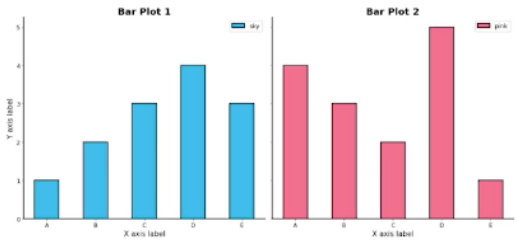
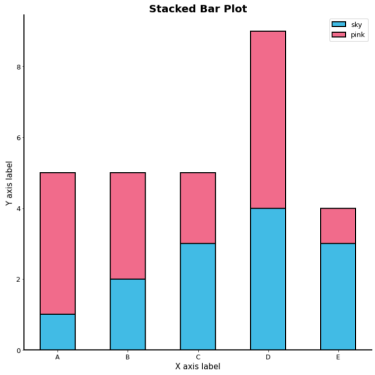
Plot

Bar Plot

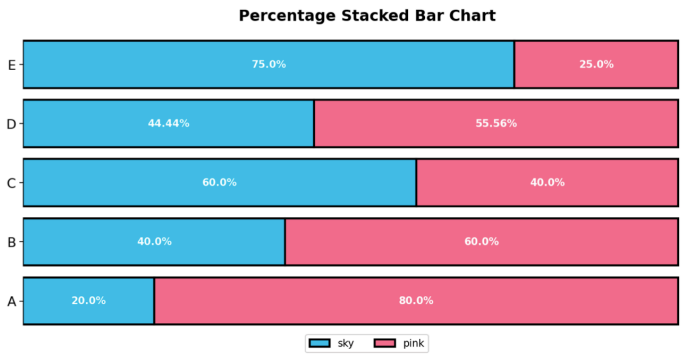
- 막대 그래프, bar chart, bar graph
- 개별 비교, 그룹 비교 모두 적합
- 수직(vertical): x축 범주, y축 값 (default)
- 수평(horizontal): y축 범주, x축 값 (범주가 많을 때 적합)
- multiple bar plot



- stacked bar plot

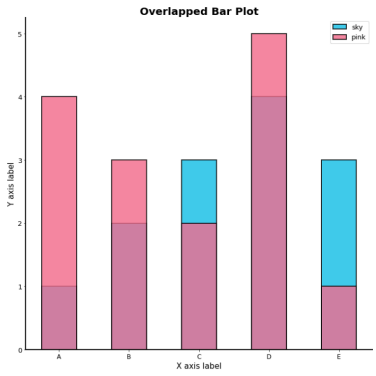


  - 각 bar에서 나타나는 그룹의 순서 항상 유지
  - 맨 밑의 bar의 분포 파악 쉬움
  - 그 외 분포 파악 어려움
  - 2개의 그룹이 positive/negative라면 축 조정 가능

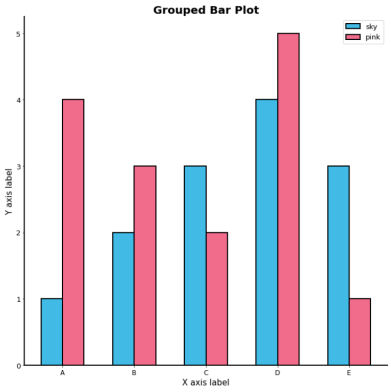


  - 응용하여 전체에서 비율을 나타내는 percentage stacked bar chart 있음

- overlapped bar plot



    - 2개 겹쳐서, 3개 이상은 파악 어려움
    - 같은 축 사용 => 비교 쉬움
    - 투명도 조정, 겹치는 부분 파악(**alpha**)
    - Bar plot보다는 Area plot에서 더 효과적
- grouped bar plot



    - 그룹별 범주에 따라 이웃되게 배치
    - 비교적 규현 까다도움

      `.set_xticks()`, `.set_xticklables()`
    - 모두 그룹이 5~7개 이하일 때 효과적

      그룹이 많으면 적은 그룹은 ETC으로 처리
- principle of proportion Ink

  실제 값과 그에 표현되는 그래픽으로 표현되는 잉크 양은 비례해야 함
    - 반드시 x축의 시작: **zero(0)**

      => 만약 차이 나타내고 싶으면, plot의 세로 비율 늘리기
    - 막 그래프에만 한정되는 원칙 X / area plot, donut chart 등등 다수의 시각화에서 적용
- 데이터 정렬하기
    - 정렬 필수
    - pandas에서는 `sort_values()` , `sort_index()`를 사용하여 정렬

1. 시계열: 시간순
2. 수치형: 크기순
3. 순서형: 범주의 순서대로

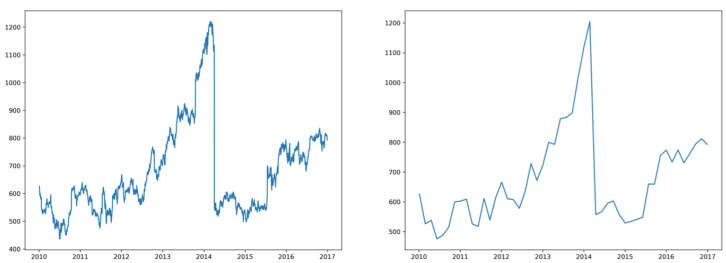
4. 명목형: 범주의 값 따라 정렬
    - 여러 가지 기분을 정렬 => 패턴 발견
    - 대시보드, interactive 제공하는 것이 유용
- 적절한 공간 활용
    - 여백, 공간 조정 / 가독성 높임
    - X/Y axis Limit `.set_xlim()`, `.set_ylime()`
    - Spines `.spines[spine].set_visible()`
    - Gap `width`
    - Legend `.legend()`
    - Margins `.margins()`
- 복잡함 & 단순함
    - 3D X
    - 정확한 차이 (EDA)
    - 큰 들에서 비교 및 추세 파악 (Dashboard)
    - 축, 디테일 등의 복잡함
        - Grid `.grid()`
        - Ticklabels `.set_ticklables()`
        - Text 어디에 추가 `.text()`, `.annotate()`
- 오차 막대 추가
    - Uncertainty 정보 추가 가능 (errorbar)
    - bar 사이 gap 0 => 히스토그램(histogram)

      `.hist()`, 연속된 느낌
    - 다양한 text 정보 활용 `.set_title()`, `.set_xlabel()`, `.set_ylabel()`

Line Plot

- 연속적으로 변화하는 값을 순서대로 점 나타내고 이를 선으로 연결한 그래프
- 시간/순서 변화에 적합, 추세 살피기 위해 사용

  => **시계열 분석에 특화**
- `.plot()` 사용
- 5개 이하의 선 사용 추천 (더 많은 선 중첩, 가독성 하락)
- 구별 요소: 색상, 마커, 선 종류
- 시시각각 변동하는 데이터 => noise로 패턴, 추세 파악 어려움



  => noise의 인지적인 방해를 줄이기 위해 **smoothing** 사용

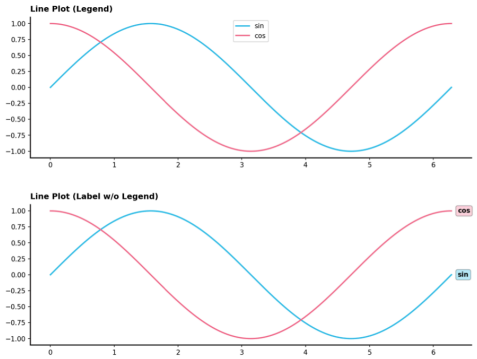
- bar plot와 달리 축을 꼭 0에 초점 둘 필요 X => 추세보기 위한 목적이므로
- grid, annotate 모두 제거
- 디테일한 정보, 표로 제공 추천
- 생략되지 않는 선에서 범위 조정, 변화율 관찰 (`.set_ylim()`)
- 규칙적인 간격이 아니라면
    - 그래프 상에서 규칙적 => 기울기 정보의 오해
    - 그래프 상에서 간격 다름 => 없는 데이터에 대해 있다고 오해

  규칙적인 간격의 데이터가 아니면 각 관측 값에 마크를 달라 오해 줄이기
- 보간: 점과 점 사이에 데이터가 없기에 이를 잇는 방법
    - 없는 데이터를 있다고 착각, 작은 차이 없앨 수 있음

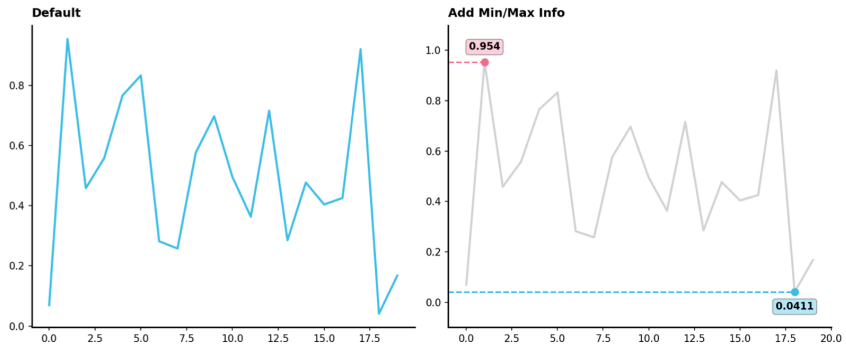
      => **일반적인 분석에서는 지양**
- 이중 축(dual axis) 사용

  한 plot에 대해 2개 축 사용
    - 같은 시간 축에 대해 서로 다른 종류의 데이터를 표현하기 위함 `.twinx()`
    - 한 데이터에 대해 다른 단위 `.secondary_zaxis()`, `.secondary_yaxis()`

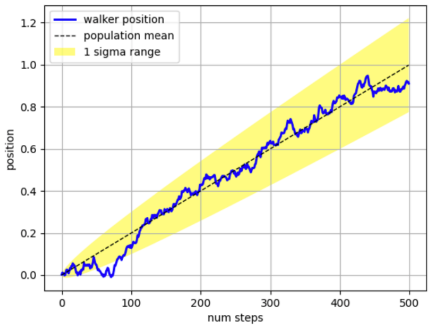
  2개의 plot 그리는 것, 이중 축 사용 => **지양**
- 범례 대신, 라인 끝 단에 레이블 추가 시, 식별 도움



- min/max, 등 마크, 텍스트, 등 일부 추가 시 도움

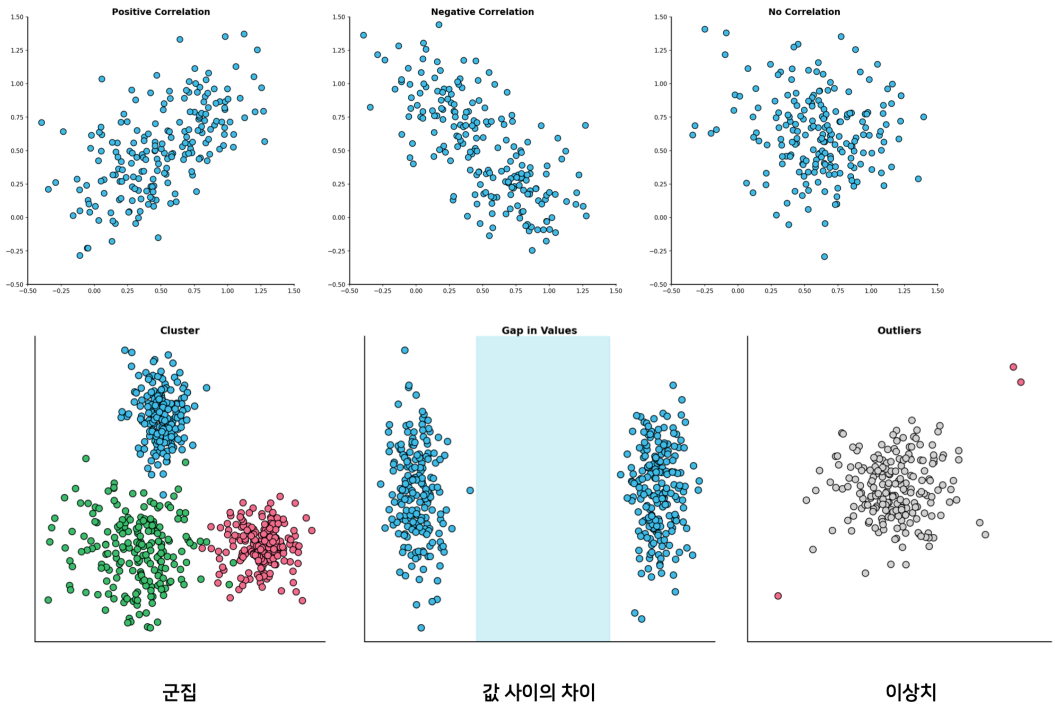


- 연한 색으로 uncertainty 표현 가능 (신뢰구간, 분산)



Scatter Plot

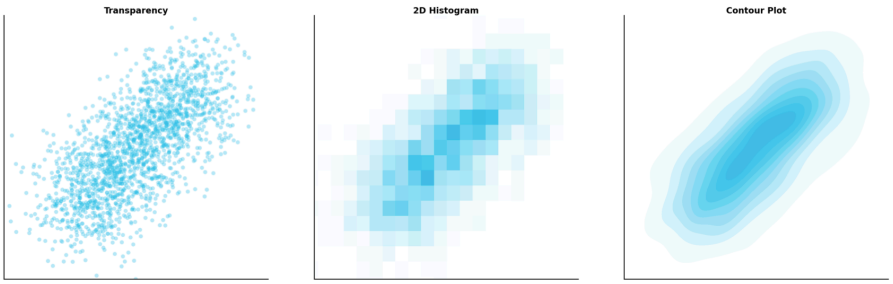
- 산점도
- 두 feature 간의 관계 알기 위해
- 점의 색, 모양, 크기 variation 사용 가능
- 목적: 상관 관계 확인 (+/-/없음)



- Overplotting

  점 많아짐 => 점 분포 파악 어려움

  - 투명도 조정
  - 지터링 (jittereing): 점 위치 약간씩 변경
  - 2차원 히스토그램: 히트맵 사용, 깔끔한 시각화
  - Contour plot: 분포, 등고선 사용하여 표현



- 색: 연속 - gradient / 이산 - 개별 색상
- 마커: 구별 어려움, 크기 안 고름
- 크기

  버블 차트(bubble chart), 구별 쉬움, 오용 쉬움

  관계보다 각 점간 비율 초점, SWOT 분석에 활용 가능
- 인과 관계(causal relation) / 상관 관계(correlation)
- **추세선**: scatter 패턴 유추 가능

  2개 이상 시, 가독성 떨어짐
- grid는 지양, 사용 시 최소한, 색: 무채색
- 범주형 포함 관계: **heatmap, bubble chart** 추천