YBIGTA 14기 교육세션

EDA

안녕하세요!

YBIGTA 디자인팀 10기 우혜원입니다!

산업공학과이며 컴퓨터과학을 복수전공하고 있습니다.

디자인팀에 합류한지 한 학기밖에 되지 않아서 기수는 높지만 실력은… 그닥….

EDA를 세시간만에 모두 가르친다는 건 솔직히 힘든 일이고 오늘은 기초적인 것만 다루어 보겠습니다.

Exploratory Data Analysis, 탐색적 자료 분석?

위키피디아에 검색해본 EDA :

탐색적 자료 분석(영어: Exploratory data analysis)은 존 튜키라는 미국의 저명한 통계학자가 창안한 자료 분석 방법론이다.
기존의 통계학이 정보의 추출에서 가설 검정 등에 치우쳐 자료가 가지고 있는 본연의 의미를 찾는데 어려움이 있어
이를 보완하고자 주어진 자료만 가지고도 충분한 정보를 찾을 수 있도록 여러가지 탐색적 자료 분석 방법을 개발하였다.
대표적인 예로 박스플롯을 들 수 있다.
탐색적 자료 분석을 통하여 자료에 대한 충분한 이해를 한 후에 모형 적합 등의 좀 더 정교한 모형을 개발할 수 있다.

EDA의
네 가지 구성 요소

Revelation. 시각적 수단을 사용하여 효과적으로 전달하자!

Residual. 흐름에서 크게 벗어나는 값을 잘 해석하자!

Re-expression. 변수를 적당한 척도로 바꾸어 파악하자!

Resistance. 아웃라이어, 결측치 등에 영향을 받지 않는 적절한 통계량을 사용하자!

#00

Data Schema 파악

본격적인 EDA에 앞서 어떤 분야의 데이터인지,

각 column이 의미하는 바가 무엇인지 정확하게 알고 넘어가야한다.

변수의 의미
& 변수의 종류

변수는 측정될 수 있는 어떠한 특징, 숫자, 양 등을 의미

모집단 내에서 어느 값이나 다양하게 가질 수 있기 때문에 '변수'라고 불림

변수의 종류에는 Numerical (수치형) & Categorical (범주형) 크게 두 가지

여기에 Date (날짜) 데이터는 크게 보면 범주형에 속하지만 따로 추가로 떼어놓고 생각해보자.

#01

Numerical (수치형)

값이 숫자로 이루어진 변수

예를 들면, YBIGTA 액팅 인원 수, GPA, 비트코인 가격 등이 있다.

다시 Discrete (이산형) & Continuous (연속형) 변수로 나누어진다.

round number로만 이루어진 경우 Discrete, 그렇지 않은 수로도 이루어진 경우 Continuous

#02

Categorical (범주형)

값이 범주의 그룹으로 이루어진 변수

예를 들면, YBIGTA 팀, 알파벳 학점, 비트코인 종류 등이 있다.

다시 Ordinal (순서형) & Nominal (명목형) 변수로 나누어진다.

범주 내에 의미적으로 순서가 있는 경우 Ordinal, 그렇지 않은 경우 Nominal

#03

Date

값이 날짜로 이루어진 변수

보통 시각화를 할 때, 다른 범주형 변수와는 다른 방식을 택하기 때문에 따로 떼어놨다.

평일 / 주말을 나누거나 요일을 추가하는 등의 작업을 자주 거친다.

ApGroupId	AdNetworkType2	Age	Clicks	Impressions	Slot	…
78db034136	S	24	3	0	S_2	…
68a0110c69	S	336	1	13	S_2	…
21af1035af	P	43	3	419	P_1	…

1. Ad라고 되어있는 것으로 보아 광고 관련 데이터
2. Clicks와 Impressions는 클릭과 노출을 뜻하므로 온라인 광고 관련 데이터
3. Click이 광고 클릭 횟수이고 Impressions이 노출 횟수라면 항상 Click ≤ Impressions이어야 하므로 첫번째 행은 잘못된 정보

EDA의
네 가지 구성 요소

Revelation. 시각적 수단을 사용하여 효과적으로 전달하자!

Residual. 흐름에서 크게 벗어나는 값을 잘 해석하자!

Re-expression. 변수를 적당한 척도로 바꾸어 파악하자!

Resistance. 아웃라이어, 결측치 등에 영향을 받지 않는 적절한 통계량을 사용하자!

#01
Resistance

이상치, 결측치, 입력 오류 등에 영향을 받지 않는,
저항성을 가진 통계량으로 데이터를 표현 (데이터가 부분적으로 바뀌어도 영향을 받지 않아야 함)

#02
Residual

Residual (잔차) : 각 값이 흐름으로부터 얼마나 벗어나 있는지를 나타내는 값

흐름을 크게 벗어나는 값이 있을 때, 왜 그러한 값이 등장했는지를 파악해야 함!

ApGroupId	AdNetworkType2	Age	Clicks	Impressions	Slot	…
78db034136	S	24	3	0	S_2	…
68a0110c69	S	336	1	13	S_2	…
21af1035af	P	43	3	419	P_1	…

Age Column에서 336이라는 Outlier가 있음을 확인!

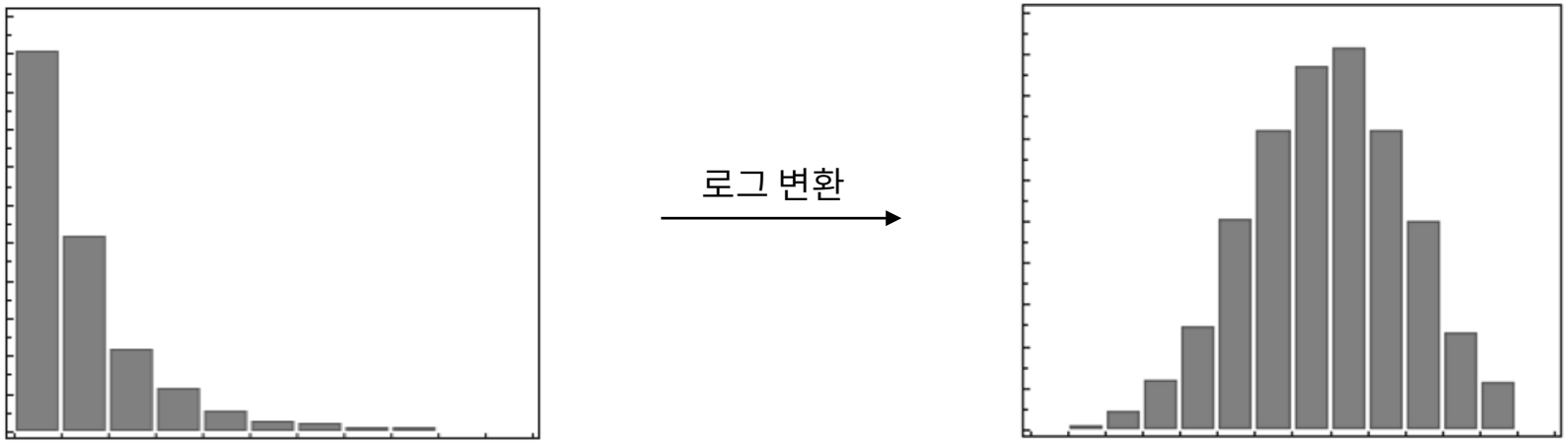
이렇게 흐름에서 크게 벗어나는 값을 잘 해석해야 하는데,

이 데이터의 경우 1. 원래 데이터의 값이 33이나 36인데 오타가 발생했거나 2. Age가 사람의 나이가 아니거나(…) 등으로 추측해볼 수 있다.

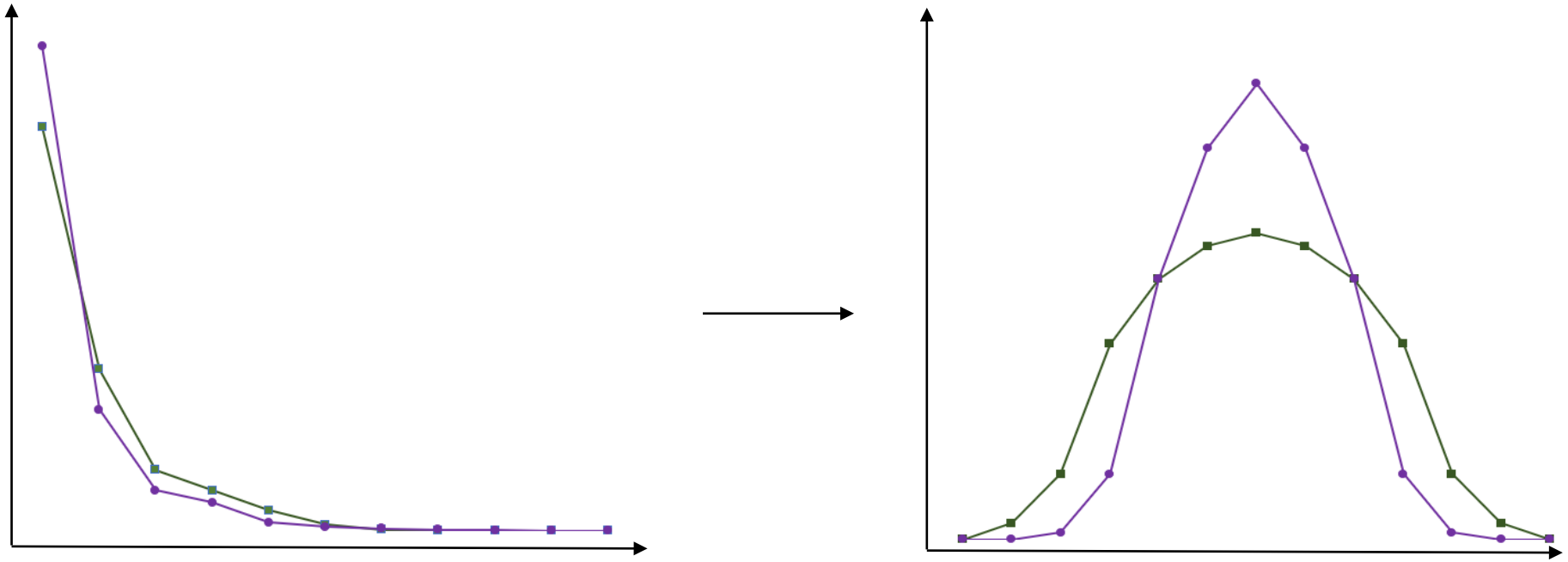
#03
Re-expression

데이터 분석 / 해석을 더 편리하게 할 수 있도록
변수를 적당한 척도로 바꾸어주는 것.

Ex)
Log-transformation



가장 대표적인 변환인 로그 변환의 경우 데이터 간 편차를 줄여주어 왜곡과 첨도(분포가 뾰족한 정도)를 줄여준다.



편차가 큰 두 그룹의 데이터를 비교할 때에도 유용하게 써먹을 수 있음!

#04
Revelation

도표(graph)라는 수단을 통하여 정보를 명확하고 효과적으로 전달하는 것

데이터 시각화! (Data Visualization)

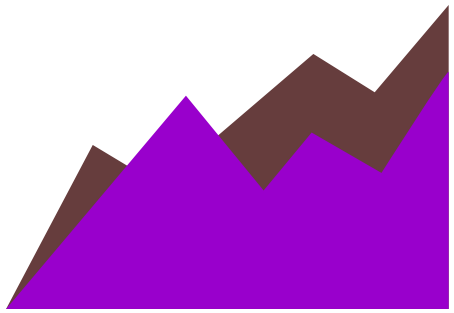
무엇을
시각화 할 것인가?

차트의 기능에 따라 크게 여덟 가지로 나눈다면 다음과 같이 나눌 수 있다.

Time-series, Ranking, Part-to-whole, Deviation, Distribution, Correlation, Nominal comparison, Geographic

각각 하나씩 들여다보자.

#01
Time-series



연속적인 시간 흐름에서의 데이터 값 변화

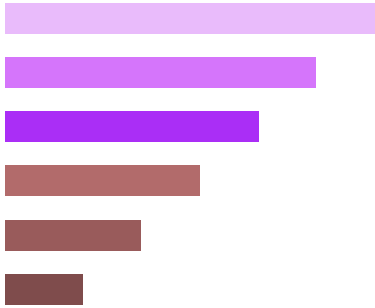
경향성을 찾고 개별적인 데이터보단 전체적인 흐름을 보는 것이 좋지만, 예외적인 데이터가 있는 경우에는 세부적으로 관찰하자.

보통 시간이 이산형인 경우에는 바 차트나 스캐터플랏, 연속형인 경우에는 라인 차트를 사용한다.

ex. 월별 판매량, 기온 변화 등

#02

Ranking



두 개 이상의 값의 상대적 크기 비교

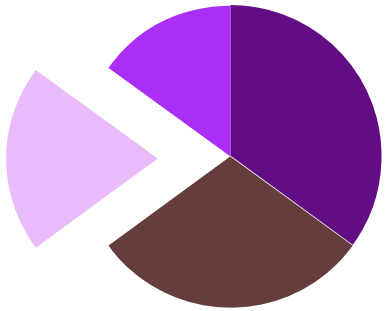
데이터를 오름차순이나 내림차순으로 정렬하여 시각화하되 강조하고 싶은 걸 상단에 두자.

보통 바 차트를 사용한다.

ex. 월별 기온 순위 등

#03

Part-to-whole



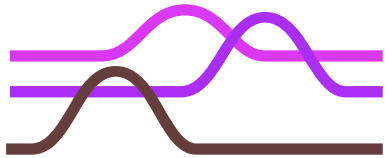
큰 전체 데이터에 대한 하위 집합 데이터의 비율 비교

전체에 대한 하위 카테고리의 부분적인 비율을 관찰하자.

주로 파이 차트나 누적 바 차트를 사용한다.

ex. 특정 제품 구매 고객 비율 등

#04
Deviation



데이터끼리 서로 얼마나 관련 있는 지, 특히 평균과 주어진 데이터의 차이

각각의 데이터를 강조하려면 바 차트를, 전체적인 관계를 강조하려면 라인 차트를 사용해보자.

단, 시계열이 아닌 데이터는 보통 바 차트만을 사용한다.

ex. 비 오는 날과 맑은 날의 놀이동산 티켓 판매 차이 등

#05
Distribution

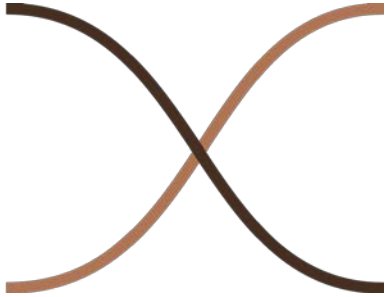
중심 값 주변의 데이터 분포

범위나 중심값, 특히 정규분포인지와 같은 전체적인 분포정도에 집중하자.

보통 히스토그램이나 덴시티 플랏을 사용하고, 선이나 막대 아래의 면적은 확률을 의미한다.

ex. 농구 선수 키 분포 등

#06
Correlation



두 개 이상의 데이터의 양, 혹은 음의 상관관계

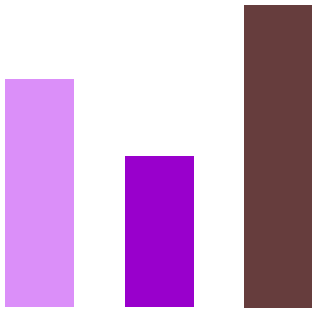
다른 변수 간의 관계를 관찰해볼 수 있지만, 그 관계가 인과관계를 의미하지는 않음에 유의하자.

보통 스캐터플랏을 사용하나, 바 차트를 사용하기도 한다.

ex. 교육 수준에 따른 연봉 수준 등

#07

Nominal Comparison



순서가 없는 하위 카테고리 간의 양적 비교

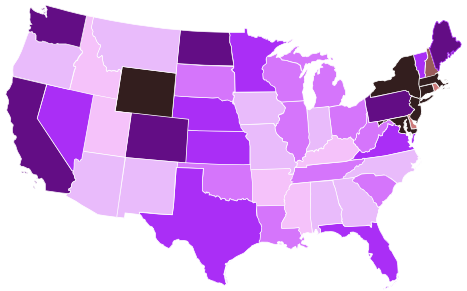
바 차트의 축은 그래프가 왜곡되지 않도록 꼭 0부터 표현하자.

보통 바 차트를 사용한다.

ex. 웹 사이트별 방문자 수 등

#08

Geographic



위치 관련 데이터

위도와 경도를 사용하여 데이터를 나타낸다.

파이썬에서 지도 시각화를 하고 싶다면 **Folium**을 사용하자.

ex. 주별 GDP 등

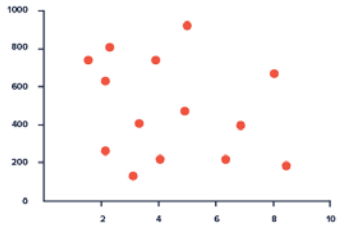
Charts

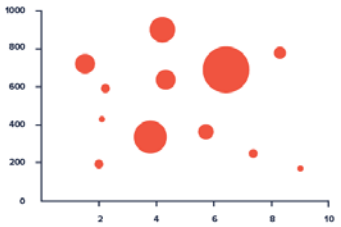
모든 차트를 다 아는 것은 굉장히 힘들고 굳이 다 알 필요도 없으므로,

대표적인 차트 몇 가지와 만들 때 주의해야할 사항을 소개하고 넘어가려고 한다.

궁금한 차트가 있다면 **이 곳**에서 검색해볼 것!

#01 Scatterplot

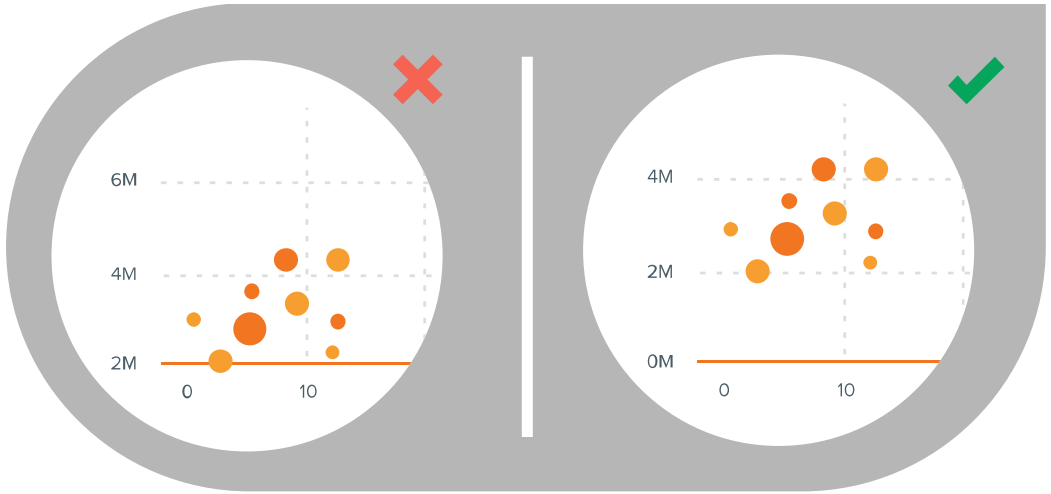




직교 좌표계 상에서 두 변수의 값을 보여주기 위한 차트

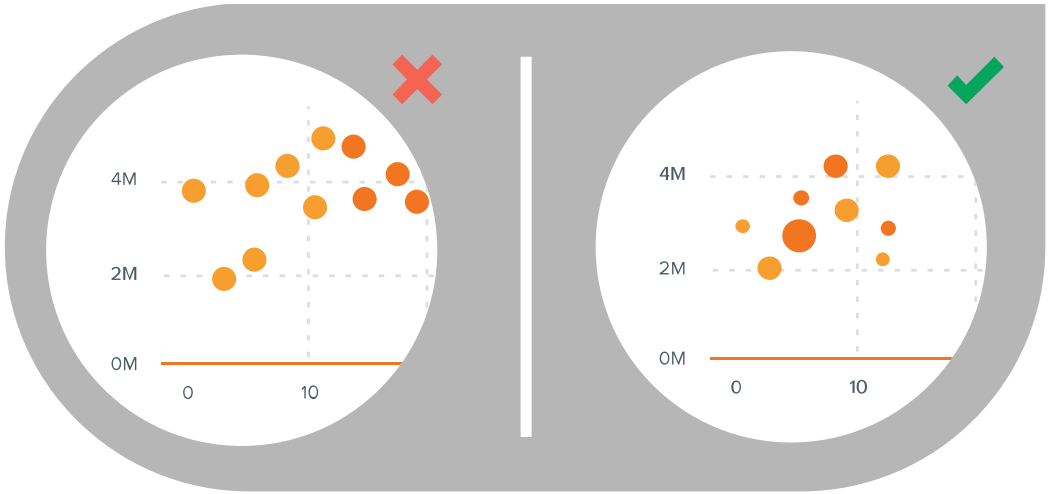
보통 상관관계나 분포를 파악하기 위해 사용하고, 특히 버블 차트의 경우에는 비교나 순위 파악에도 유용하다.

보통 두 개의 연속형 변수를 인풋 값으로 갖는다.



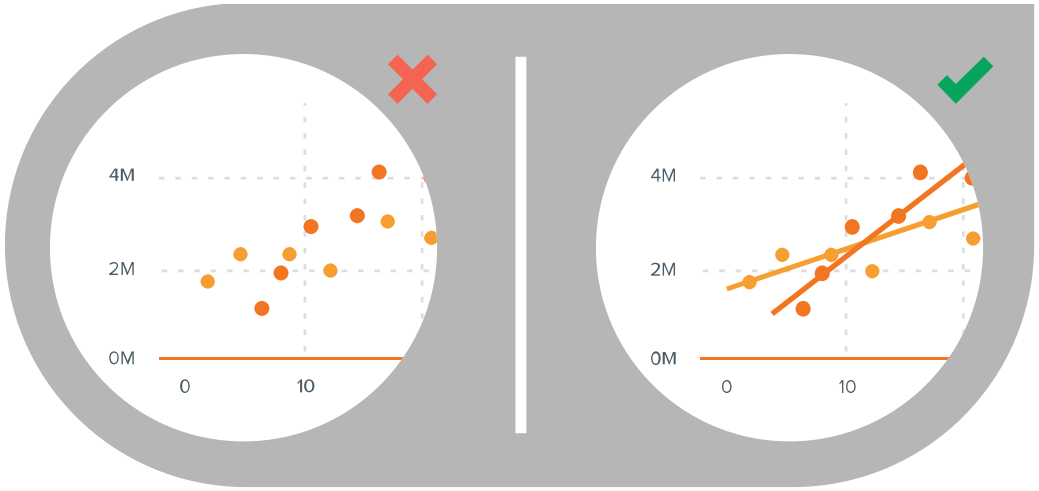
y축은 0에서부터 시작하자

그렇지 않으면 원이 반토막 날 수 있다



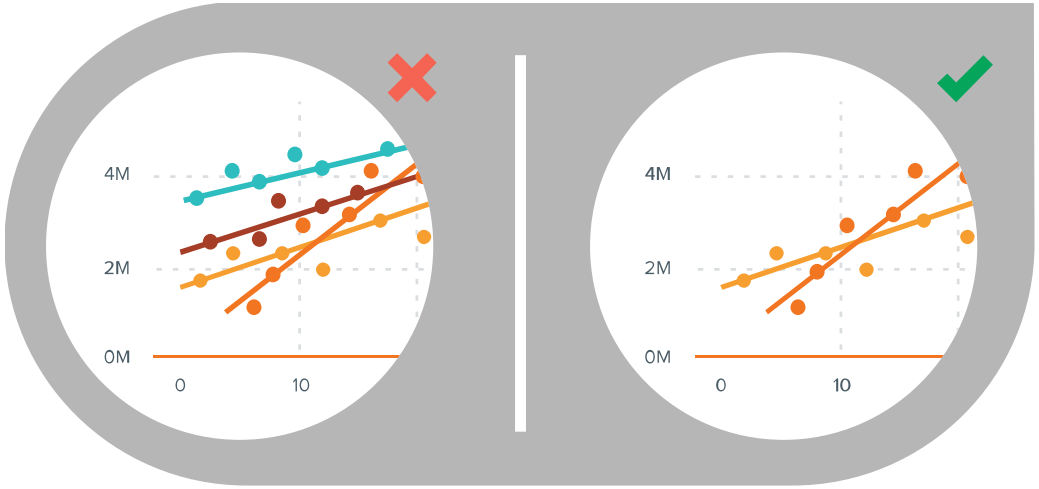
더 많은 변수를 포함하자

색을 조절하거나 크기를 조절하는 등 추가적인 데이터를 포함하자



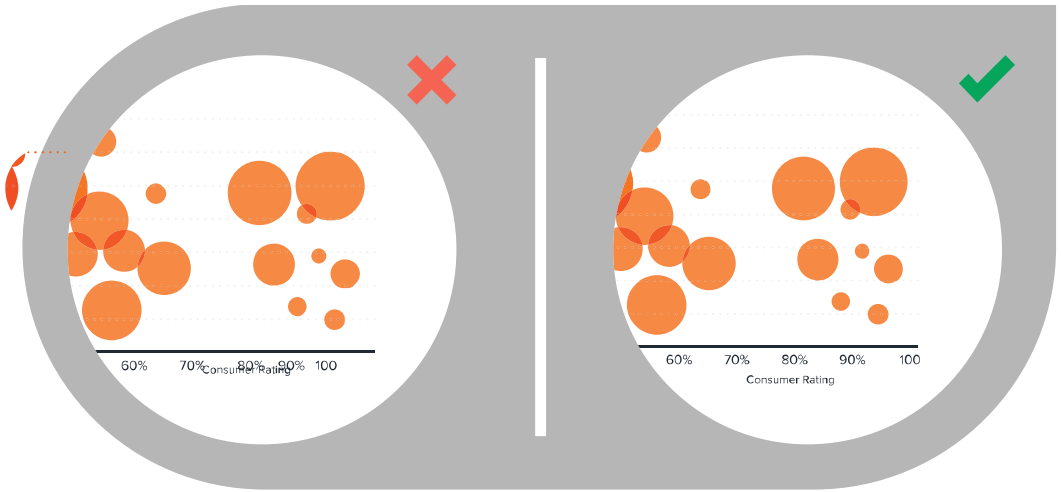
추세선을 사용하자

경향성을 보여주기 위하여 변수간 상관관계를 그리자



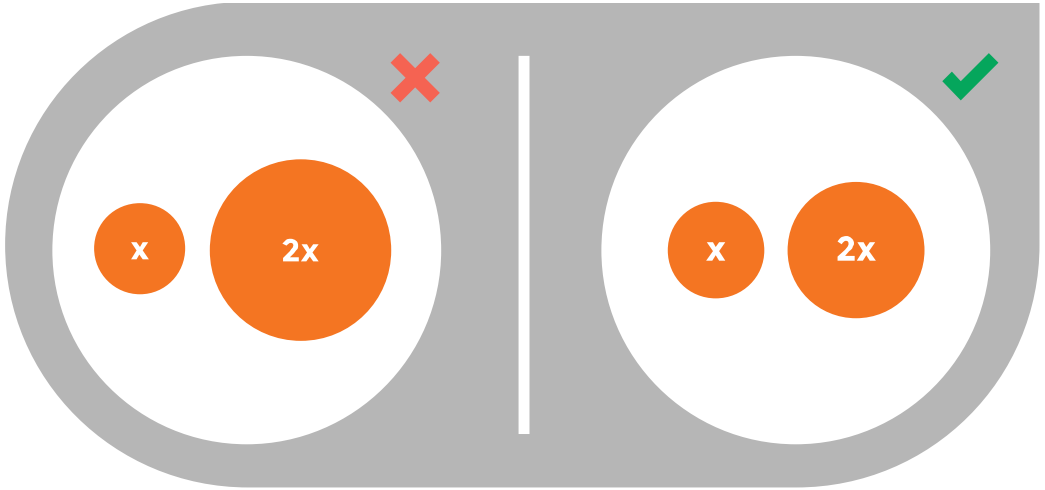
추세선은 두 개 이상 비교하지 말자

너무 많은 선은 해석하기 어렵다



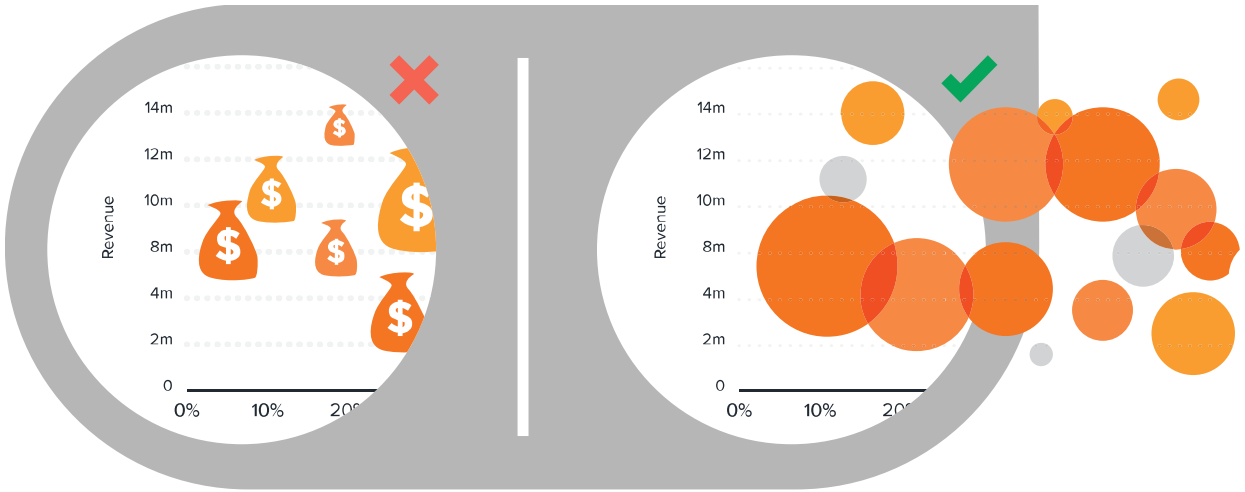
라벨 이름이 보이는 지 확인하자

어떤 관계인지 알 수 있도록 라벨을 쉽게 인식되게 달자



버블 크기를 적절하게 조절하자

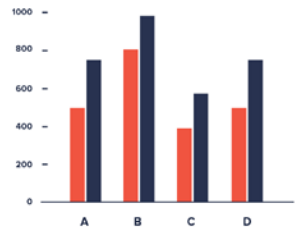
직경이 아니라 넓이를 기준으로 버블 크기를 조절하자

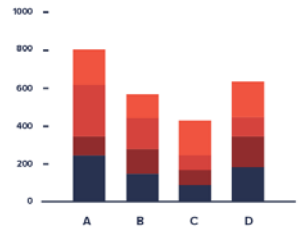


되도록 원을 사용하자

너무 디테일한 모양은 차트의 왜곡을 초래한다

#02
Bar Chart

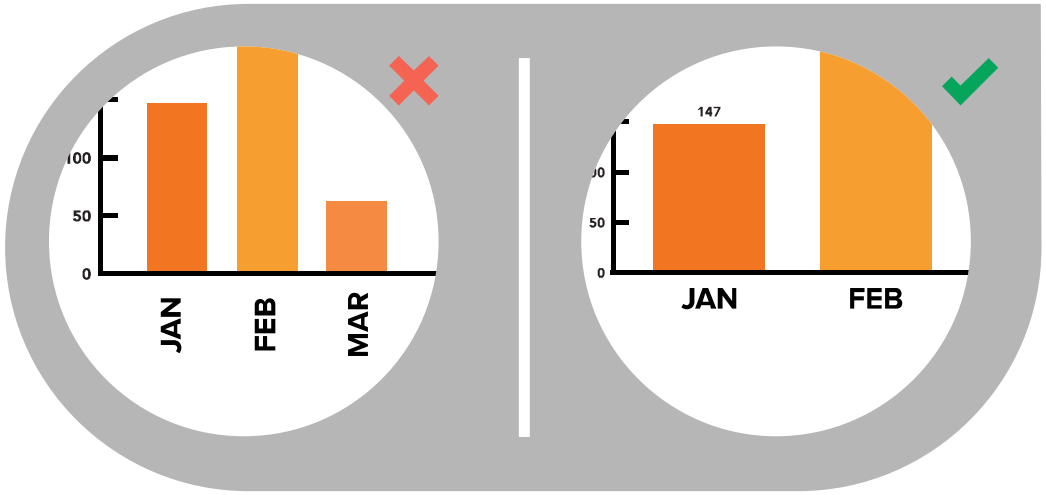




상대적으로 직사각형의 길이로 값을 나타내는 차트

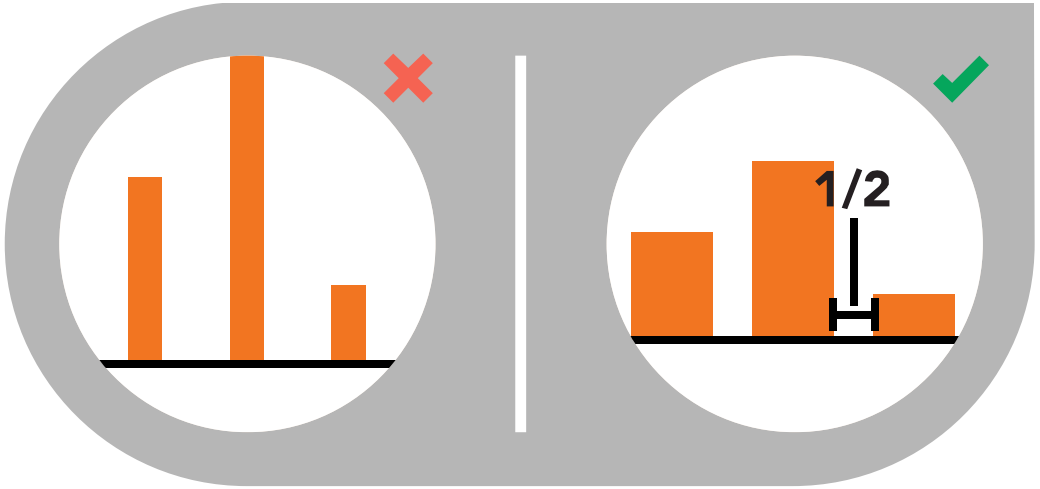
거의 대부분의 시각화를 할 수 있다.

보통 한 개의 수치형 변수와 범주형 변수를 인풋 값으로 갖는다.



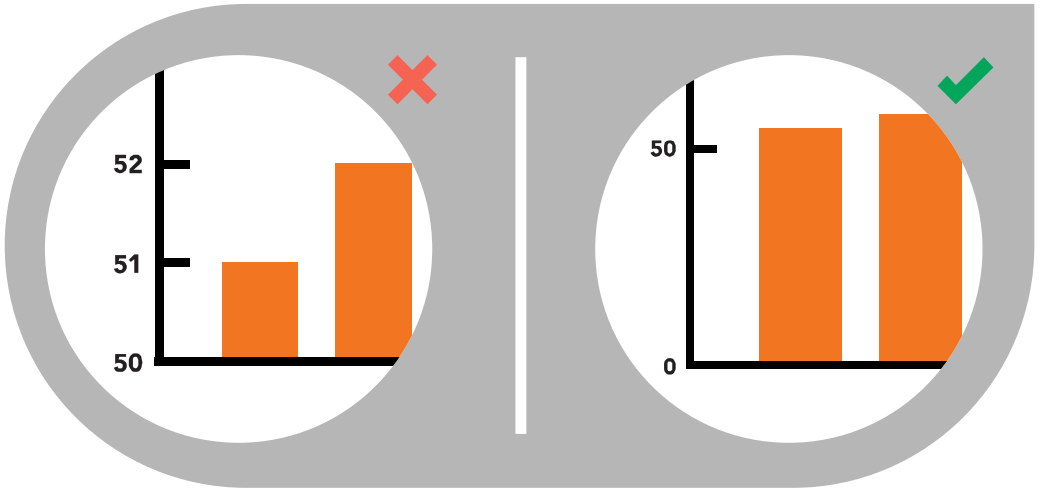
라벨은 가로로 쓰자

대각선이나 세로는 알아보기 힘들다



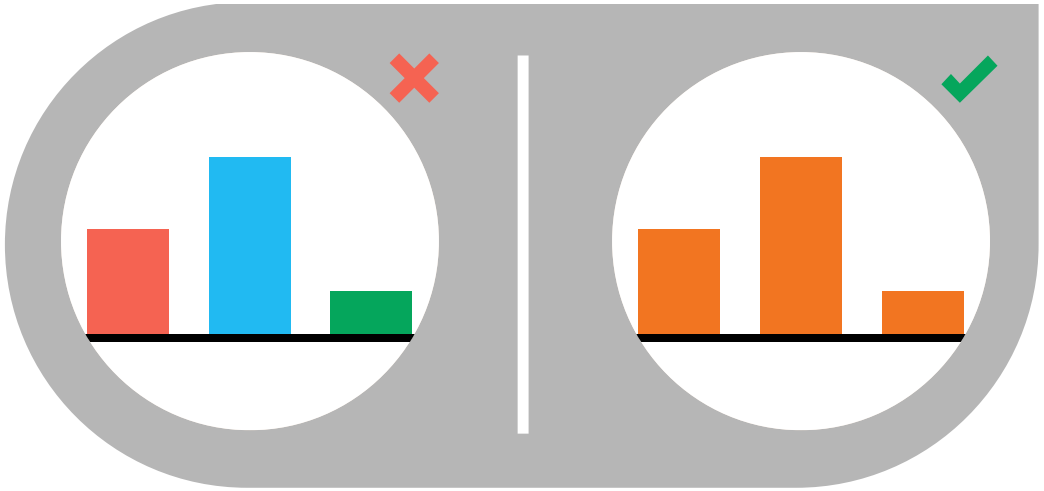
바 사이의 간격을 적절하게 설정하자

바 길이보다 간격이 더 넓은 것을 지양하자



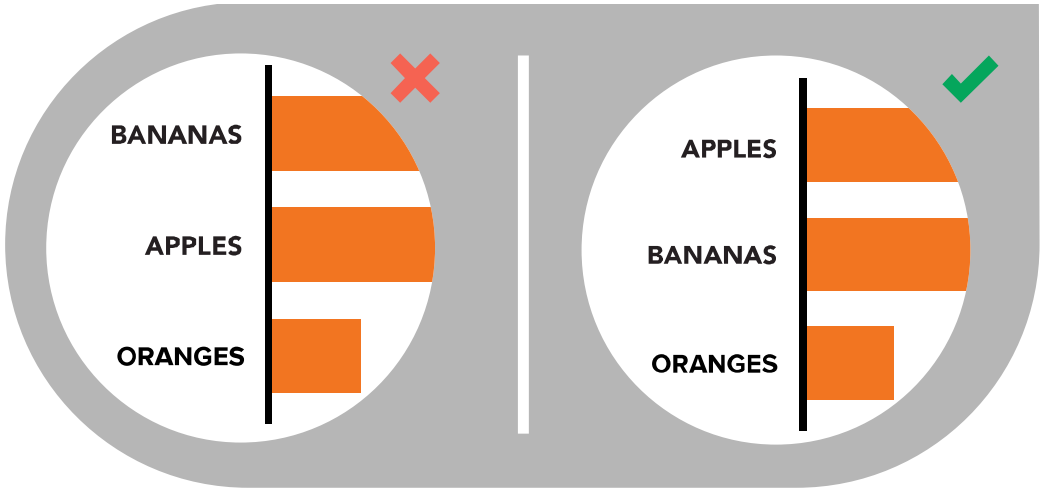
y축은 0에서부터 시작하자

0이상에서 시작하면 전체 값이 충분히 반영되지 않는다



일관된 색을 사용하자

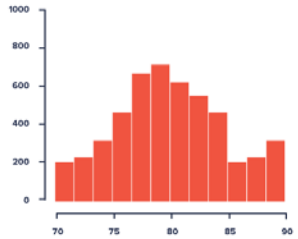
강조하고 싶은 경우에만 다른 색을 사용하자



적절하게 배열하자

알파벳순, 값의 오름차순, 내림차순 등으로 배열하자

#03 Histogram

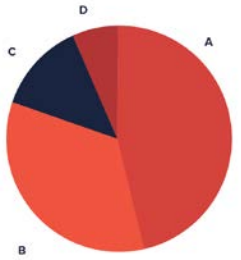


수치형 데이터를 비닝하여 해당 구간의 개수를 직사각형의 길이로 나타내는 차트

보통 분포를 파악하기 위해 사용한다.

보통 구간과 해당 구간의 데이터 수를 인풋 값으로 갖는다.

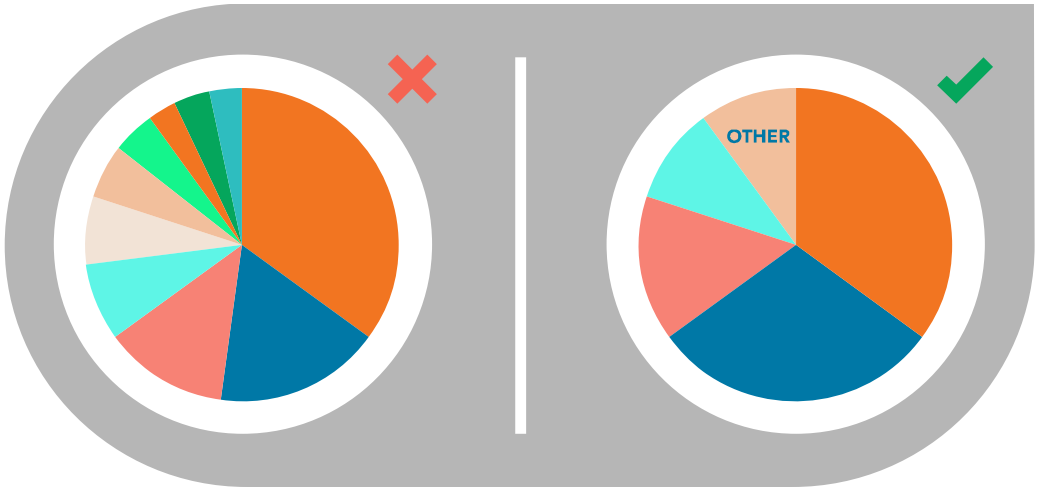
#04
Pie Chart



원의 각도의 크기로 값의 비율을 나타내는 차트

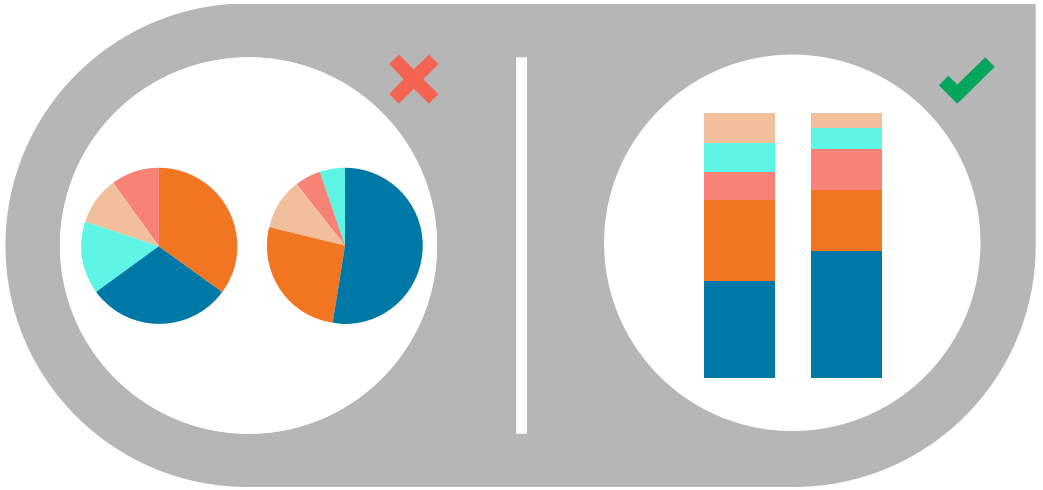
보통 전체에 대한 부분 비율 파악에 유용하다.

보통 한 개의 범주형 변수와 해당 값에 대한 비율을 인풋 값으로 갖는다.



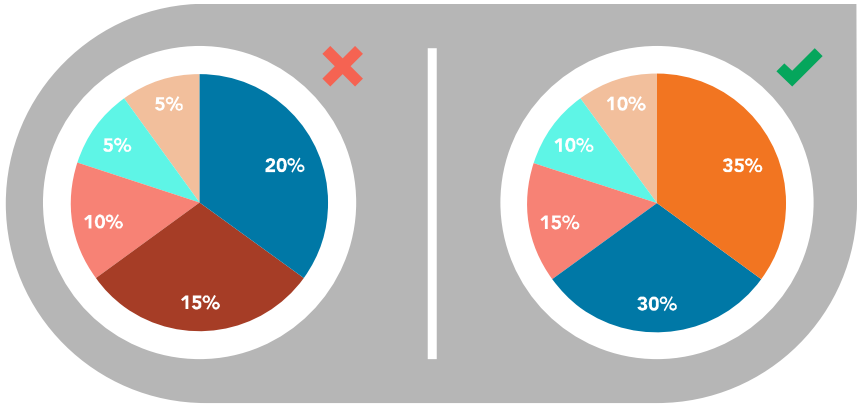
다섯 개 이상 비교하지 말자

값이 작아질 수록 구분하기 힘들어진다



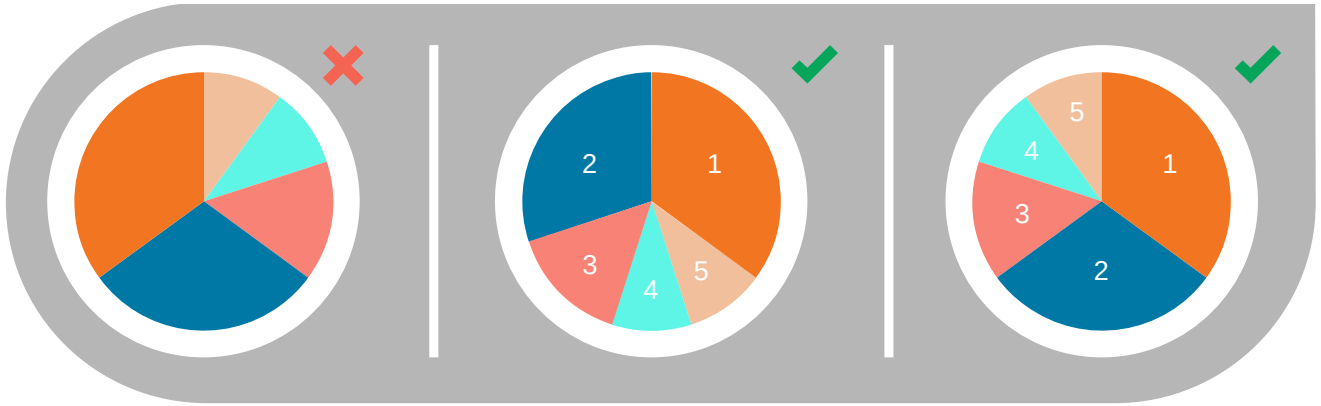
비교 목적으로 파이 차트를 사용하지 말자

사이즈 별 비교가 굉장히 어렵다



합이 100%가 되도록 하자

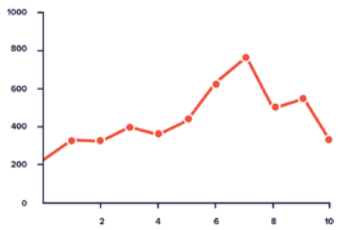
비율 값이 100%가 되는지 늘 확인하자

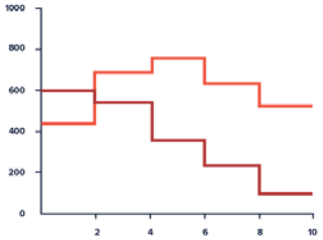


슬라이스 순서를 올바르게 하자

반시계 방향 혹은 시계 방향에 맞게 설정하자

#05
Line Chart

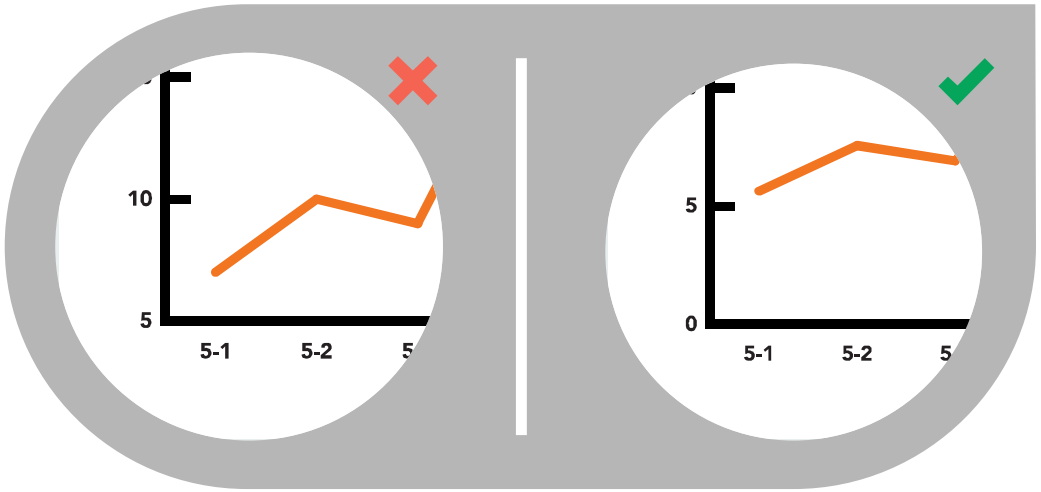




순서가 있는 값의 연속적 데이터를 선으로 나타내는 그래프

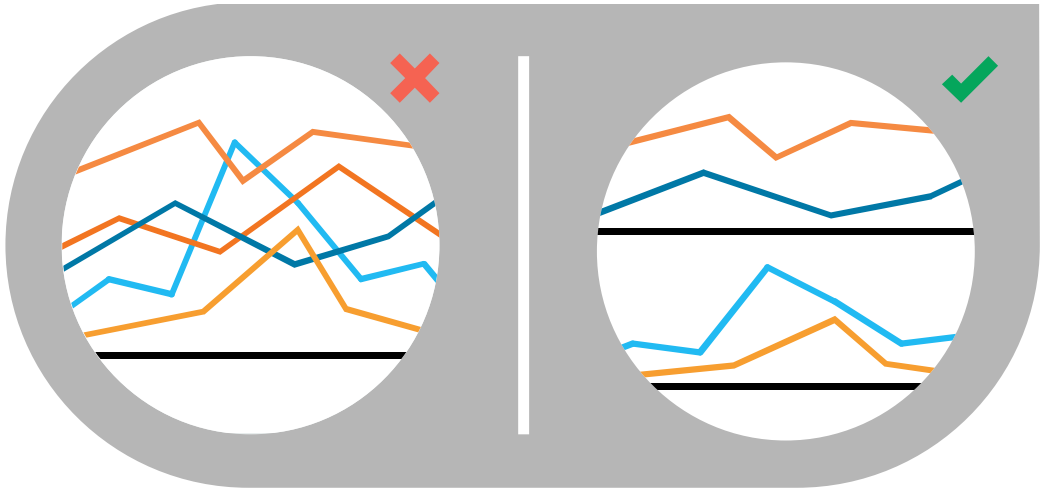
보통 분포나 시계열 데이터를 파악하기 위해 사용한다.

보통 한 개의 순서가 있는 변수와 수치형 변수를 인풋 값으로 갖는다.



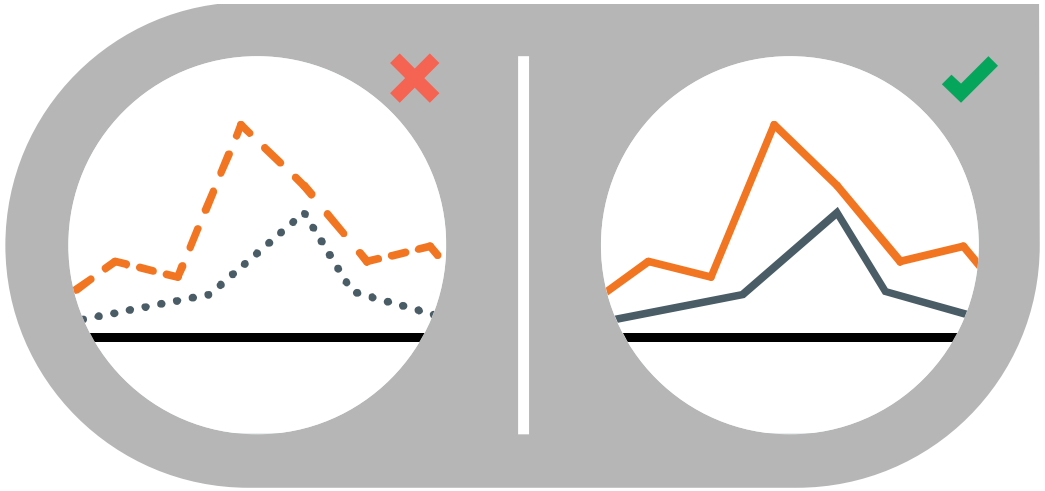
되도록 0을 포함하는 그래프를 그리자

미묘한 변동도 유의미한 경우가 있다



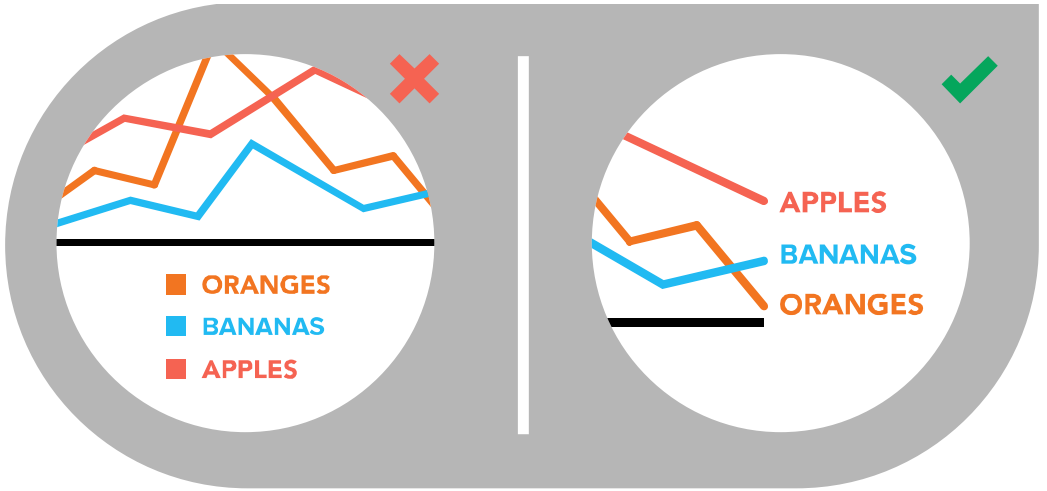
선을 네 개 이상 사용하지 말자

차라리 서브플랏으로 나누어 그리자



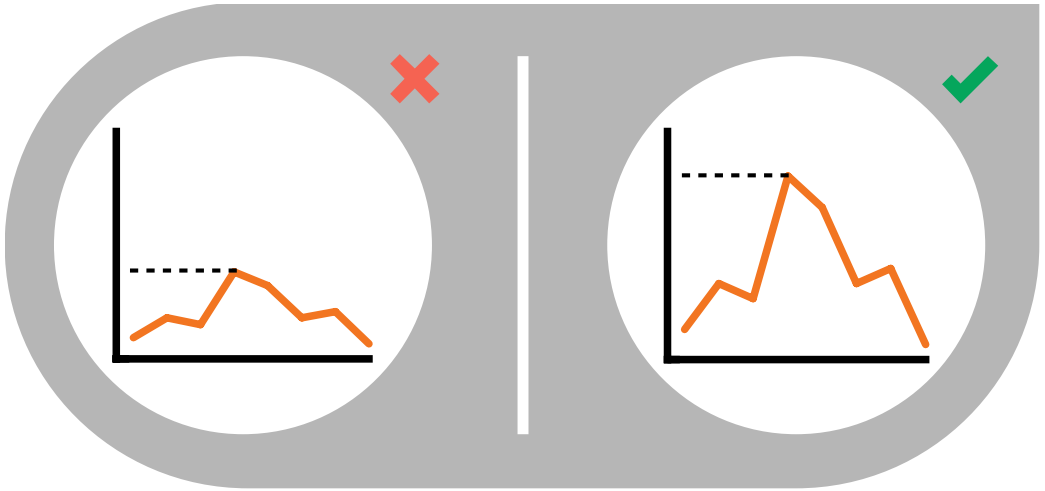
실선만 사용하자

점선은 거슬린다



선 옆에 라벨링 하자

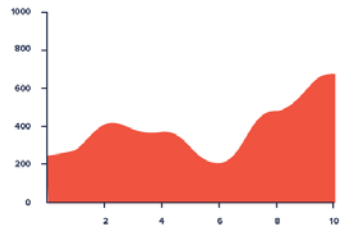
범례를 따로 두면 인지하는 데 더 어렵다

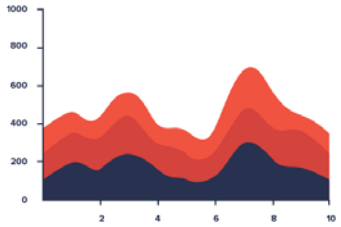


올바른 높이를 사용하자

전체 차트의 삼분의 이 정도를 차지하게 그리자

#06
Area Chart

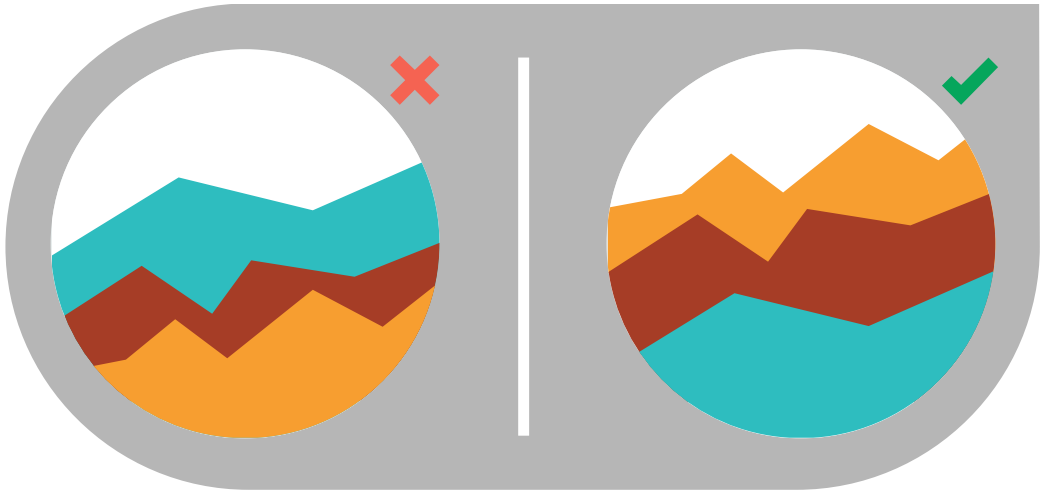




라인 차트와 거의 흡사하나 볼륨을 표현할 수 있다는 데 차이가 있는 차트

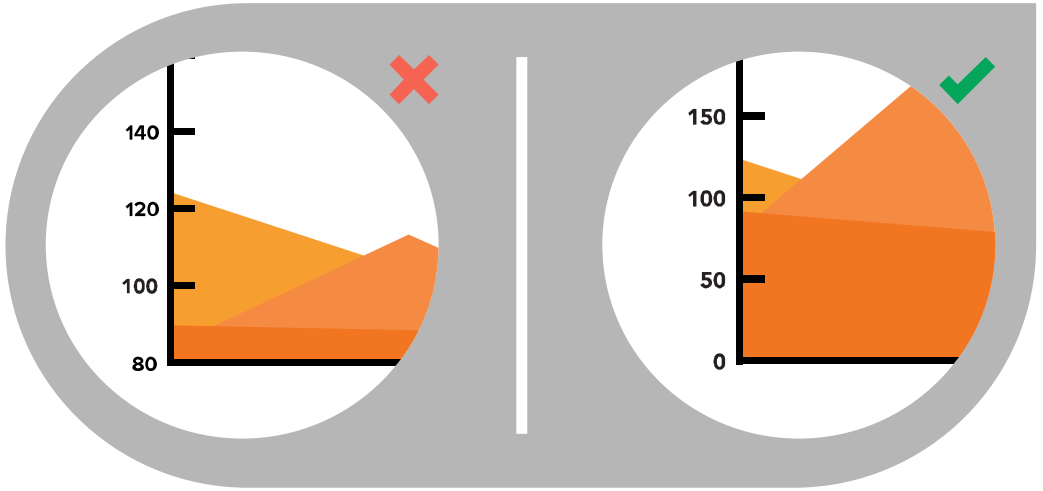
보통 비교, 분포나 시계열 데이터를 파악하기 위해 사용한다.

보통 한 개의 순서가 있는 변수와 연속형 변수를 인풋 값으로 갖는다.



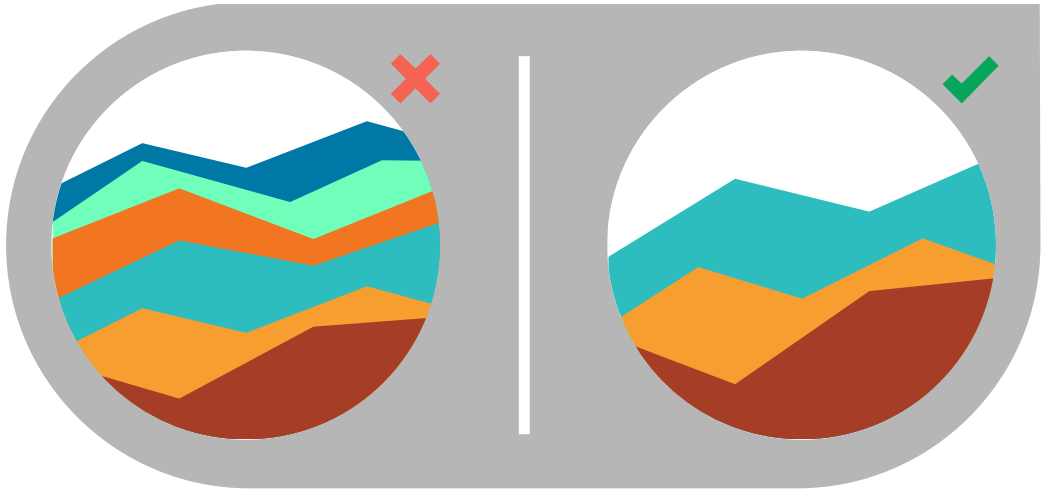
읽기 쉽게 만들자

가장 큰 값을 가장 뒤에 배치하자



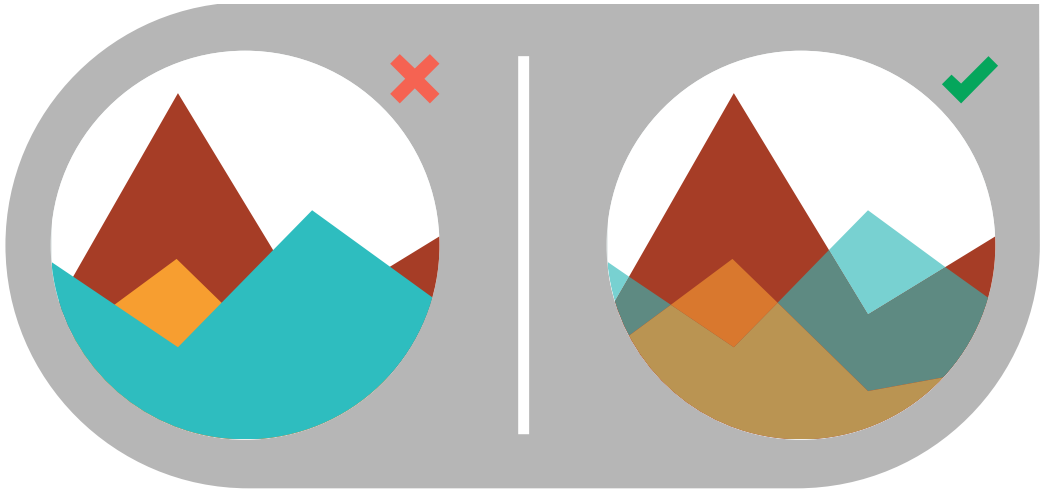
y축은 0에서부터 시작하자

0이상에서 시작하면 전체 값이 충분히 반영되지 않는다



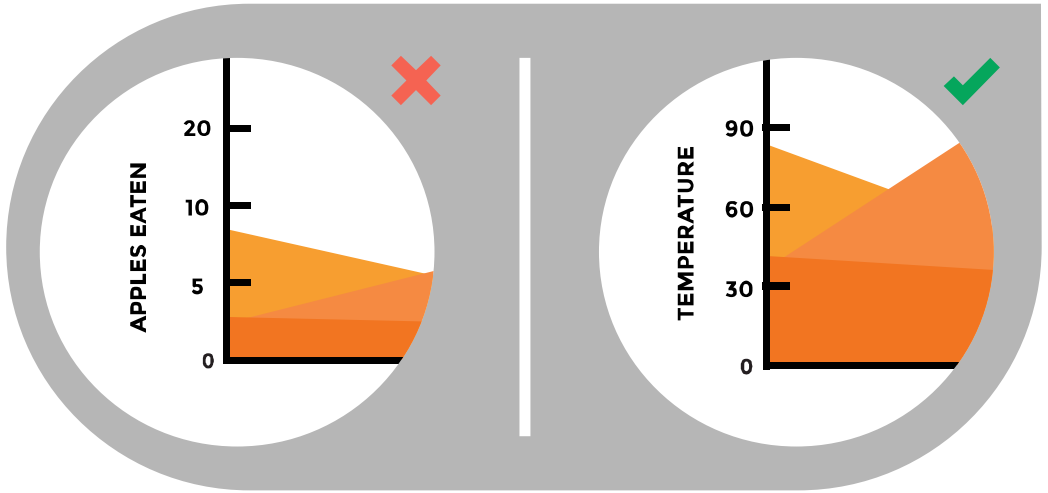
네 개 이상 그리지 말자

차이를 알아보기 더 힘들어진다



투명도를 조절해보자

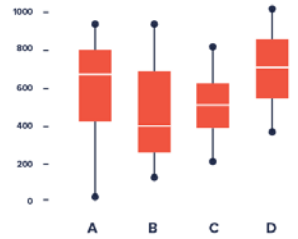
겹치는 부분이 발생하는 경우에 좋다

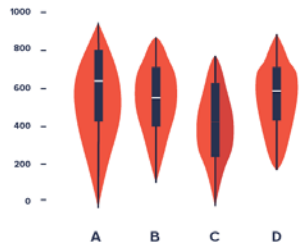


이산형 변수를 시각화하지 말자

선은 연속형일 때만 유의미한 중간값이다

#07
Box Plot



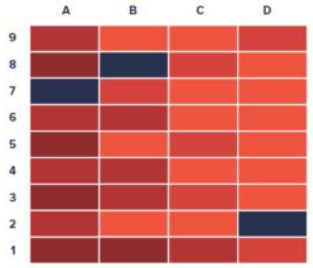


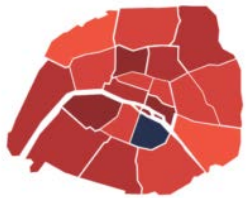
사분위수로 분포를 보여주기 위한 차트

보통 분포를 파악하고 다른 값과 비교하는 데 유용하다.

보통 두 개 이상의 범주형 변수에 대한 연속형 분포를 인풋 값으로 갖는다.

#08 Heatmap

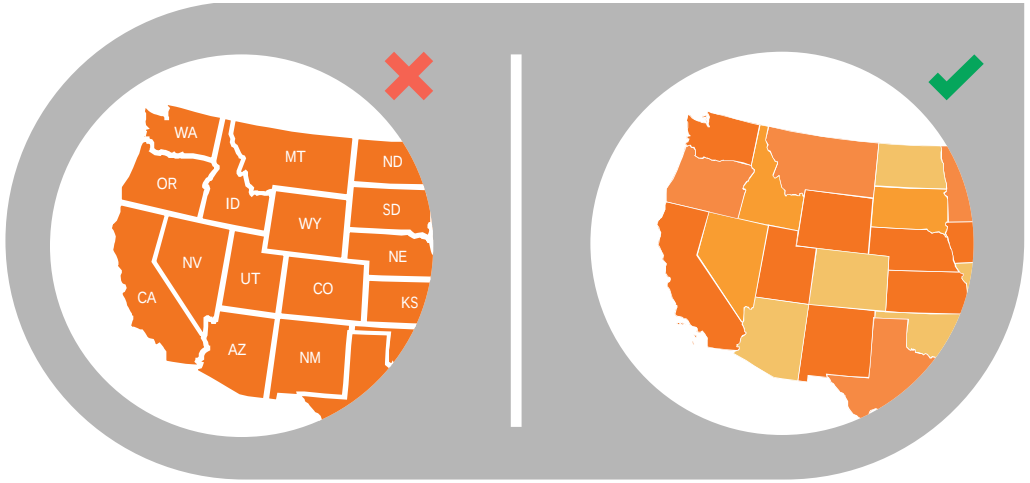




다양한 값에 대하여 색깔 변화를 통하여 값의 분포를 나타내는 차트

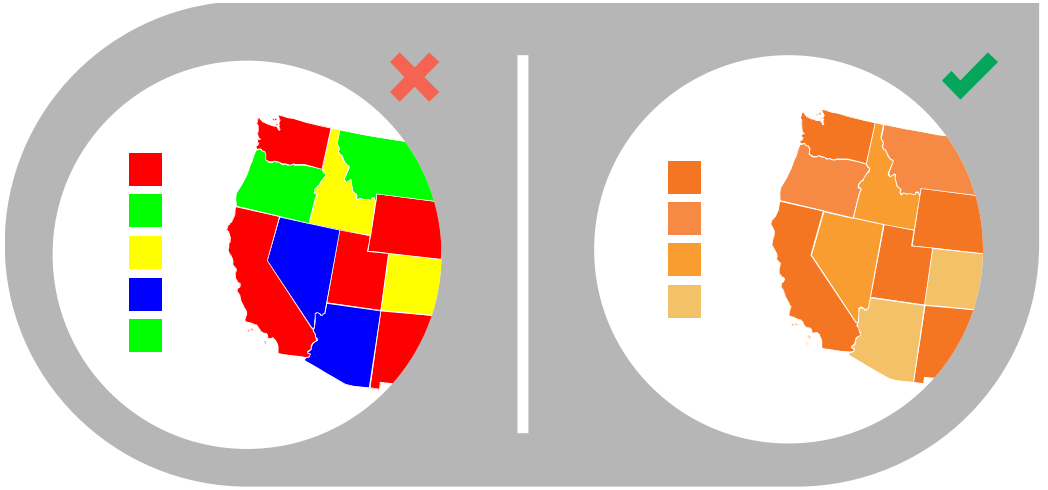
보통 시계열, 상관관계, 분포, 비교에 유용하다.

보통 두 가지 범주와 수치형 교차값을 인풋 값으로 갖는다.



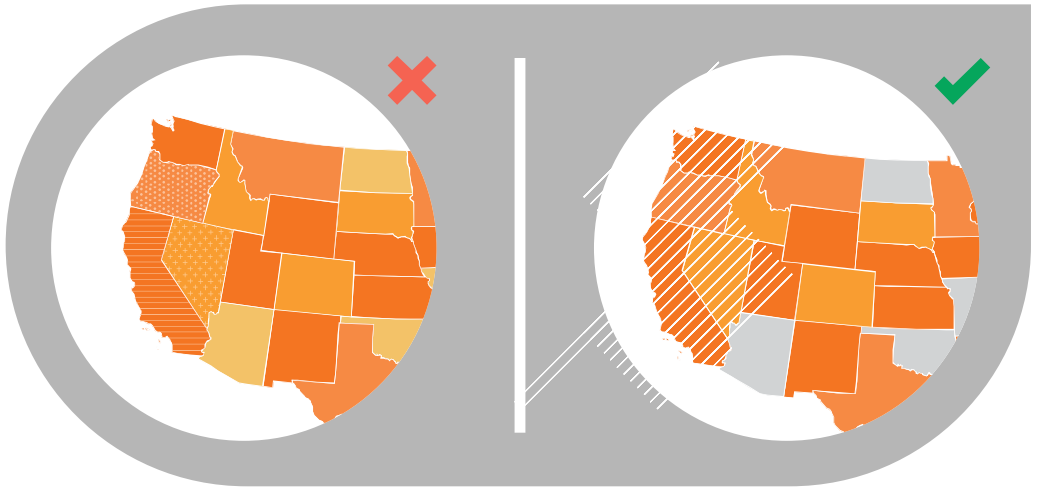
단순한 지도를 사용하자

목적은 구역이 아니라 데이터를 구분하기 위함이다



무지개를 버리자

한 두개의 스펙트럼으로 이루어진 색을 사용하자



패턴은 의미있게 사용하자

너무 많으면 무의미해진다

적절한 범위를 선택하자

극단 값의 범위를 확장하자

그렇다면,
어떤 툴로
시각화를 해야 할까?

솔직히 시각화 부분에선 알이 파이썬보다 우수하다.

파이썬에는 다양한 시각화 라이브러리가 존재하고 입맛에 맞는 걸 사용하면 된다.

대표적인 라이브러리를 알아보자면,

밖에서 많이 쓰는 거 = matplotlib, seaborn / 인터랙티브 = bokeh, pygal, cufflinks