

삼성전자 AI전문가 교육과정 사전교육 - 데이터 분석 및 시각화

Altair 실습 2 - Mark

Marks

- 채널을 다루는 방법은 이미 배웠고, 이번에는 Altair의 마크를 다루어 봅시다.

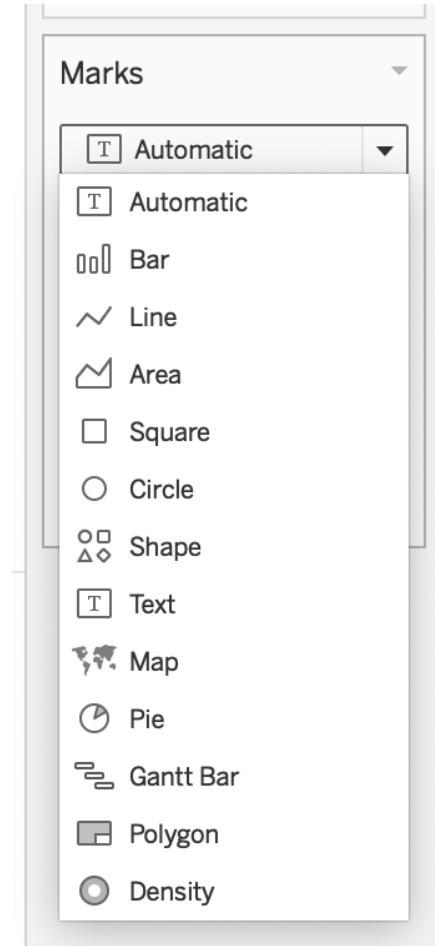
```
import altair as alt
from vega_datasets import data

url = data.cars.url

alt.Chart(url).mark_circle
    color='red',
    opacity=0.3
).encode(
    x='Horsepower:Q',
    y='Miles_per_Gallon:Q'
)
```

Marks

- area
- bar
- circle
- geoshape
- line
- point
- rect
- rule
- square
- text
- tick



Review : 데이터 불러오기

```
import altair as alt
import pandas as pd

from vega_datasets import data
gapminder = data.gapminder()

print(gapminder.shape)
print(gapminder.head())

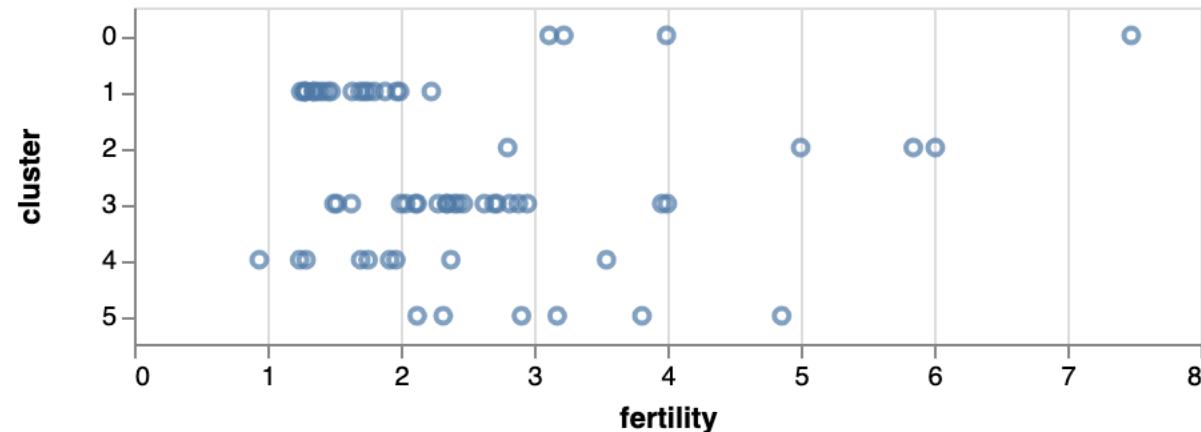
data2000 = gapminder.loc[gapminder["year"] == 2000]
print(data2000.head())
```

(693, 6)						
	year	country	cluster	pop	life_expect	fertility
0	1955	Afghanistan	0	8891209	30.332	7.7
1	1960	Afghanistan	0	9829450	31.997	7.7
2	1965	Afghanistan	0	10997885	34.020	7.7
3	1970	Afghanistan	0	12150000	35.088	7.7
4	1975	Afghanistan	0	14132019	38.438	7.7
코드 셀 추가(⌘Enter)						
	year	country	cluster	pop	life_expect	fertility
9	2000	Afghanistan	0	23898198	42.129	7.4792
20	2000	Argentina	3	37497728	74.340	2.3500
31	2000	Aruba	3	69539	73.451	2.1240
42	2000	Australia	4	19164620	80.370	1.7560
53	2000	Austria	1	8113413	78.980	1.3820

Review : 인코딩을 명시하는 두 가지 방법

- 아래 두 코드는 같은 역할을 하는 코드입니다.

```
alt.Chart(data2000).mark_point().encode(  
    alt.X('fertility:Q'),  
    alt.Y('cluster:N')  
)  
  
alt.Chart(data2000).mark_point().encode(  
    x='fertility:Q',  
    y='cluster:N'  
)
```



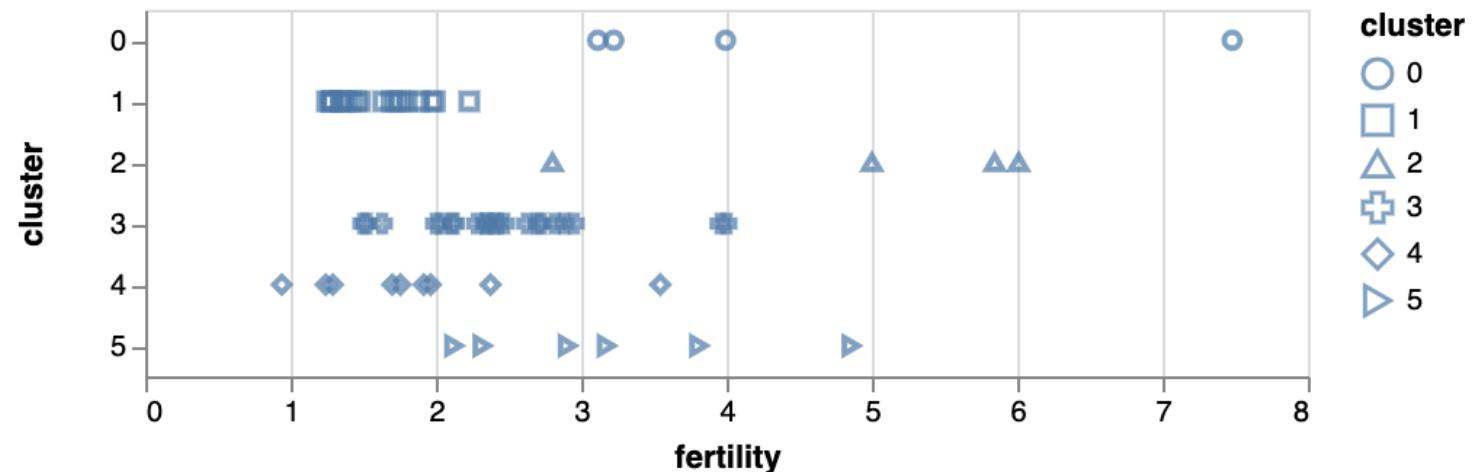
Point Mark

- 포인트 마크는 점 하나로 데이터를 나타냅니다.
- 사용 가능한 채널
 - x
 - y
 - color
 - size
 - shape

Point Mark

- 아래 차트를 점 도표(Dot plot)라고 부릅니다.

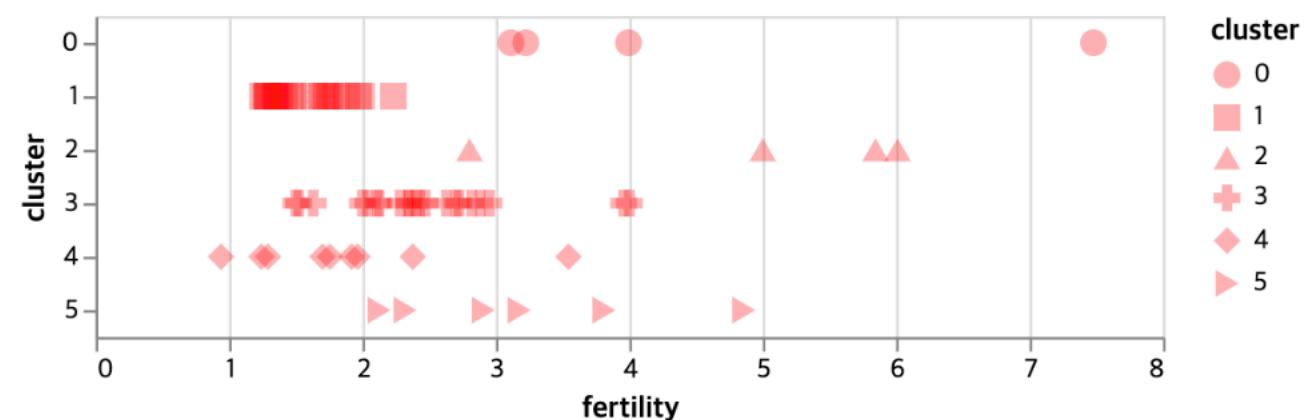
```
alt.Chart(data2000).mark_point().encode(  
    alt.X('fertility:Q'),  
    alt.Y('cluster:N'),  
    alt.Shape('cluster:N')  
)
```



Point Mark

- 포인트 마크에 스타일 적용하기
 - 이 때 size는 포인트의 마크의 넓이를 의미합니다

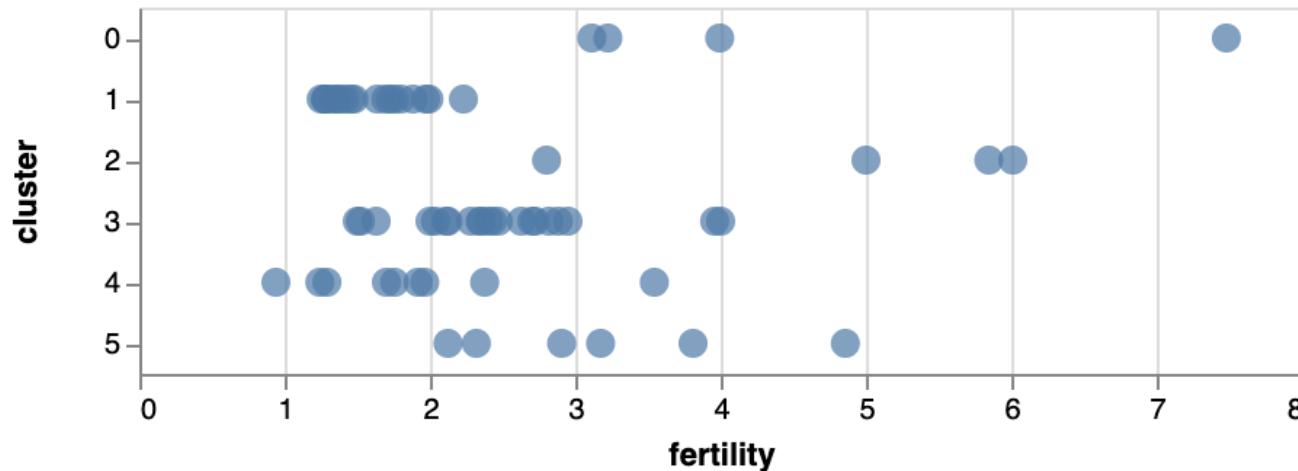
```
alt.Chart(data2000).mark_point(  
    filled=True,  
    color="red",  
    opacity= 0.3,  
    size=100  
).encode(  
    alt.X('fertility:Q'),  
    alt.Y('cluster:N'),  
    alt.Shape('cluster:N')  
)
```



Circle Mark

- 원 마크는 포인트 마크와 같으나, shape 옵션이 'circle'로, filled 옵션이 True로 설정되었다고 보면 됩니다.

```
alt.Chart(data2000).mark_circle(size=100).encode(  
    alt.X('fertility:Q'),  
    alt.Y('cluster:N')  
)
```

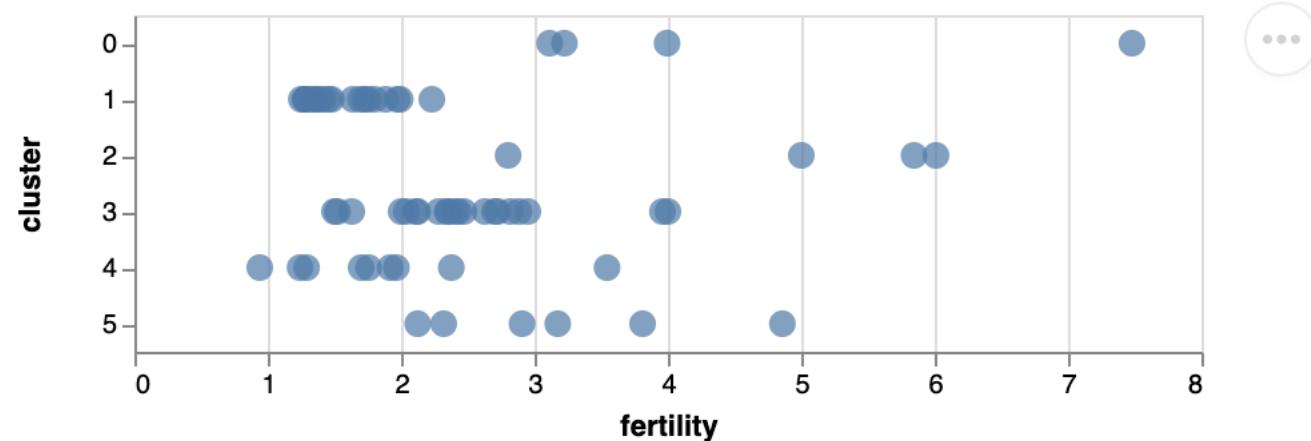


Circle Mark

- 아래 두 코드는 같습니다.

```
alt.Chart(data2000).mark_circle(size=100).encode(  
    alt.X('fertility:Q'),  
    alt.Y('cluster:N')  
)
```

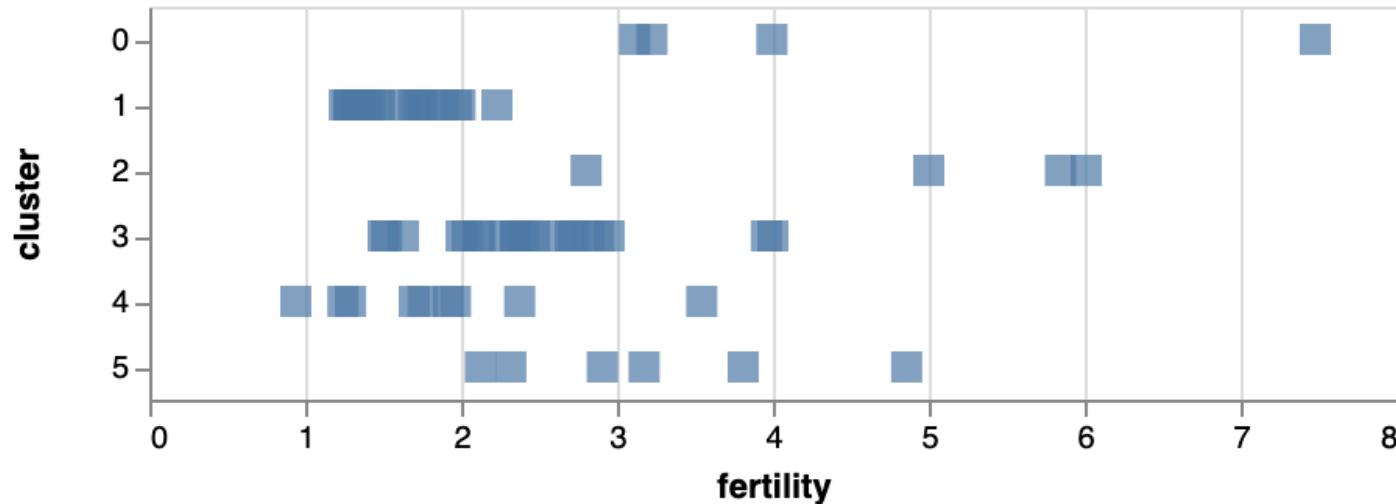
```
alt.Chart(data2000).mark_point(size=100, shape='circle', filled=True).encode(  
    alt.X('fertility:Q'),  
    alt.Y('cluster:N'),  
)
```



Square Mark

- 사각형 마크의 경우에도 포인트 마크와 같으나, 역시 shape옵션과 filled 옵션이 “square”와 True로 설정되었다고 보시면 됩니다.

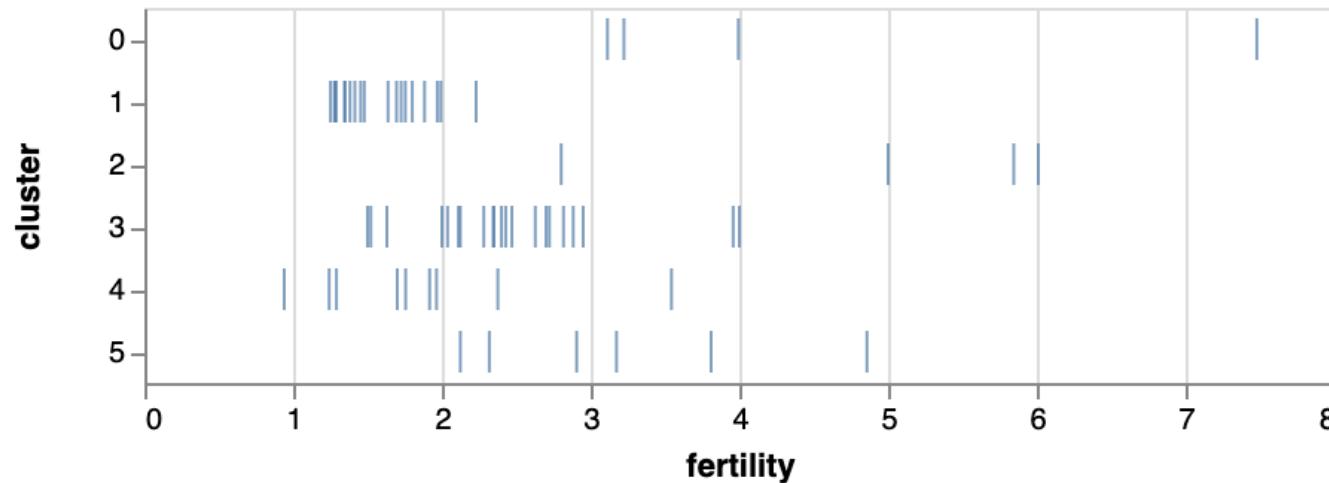
```
alt.Chart(data2000).mark_square(size=100).encode(  
    alt.X('fertility:Q'),  
    alt.Y('cluster:N'),  
)
```



Tick Mark

- 틱 마크의 경우 점 대신 얇은 선을 통해 데이터의 위치를 표시합니다.
- 이를 바코드 차트 혹은 strip plot이라고 부르기도 합니다.

```
alt.Chart(data2000).mark_tick().encode(  
    alt.X('fertility:Q'),  
    alt.Y('cluster:N')  
)
```

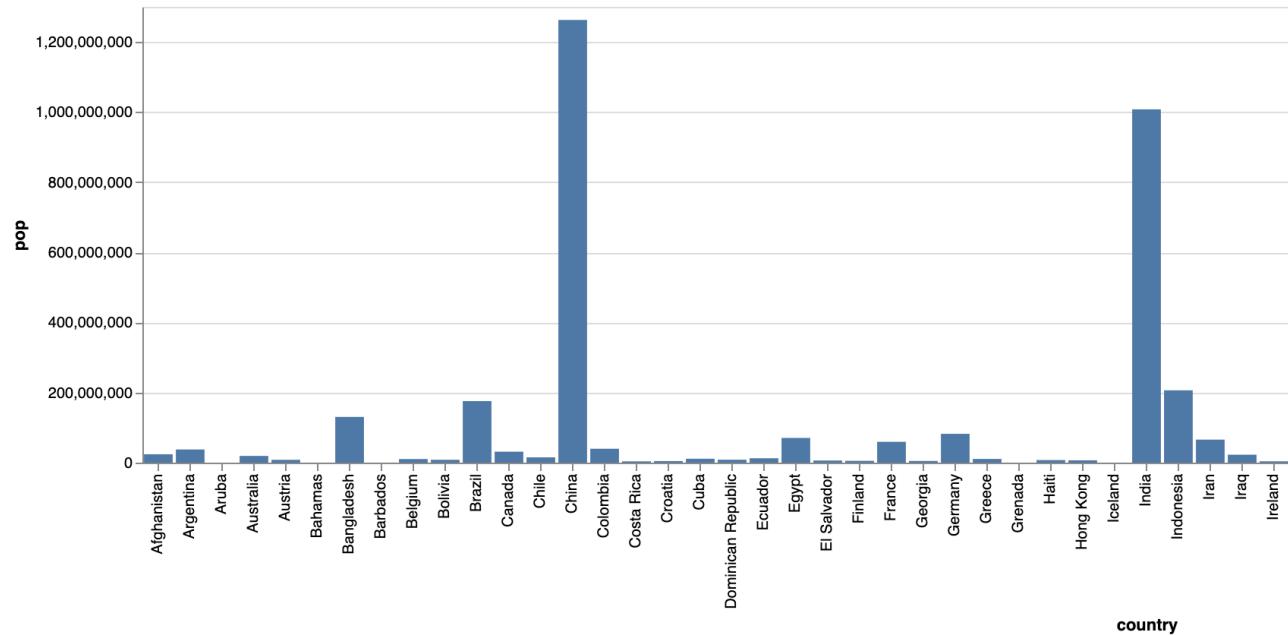


Bar Mark

- 막대 마크의 경우 데이터를 막대로 표현합니다.
- 사용 가능한 채널
 - x
 - y
 - x2
 - y2
 - color

Bar Mark

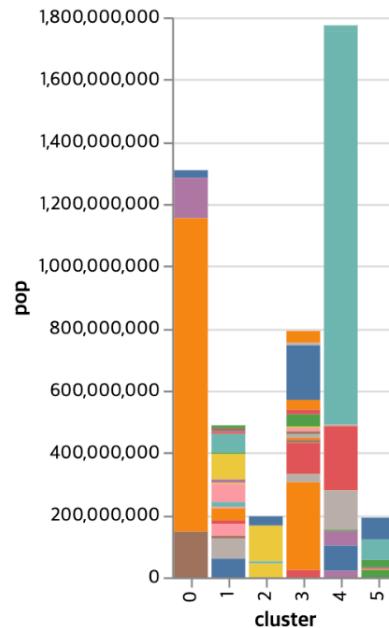
```
alt.Chart(data2000).mark_bar().encode(  
    alt.X('country:N'),  
    alt.Y('pop:Q')  
)
```



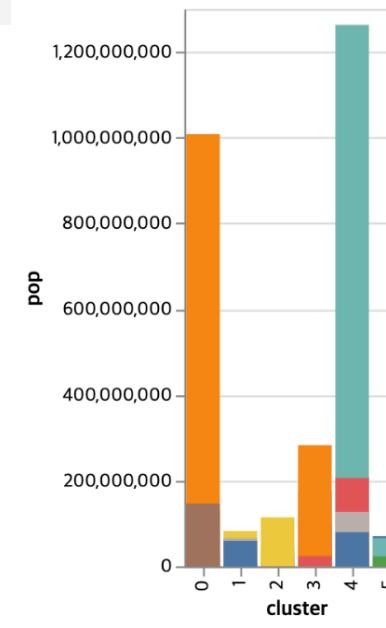
Bar Mark

- Color로 누적 막대 그래프를 그릴 수 있습니다.
- 이 경우, 자동으로 y 인코딩에 stack="zero"가 적용됩니다.

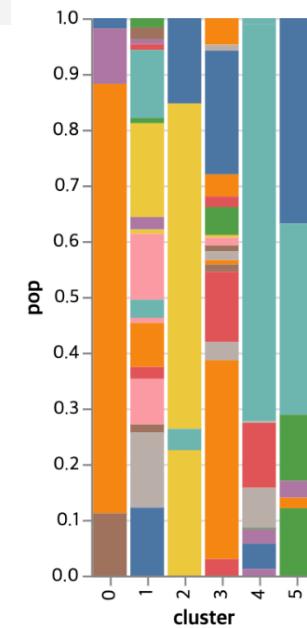
```
alt.Chart(data2000).mark_bar().encode(
    alt.X('cluster:N'),
    alt.Y('pop:Q'),
    alt.Color('country:N', legend=None),
    alt.Tooltip('country:N')
)
```



```
alt.Chart(data2000).mark_bar().encode(
    alt.X('cluster:N'),
    alt.Y('pop:Q', stack=None),
    alt.Color('country:N', legend=None),
    alt.Tooltip('country:N')
)
```



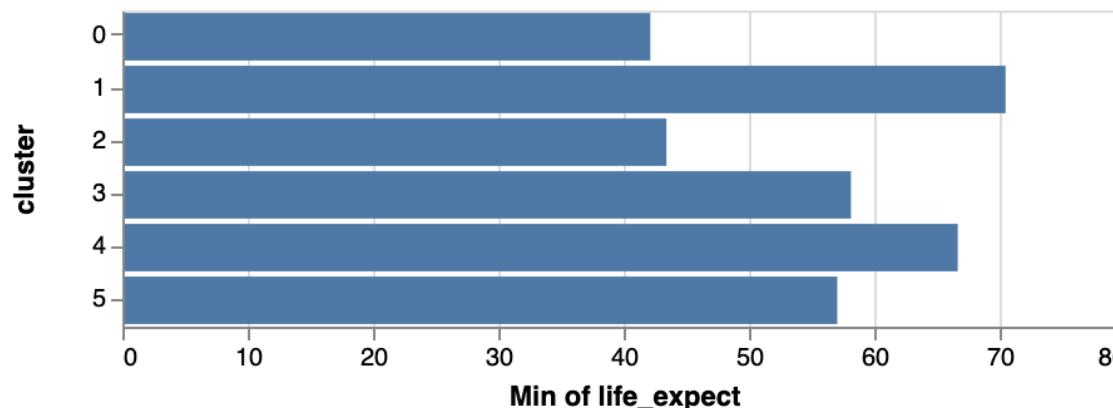
```
alt.Chart(data2000).mark_bar().encode(
    alt.X('cluster:N'),
    alt.Y('pop:Q', stack="normalize"),
    alt.Color('country:N', legend=None),
    alt.Tooltip('country:N')
)
```



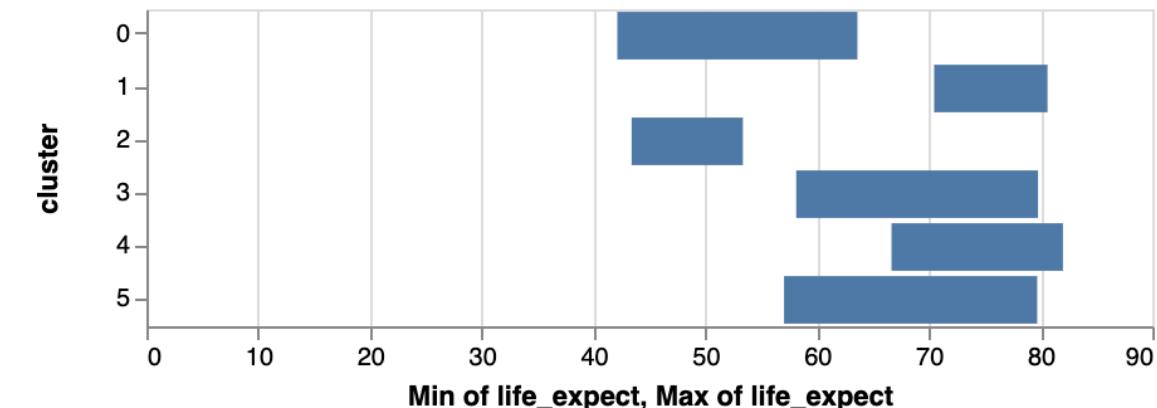
Bar Mark

- X2의 경우 막대의 오른쪽 끝점을 지정할 때 사용합니다.
- 지정할 경우 x가 막대의 왼쪽 끝에, x2가 막대의 오른쪽 끝에 대응됩니다.
- 지정하지 않을 경우, 0이 막대의 왼쪽 끝에, x가 막대의 오른쪽 끝에 대응됩니다.

```
alt.Chart(data2000).mark_bar().encode(  
    alt.X('min(life_expect):Q'),  
    alt.Y('cluster:N')  
)
```



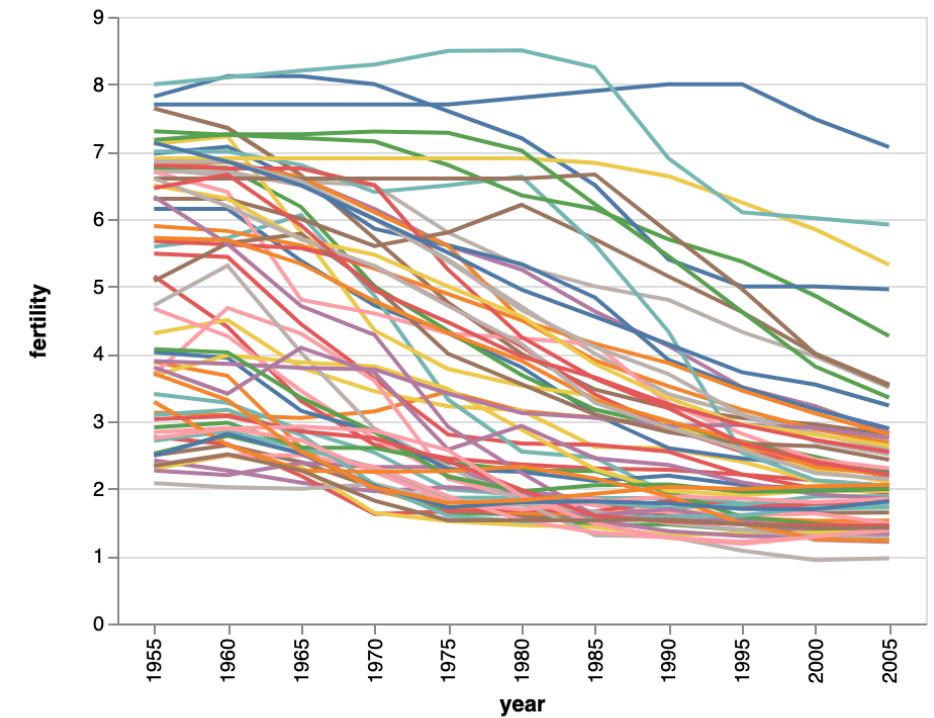
```
alt.Chart(data2000).mark_bar().encode(  
    alt.X('min(life_expect):Q'),  
    alt.X2('max(life_expect):Q'),  
    alt.Y('cluster:N')  
)
```



Line Mark

- 선 마크는 포인트 마크를 사이를 선으로 잇습니다.

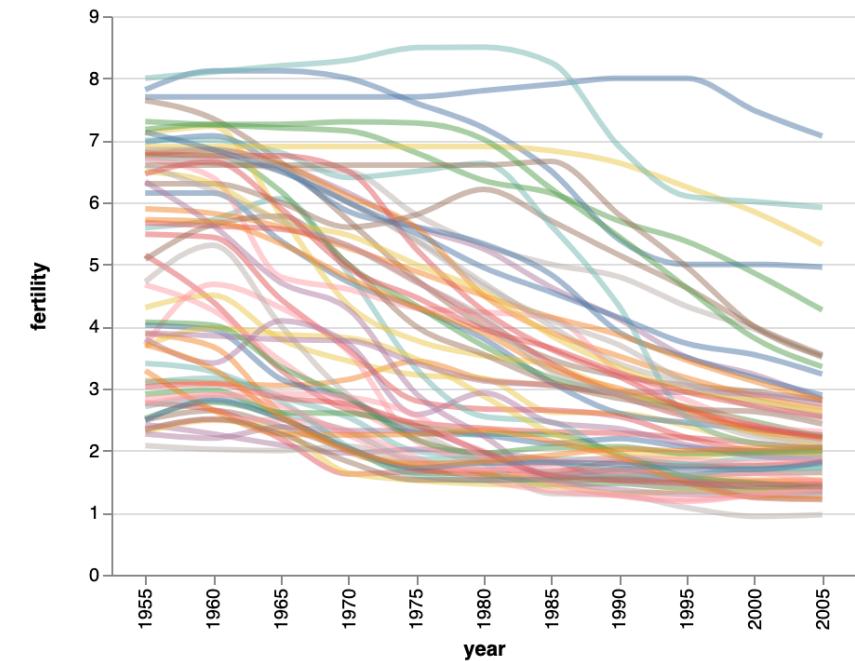
```
alt.Chart(gapminder).mark_line().encode(  
    alt.X('year:Q'),  
    alt.Y('fertility:Q'),  
    alt.Color('country:N', legend=None),  
    alt.Tooltip('country:N')  
).properties(  
    width=400  
)
```



Line Mark

- interpolate 옵션을 지정하여 점 사이를 부드럽게 보간할 수 있습니다.

```
alt.Chart(gapminder).mark_line(interpolate='monotone')
    .encode(
        alt.X('year:Q'),
        alt.Y('fertility:Q'),
        alt.Color('country:N', legend=None),
        alt.Tooltip('country:N')
    ).properties(
        width=400
    )
```



Line Mark

- interpolate에 사용할 수 있는 옵션은 아래와 같습니다.

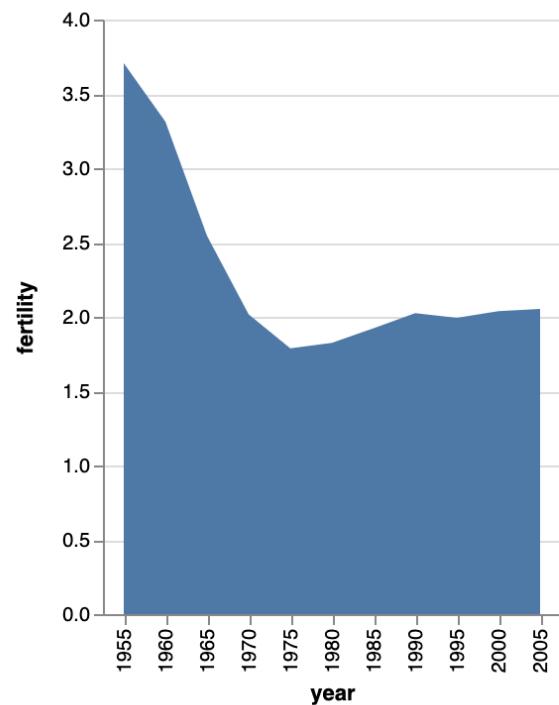
- `"linear"` : piecewise linear segments, as in a polyline.
- `"linear-closed"` : close the linear segments to form a polygon.
- `"step"` : alternate between horizontal and vertical segments, as in a step function.
- `"step-before"` : alternate between vertical and horizontal segments, as in a step function.
- `"step-after"` : alternate between horizontal and vertical segments, as in a step function.
- `"basis"` : a B-spline, with control point duplication on the ends.
- `"basis-open"` : an open B-spline; may not intersect the start or end.
- `"basis-closed"` : a closed B-spline, as in a loop.
- `"cardinal"` : a Cardinal spline, with control point duplication on the ends.
- `"cardinal-open"` : an open Cardinal spline; may not intersect the start or end, but will intersect other control points.
- `"cardinal-closed"` : a closed Cardinal spline, as in a loop.
- `"bundle"` : equivalent to basis, except the tension parameter is used to straighten the spline.
- `"monotone"` : cubic interpolation that preserves monotonicity in y.

Area Mark

- 영역 마크는 막대 마크와 선 마크를 합친 것입니다.
 - 막대 마크처럼 아래 영역에 색을 칠하고, 쌓을 수 있습니다.
 - 선 마크처럼 기울기를 이용하여 데이터의 변화를 보여줄 수 있습니다.

```
dataUS = gapminder.loc[gapminder["country"] == "United States"]

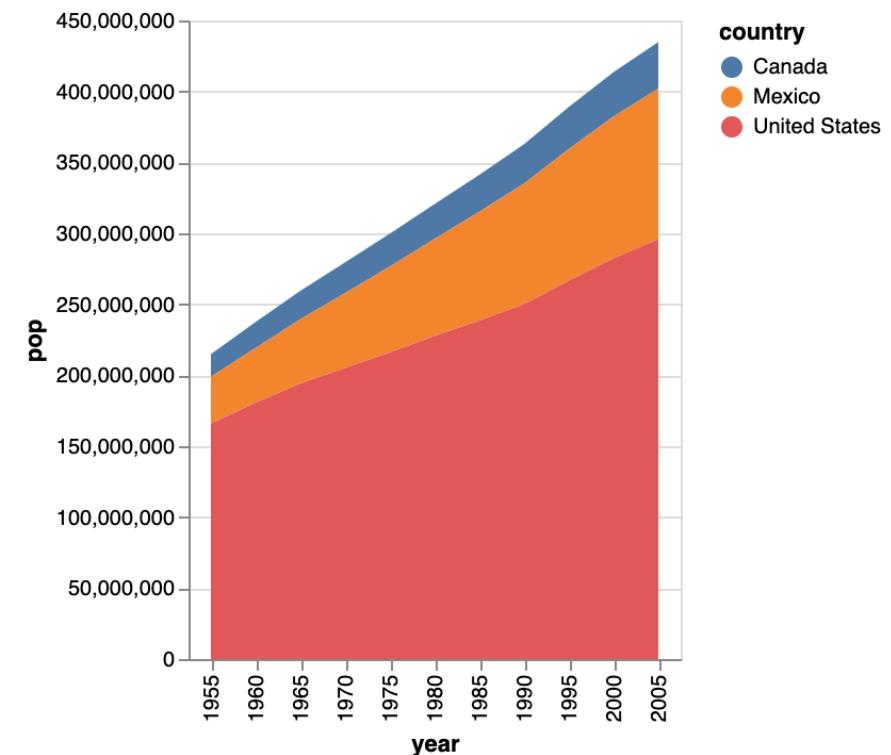
alt.Chart(dataUS).mark_area().encode(
    alt.X('year:0'),
    alt.Y('fertility:Q')
)
```



Area Mark

- 막대 마크 처럼 쌓거나

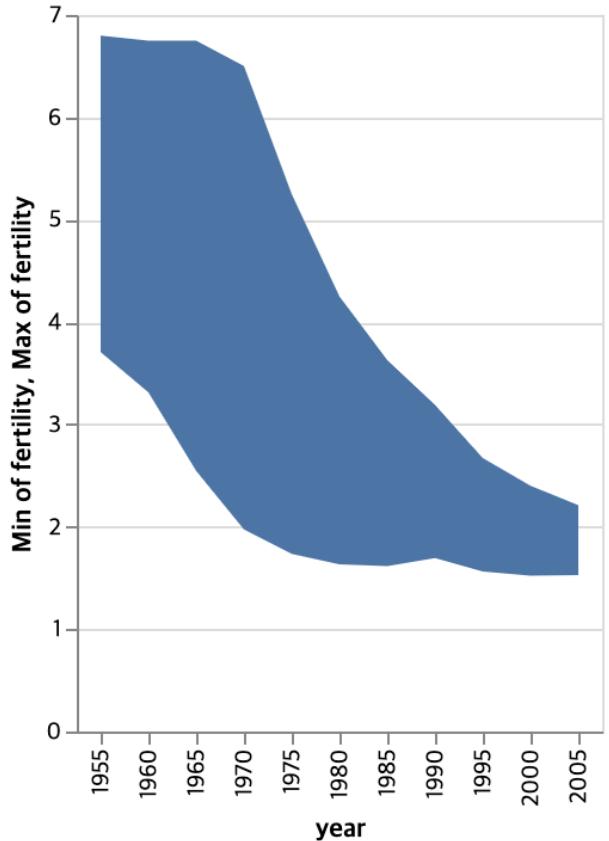
```
dataNA = gapminder.loc[  
    (gapminder['country'] == 'United States') |  
    (gapminder['country'] == 'Canada') |  
    (gapminder['country'] == 'Mexico')  
]  
  
alt.Chart(dataNA).mark_area().encode(  
    alt.X('year:0'),  
    alt.Y('pop:Q'),  
    alt.Color('country:N')  
)
```



Area Mark

- Y2를 사용할 수 있습니다.

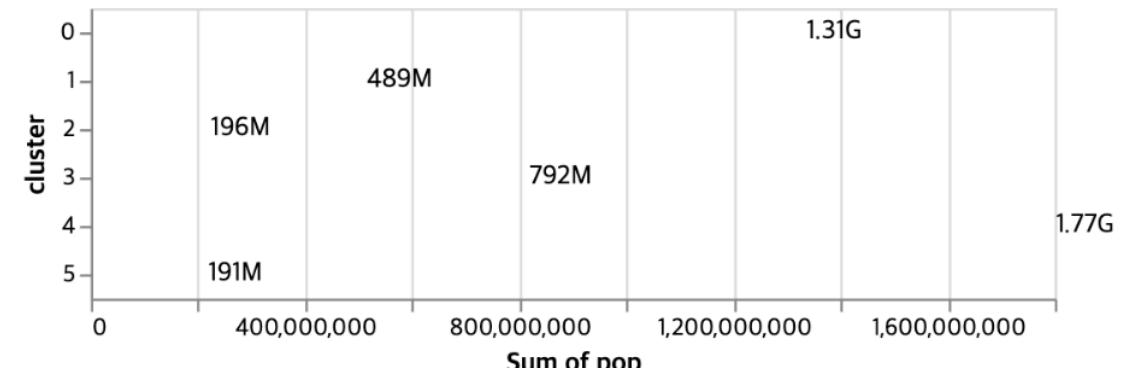
```
alt.Chart(dataNA).mark_area().encode(  
    alt.X('year:0'),  
    alt.Y('min(fertility):Q'),  
    alt.Y2('max(fertility):Q')  
)
```



Text Mark

- 차트 위에 텍스트를 표시하기 위해 사용합니다.
- 별로 쓸모가 없어 보이시나요?

```
texts = alt.Chart(data2000).mark_text(  
    align='left',  
    baseline='middle',  
    dx=6  
) .encode(  
    alt.X('sum(pop):Q'),  
    alt.Y('cluster:N'),  
    alt.Text('sum(pop):Q', format='.3s')  
)  
  
texts
```

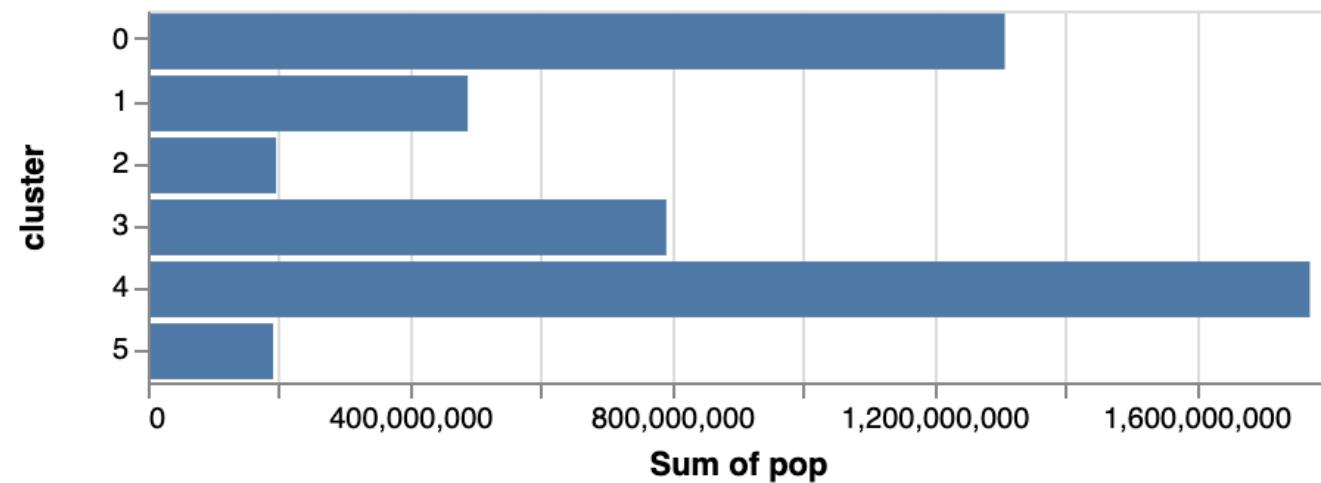


Text Mark

- 아래처럼 막대 그래프를 하나 만들고

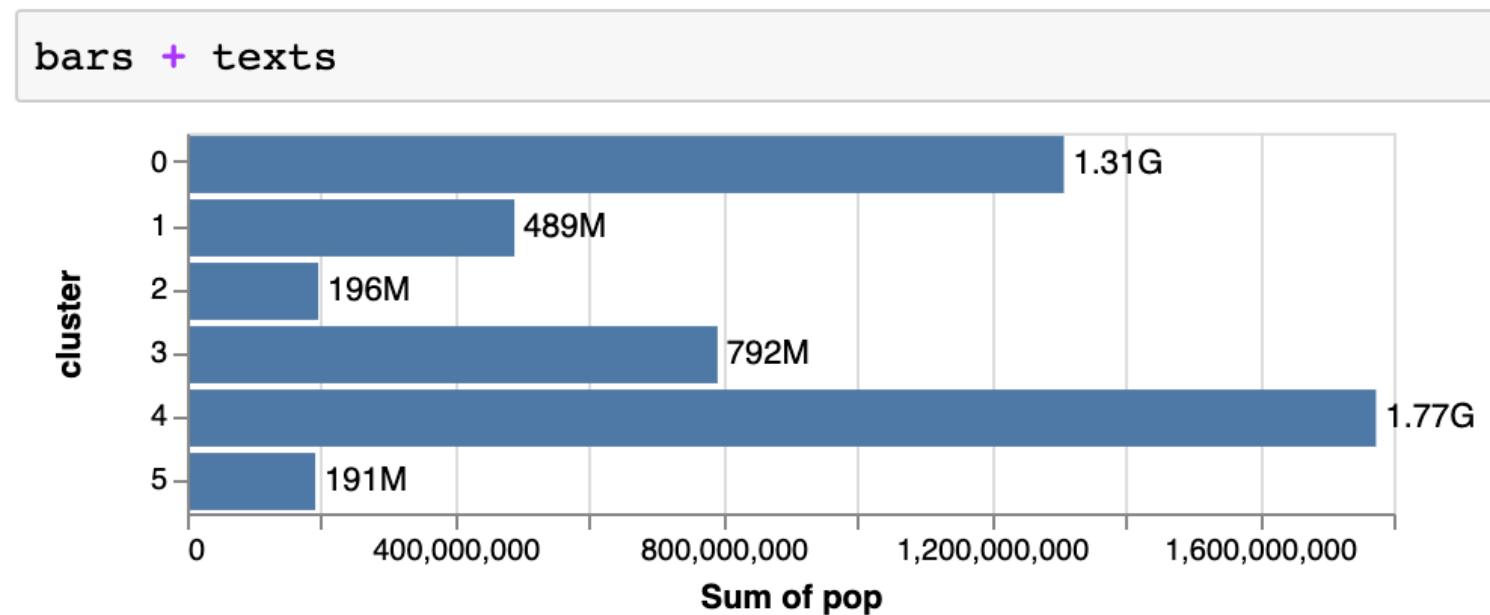
```
bars = alt.Chart(data2000).mark_bar().encode(  
    alt.X('sum(pop):Q'),  
    alt.Y('cluster:N')  
)
```

```
bars
```



Text Mark

- 합쳐서 보여줄 수 있습니다!



Data Transformation

- Altair의 data transformation을 이용하여 원본 데이터를 변환할 수 있습니다.
- 물론 이는 Python 상에서 변환한 데이터프레임을 인자로 넘겨도 됩니다.
- 가장 유용한 4개의 변환을 다룹니다
 - 데이터 Binning
 - 데이터 Aggregation
 - transform_calculate()
 - transform_filter()

데이터 불러오기

- 이번에는 영화 데이터를 사용해봅시다.

(3201, 16)

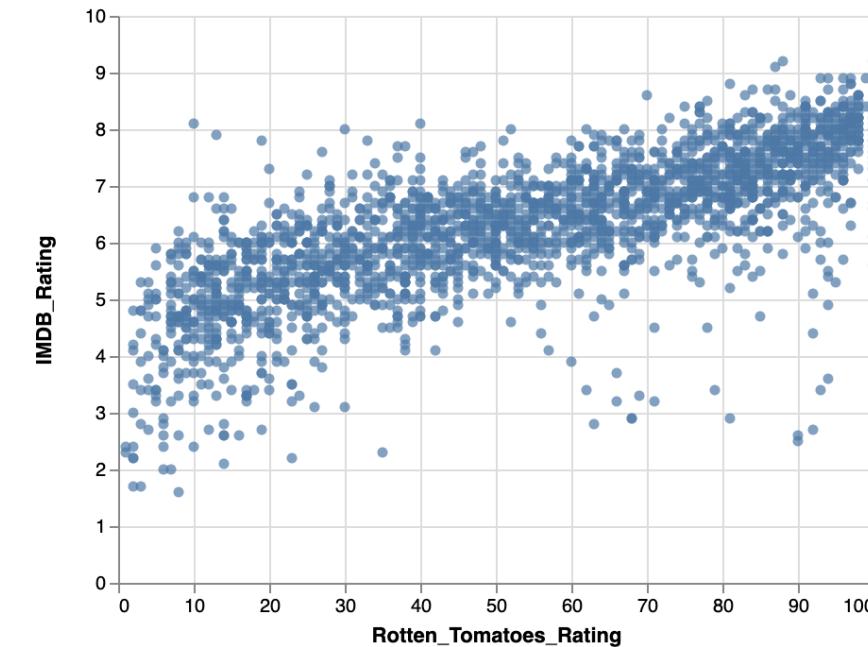
	Title	US_Gross	Worldwide_Gross	US_DVD_Sales	Production_Budget	Release_Date	MPAA_Rating	Runni
0	The Land Girls	146083.0	146083.0	NaN	8000000.0	Jun 12 1998		R
1	First Love, Last Rites	10876.0	10876.0	NaN	300000.0	Aug 07 1998		R
2	I Married a Strange Person	203134.0	203134.0	NaN	250000.0	Aug 28 1998		None
3	Let's Talk About Sex	373615.0	373615.0	NaN	300000.0	Sep 11 1998		None
4	Slam	1009819.0	1087521.0	NaN	1000000.0	Oct 09 1998		R

```
movies = data.movies()  
  
print(movies.shape)  
movies.head()
```

데이터 Binning

- 데이터 빙팅은 연속된 값을 가지는 필드를 구간으로 나누는 변환입니다.
- 데이터에는 Rotten Tomatoes와 IMDB라는 두 개의 영화 리뷰 사이트의 점수가 있습니다.
이 둘은 비례할까요?

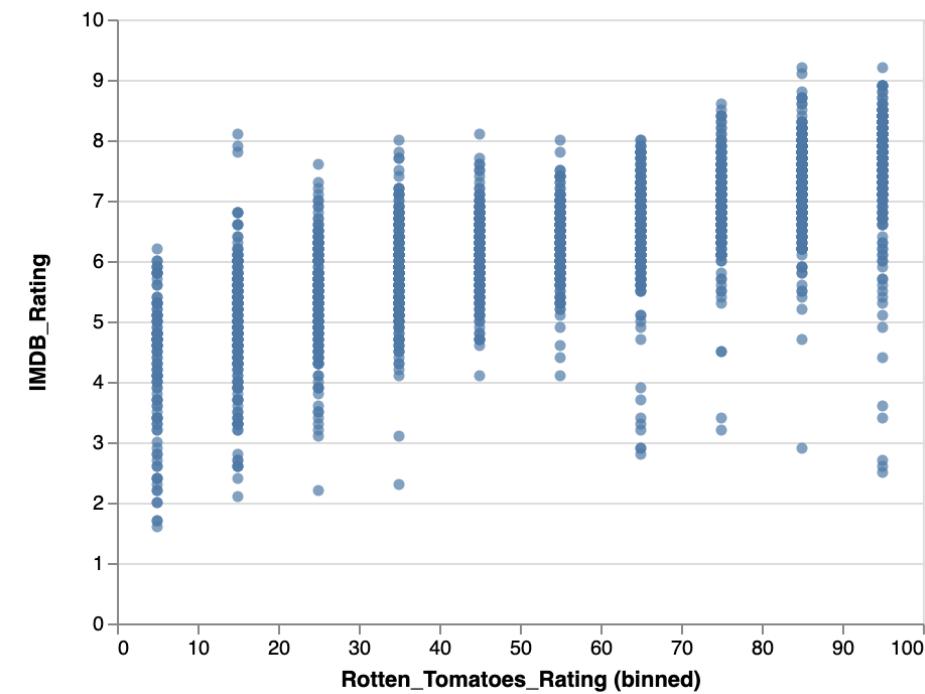
```
alt.Chart(movies).mark_circle().encode(  
    alt.X('Rotten_Tomatoes_Rating:Q'),  
    alt.Y('IMDB_Rating:Q')  
)
```



데이터 Binning

- 가로 축 (Rotten Tomatoes 점수)를 binning해 봅시다.

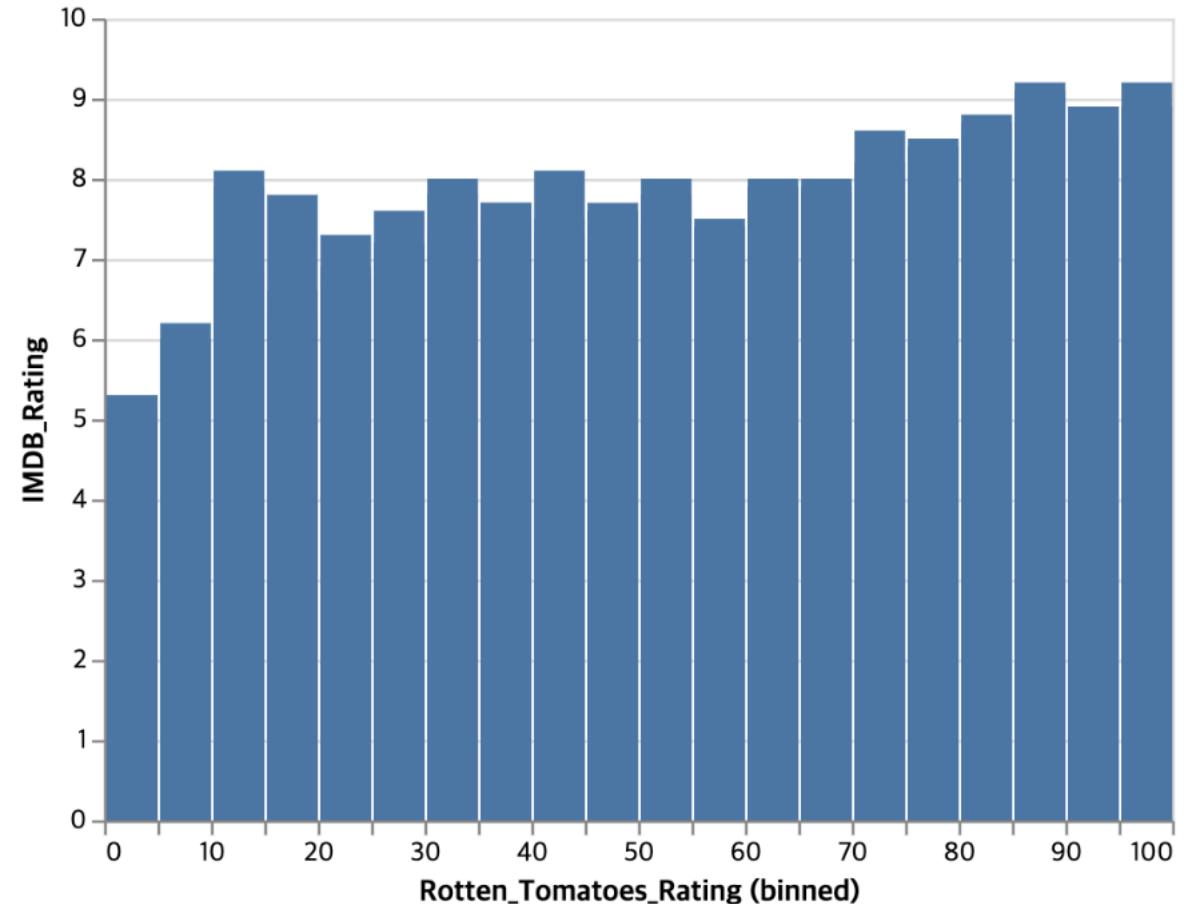
```
alt.Chart(movies).mark_circle().encode(  
    alt.X(  
        'Rotten_Tomatoes_Rating:Q',  
        bin=alt.BinParams(maxbins=20)  
    ),  
    alt.Y('IMDB_Rating:Q')  
)
```



데이터 Binning

- 비닝을 통해 히스토그램을 그릴 수 있습니다.

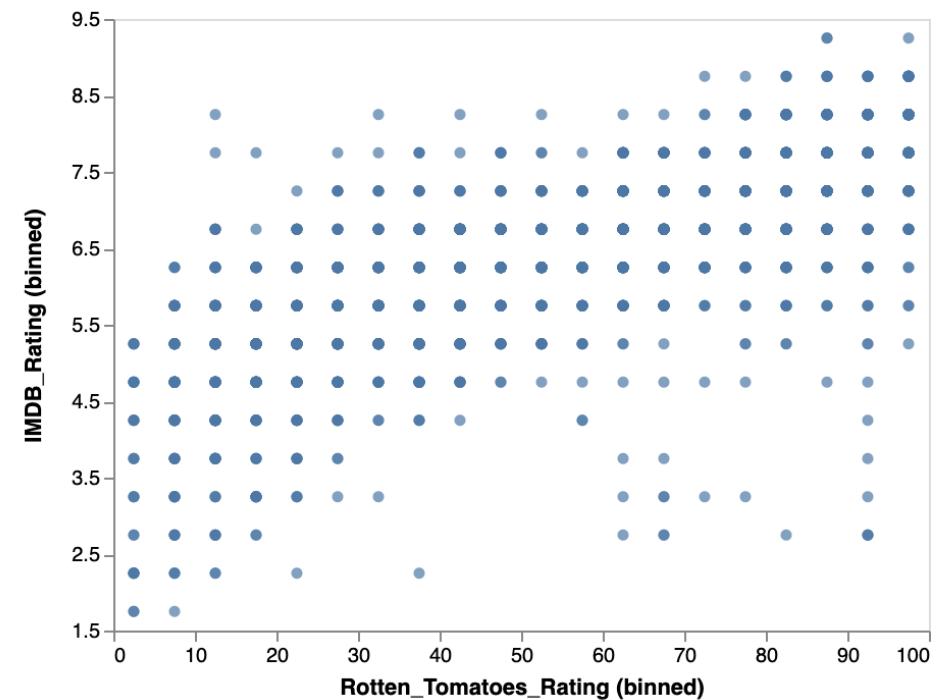
```
alt.Chart(movies).mark_bar().encode(  
    alt.X(  
        'Rotten_Tomatoes_Rating:Q',  
        bin=alt.BinParams(maxbins=20)  
    ),  
    alt.Y('IMDB_Rating:Q')  
)
```



데이터 Binning

- 다시 돌아와서, 두 리뷰 점수에 모두 비닝을 적용해 볼까요?

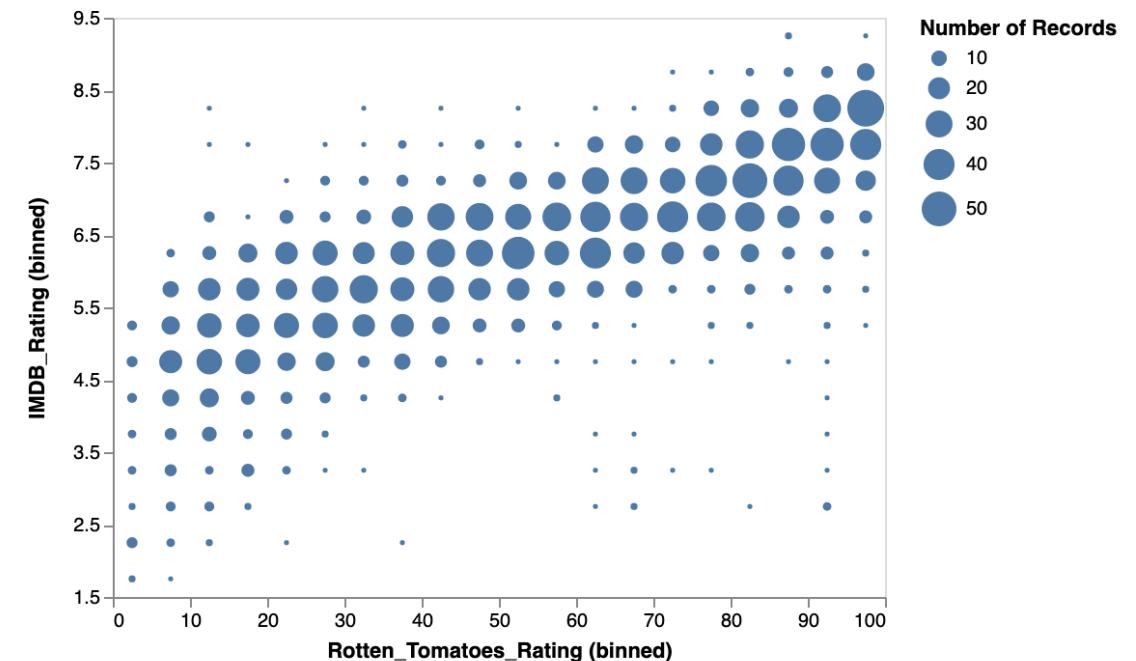
```
alt.Chart(movies).mark_circle().encode(  
    alt.X(  
        'Rotten_Tomatoes_Rating:Q',  
        bin=alt.BinParams(maxbins=20)),  
    alt.Y(  
        'IMDB_Rating:Q',  
        bin=alt.BinParams(maxbins=20)),  
)
```



데이터 Binning

- 크기에 데이터의 개수를 매핑해 봅시다.

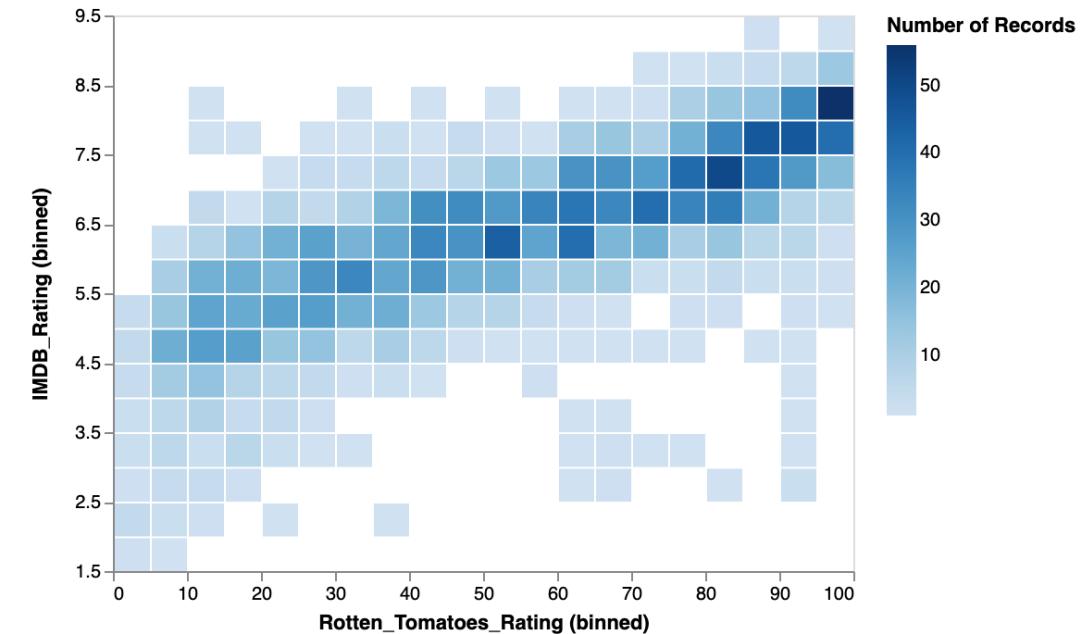
```
alt.Chart(movies).mark_circle().encode(  
    alt.X(  
        'Rotten_Tomatoes_Rating:Q',  
        bin=alt.BinParams(maxbins=20)),  
    alt.Y(  
        'IMDB_Rating:Q',  
        bin=alt.BinParams(maxbins=20)),  
    alt.Size('count()')  
)
```



데이터 Binning

- 크기 대신 색깔을 사용하고, 마크를 사각형으로 바꾸어 봅시다.
- 이번엔 열 지도(heatmap)을 만들었습니다!

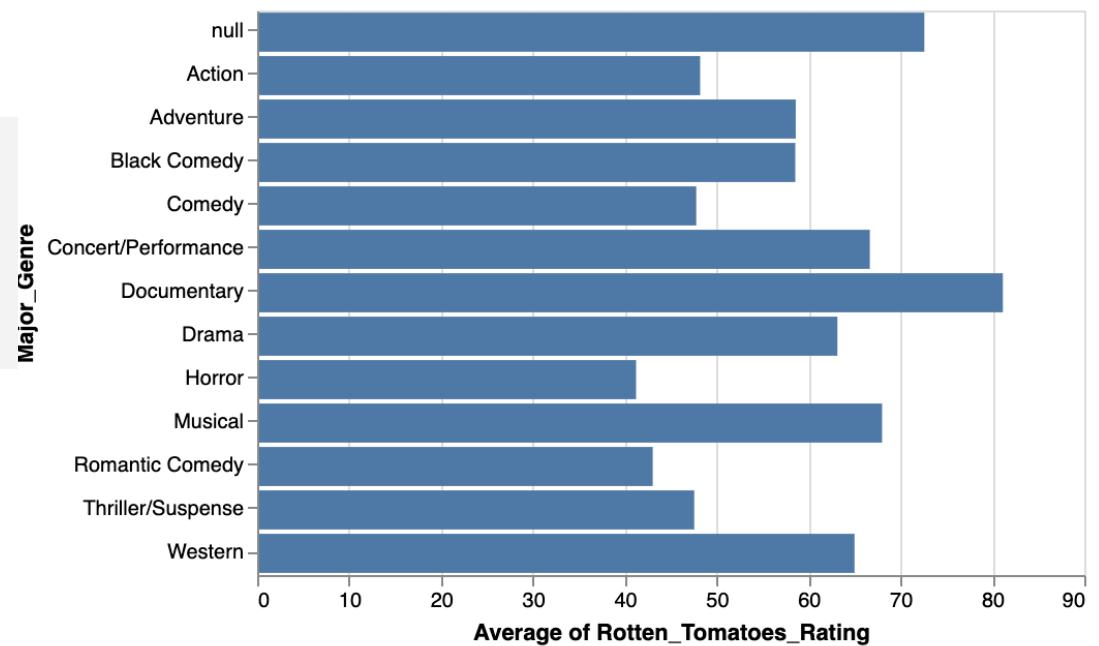
```
alt.Chart(movies).mark_bar().encode(  
    alt.X(  
        'Rotten_Tomatoes_Rating:Q',  
        bin=alt.BinParams(maxbins=20)),  
    alt.Y(  
        'IMDB_Rating:Q',  
        bin=alt.BinParams(maxbins=20)),  
    alt.Color('count()')  
)
```



데이터 Aggregation

- Aggregation변환은 여러 개의 값에 집합 함수를 적용하여 하나의 값으로 합칠 때 사용합니다.

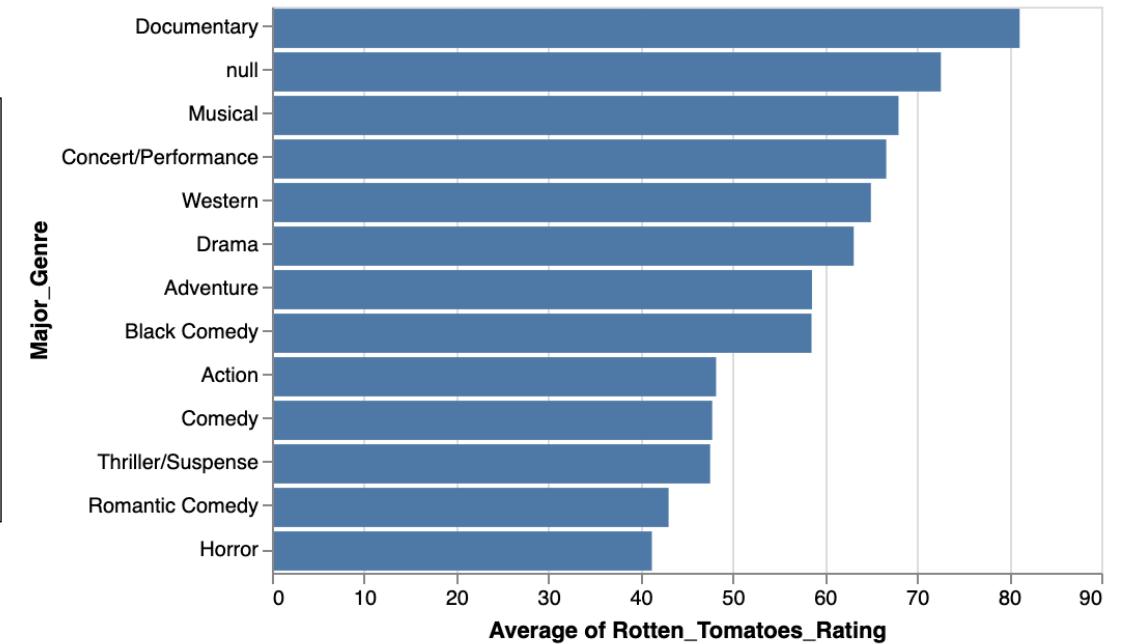
```
alt.Chart(movies).mark_bar().encode(  
    alt.X('average(Rotten_Tomatoes_Rating):Q'),  
    alt.Y('Major_Genre:N')  
)
```



데이터 Aggregation

- y축을 평균 로튼토마토 평점 순으로 정렬하여 봅시다.

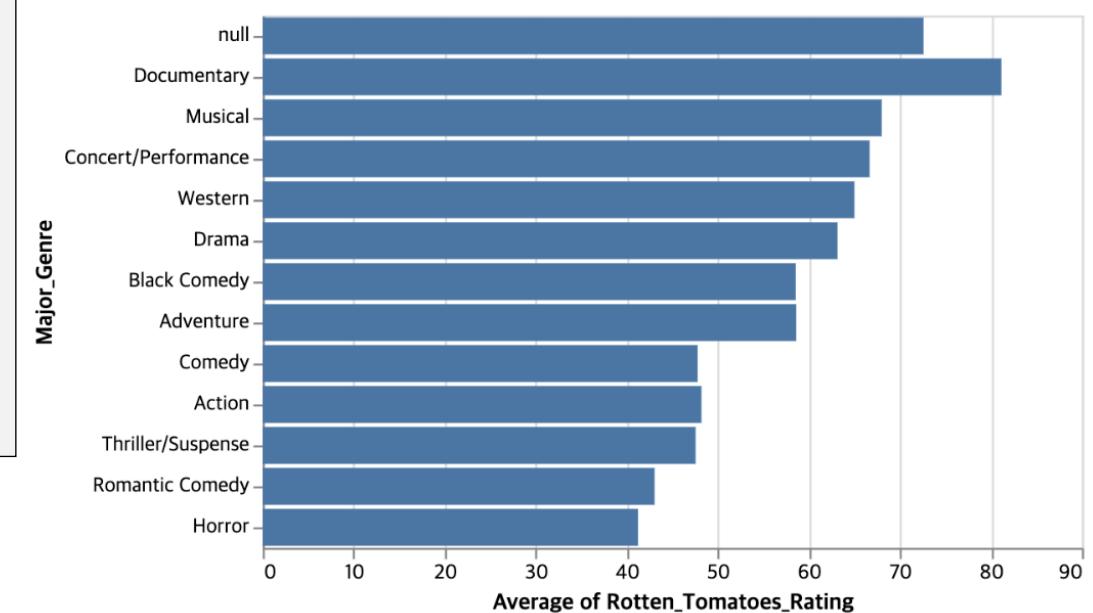
```
alt.Chart(movies).mark_bar().encode(  
    alt.X('average(Rotten_Tomatoes_Rating):Q'),  
    alt.Y('Major_Genre:N', sort=alt.EncodingSortField(  
        op='average',  
        field='Rotten_Tomatoes_Rating',  
        order='descending'  
    ))
```



데이터 Aggregation

- min, max, average, median, missing, distinct, q1, q3, stdev, stderr, sum, variance 등 다양한 집합 함수를 사용할 수 있습니다.

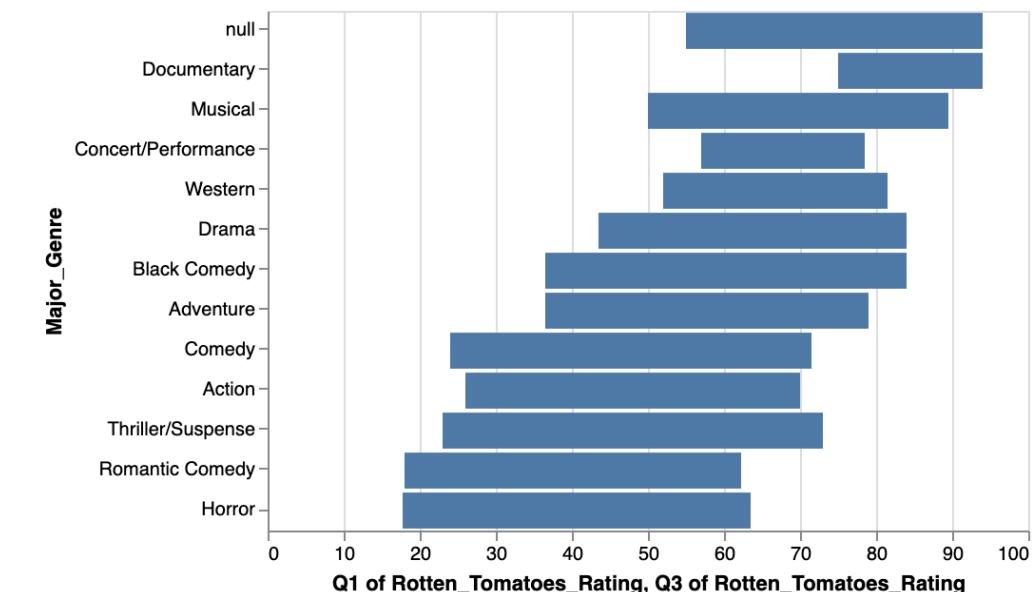
```
alt.Chart(movies).mark_bar().encode(  
    alt.X('average(Rotten_Tomatoes_Rating):Q'),  
    alt.Y('Major_Genre:N', sort=alt.EncodingSortField(  
        op='median',  
        field='Rotten_Tomatoes_Rating',  
        order='descending'  
    ))  
)
```



데이터 Aggregation

- X2를 사용할 수 있습니다.

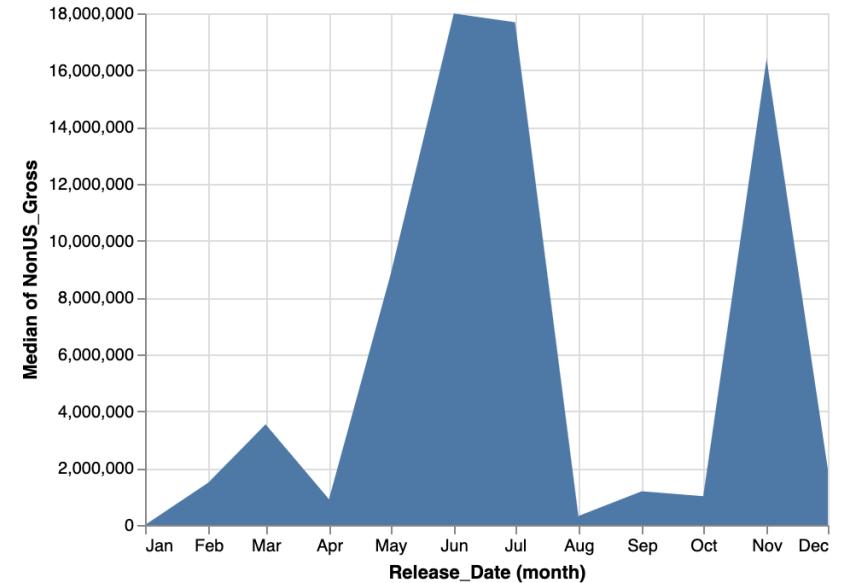
```
alt.Chart(movies).mark_bar().encode(  
    alt.X('q1(Rotten_Tomatoes_Rating):Q'),  
    alt.X2('q3(Rotten_Tomatoes_Rating):Q'),  
    alt.Y('Major_Genre:N', sort=alt.EncodingSortField(  
        op='average',  
        field='Rotten_Tomatoes_Rating',  
        order='descending'  
)  
)
```



transform_calculate()

- transform_calculate()로 기존의 필드를 이용해 새로운 필드를 만들 수 있습니다.

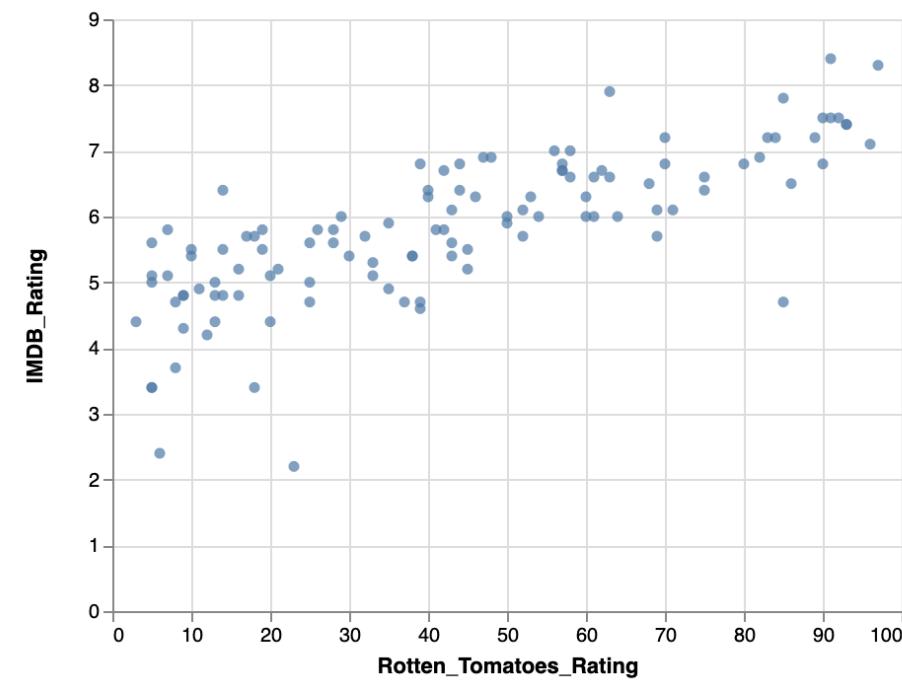
```
alt.Chart(movies).mark_area().transform_calculate(  
    NonUS_Gross='datum.WORLDWIDE_GROSS - datum.US_GROSS'  
).encode(  
    alt.X('month(Release_Date):T'),  
    alt.Y('median(NonUS_Gross):Q')  
)
```



transform_filter()

- transform_filter()는 조건을 만족하는 행들만 남기는 기능입니다.

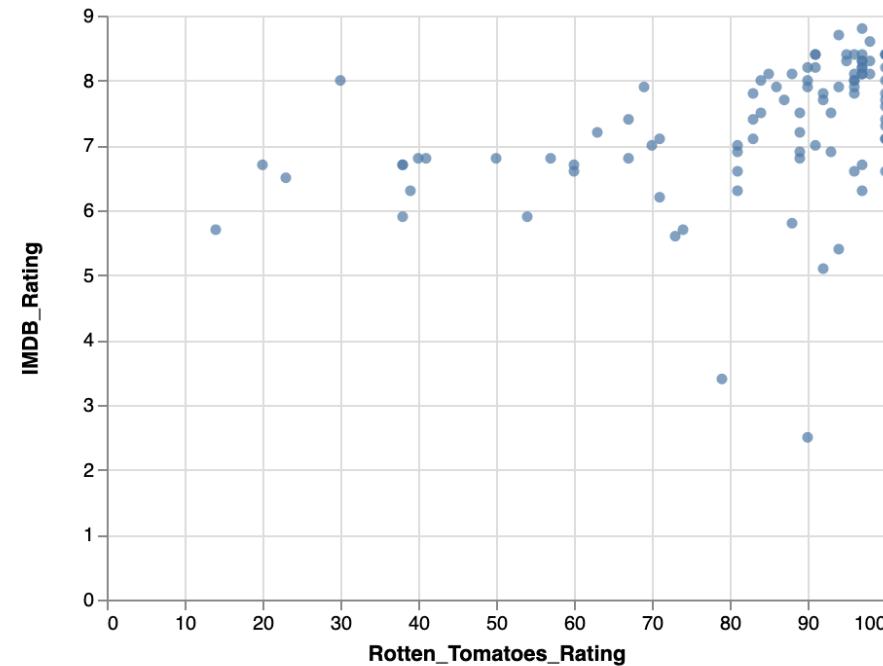
```
alt.Chart(movies).mark_circle().encode(  
    alt.X('Rotten_Tomatoes_Rating:Q'),  
    alt.Y('IMDB_Rating:Q')  
).transform_filter(  
    'datum.Major_Genre == "Romantic Comedy"'  
)
```



transform_filter()

- <, >, <=, >=, &&, || 등 JavaScript에서 사용하는 다양한 조건 연산자를 사용할 수 있습니다.

```
alt.Chart(movies).mark_circle().encode(  
    alt.X('Rotten_Tomatoes_Rating:Q'),  
    alt.Y('IMDB_Rating:Q')  
)  
.transform_filter(  
    'year(datum.Release_Date) < 1970'  
)
```



(응용) transform_filter()로 인터랙티브 차트 만들기

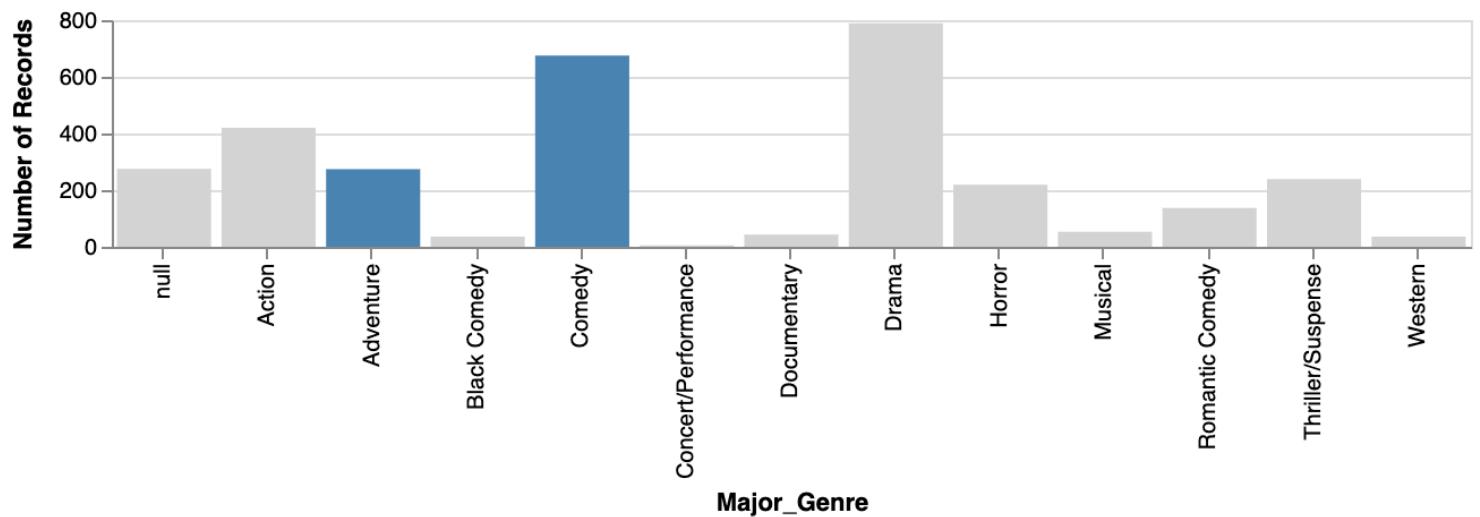
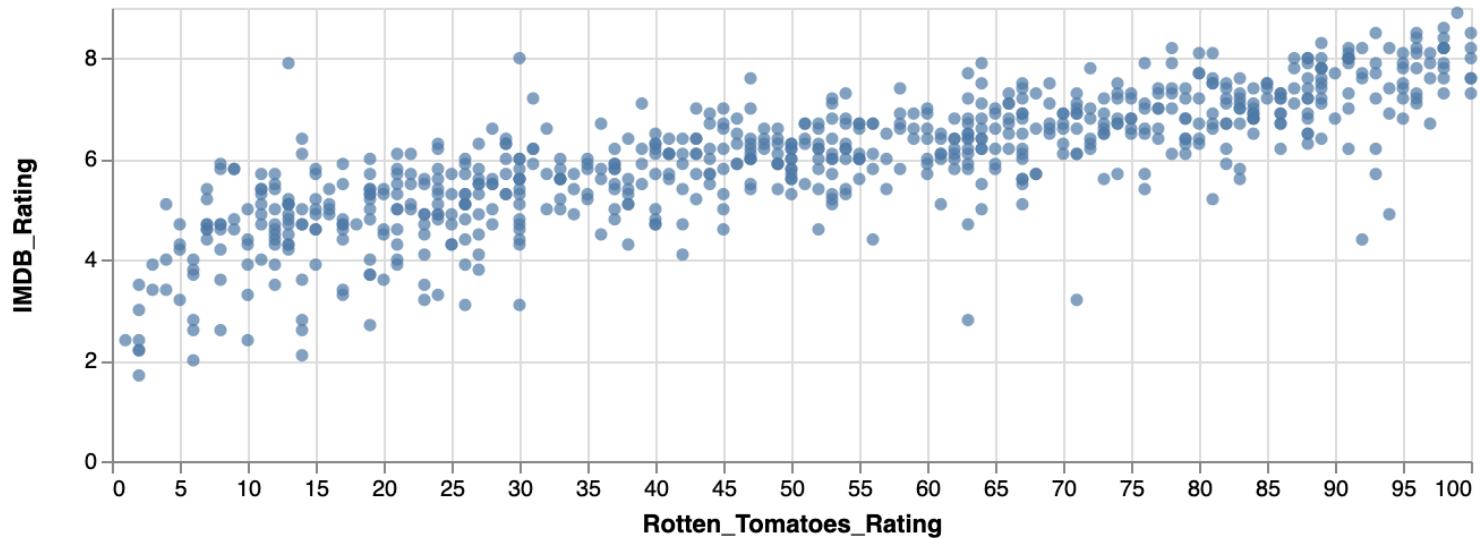
```
selection = alt.selection_multi(fields=['Major_Genre'])

top=alt.Chart().mark_circle().encode(
    x='Rotten_Tomatoes_Rating:Q',
    y='IMDB_Rating:Q'
).properties(
    width=600, height=200
).transform_filter(
    selection
)

bottom = alt.Chart().mark_bar().encode(
    x='Major_Genre:O',
    y='count()',
    color=alt.condition(selection, alt.value('steelblue'), alt.value('lightgray'))
).properties(
    width=600, height=100,
    selection=selection
)

alt.vconcat(
    top, bottom,
    data=movies
)
```

(응용) transform_filter()로 인터랙티브 차트 만들기



관련 링크

- <https://altair-viz.github.io/>
- <https://altair-viz.github.io/gallery/index.html>
 - Altair로 그린 여러가지 차트를 볼 수 있는 갤러리
- https://altair-viz.github.io/user_guide/troubleshooting.html
 - 설치 문제 발생시 한번 확인하면 좋음
- https://github.com/altair-viz/vega_datasets
 - 테스트용 작은 데이터를 모아둔 리파지토리