

2일차 과제

[Day2]

Day1 에서 사용한 스타벅스 메뉴 데이터와 매장 정보 데이터를 이용해 좀 더 다양한 시각화와 분석을 해보는 날입니다. 오늘 실습을 통해 행과 열, 마크의 개념을 좀 더 이해해보고 계산식을 맛보세요~

아래 파일을 다운로드 받으세요. (최신 버전 다운로드를 권장합니다)

V2020.1 이상 : @DAY2 워드클라우드&박스플롯_Start.twbx

V2018.1 : @DAY2 □□□□□□□□□□&□□□□□□□□□□_Start_v2018.1.twbx

2일차 과제

🔗 1. 워드클라우드

🔗 2. 박스플롯

🔗 3. 계산된 필드 만들기

🔗 [추가 도전 과제]

1. 워드클라우드

워드클라우드란 많은 키워드 속에서 분석 목적에 따라 핵심 키워드를 표현하는데 유용한 시각화입니다. 워드클라우드도 값의 자세한 비교보다는 데이터의 트렌드를 나타내는데 좀 더 유용하겠죠?

워드클라우드를 이용해 가장 “칼로리”가 높은 “메뉴명”을 한 눈에 살펴봅시다.

다음과 같은 고민을 해 보시고 아래와 같이 시각화를 완성해 보세요.

1. 워드 클라우드를 위해 우리는 어떤 마크의 유형을 사용해야 할까요?
2. 메뉴명을 표현하기 위해서 마크의 속성 중 어디에 메뉴명을 가져다 놓으면 될까요?
3. 마크의 어떤 속성을 사용해 칼로리를 표현하면 좋을까요?

(시각화를 위한 HINT를 참조하세요.)

사용 Data = 스타벅스 메뉴

[illegible]

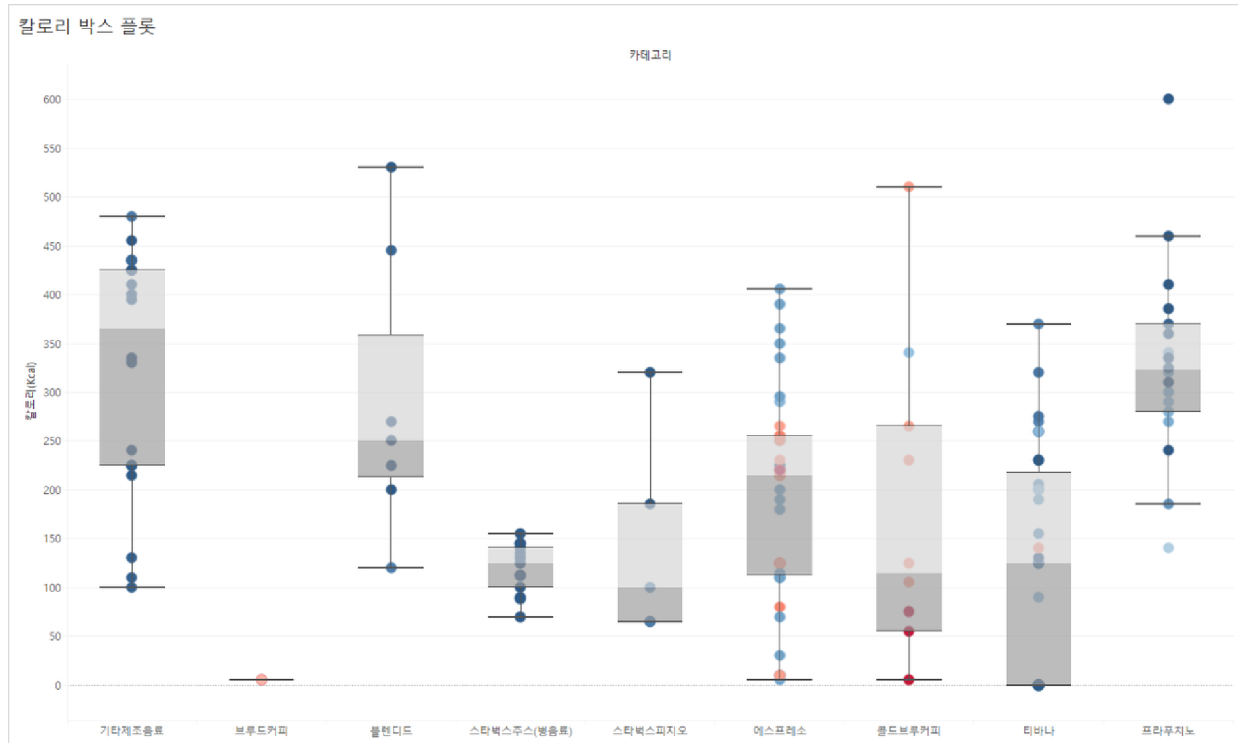
1. 마크의 유형을 “**텍스트**”로 선택해 주세요.
2. **제품명**을 텍스트에 가져다 놓아주세요.
2. 마크의 색상을 **칼로리**로 표현하되, **칼로리가 높을수록 붉은 색, 칼로리가 낮을수록 푸른색**을 나타내도록 해주세요.
3. 마크의 **크기** 또한 **칼로리**로 표현해주세요.

2. 박스플롯

사용 Data = 스타벅스 메뉴

박스플롯은 데이터의 분포 상태와 이상치를 동시에 보여주면서 서로 다른 데이터 군을 쉽게 비교할 수 있는 좋은 시각화입니다. 여러 개의 데이터를 한 눈에 표현할 수 있어 값을 비교하기에 유용하겠죠?

아래와 같은 시각화를 완성해 보세요. (어렵게 느껴지신다면, 시각화를 위한 HINT를 참조하세요.)

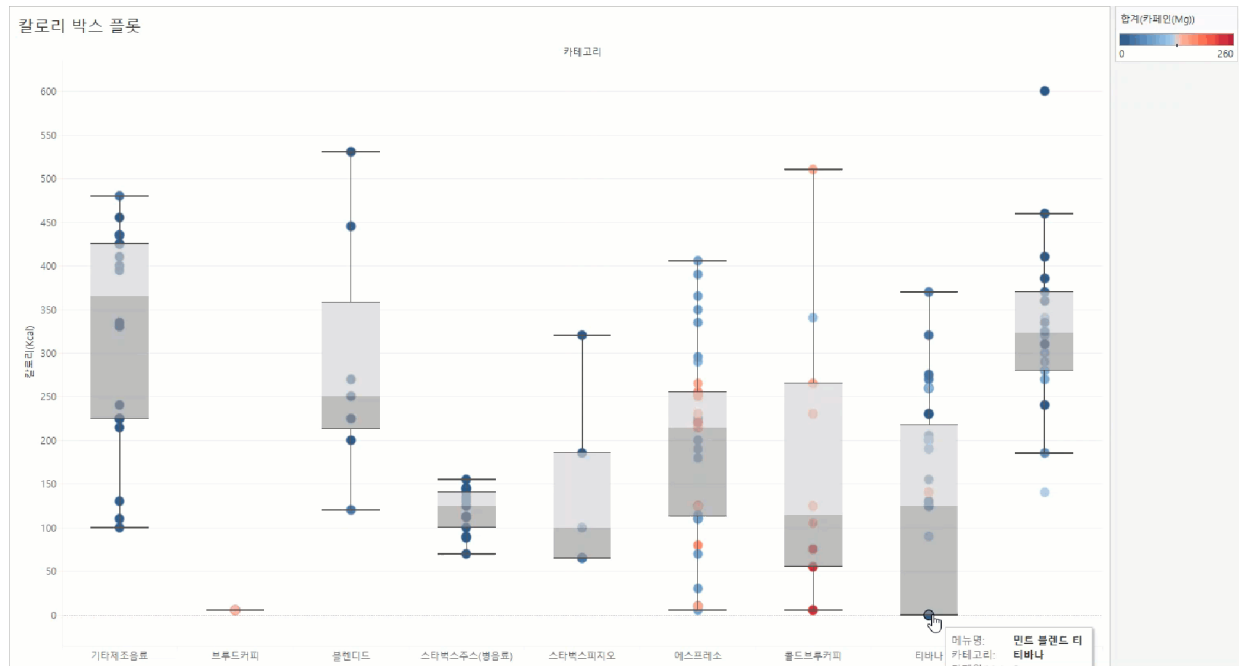


[시각화를 위한 HINT]

1. **카테고리 별로 메뉴명 수준까지의 합계 칼로리**를 나타내세요. (스스로 도전해보시고, 아래 Step을 참조하세요)
 - 1) 카테고리를 열 선반에, 칼로리를 행 선반에 놓아주세요. → 카테고리별 칼로리 합계를 막대차트로 표현
 - 2) 메뉴명 수준에서 살펴봐야 하니 메뉴명을 마크 선반에 있는 세부 정보에 넣어주세요.
 - 각 막대가 메뉴명으로 쪼개짐 (4. **당분 함유량과 칼로리 상관관계: 1일차 과제** 에서 집계 기준을 변경하는 것과 같은 원리)
 - 3) 마크의 형태는 “원”으로 변경해 주세요.
 - 4) 마크의 색상은 “합계 카페인”으로 표현해 주세요. 카페인이 높을수록 붉은색, 낮을 수록 푸른색이면 더 좋을 것 같네요.
2. “분석 탭”에 있는 박스플롯을 드래그해서 시각화에 놓아 보세요.

[Day1] 질문.. 칼로리가 낮고 카페인이 높지 않은 메뉴를 마시고 싶다면? 에 대한 답을 다르게 찾아보실 수 있으셨나요?

티바나 카테고리에 있는 메뉴가 도움이 될 것 같네요~ 아래 그림처럼 티바나 카테고리에서 칼로리가 낮은 마크에서 드래그 한 후에 데이터 보기를 실행해 보세요.



3. 계산된 필드 만들기

분석에 필요한 필드가 데이터 원본에 이미 포함되어 있지 않을 수도 있습니다. 이 때, 계산을 사용하면 데이터 원본에 이미 존재하는 데이터에서 새 데이터를 만들 수 있을 뿐만 아니라 데이터에 대한 계산을 수행할 수 있습니다. 이를 통해 복잡한 분석을 수행하고 데이터 원본에 고유한 필드를 즉석에서 추가할 수 있습니다.

예를 들어 데이터 원본에 “매출” 및 “수익” 값에 대한 필드는 있지만 “수익률”에 대한 필드는 없다고 가정합니다. 이 경우 아래와 같이 “매출”과 “수익” 필드를 사용하여 “수익률”이라는 계산된 필드를 만들 수 있습니다.

수익률

주문+반종 추출

$$\text{SUM}([\text{수익}]) / \text{SUM}([\text{매출}])$$

계산이 유효합니다.

적용

확인

측정값

- # 매출
- # 수량
- # 수익
- # 수익률**
- # 할인율
- 🌐 경로(생성됨)

계산된 필드의 결과가 숫자일 경우 아이콘이 **=#** 라고 나타납니다.

또 다른 예를 들어볼까요?

예를 들어 색상을 사용하여 데이터를 수익성이 있는 것과 수익성이 없는 것으로 빠르게 분류하고 싶습니다. 그럼 아래와 같은 계산을 사용해서 계산된 필드를 만들고 마크 선반의 색상에 추가하면 되겠죠?

수익성 여부

주문+반종 추출

✕

```
IF SUM([수익]) > 0
THEN "Profitable"
ELSE "Nonprofitable"
END
```

계산이 유효합니다.

적용

확인

측정값

- # 매출
- # 수량
- # 수익
- # 수익률
- =Abc 수익성 여부**
- # 할인율

계산된 필드의 결과가 문자일 경우 아이콘이 **=Abc** 라고 나타납니다.

그럼 우리도 계산식을 만들어서 평균카페인 함유량이 80mg 보다 높은 카테고리와 아닌 카테고리를 분류해서 볼 수 있도록 새로운 필드를 만들어 봅시다!

아래와 같은 시각화를 완성해 보세요. (어렵게 느껴지신다면, 시각화를 위한 HINT를 참조하세요.)

사용 Data = 스타벅스 메뉴

계산된 필드 만들기

카테고리 ☐

카테고리	평균 칼로리(Kcal)	카페인 <= 80mg	카페인 > 80mg
프라푸지노	325	Yes	No
기타제조음료	320	Yes	No
블렌디드	290	Yes	No
에스프레소	190	No	Yes
클드브루커피	170	No	Yes
스타벅스피지오	150	Yes	No
스타벅스주스(병음료)	120	Yes	No
티바나	120	Yes	No
브루드커피	10	No	Yes

평균 칼로리(Kcal) ☐

집계(지정 카페인 용량)

- 카페인 <= 80mg
- 카페인 > 80mg

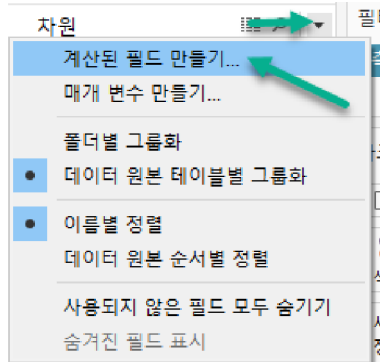
[시각화를 위한 HINT]

1. “카테고리” 별 “평균 칼로리”를 먼저 시각화로 나타내 주세요.

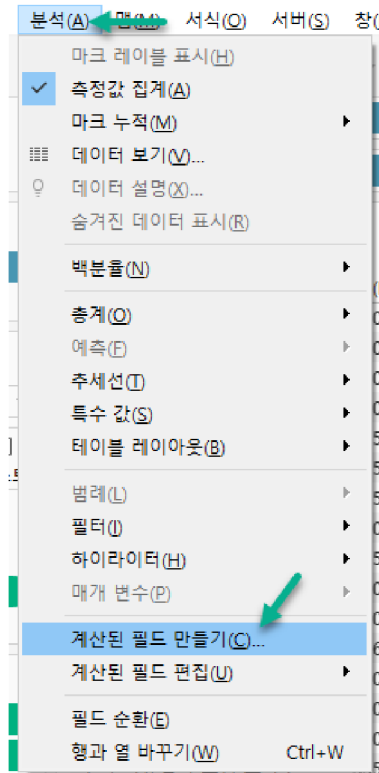
카테고리는 행 선반에, 평균 칼로리는 열 선반에 놓아주세요.

2. 계산된 필드 만들기를 실행해주세요. 계산된 필드를 만드는 방법은 크게 4가지가 있습니다. 어느 방법을 사용해도 무방합니다.

1) 차원 옆 > 아래 화살표 > 계산된 필드 만들기



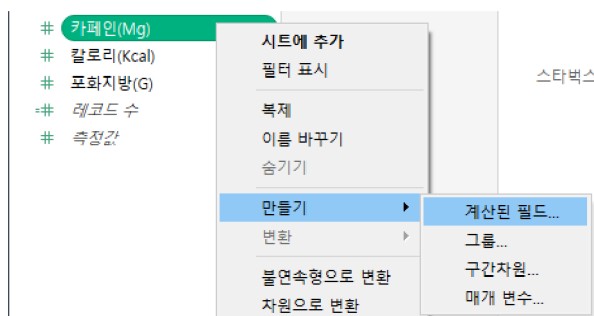
2) 메뉴 > 분석 > 계산된 필드 만들기



3) 데이터 창 > 마우스 오른쪽 버튼 > 계산된 필드 만들기



4) 사용할 필드 > 마우스 오른쪽 버튼 > 계산된 필드 만들기

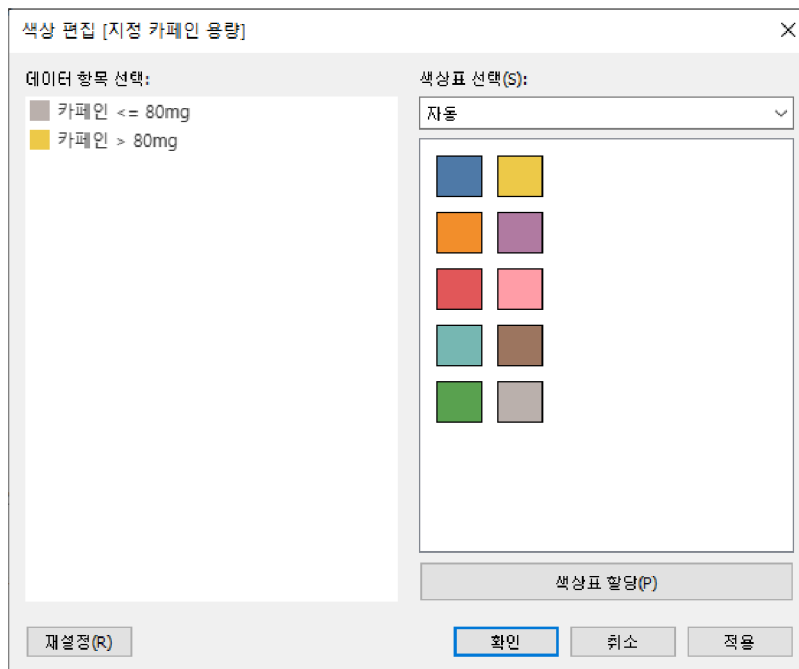


2. 아래와 같이 계산식 이름과 계산식을 입력해 주세요.

아래 계산식은 **평균 카페인**이 80보다 크면 “카페인 > 80mg” 라는 값을, 80보다 작거나 같을 경우 “카페인 <= 80mg”을 부여하는 식입니다.



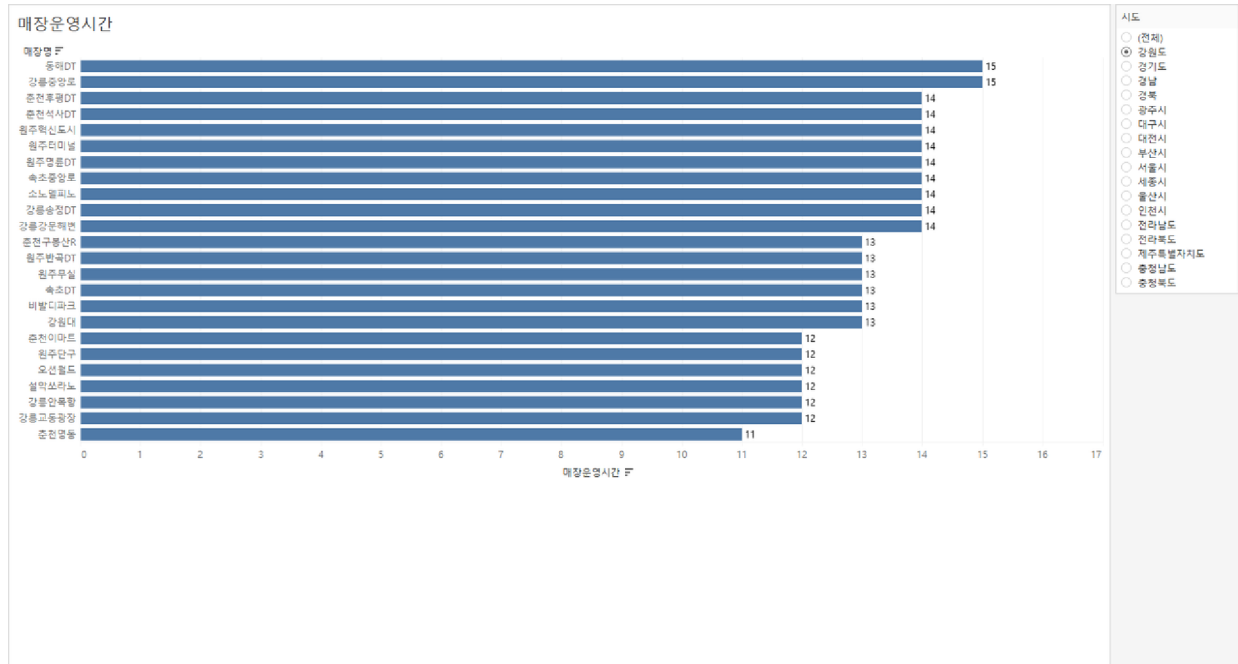
3. 만들어진 “지정 카페인 용량” 필드를 색상에 가져다 놓아주세요.
4. 카페인인 높은 카테고리를 강조하기 위해서 색상을 변경해 봅시다.
 - 1) 마크 선반의 색상을 클릭해서 색상 편집을 실행해 주세요.
 - 2) 아래와 같이 색상을 부여해주세요. 각각 데이터 항목을 선택한 후 해당 색상을 클릭하시면 됩니다.



[추가 도전 과제]

“매장명” 별로 “매장운영시간”을 표현하고, 시도를 필터로 걸어 각 시도의 매장 별 운영 시간을 확인해 보세요~

사용 Data = 스타벅스 매장 정보



[HINT]

“매장운영시간”은 “영업시작시간”과 “영업종료시간”의 차이를 계산해서 표현하면 되겠죠?

날짜 함수 중 **DATEDIFF** 함수를 사용해 두 날짜 간의 “시간(hour)” 차이를 계산해 보세요.

https://help.tableau.com/current/pro/desktop/ko-kr/functions_functions_date.htm

아래 공백에 어떤 내용들이 들어가면 될까요?

매장운영시간

스타벅스 매장 정보

DATEDIFF (' , [, [])

↑ ↑ ↑

계산이 유효합니다.

적용

확인

날짜

검색 텍스트 입력

DATEADD
DATEDIFF
DATENAME
DATEPARSE
DATEPART
DATETRUNC
DAY
ISDATE
ISOQUARTER
ISOWEEK
ISOWEEKDAY
ISOYEAR

DATEDIFF(날짜 부분, 시작_날짜, 끝_날짜, [주_시작])

두 날짜 사이의 차이를 반환합니다. 여기서는 끝_날짜에서 시작_날짜를 뺍니다. 차이는 날짜_부분의 단위로 표현됩니다. 주_시작을 생략하면 주 시작일이 데이터 원본에 대해 구성된 시작일에 의해 결정됩니다.

예: DATEDIFF('month', #2004-07-15#, #2004-04-03#, 'sunday') = -3

함수 도움말에 표현되어 있는 **날짜_부분**(영어로 **date_part**)은 아래 URL 또는 표를 통해서 확인하세요.

https://help.tableau.com/current/pro/desktop/ko-kr/functions_functions_date.htm#datepart-%EA%B0%92

DATE_PART	값
'year'	4자리 연도
'quarter'	1-4
'month'	1-12 또는 "1월", "2월" 등
'dayofyear'	일년 중 몇째 날. 1월 1일은 1, 2월 1일은 32 등으로 계산됩니다.
'day'	1-31
'weekday'	1-7 또는 "일요일", "월요일" 등
'week'	1-52
'hour'	0-23
'minute'	0-59
'second'	0-60
'iso-year'	4자리 ISO 8601 연도
'iso-quarter'	1-4
'iso-week'	1-52, 주의 시작은 항상 월요일
'iso-weekday'	1-7, 주의 시작은 항상 월요일