

3-차트의 요소

4. More Tips

안수빈

서울대학교 컴퓨터공학부

Contents

1. Grid 이해하기

- 1.1 Default Grid
- 1.2 다양한 타입의 Grid

2. 심플한 처리

- 2.1 선 추가하기
- 2.2 면 추가하기

3. Setting 바꾸기

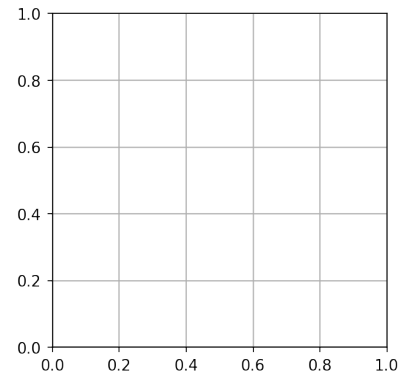
- 3.1 Theme

1. Grid 이해하기

격자도 다양하게 사용해보자

1.1 Default Grid

- 기본적인 **Grid**는 축과 평행한 선을 사용하여 거리 및 값 정보를 보조적으로 제공

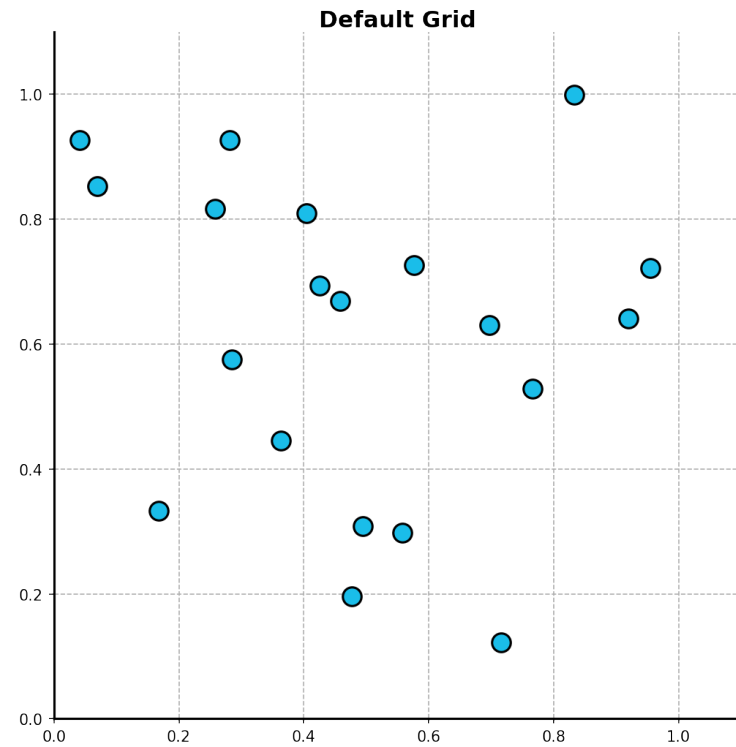


- 색은 다른 표현들을 방해하지 않도록 **무채색** (`color`)
- 항상 Layer 순서 상 **맨 밑**에 오도록 조정 (`zorder`)
- **큰** 격자/**세부** 격자 (`which='major', 'minor', 'both'`)
- X축? Y축? 동시에 (`axis='x', 'y', 'both'`)

- 전형적인 Grid는 아니지만 여러 형태의 Grid가 존재
 - 두 변수의 합이 중요하다면 $x+y = c$
 - 비율이 중요하다면 $y = cx$
 - 두 변수의 곱이 중요하다면 $xy = c$
 - 특정 데이터를 중심으로 보고 싶다면 $(x-x')^2 + (y-y')^2 = c$
- 전형적이지 않고, 구현도 까다롭지만
 - numpy + matplotlib으로 쉽게 구현 가능
 - 재미있는 예시는 <https://medium.com/nightingale/gotta-gridem-all-2f768048f934>

1.2 다양한 타임의 Grid

1. Grid 이해하기



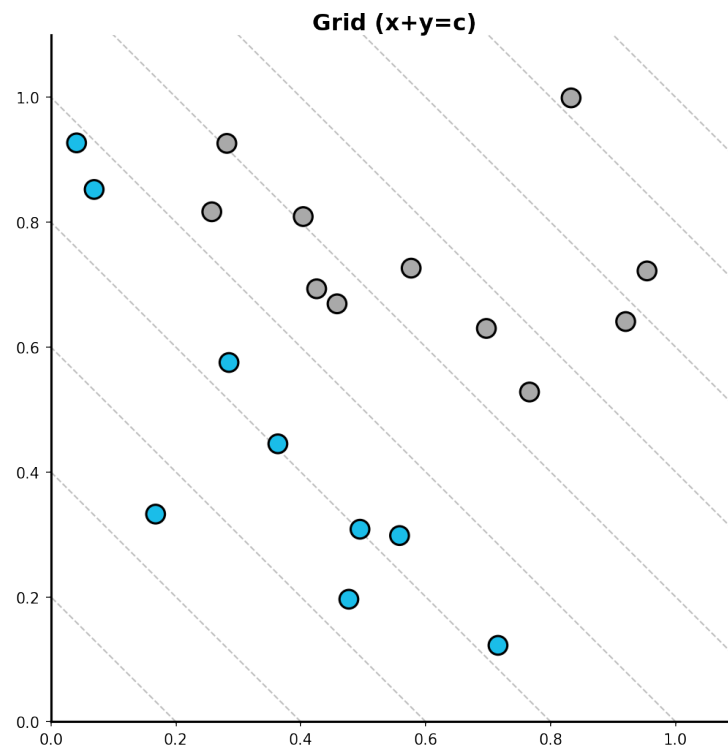
`ax.grid(zorder=0)`

기본적인 Grid 정보

1.2 다양한 타임의 Grid

1. Grid 이해하기

- $X+Y = C$ 를 사용한 Grid
- 회색 선에 걸치는 값은 $X+Y$ 값이 동일
- Feature의 절대적 합이 중요한 경우?
 - 공격 + 수비 합으로 평가
 - 국어+수학 비중 평가



```
x_start = np.linspace(0, 2.2, 12, endpoint=True)
```

```
slope = 2
```

```
# 절편 값을 지정하여 이어주는게 포인트
```

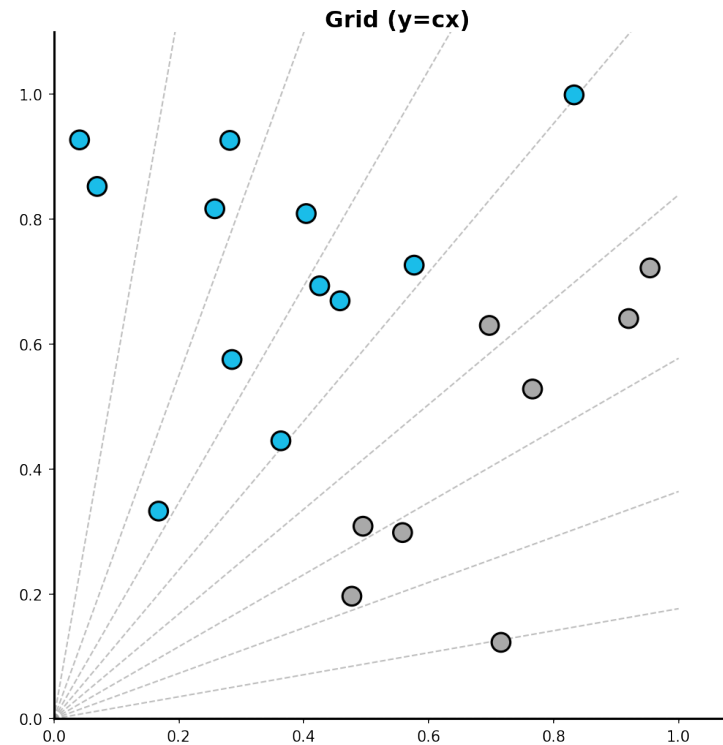
```
for xs in x_start:
```

```
    ax.plot([xs, 0], [0, xs],  
            linestyle='--',  
            color='gray',  
            alpha=0.5,  
            linewidth=1)
```

1.2 다양한 타입의 Grid

1. Grid 이해하기

- $Y = CX$ 를 사용한 Grid
- 가파를 수록 Y/X 가 커짐
- Feature의 비율이 중요한 경우



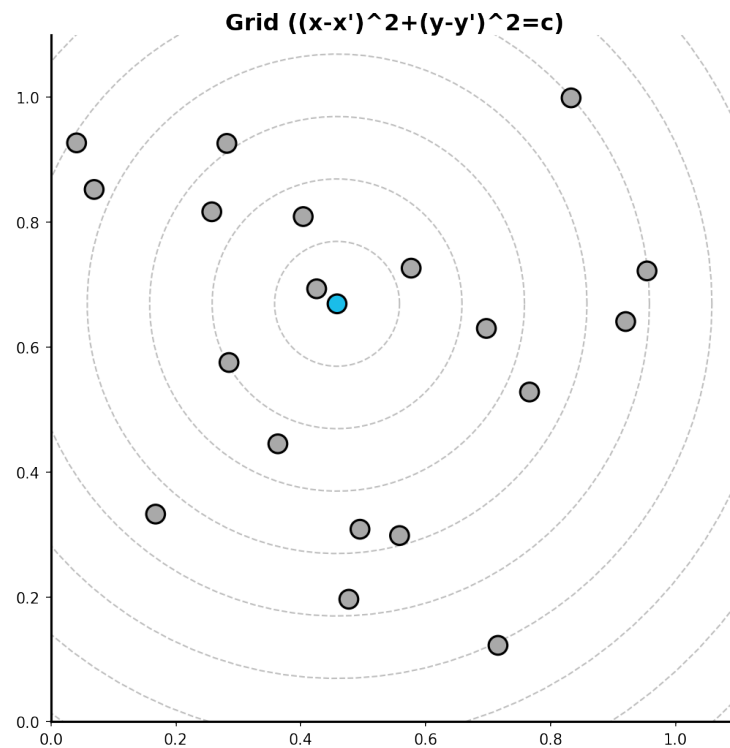
```
# 균일한 기울기 값을 삼각함수로 만들기  
radian = np.linspace(0, np.pi/2, 10,  
                    endpoint=True)
```

```
for rad in radian:  
    ax.plot([0,1], [0, np.tan(rad)],  
           linestyle='--',  
           color='gray',  
           alpha=0.5,  
           linewidth=1)
```


1.2 다양한 타입의 Grid

1. Grid 이해하기

- 동심원을 사용
- 특정 지점에서 거리를 살펴볼 수 있음
- 가장 가까운 포인트를 찾거나
- 한 데이터에서 특정 범위의 데이터



```
rs = np.linspace(0, 2, 21, endpoint=True)
```

```
# 균일한 거리에 따라 동심원을 만들어주기
```

```
for r in rs:
```

```
    xx = r*np.cos(np.linspace(0, 2*np.pi, 100))
```

```
    yy = r*np.sin(np.linspace(0, 2*np.pi, 100))
```

```
    ax.plot(xx+x[2], yy+y[2],
```

```
            linestyle='--',
```

```
            color='gray',
```

```
            alpha=0.5,
```

```
            linewidth=1)
```

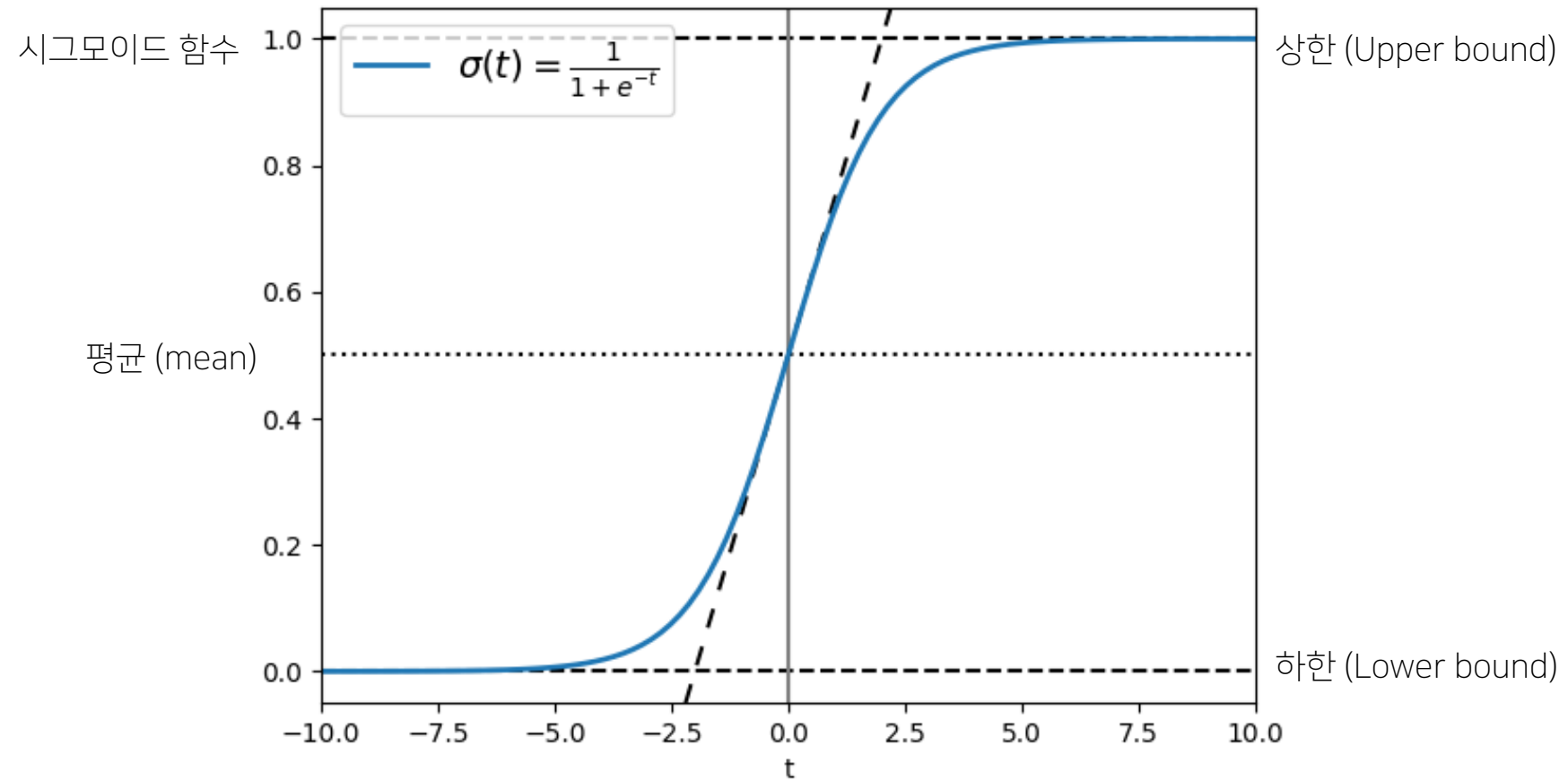
2.

심플한 처리

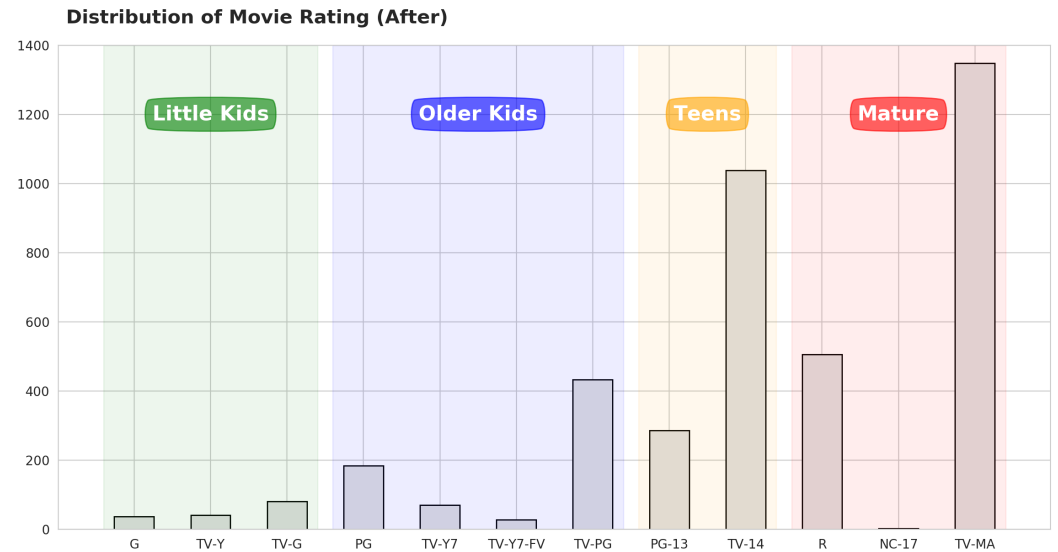
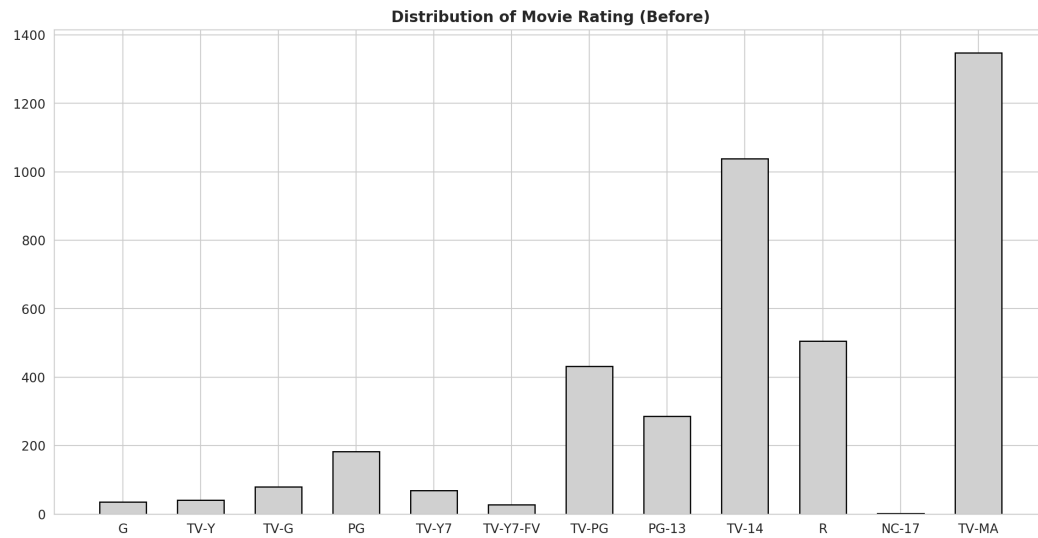
어렵지 않게 더 많은 정보와 주의(attention)를 줄 수 있는 방법들

2.1 선 추가하기

2. 심플한 처리



Netflix 영화 상영 등급 분포



3.

Setting 바꾸기

한 번에 설정 바꾸기

3.1 Theme

3. Setting 바꾸기

대표적으로 많이 사용하는 테마

