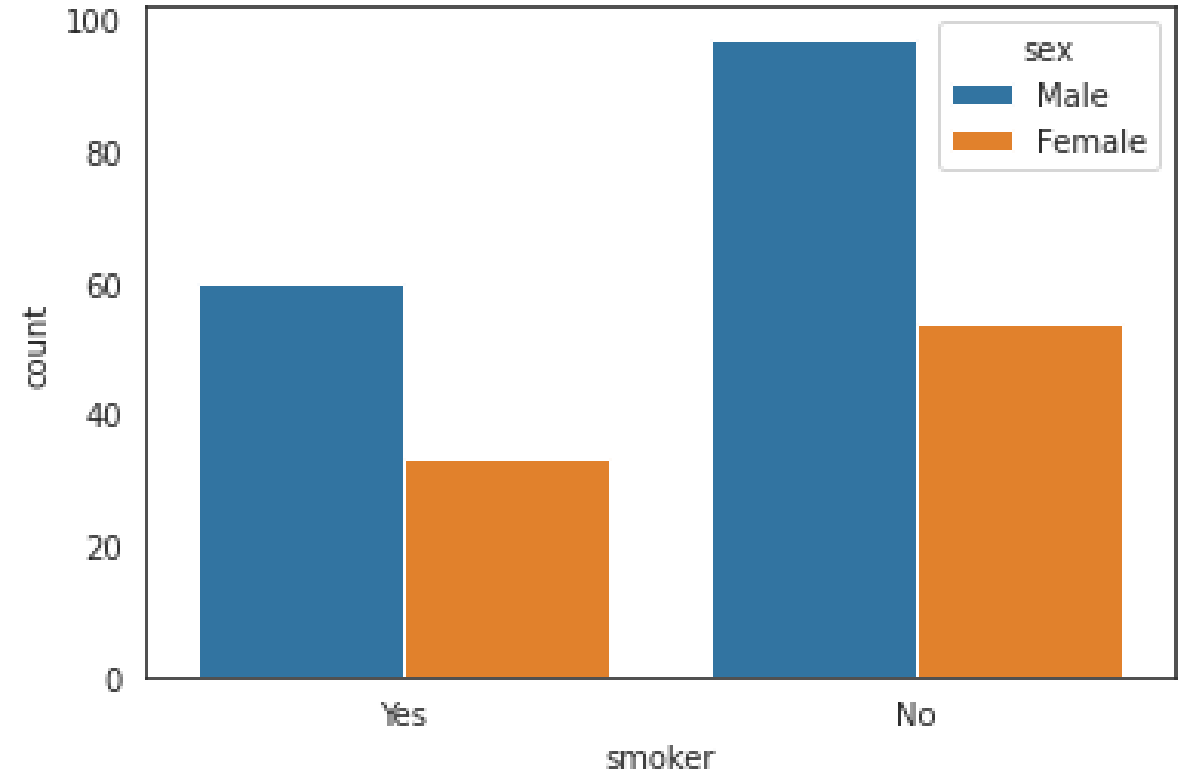
흡연 유무의 따른 수

# Seaborn – countplot

- 범주형 데이터의 항목별 개수를 그래프로 표현

```
tips=sns.load_dataset("tips")
```

```
sns.countplot(data=tips,x='smoker',hue='sex')
```



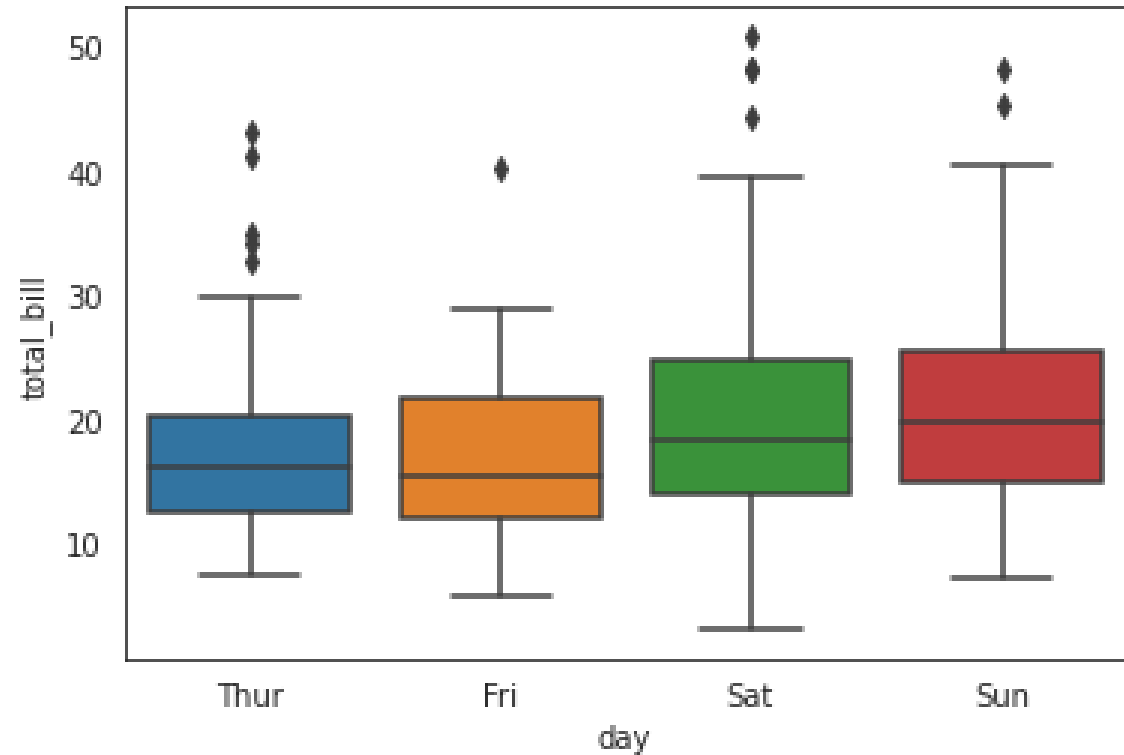
흡연 유무의 따른 수: 하위 분류 : 성별

# Seaborn – boxplot

- 데이터의 분포를 나타내면서 밀집 정도 표현
- 25%~75%까지의 데이터를 박스 형태에 위치

```
tips=sns.load_dataset("tips")
```

```
sns.boxplot(data=tips, x='day',y='total_bill')
```



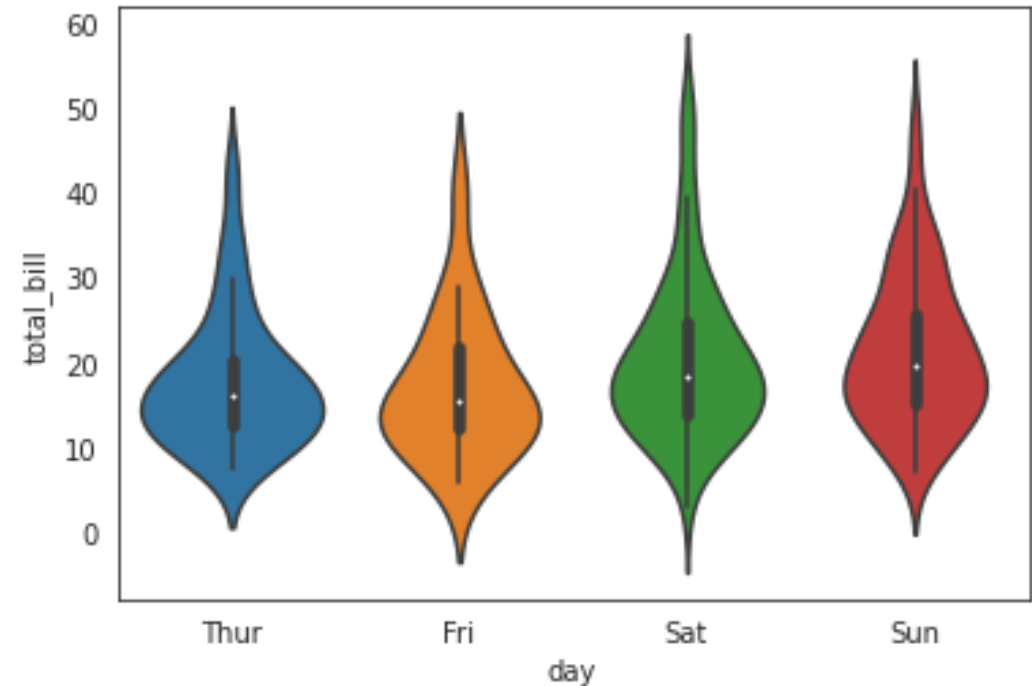
요일별 전체 금액의 분포

# Seaborn – violinplot

- 데이터의 분포를 바이올린과 비슷한 형태로 보여주는 그래프

```
tips=sns.load_dataset("tips")
```

```
sns.violinplot(data=tips, x='day',y='total_bill')
```



요일별 전체 금액의 분포



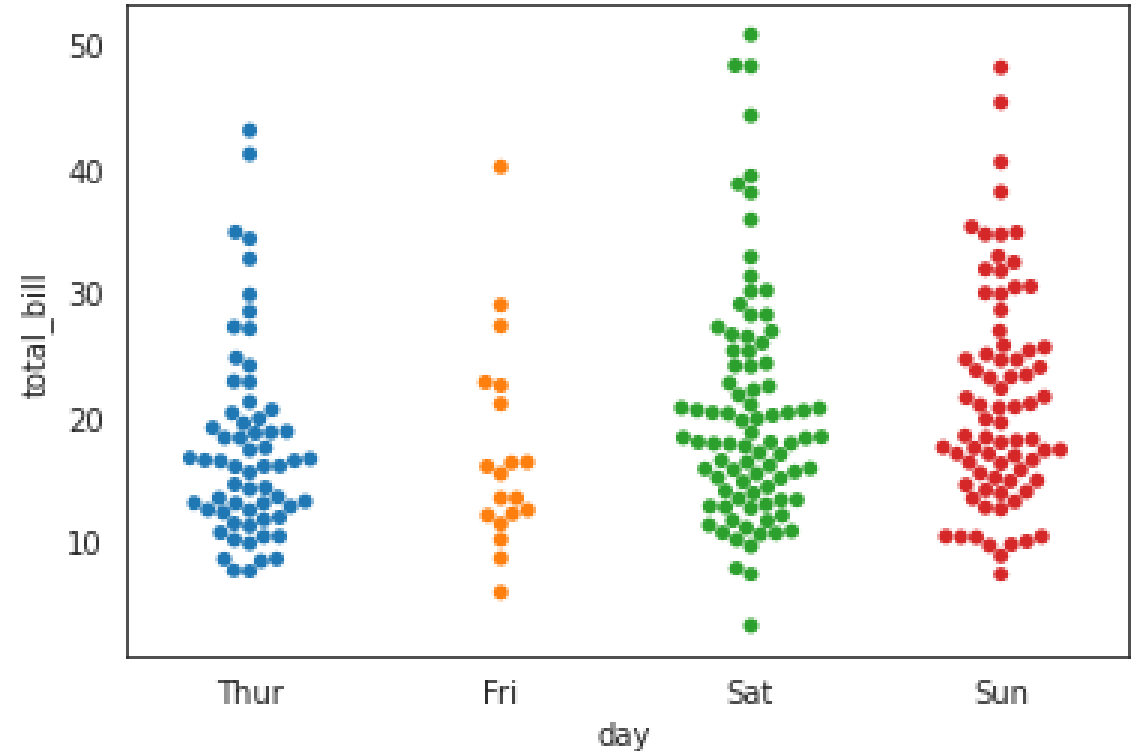
# Seaborn – swarmplot

- 데이터의 분포를 산점도를 이용해 나타냄

```
tips=sns.load_dataset("tips")
```

```
sns.swarmplot(data=tips, x='day',y='total_bill')
```

요일별 전체 금액의 분포

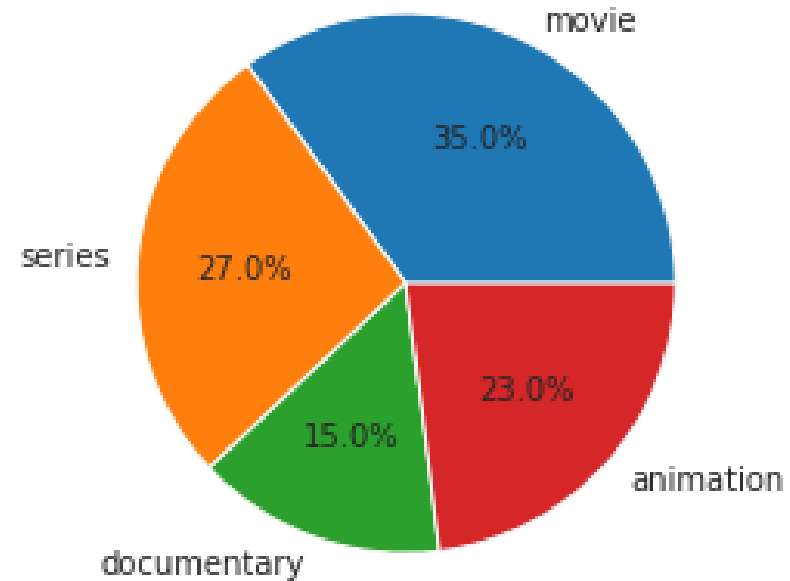


# Matplotlib pie chart

인하공전 컴퓨터 정보 과

## ■ 비율 비교에 효과 적임

```
import matplotlib.pyplot as plt  
ratio = [35, 27, 15, 23]  
labels = ['movie', 'series', 'documentary', 'animation']  
plt.pie(ratio, labels=labels, autopct='%.1f%%')  
  
plt.show()
```



동영상 장르별 조회 비율

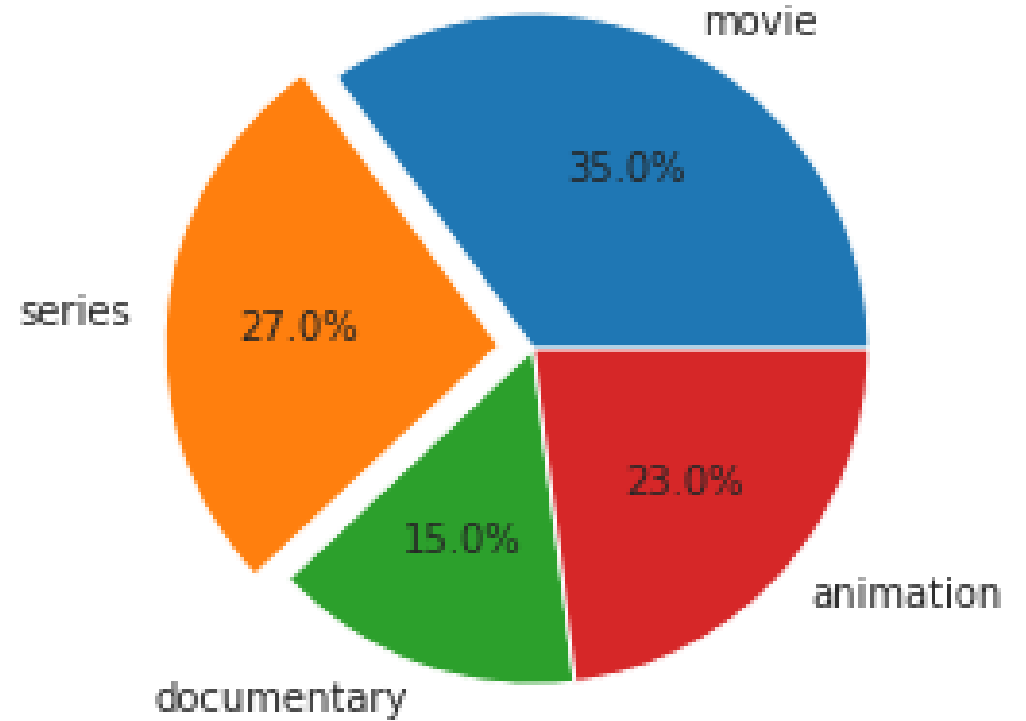
# Matplotlib pie chart

인하공전 컴퓨터 정보 과

```
import matplotlib.pyplot as plt
ratio = [35, 27, 15, 23]
labels = ['movie', 'series', 'documentary', 'animation']
explode = [0, 0.1, 0, 0.0]
plt.pie(ratio, labels=labels,
        explode=explode, autopct='%0.1f%%')

plt.show()
```

동영상 장르별 조회 비율



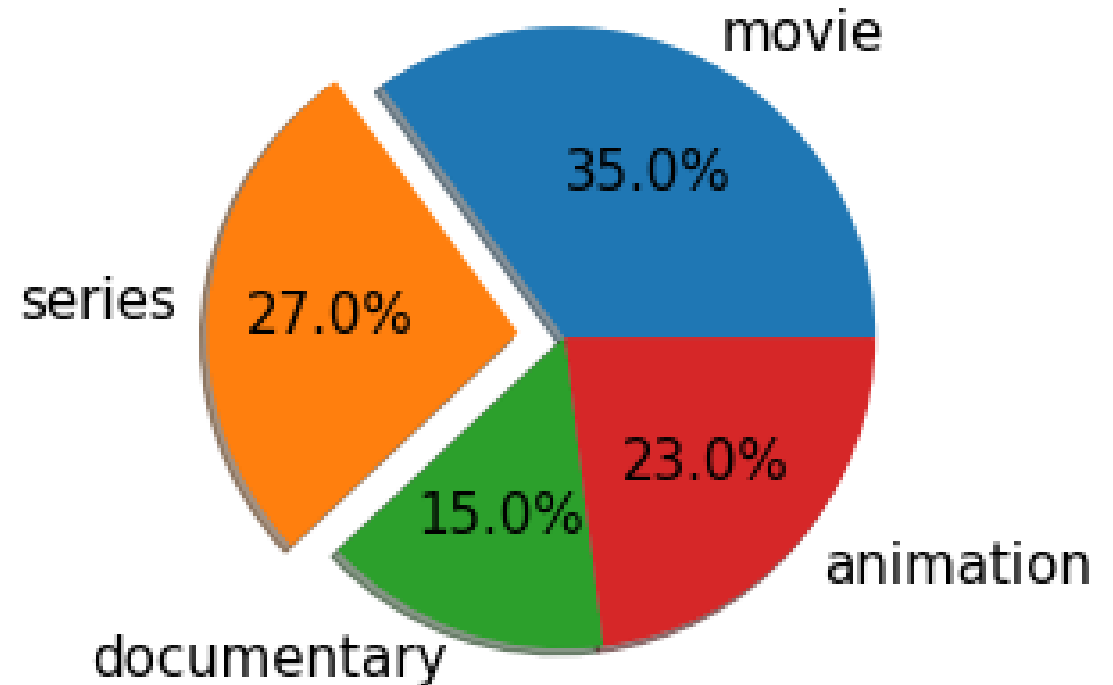
# Matplotlib pie chart

인하공전 컴퓨터 정보 과

```
import matplotlib.pyplot as plt
ratio = [35, 27, 15, 23]
labels = ['movie', 'series', 'documentary', 'animation']
explode = [0, 0.15, 0, 0.0]
plt.pie(ratio, labels=labels,
shadow=True,explode=explode,autopct='%1f%%',textprops={'fontsize': 15})

plt.show()
```

동영상 장르별 조회 비율

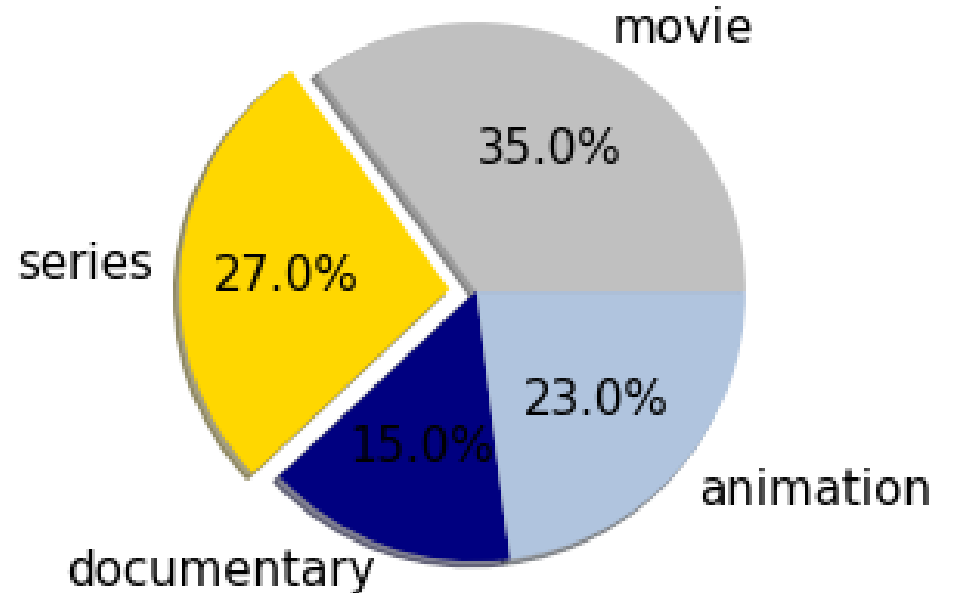




























































































# Matplotlib pie chart

```
import matplotlib.pyplot as plt
ratio = [35, 27, 15, 23]
labels = ['movie', 'series', 'documentary', 'animation']
explode = [0, 0.1, 0, 0.0]
colors=['silver', 'gold', 'navy', 'lightsteelblue']
plt.pie(ratio, labels=labels,
shadow=True,explode=explode,colors=colors,autopct='%1f%%',textprops={'fontsize': 15})

plt.show()
```

동영상 장르별 조회 비율



	black		bisque		forestgreen		slategrey
	dimgray		darkorange		limegreen		lightsteelblue
	dimgrey		burlywood		darkgreen		cornflowerblue
	gray		antiquewhite		green		royalblue
	grey		tan		lime		ghostwhite
	darkgray		navajowhite		seagreen		lavender
	darkgrey		blanchedalmond		mediumseagreen		midnightblue
	silver		papayawhip		springgreen		navy
	lightgray		moccasin		mintcream		darkblue
	lightgrey		orange		mediumspringgreen		mediumblue
	gainsboro		wheat		mediumaquamarine		blue
	whitesmoke		oldlace		aquamarine		slateblue
	white		floralwhite		turquoise		darkslateblue
	snow		darkgoldenrod		lightseagreen		mediumslateblue
	rosybrown		goldenrod		mediumturquoise		mediumpurple
	lightcoral		cornsilk		azure		rebeccapurple
	indianred		gold		lightcyan		blueviolet
	brown		lemonchiffon		paleturquoise		indigo
	firebrick		khaki		darkslategray		darkorchid
	maroon		palegoldenrod		darkslategrey		darkviolet
	darkred		darkkhaki		teal		mediumorchid
	red		ivory		darkcyan		thistle
	mistyrose		beige		aqua		plum
	salmon		lightyellow		cyan		violet
	tomato		lightgoldenrodyellow		darkturquoise		purple
	darksalmon		olive		cadetblue		darkmagenta
	coral		yellow		powderblue		fuchsia
	orangered		olivedrab		lightblue		magenta
	lightsalmon		yellowgreen		deepskyblue		orchid
	sienna		darkolivegreen		skyblue		mediumvioletred
	seashell		greenyellow		lightskyblue		deeppink
	chocolate		chartreuse		steelblue		hotpink
	saddlebrown		lawngreen		aliceblue		lavenderblush
	sandybrown		honeydew		dodgerblue		palevioletred
	peachpuff		darkseagreen		lightslategray		crimson
	peru		palegreen		lightslategrey		pink
	linen		lightgreen		slategray		lightpink

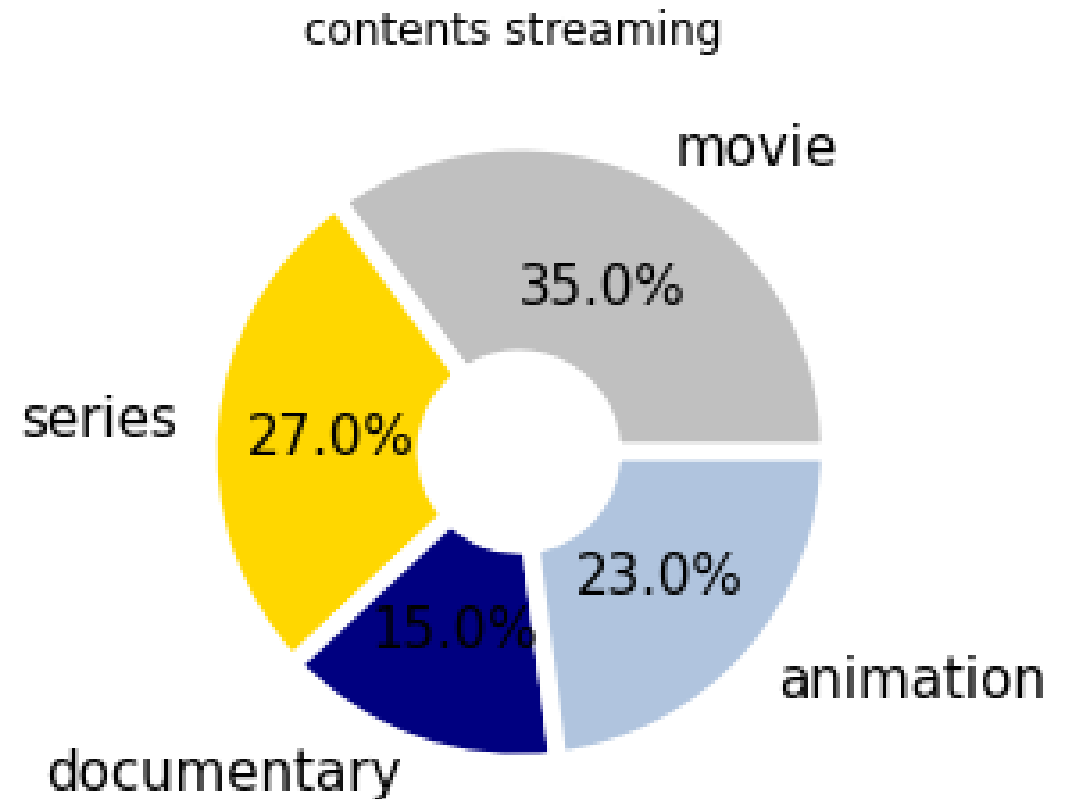
# Matplotlib pie chart

인하공전 컴퓨터 정보 과

```
import matplotlib.pyplot as plt
ratio = [35, 27, 15, 23]
labels = ['movie', 'series', 'documentary', 'animation']
explode = [0, 0.1, 0, 0.0]
colors=['silver', 'gold', 'navy', 'lightsteelblue']
wedgeprops={'width': 0.7, 'edgecolor': 'w', 'linewidth': 5}
plt.pie(ratio, labels=labels, colors=colors, autopct='%1f%%',
wedgeprops=wedgeprops, textprops={'fontsize': 15})
plt.title('contents streaming ')

plt.show()
```

동영상 장르별 조회 비율



# Matplotlib ,seaborn 한글 폰트 for colab

인하공전 컴퓨터 정보 과

```
# 폰트 설치
```

```
import matplotlib.font_manager as fm
```

```
!apt-get -qq -y install fonts-nanum > /dev/null
```

```
fontpath = '/usr/share/fonts/truetype/nanum/NanumBarunGothic.ttf'
```

```
font = fm.FontProperties(fname=fontpath, size=9)
```

```
fm._rebuild()
```

```
#런타임 재시작
```

```
import os
```

```
os.kill(os.getpid(), 9)
```

```
# 폰트 설정
```

```
import matplotlib.pyplot as plt
```

```
import matplotlib as mpl
```

```
import matplotlib.font_manager as fm
```

```
# 마이너스 표시 문제
```

```
mpl.rcParams['axes.unicode_minus'] = False
```

```
# 한글 폰트 설정
```

```
path = '/usr/share/fonts/truetype/nanum/NanumGothicBold.ttf'
```

```
font_name = fm.FontProperties(fname=path, size=18).get_name()
```

```
plt.rc('font', family=font_name)
```

```
fm._rebuild()
```



# pandas area plot (면적 그래프)

- 데이터의 합계와 비율이 전체적으로 어떻게 변하는지 파악 할 수 있다.

```
%reset -f
import pandas as pd
import matplotlib.pyplot as plt
```

```
df=pd.read_csv('/content/contents.csv')
print(df)
```

컨텐츠 조회수

	영화	뮤직비디오	음악
0	30	10	6
1	20	23	7
2	30	34	8
3	35	23	13
4	45	20	14
5	40	21	15
6	48	15	16
7	50	14	20
8	43	13	18
9	33	12	16
10	21	16	14
11	15	18	12

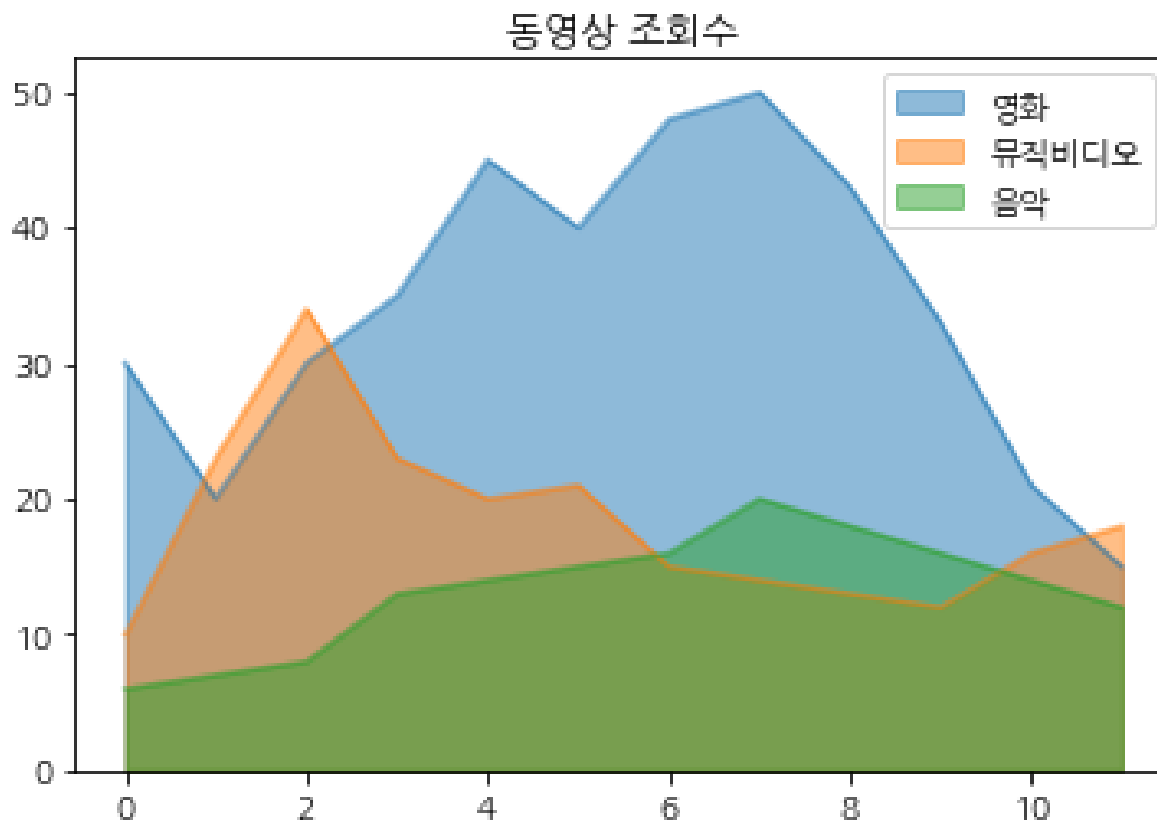
동영상 장르별 조회 비율

# Pandas (matplotlib) area plot (면적 그래프)

인하공전 컴퓨터 정보 과

- 데이터의 합계와 비율이 전체적으로 어떻게 변하는지 파악 할 수 있다.

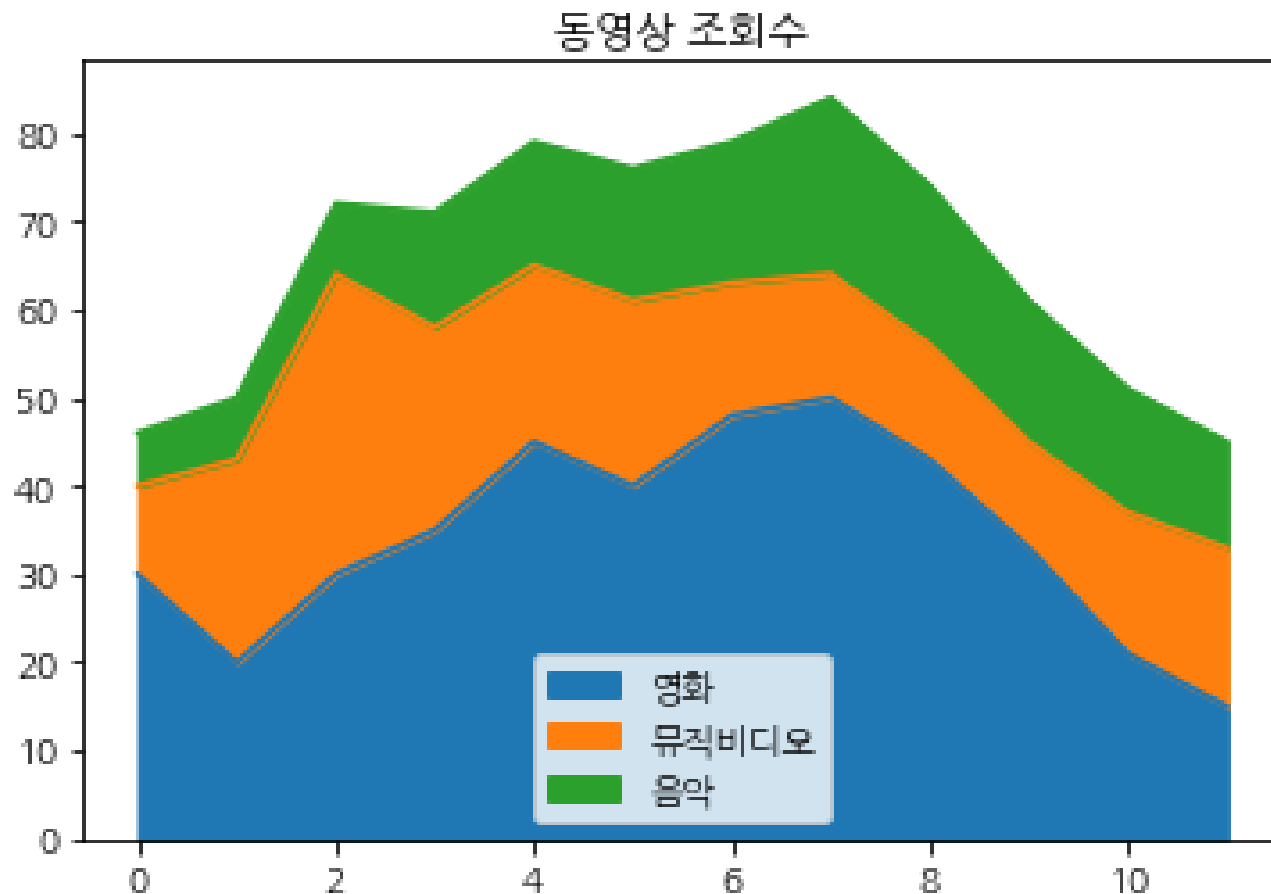
```
df.plot(kind='area',stacked=False)  
plt.title('동영상 조회수')  
plt.show()
```



# Pandas (matplotlib) area plot (면적 그래프)

인하공전 컴퓨터 정보 과

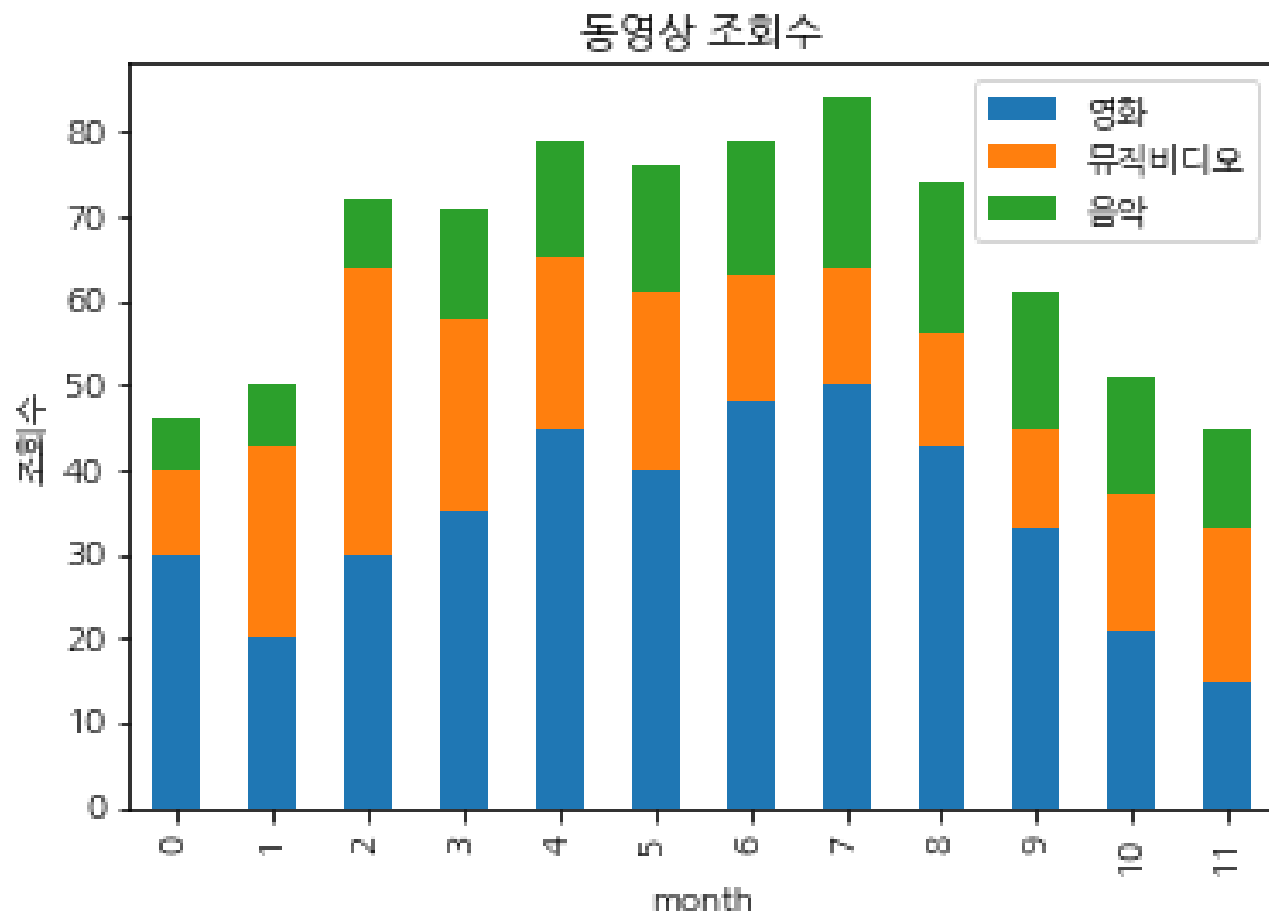
```
df=pd.read_csv('/content/contents.csv')  
print(df)  
df.plot(kind='area',stacked=True)  
plt.title('동영상 조회수')  
plt.show()
```



# Pandas (matplotlib) bar (누적 막대)

인하공전 컴퓨터 정보 과

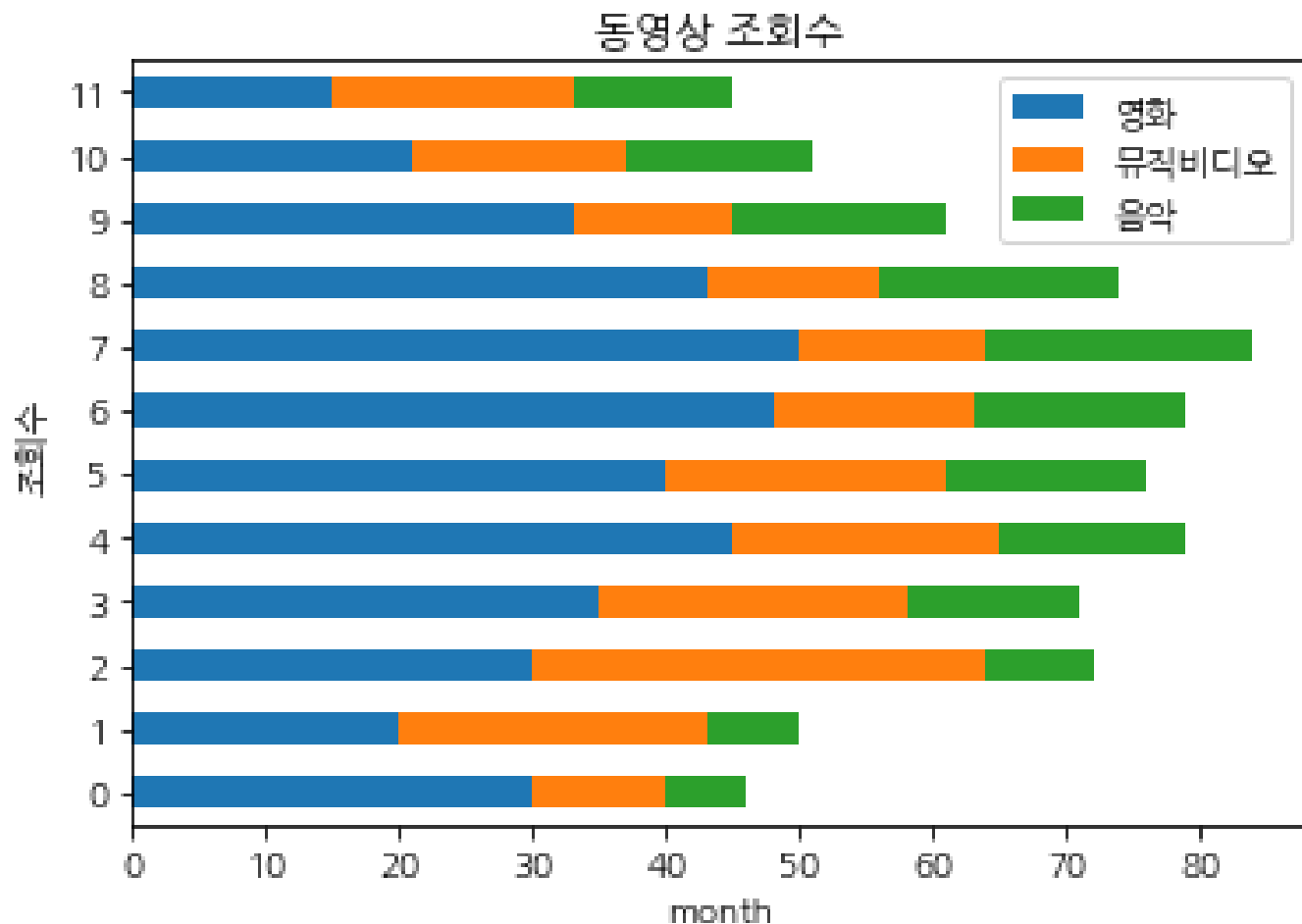
```
df=pd.read_csv('/content/contents.csv')
df.plot(kind='bar',stacked=True)
plt.title('동영상 조회수')
plt.xlabel('month')
plt.ylabel('조회수')
plt.show()
```



# Pandas (matplotlib) bar (누적 막대-가로)

인하공전 컴퓨터 정보 과

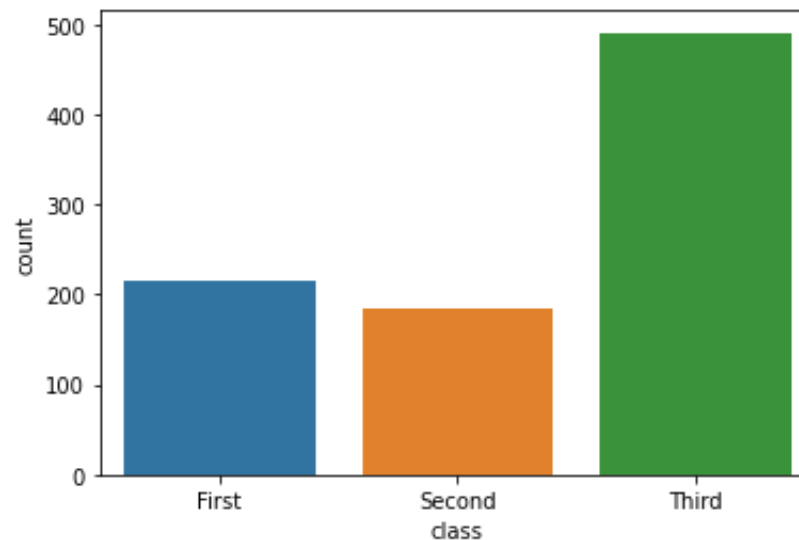
```
df=pd.read_csv('/content/contents.csv')
df.plot(kind='barh',stacked=True)
plt.title('동영상 조회수')
plt.xlabel('month')
plt.ylabel('조회수')
plt.show()
```



- 1~10. (8,9,10은 선택 과제)

\*.ipynb, \*.py 제출

1.class 별로 승객 수를 나타내는 countplot을 구하시오(image 참조)



```
import warnings
warnings.simplefilter(action='ignore',
category=FutureWarning)
import matplotlib.pyplot as plt
import seaborn as sns
```

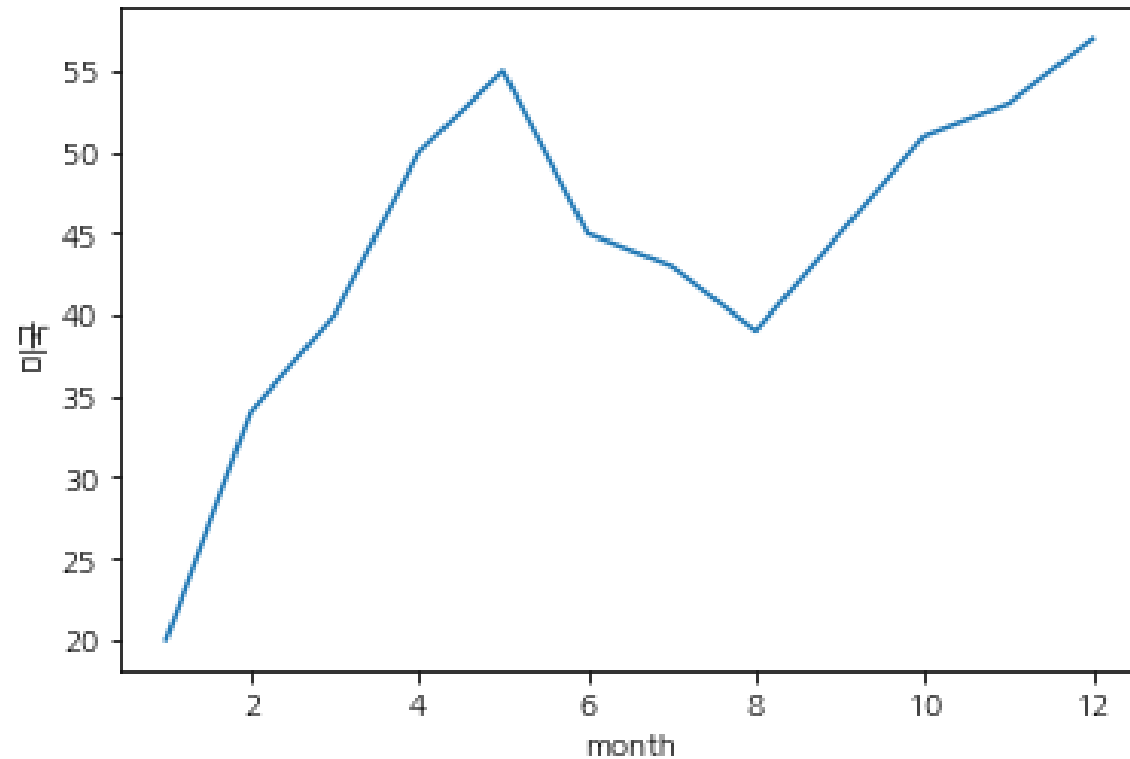
```
filename = '/content/welfareClean.csv'
```

	성별	생일	결혼	유무	종교	유무	직업	코드	소득	지역구	나이	직업	연령대
0	남성	1948	무응답		없음		942.0	120.000000		서울	73	경비원 및 검표원	노년
1	남성	1945	이혼		없음		942.0	220.200000		서울	76	경비원 및 검표원	노년
2	남성	1946	결혼		없음		942.0	139.000000		서울	75	경비원 및 검표원	노년
3	남성	1953	결혼		없음		942.0	150.000000		서울	68	경비원 및 검표원	노년
4	남성	1960	결혼		있음		942.0	166.000000		서울	61	경비원 및 검표원	노년
...	...	...	...	...	...	...	...	...	...	...	...	...	...
7524	여성	1950	결혼		있음		819.0	241.619016		강원/충북	71	기타 식품가공관련 기계조작원	노년
7525	남성	1960	결혼		있음		111.0	250.000000	광주/전남/전북/제주도	61	의회의원 고위공무원 및 공공단체임원	노년	
7526	남성	1960	결혼		없음		111.0	1250.000000		서울	61	의회의원 고위공무원 및 공공단체임원	노년
7527	남성	1992	무응답		있음		876.0	280.000000		부산/경남/울산	29	선박 갑판승무원 및 관련 종사원	청년
7528	남성	1935	결혼		있음		876.0	156.000000		부산/경남/울산	86	선박 갑판승무원 및 관련 종사원	노년

7529 rows x 10 columns

```
df=pd.read_csv('/content/입국자.csv')  
print(df)
```

	month	미국	중국	일본
0	1	20	10	15
1	2	34	15	14
2	3	40	15	15
3	4	50	16	17
4	5	55	17	18
5	6	45	18	13
6	7	43	22	12
7	8	39	14	9
8	9	45	13	10
9	10	51	10	15
10	11	53	12	20
11	12	57	9	22





# 수고하셨습니다

---

[jhmin@inhatec.ac.kr](mailto:jhmin@inhatec.ac.kr)