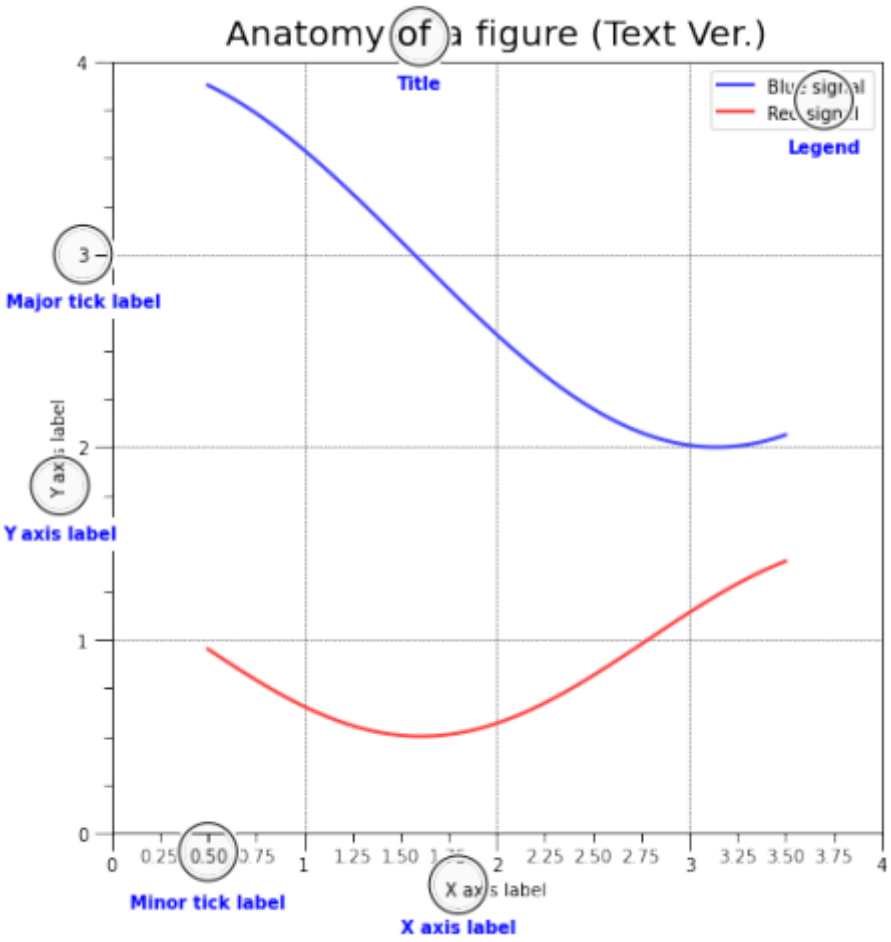
(3-1) Text 사용하기

시각화에 Text를 추가해 설명하면 의도를 전달하고 오해를 방지할 수 있다.

하지만 Text를 과하게 사용한다면 오히려 이해를 방해할 수도 있다.

이번 강의는 Matplotlib에서 제공하는 API를 바탕으로 Text 이해하는 것이 목표



- Title : 가장 큰 주제를 설명
- Label : 축에 해당하는 데이터 정보를 제공
- Tick Label : 축에 눈금을 사용하여 스케일 정보를 추가
- Legend : 한 그래프에서 2개 이상의 서로 다른 데이터를 분류하기 위해서 사용하는 보조 정보
- Annotation(Text) : 그 외의 시각화에 대한 설명을 추가

3-1.Text 코드 실습

[ ] ↳ 숨겨진 셀 25개

(3-2) Color 사용하기

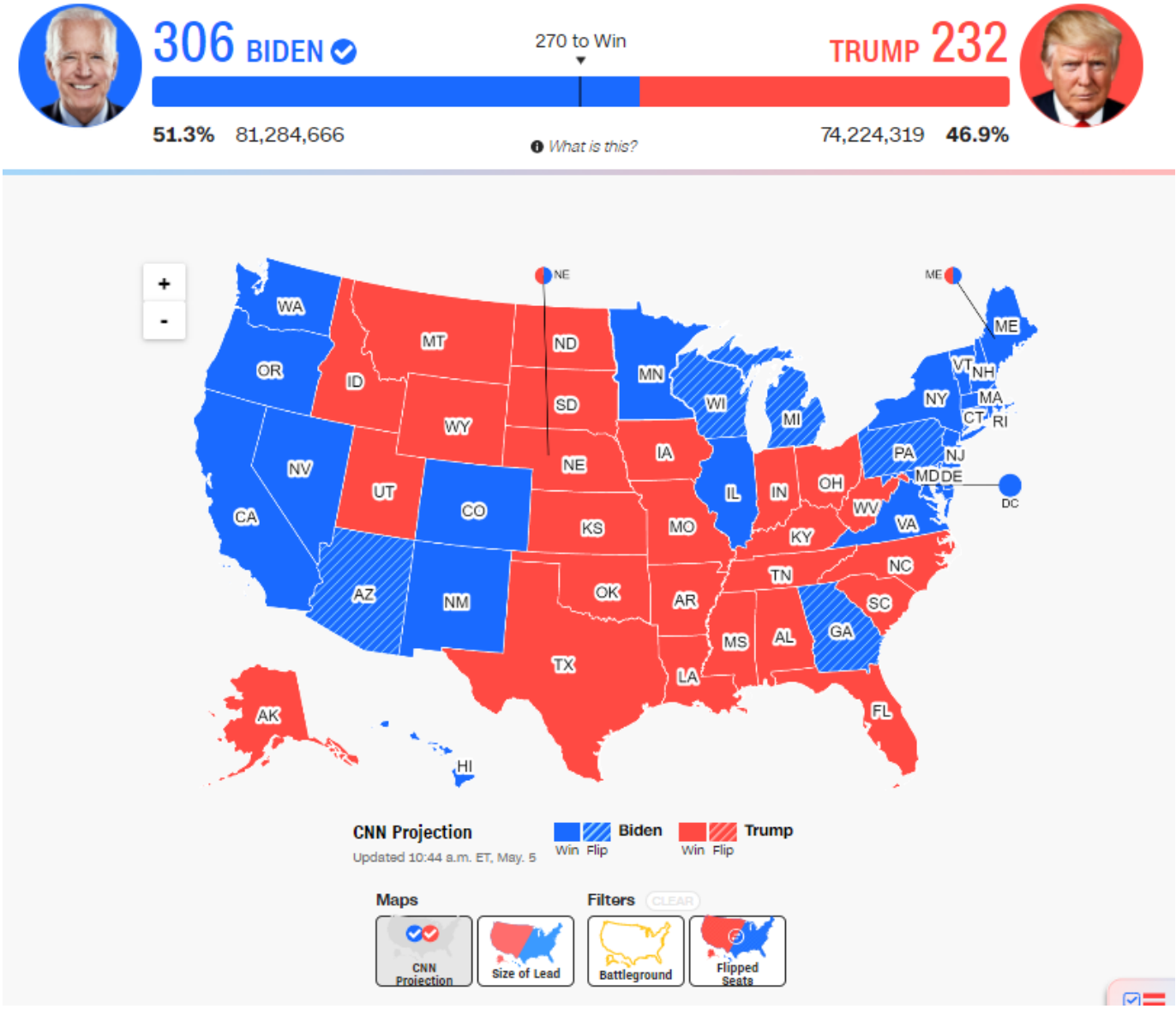
Color에 대한 이해

가장 효과적인 시각화 채널

1. 위치
2. **색**

시각화에서 중요한 것은 독자에게 원하는 인사이트를 전달!

- 전하고 싶은 내용 모두 전달 + 그 과정에서 오해 없어야

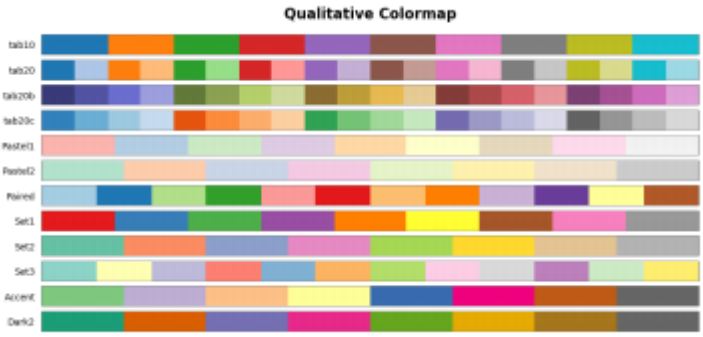


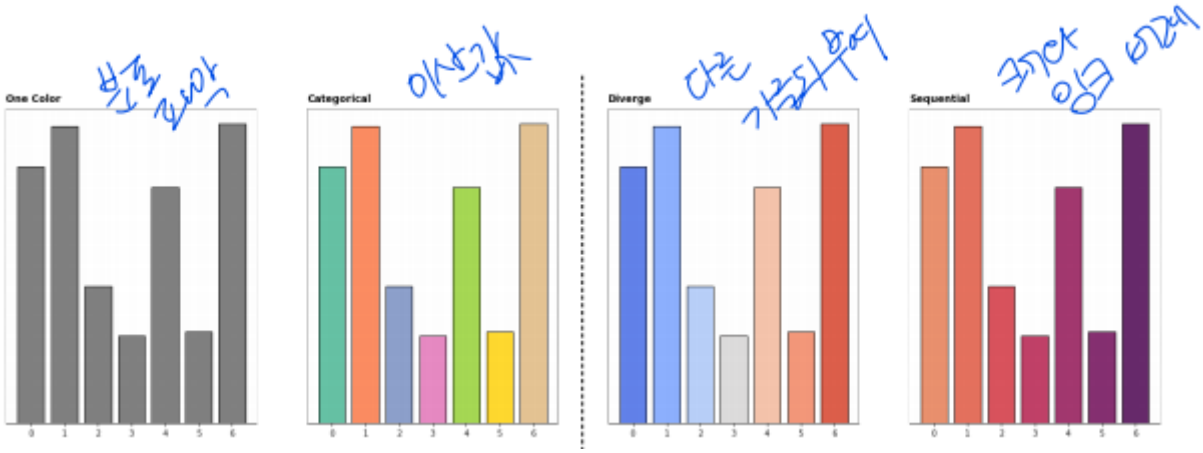
이미지 출처 : https://edition.cnn.com/election/2020/results/president

Color Palette의 종류

범주형 (Categorical)

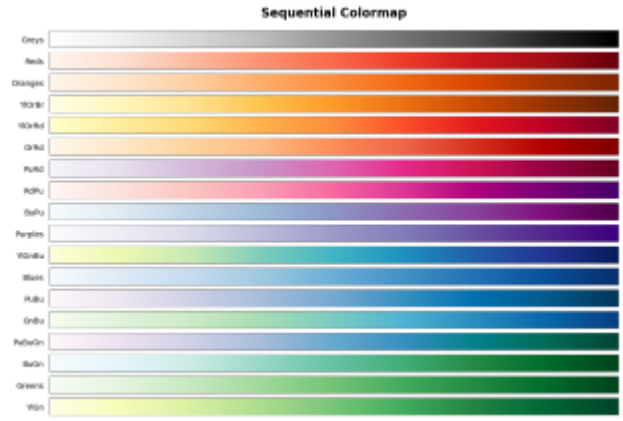
- Discrete, Qualitative 등으로도 불림
- 독립된 색상으로 구성되어 범주형 변수에 사용.
- 최대 10개의 색상까지 사용
    - 그 이상은 기타 로 묶기
- 색의 차이로 구분
    - 채도, 명도를 개별적 조정은 지양





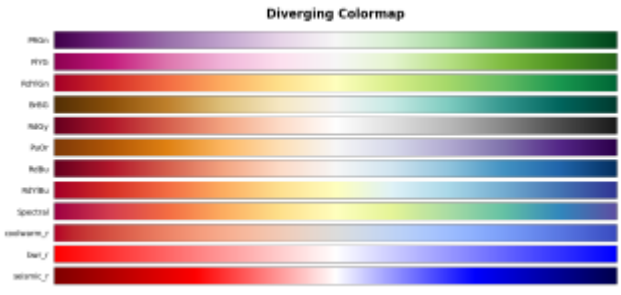
연속형 (Sequential)

- 정렬된 값을 가지는 순서형, 연속형 변수에 적합
- 연속적인 색상(gradient) 사용해 값을 표현
    - 어두운 배경에서는 밝은 색이 큰 값을 표현
    - 밝은 배경에서는 어두운 색이 큰 값을 표현
- 색상은 단일 색조로 표현
- 균일한 색상 변화가 중요

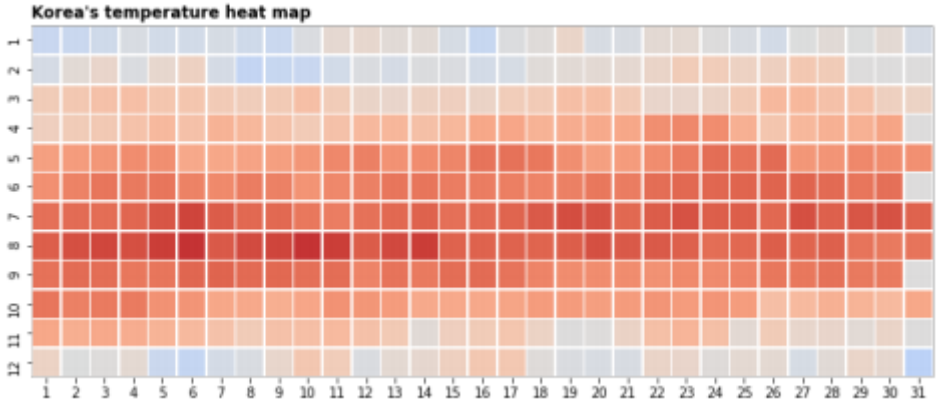


발산형 (Diverge)

- 연속형과 유사하지만 중앙을 기준으로 발산(다른 색을 사용)
    - 상반된 값(ex. 기온), 서로 다른 2개(ex. 지지율)을 표현하는 데 적합
- 양 끝으로 갈수록 채도가 높아짐
- 중앙의 색은 양쪽의 점에서 편향되지 않아야 함
    - 무채색일 필요 없음



아래는 평균 기온 발산형 컬러 팔레트를 사용



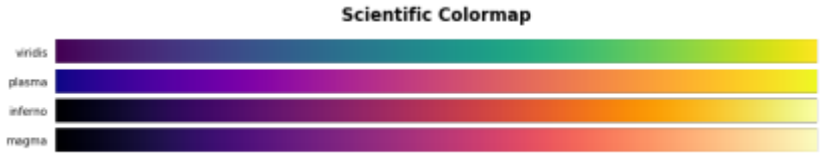
그 외 색 Tips

강조, 그리고 색상 대비

1. 강조를 위해 Highlighting 사용
2. 강조를 위한 방법 중 하나. 색상 대비(Color Contrast) 사용
    - 명도 대비 : 밝은 색과 어두운 색을 같이 배치하면 밝은 색은 더 밝게, 어두운 색은 더 어둡게 보임
        - 회색 vs 검정
    - 색상 대비 : 가까운 색은 차이가 더 크게 보임
        - 파랑 vs 보라, 빨강 vs 보라
    - 채도 대비 : 채도의 차이 활용
        - 회색 vs 주황
    - 보색 대비 : 정반대 색상을 사용 => 더 선명해 보임
        - 빨강 vs 초록

색각 이상

- 색맹 : 삼원색 중에 특정 색 감지 불가
- 색약 : 부분적 인지 이상
- 색 인지가 중요한 분야(과학/연구 등)는 이에 대한 고려가 필수
    - 마그마, 플라즈마



3-2. Color 코드 실습

[ ] ↳ 숨겨진 셀 29개

(3-3) Facet 사용하기

Facet

Facet : 분할을 의미, 화면 분할

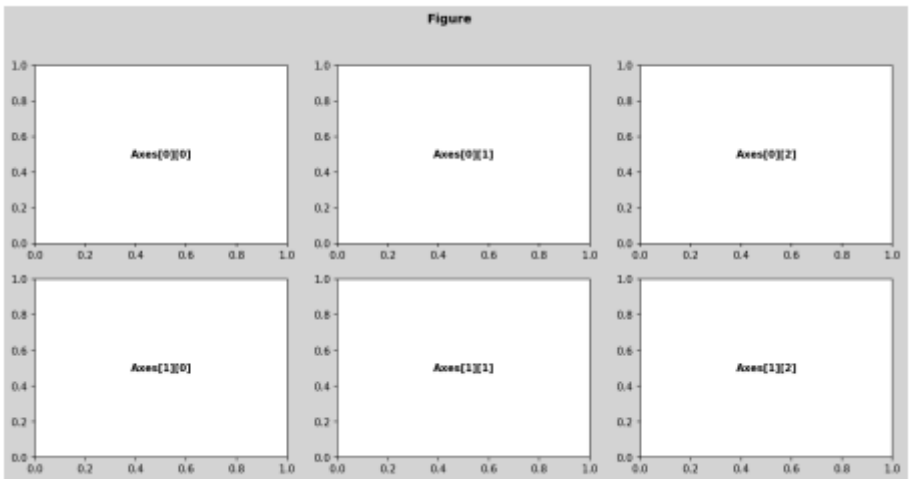
화면의 View를 분할, 추가해 다양한 관점을 전달 가능

- 하나의 데이터셋을 서로 다른 인코딩으로 다른 인사이트를 보여줌
- 같은 방법으로 동시에 여러 feature를 확인 가능
- 큰 틀에서 볼 수 없는 **부분 집합을 세세하게 보여줌**

Matplotlib에서 구현

Figure와 Axes

- Figure: 큰 틀, 언제나 1개
- Ax는 각 그래프가 들어가는 공간, 1개 이상



NxM subplots

가장 쉬운 3가지 방법

- plt.subplot()
- plt.figure() + fig.add_subplot()
- plt.subplots()

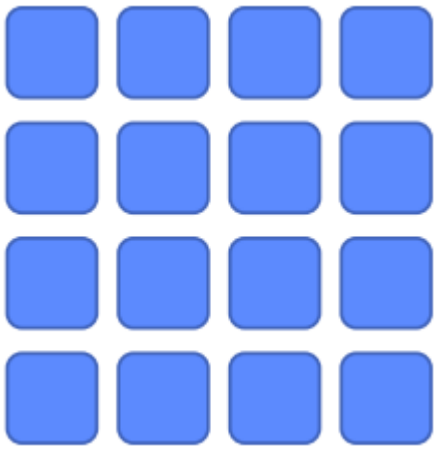
쉽게 조정할 수 있는 요소

- figuresize
- dpi
- sharex, sharey
- squeeze
- aspect

Grid Spec의 활용

Grid Spec : 그리드 형태의 subplots, 아래 이미지가 Grid Spec

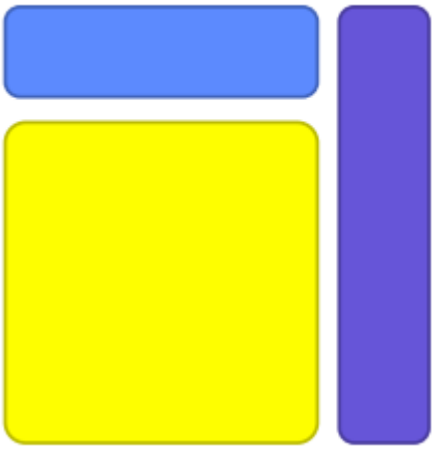
- 기존 Subplots로 4 x 4 생성



아래 서브플롯 표현 방법

1. Slicing 사용
    - fig.subplot2grid()
        - 파랑 : axes[0, :3]
        - 노랑 : axes[1:, :3]
        - 보라 : axes[3, :]
2. x, y, dx, dy를 사용
    - 시작 위치 x, y와 차이 dx, dy로도 표현

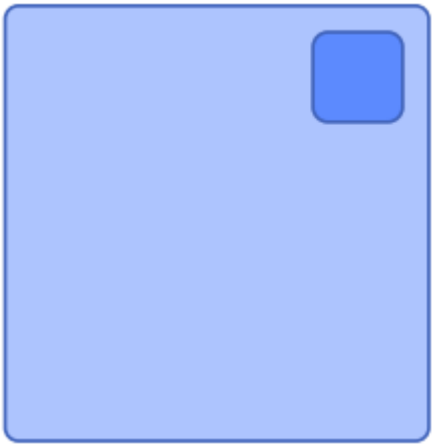
- fig.subplot2grid()
  - (x, y), dx, dy 포맷으로 작성
  - 파랑 : (0, 0), 1, 3
  - 노랑 : (1, 0), 3, 3
  - 보라 : (0, 3), 1, 4



내부에 그리기

ax.inset_axes()

- Ax 내부에 서브플롯을 추가
- 미니맵과 같은 형태로 추기
- 외부 정보를 적은 비중으로 추가



make_axes_locatable(ax)

- 그리드를 사용하지 않고 사이드에 추가

- 단, 방향의 통계정보를 제공 가능
- 제목 등의 텍스트 추가도 가능

o 그리드를 사용하지 않고 사이드에 추가 o 단, 방향의 통계정보를 제공할수도 o 제목 등의 텍스트 추가도 가능 o make_axes_locatable(ax)

3-3-Facet 코드 실습

[ ] ↳ 숨겨진 셀 *31*개

(3-4) More Tips

Grid 이해

Default Grid

- 기본 Grid는 축과 평행한 선을 사용하여 거리 및 값 정보를 보조적으로 제공
- 다른 표현 방해 방지를 위해 무채색 (color)
- 항상 Layer 순서 상 맨 밑 (zorder)
- 큰 격자 / 세부 격자 있음 (which='major', 'minor', 'both')
- X축 / Y축 따로 그리드 사용 가능 (axis='x' , 'y', 'both')

다양한 타입의 Grid

여러 형태의 Grid가 존재

- 두 변수의 합 표현 : x + y = c
- 비율 표현 : y = cx
- 두 변수의 곱 표현 : xy = c
- 특정 데이터를 중심 표현 : (x-x')^2 + (y-y')^2 = c

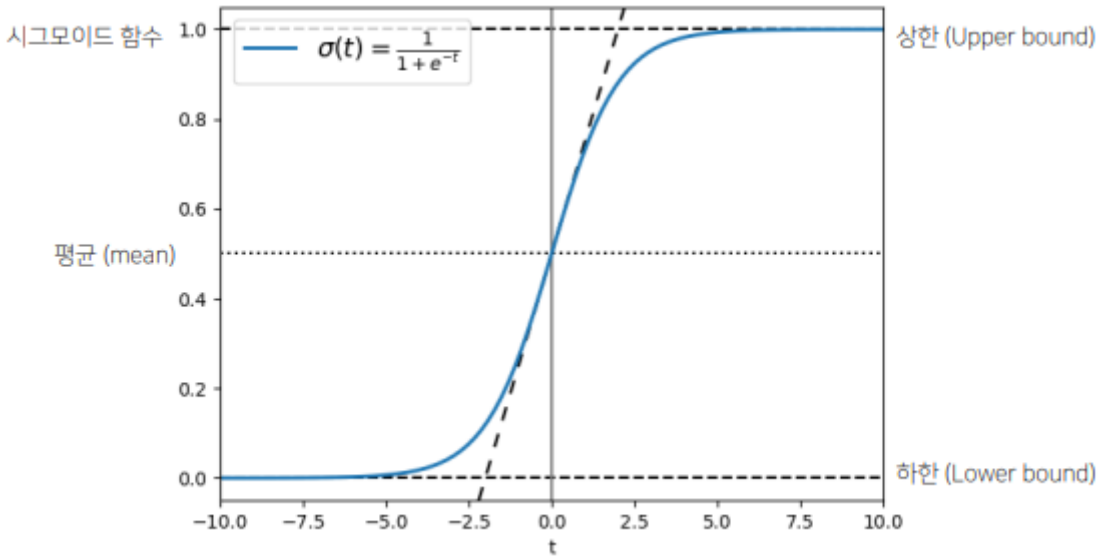
까다로운 구현

- numpy + matplotlib으로 구현 가능
- 예시 : https://medium.com/nightingale/gotta-gridem-all-2f768048f934

심플한 처리

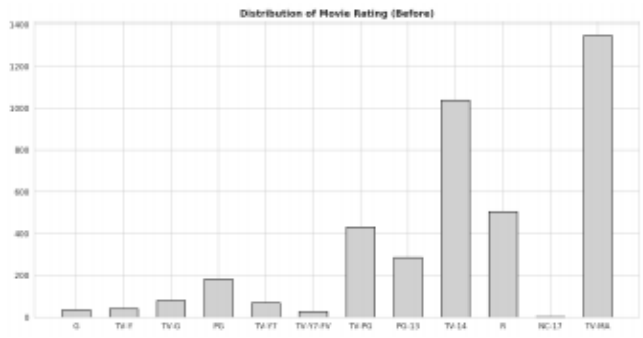
선 추가하기

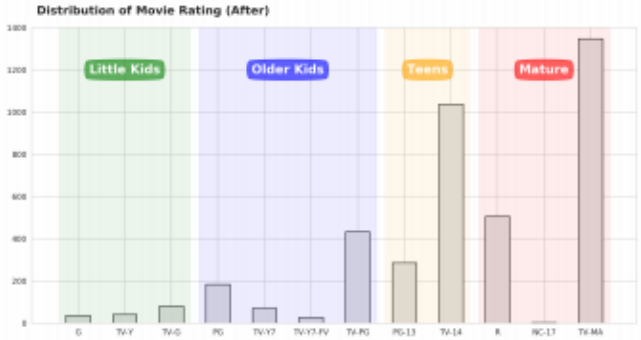
보조선 추가로 수치의 값 확인 용이해짐



면 추가하기

면 추가로 가독성 Up

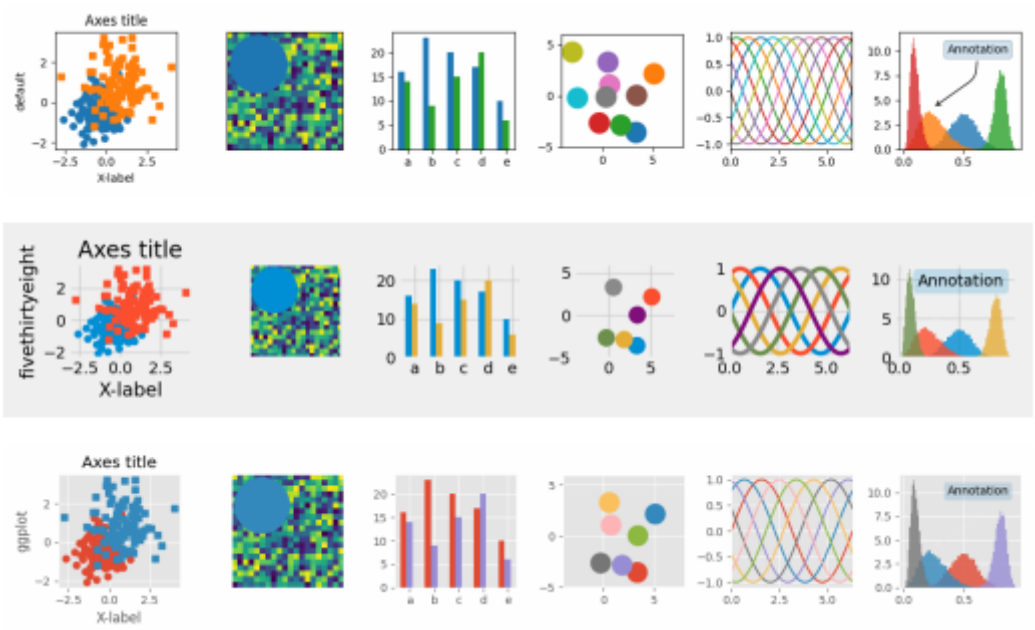




Setting 바꾸기

Theme

아래 이미지는 많이 사용하는 테마



3-4-More Information 코드 실습

[ ] ↳ 숨겨진 셀 *44*개