

삼성전자 AI전문가 교육과정 사전교육 - 데이터 분석 및 시각화

Altair 실습 3 - selection

Altair 실습 3

Interaction

Selection, condition, bind

- Altair가 Vega-Lite로 부터 계승한 중요한 요소중 하나는 declarative grammar
- 3 core concepts: selection(), condition(), bind
- Selection()
 - Selection object는 마우스나 인풋 기기들로부터 interaction을 capture합니다.
 - Input은 마우스 클릭이나, 드래그 같은 동작들이 포함됩니다.
 - Drop-down, radio button 또는 슬라이더와 같은 element 역시 input이 될 수 있습니다.
 - Selection 자체만 사용도 가능하지만 차트의 element를 바꾸고 싶다면 selection을 condition에 연결해야 합니다.

Selection, condition, bind

- Condition()
 - selection의 input을 가져가서
 - 이를 바탕으로 차트의 element를 변경
- Bind
 - selection과 차트의 input element 사이에 two-way binding을 설정하게 해주는 selection의 property
- Interactive chart는 이러한 요소를 하나 이상 사용하여 사용자와 데이터간의 interaction을 지원합니다.

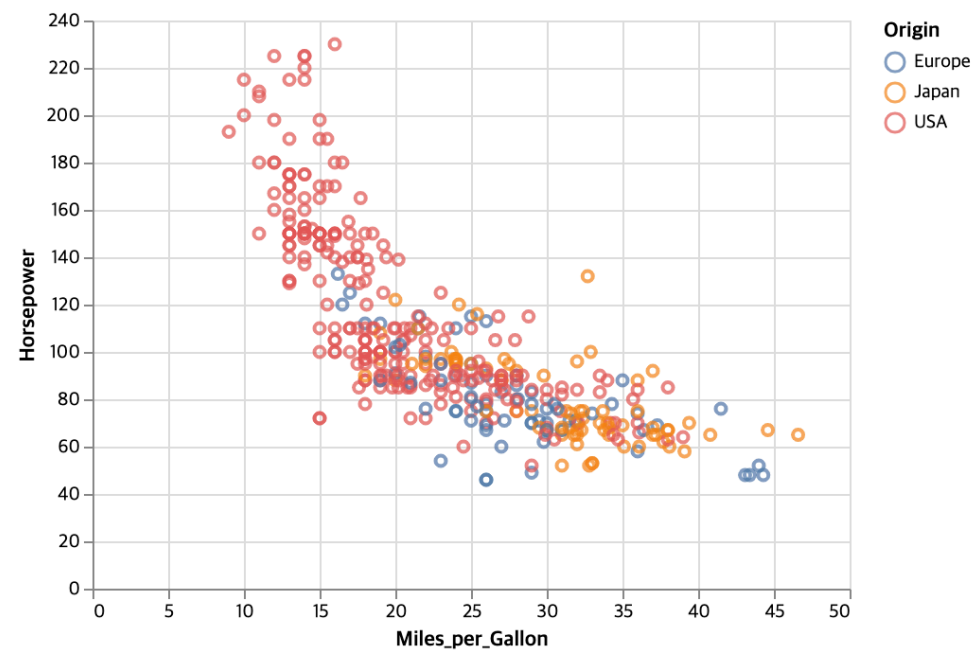
Linked-Brush Scatter-Plot

```
import altair as alt
import pandas as pd

from vega_datasets import data

cars = data.cars.url

alt.Chart(cars).mark_point().encode(
    x='Miles_per_Gallon:Q',
    y='Horsepower:Q',
    color='Origin:N'
)
```



Linked-Brush Scatter-Plot

- `selection_interval()`을 사용하여 interval selection을 만들어줍니다.

```
brush = alt.selection_interval() # selection of type "interval"
```

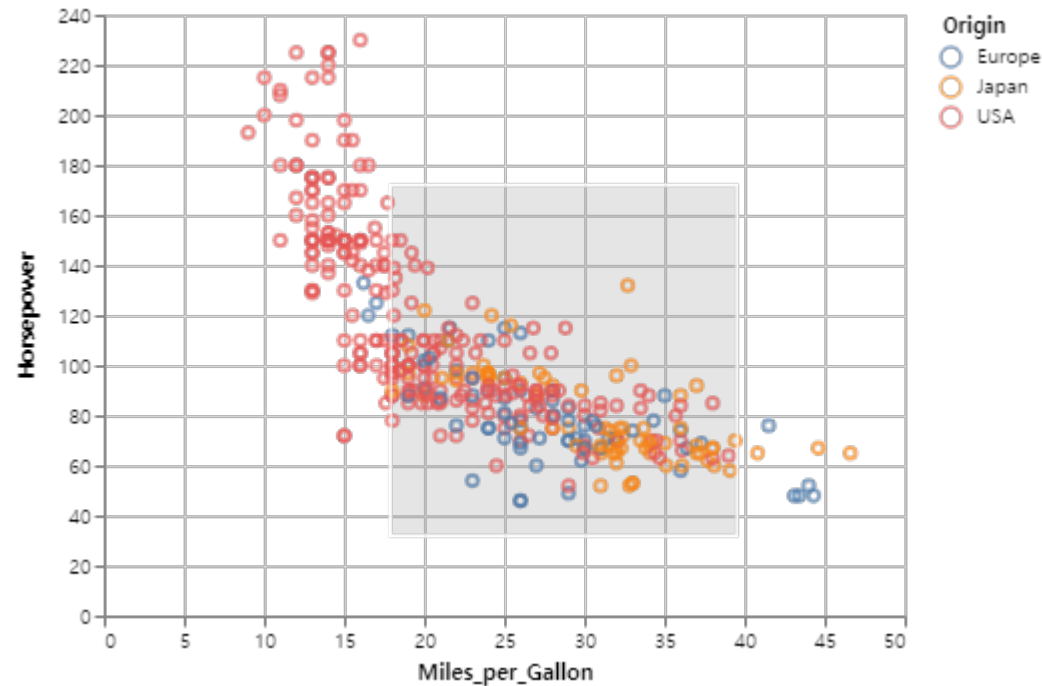
- 만들어진 brush를 selection property를 사용하여 차트에 bind 해봅시다.

```
cars = data.cars.url

alt.Chart(cars).mark_point().encode(
    x='Miles_per_Gallon:Q',
    y='Horsepower:Q',
    color='Origin:N'
).add_selection(
    brush
)
```

Linked-Brush Scatter-Plot

- 마우스를 사용하여 영역을 선택하고, 선택된 부분을 움직일 수 있게 되었습니다.
 - 아직 condition에 연결되지 않았으므로 차트의 element에는 변화가 없습니다.



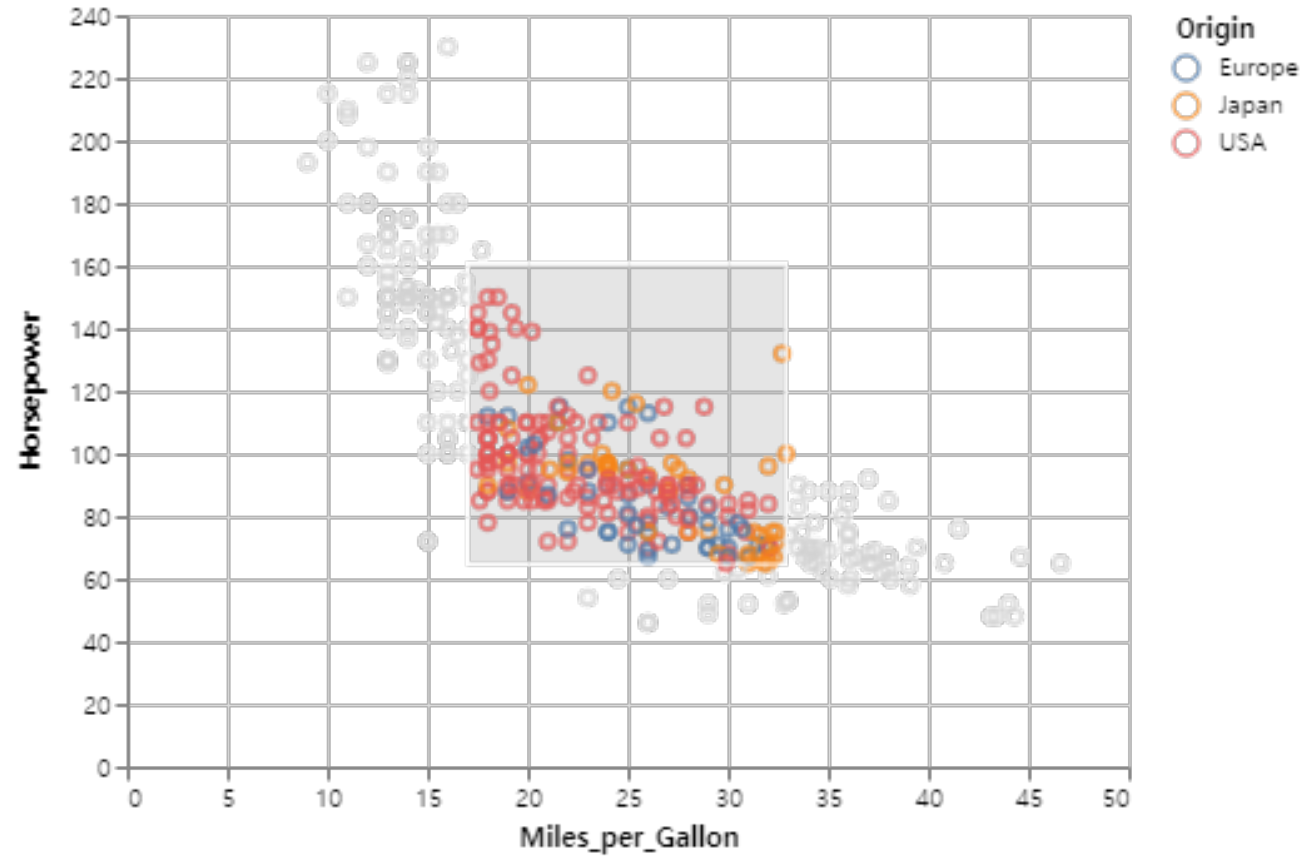
Linked-Brush Scatter-Plot

- 차트에서 만들어진 selection을 사용하기 위해서는 차트 내부에서 이를 참고해야 합니다.
- Condition()을 사용하여 Conditional color encoding을 구현해 봅시다.
 - 먼저 Color를 “Origin” column에 연결해주고
 - Selection 영역 밖의 point들은 “lightgray”로 색을 설정하도록 합시다.

```
alt.Chart(cars).mark_point().encode(  
  x='Miles_per_Gallon:Q',  
  y='Horsepower:Q',  
  color=alt.condition(brush, 'Origin:N', alt.value('lightgray'))  
)  
.add_selection(  
  brush  
)
```


Linked-Brush Scatter-Plot

- Linked-Brush Scatter-Plot



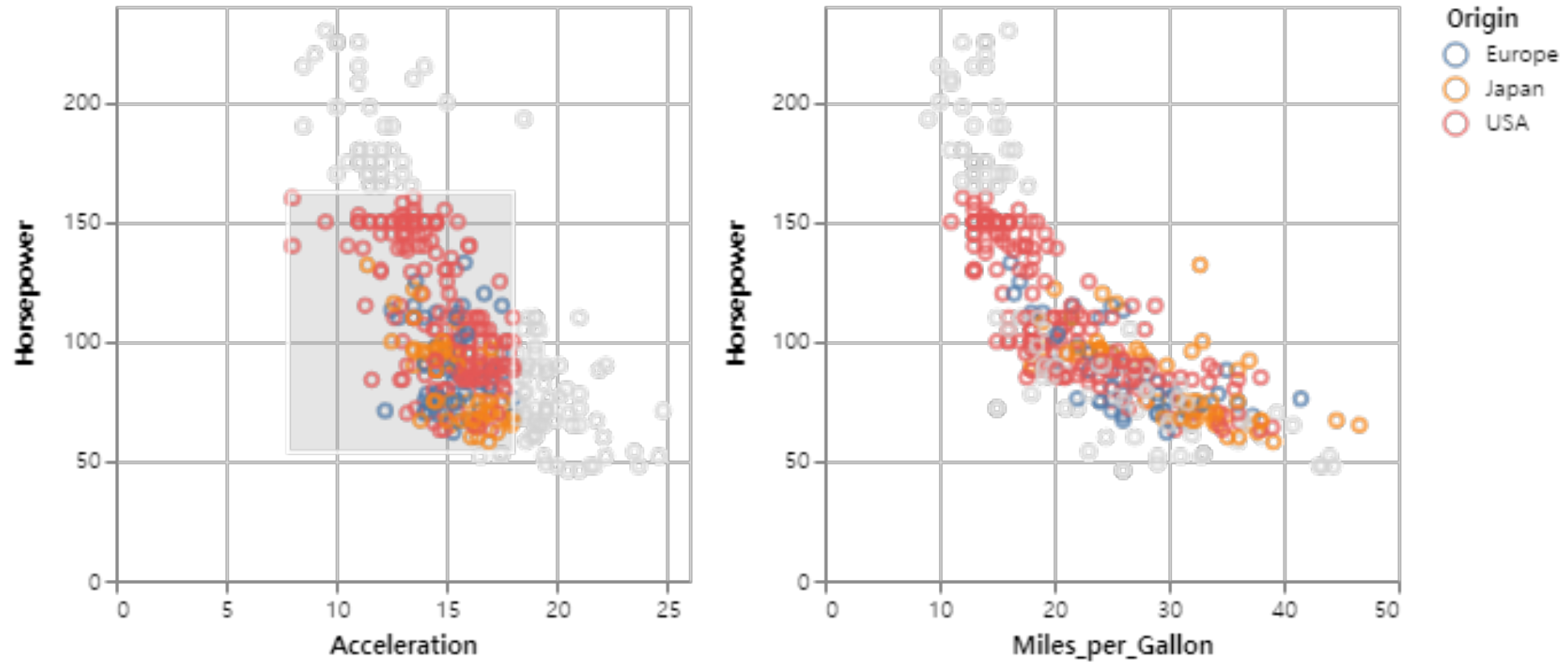
Linked-Brush Scatter-Plot

- 이제 차트가 selection에 반응하여 point들의 색을 바꿀 수 있게 되었습니다.
- 이번에는 여러개의 차트들이 selection에 연동되게 해봅시다.

```
chart = alt.Chart(cars).mark_point().encode(  
    y='Horsepower:Q',  
    color=alt.condition(brush, 'Origin:N', alt.value('lightgray'))  
)  
.properties(  
    width=250,  
    height=250  
)  
.add_selection(  
    brush  
)  
  
chart.encode(x='Acceleration:Q') | chart.encode(x='Miles_per_Gallon:Q')
```

Linked-Brush Scatter-Plot

- Linked-Brush Scatter-Plot

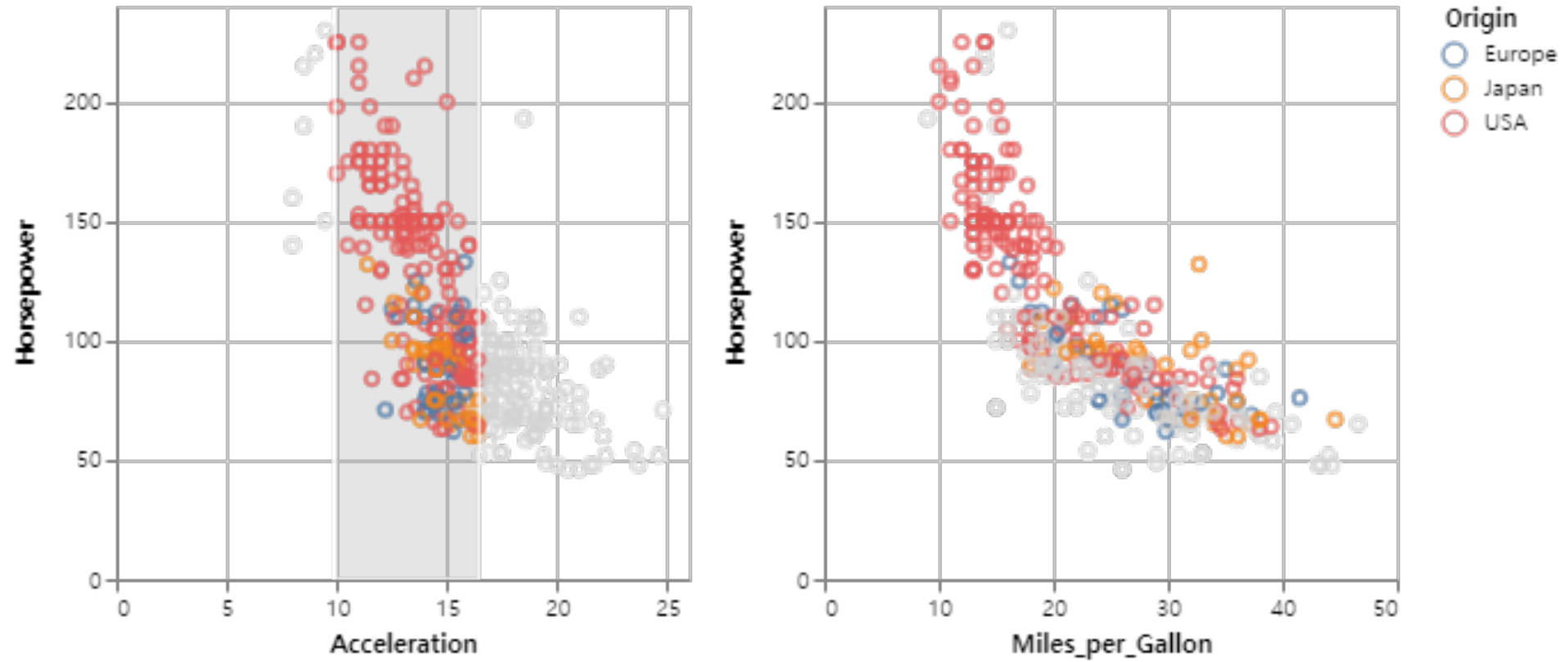


Linked-Brush Scatter-Plot

- 두 차트가 동일한 selection object를 참고하고 있기 때문에 renderer는 selection을 panel들 간에 묶어줄 수 있습니다.
- 이러한 시각화는 dataset간의 관계를 탐색할때 유용하게 사용될 수 있습니다.
- 아래와 같이 selection을 수정하여 brush가 특정 encoding에만 묶이도록 할 수 도 있습니다.

```
brush = alt.selection_interval(encodings=['x'])
```

Linked-Brush Scatter-Plot



Altair 실습 3

Selection Types

Selection Types

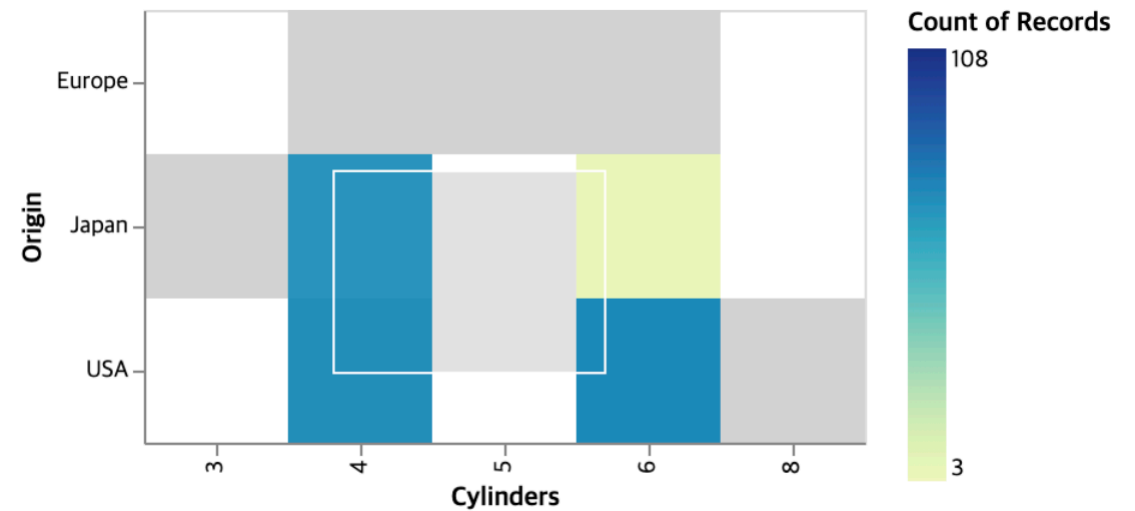
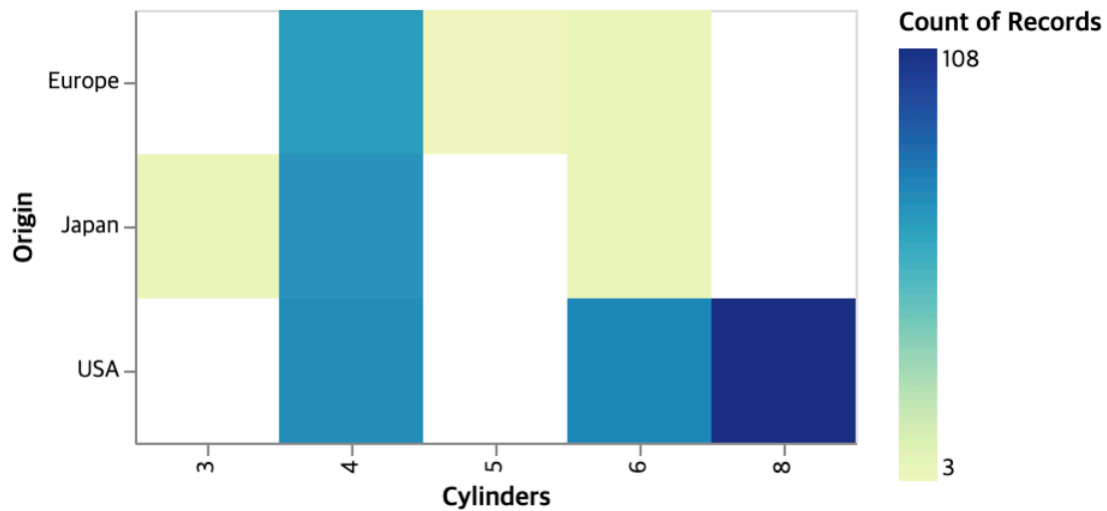
- 이번에는 Altair에서 지원하는 다양한 Selection type들을 알아보고 이를 사용해보도록 합시다.
- 차트를 생성하고 차트의 color와 selection object를 연결해주는 함수를 만듭시다.

```
def make_example(selector):  
    cars = data.cars.url  
  
    return alt.Chart(cars).mark_rect().encode(  
        x='Cylinders:O',  
        y='Origin:N',  
        color=alt.condition(selector, 'count()', alt.value('lightgray'))  
    ).properties(  
        width=300,  
        height=180  
    ).add_selection(  
        selector  
    )
```

Interval selection

- Scatter-plot 예제에서 사용하였던 `selection_interval()`을 사용해봅시다.

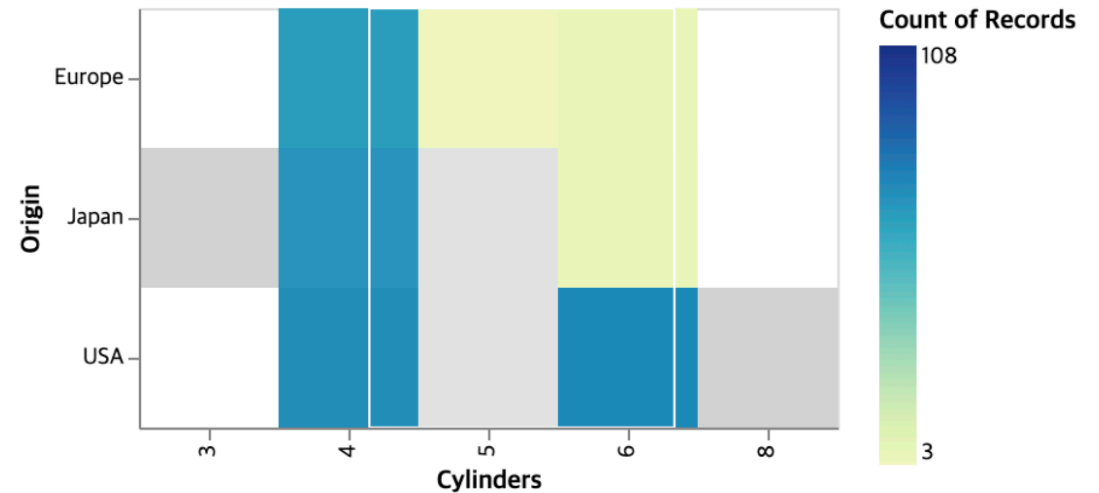
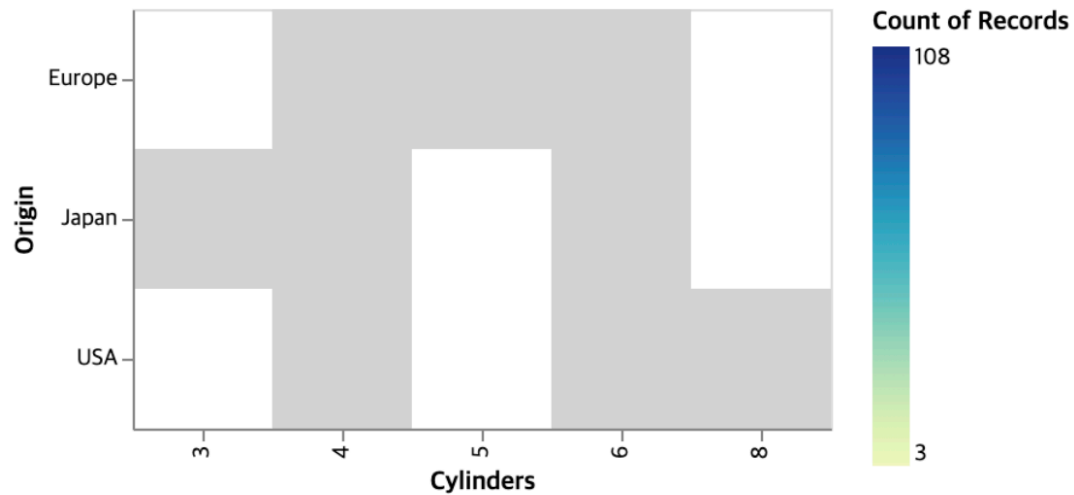
```
interval = alt.selection_interval()  
make_example(interval)
```



Interval selection

- Selection interval를 x축으로 제한해보고 어떤 차이가 있는지 비교해 봅시다.
 - 좀 전에 했던 것!
 - Empty attribute도 넣어봅시다

```
interval = alt.selection_interval(encodings=['x'], empty='none')  
make_example(interval)
```



Interval selection

- 이번에는 selection interval를 차트의 scale에 bind 해보고 어떤 일이 일어나는지 살펴봅시다.

```
scales = alt.selection_interval(bind="scales")

alt.Chart(cars).mark_point().encode(
    x="Horsepower:Q",
    y="Miles_per_Gallon:Q",
    color="Origin:N"
).add_selection(
    scales
)
```

Interval selection

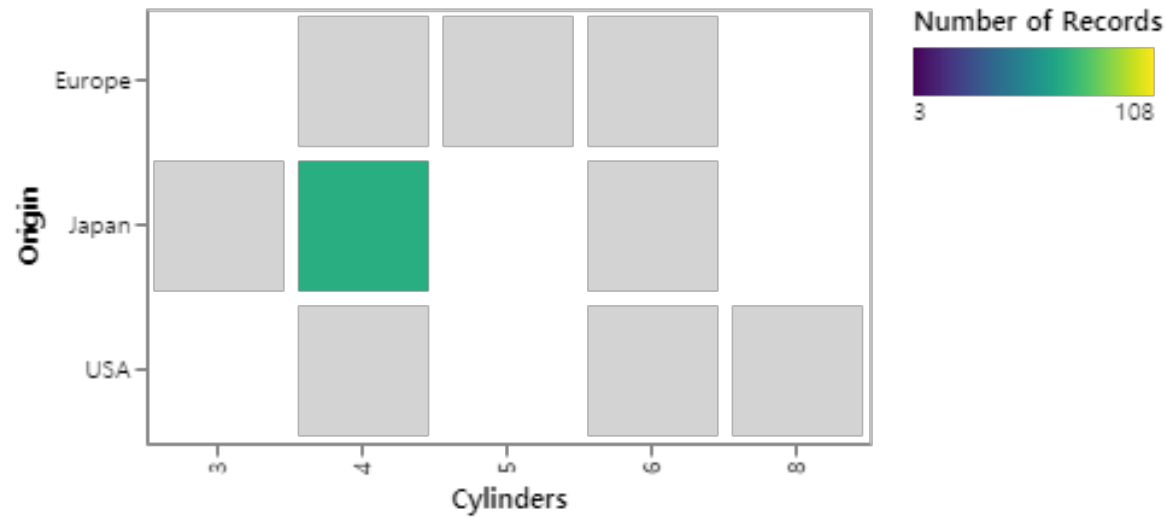
- 시각화에서 자주 쓰이는 interaction이므로 Altair는 `chart.interactive()` 메소드를 통해 동일한 기능을 간결하게 제공합니다.

```
alt.Chart(cars).mark_point().encode(  
    x="Horsepower:Q",  
    y="Miles_per_Gallon:Q",  
    color="Origin:N"  
).interactive()
```

Single selection

- Single selection은 마우스를 사용하여 한개의 element를 선택하게 해줍니다.
 - 기본적으로 마우스 클릭을 사용

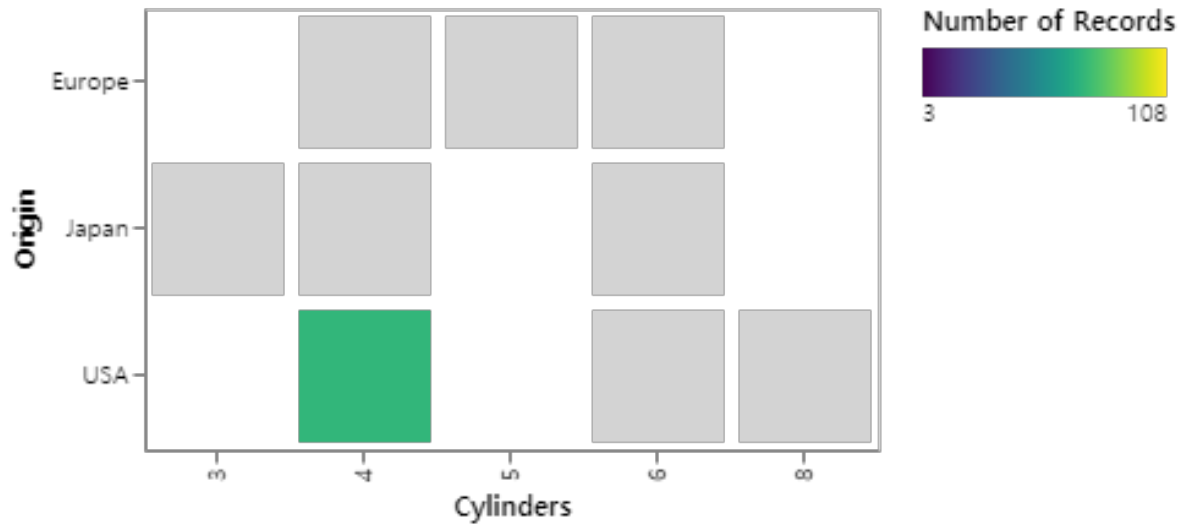
```
single = alt.selection_single()  
make_example(single)
```



Single selection

- 이번에는 mouseover를 하였을 때 highlight가 되도록 해봅시다.
 - Nearest flag를 끈 경우와 켜진 경우의 차이도 보도록 합시다.

```
single_nearest = alt.selection_single(on="mouseover", nearest=False)  
make_example(single_nearest)
```

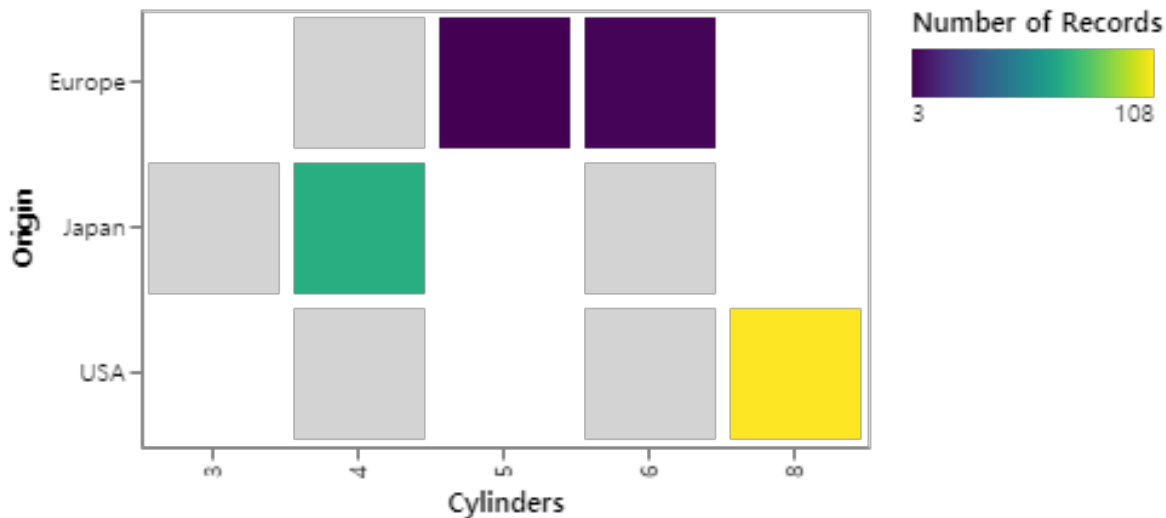


Multiple selection

- Multiple selection은 차트에서 여러개의 element들을 선택할 수 있게 해줍니다.
 - 기본적으로 shift키를 누르고 클릭함으로써 element들을 추가하고 제외할 수 있습니다.
 - Selection안에 toggle=False 옵션을 켜보고 어떤 일이 발생하는지 살펴봅시다.

```
multi = alt.selection_multi()  
make_example(multi)
```

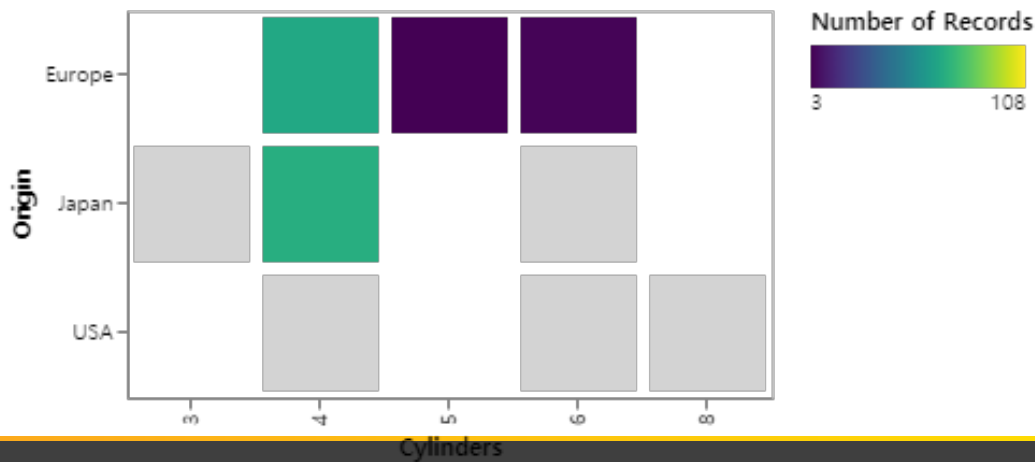
```
multi = alt.selection_multi(toggle=False)  
make_example(multi)
```



Multiple selection

- 이번에는 앞선 예제들을 응용해서 마우스가 hovering을 하면 element들이 선택되는 예제를 만들어 봅시다.
 - 새로운 element가 선택되었을 때 이전에 선택한 element가 풀리면 안되므로 toggle parameter는 false로
 - Empty selection은 element를 포함하지 않도록
 - toggle option 없이도 해보세요!!

```
multi_mouseover = alt.selection_multi(on="mouseover", toggle=True, empty="none")  
make_example(multi_mouseover)
```



Selection targets

- Selection을 사용함에 있어서 사용자는 selection이 무엇을 가리킬지 정해야 한다.
- 이러한 동작은 fields나 encodings argument를 사용하여 정할 수 있다.
- 이제 다시 scatter-plot 예제로 돌아와서, interactive legend를 가지도록 아까 만들었던 scatter-plot을 수정해보자
 - Selection이 “Origin” field를 targeting하도록 하면 된다.
 - 선택된 포인트들 외에는 lightgray로 색을 칠하도록 하고
 - Legend도 하나 만들어서 selection을 붙여주자.

Selection targets

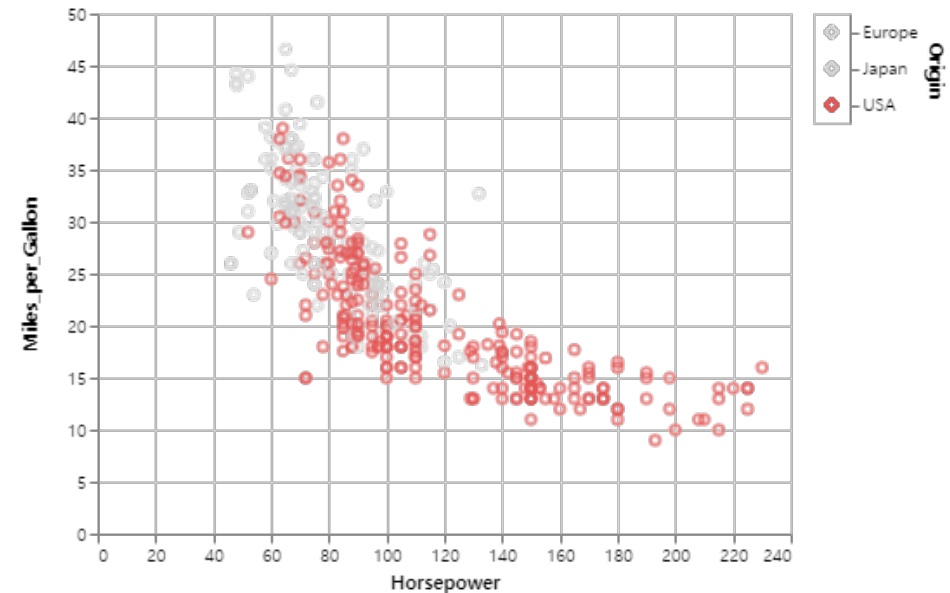
```
selection = alt.selection_multi(fields=["Origin"])
color = alt.condition(selection,
                        alt.Color('Origin:N', legend=None),
                        alt.value('lightgray'))

scatter = alt.Chart(cars).mark_point().encode(
    x='Horsepower:Q',
    y="Miles_per_Gallon:Q",
    color=color,
    tooltip="Name:N"
)
```

```
legend = alt.Chart(cars).mark_point().encode(
    y=alt.Y('Origin:N', axis=alt.Axis(orient="right")),
    color=color
).add_selection(
    selection
)

scatter | legend
```

- Encoding을 사용해서 동일한 동작을 수행하려면 어떻게 해야 할까?



Selection targets

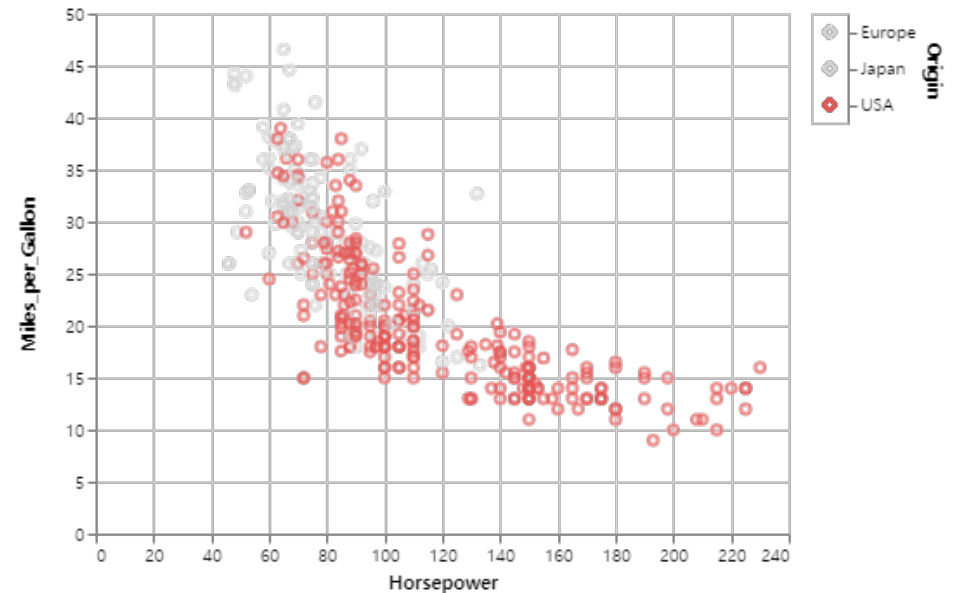
```
selection = alt.selection_multi(encodings=["color"])
color = alt.condition(selection,
                        alt.Color('Origin:N', legend=None),
                        alt.value('lightgray'))

scatter = alt.Chart(cars).mark_point().encode(
    x='Horsepower:Q',
    y="Miles_per_Gallon:Q",
    color=color,
    tooltip="Name:N"
)
```

```
legend = alt.Chart(cars).mark_point().encode(
    y=alt.Y('Origin:N', axis=alt.Axis(orient="right")),
    color=color
).add_selection(
    selection
)

scatter | legend
```

- Color encoding을 가리키게하면 된다.
- 차트의 color encoding이 'Origin'과 연결되어 있으므로



Selection targets

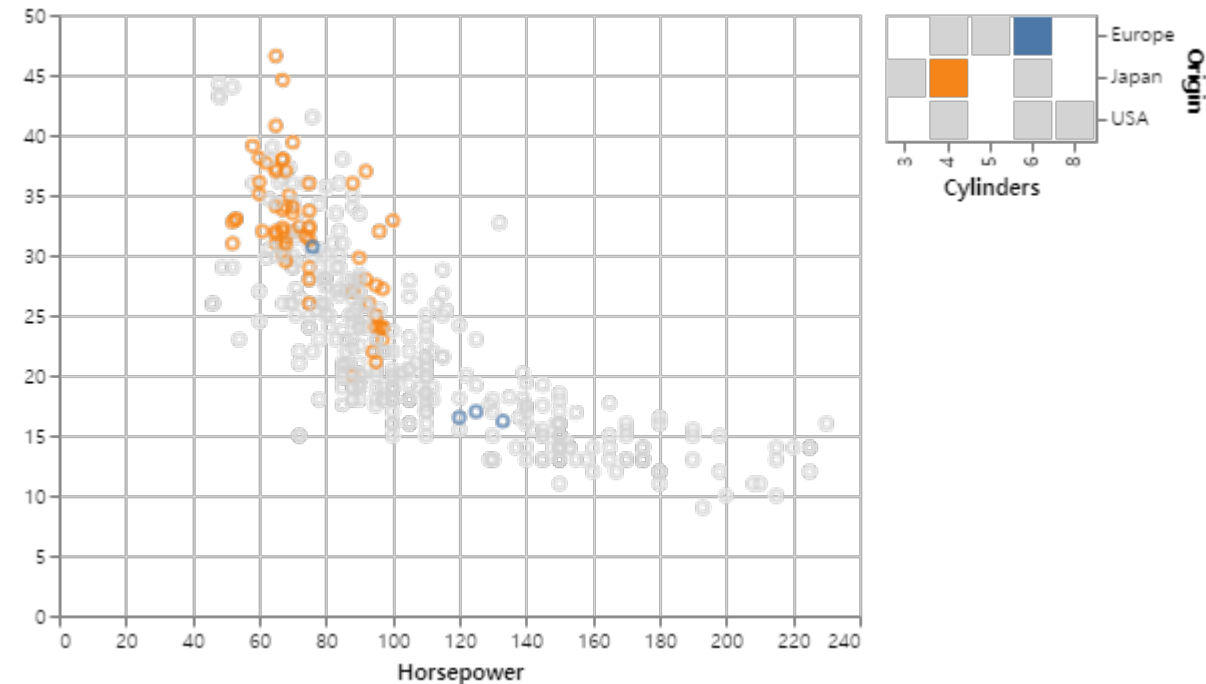
- 여러 개의 field 또는 encoding이 selection에 포함되게 할 수도 있다.

```
selection = alt.selection_multi(fields=["Origin", "Cylinders"])
color = alt.condition(selection,
    alt.Color('Origin:N', legend=None),
    alt.value('lightgray'))

scatter = alt.Chart(cars).mark_point().encode(
    x='Horsepower:Q',
    y='Miles_per_Gallon:Q',
    color=color,
    tooltip='Name:N'
)

legend = alt.Chart(cars).mark_rect().encode(
    y=alt.Y('Origin:N', axis=alt.Axis(orient="right")),
    x='Cylinders:O',
    color=color
).add_selection(selection)

scatter | legend
```



Altair 실습 3

Binding

Binding

- Single selection의 경우 input과 selection간에 binding을 만드는데 dropdown과 같은 input element를 사용할 수 있다.
- Binding과 input element들은 client side에서 data를 filtering하는데 유용하게 사용될 수 있습니다.

Input Element	Description
<code>binding_checkbox</code>	Renders as checkboxes allowing for multiple selections of items.
<code>binding_radio</code>	Radio buttons that force only a single selection
<code>binding_select</code>	Drop down box for selecting a single item from a list
<code>binding_range</code>	Shown as a slider to allow for selection along a scale.

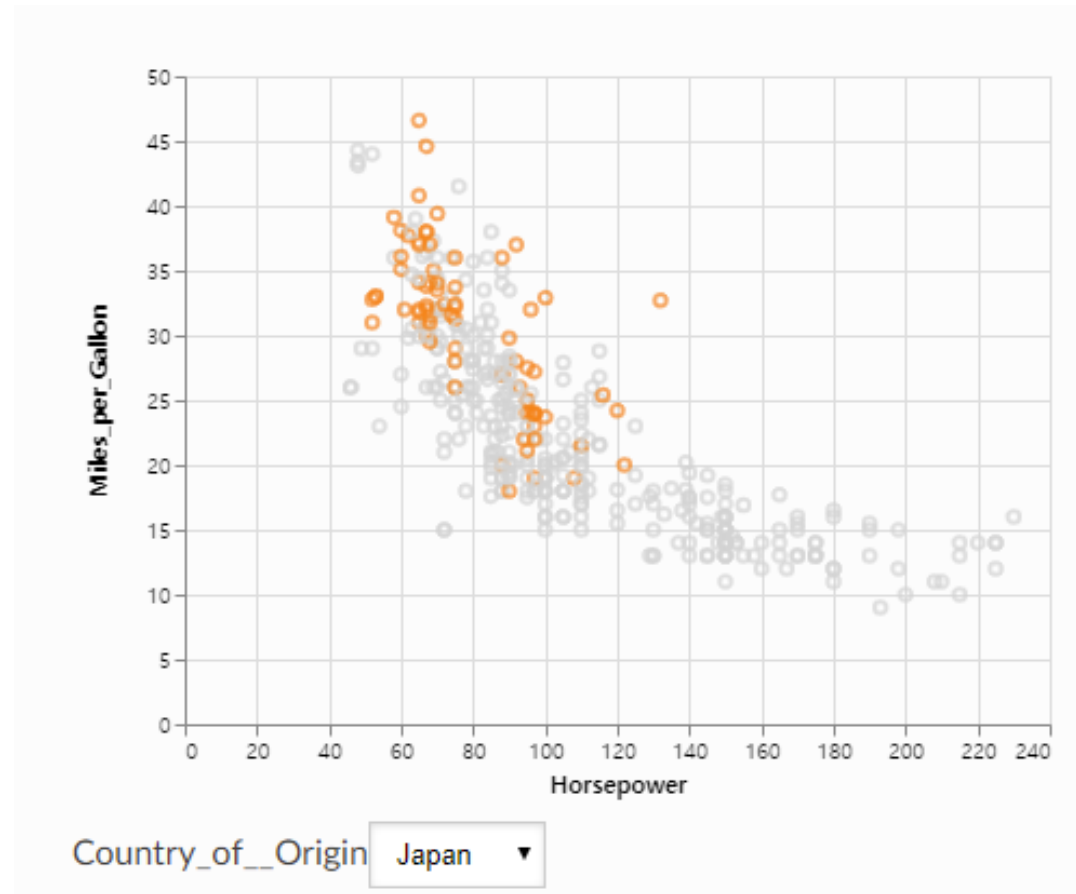
Binding

- 아래의 예제는 bind, selection, condition이 모두 적용된 예제이다.

```
input_dropdown = alt.binding_select(options=["Europe", "Japan", "USA"])
selection = alt.selection_single(fields=["Origin"], bind=input_dropdown)
color = alt.condition(selection,
                        alt.Color('Origin:N', legend=None),
                        alt.value('lightgray'))

alt.Chart(cars).mark_point().encode(
    x='Horsepower:Q',
    y="Miles_per_Gallon:Q",
    color=color,
    tooltip="Name:N"
).add_selection(
    selection
)
```

Binding

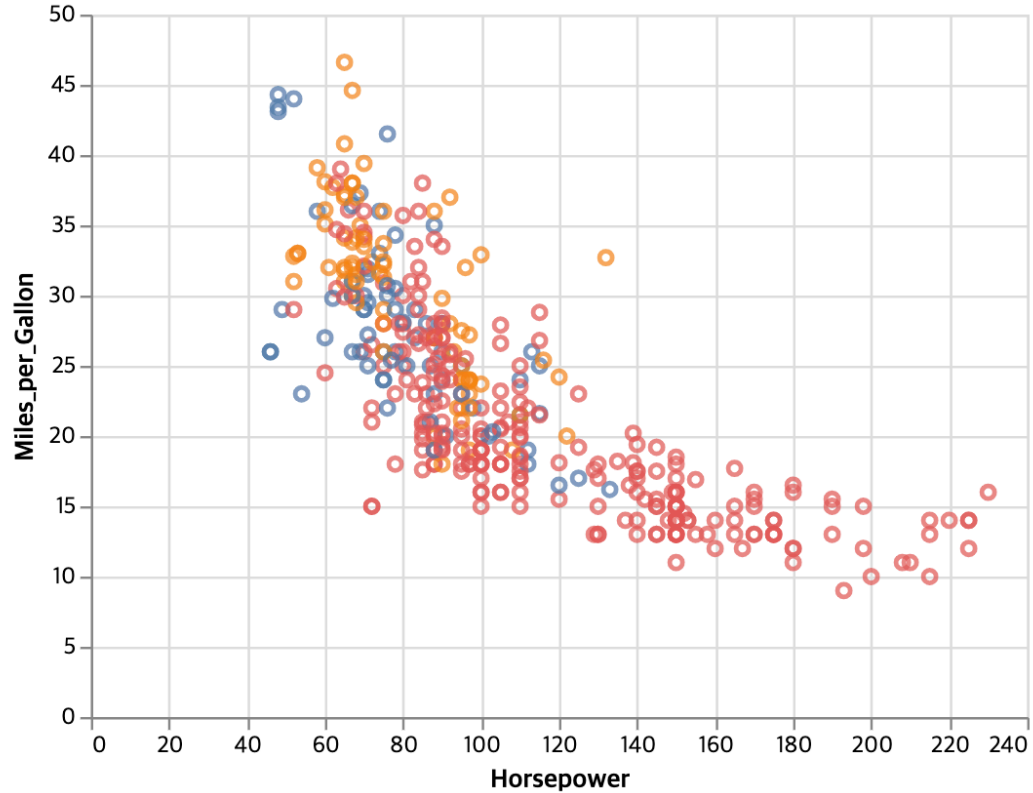
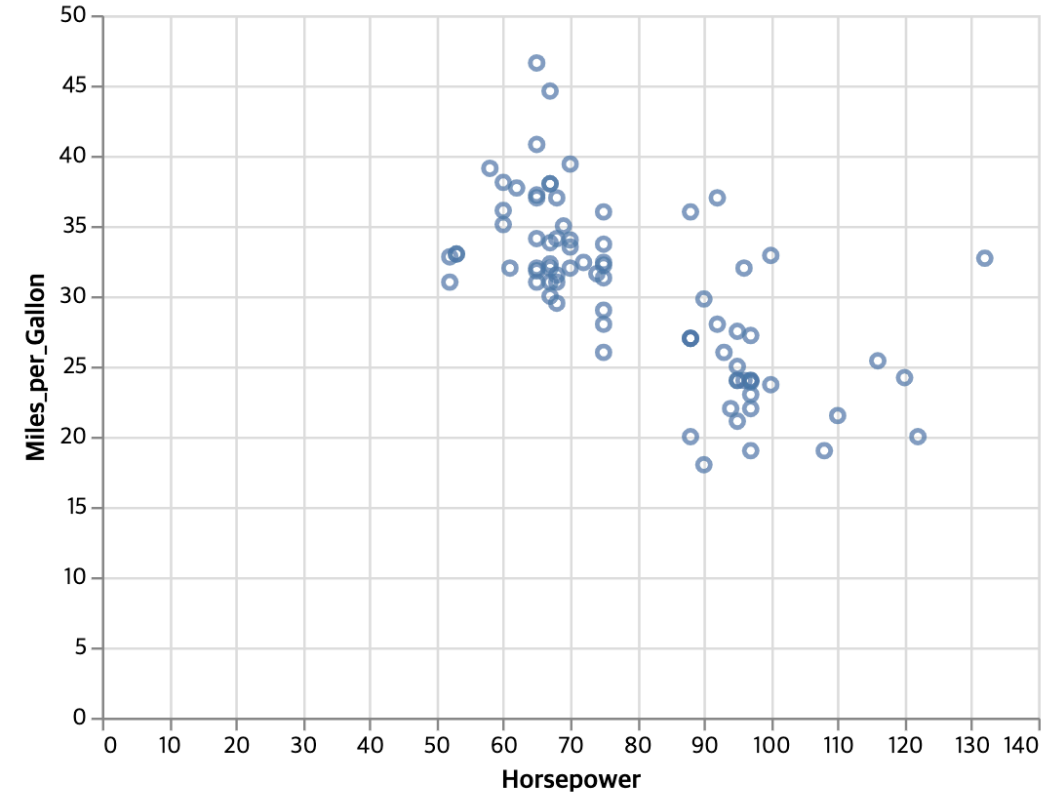


Binding

- 이제 위에서 구현한 Scatterplot 예제를 살짝 수정하여 Dropdown box에서 선택된 item만이 Scatterplot에 표시되도록 해봅시다.
- Transform filter를 적용하시면 손쉽게 구현하실 수 있습니다.

```
alt.Chart(cars).mark_point().encode(  
    x='Horsepower:Q',  
    y='Miles_per_Gallon:Q',  
    color=color,  
    tooltip='Name:N'  
)  
.add_selection(  
    selection  
)  
.transform_filter(  
    selection  
)
```


Binding

Country of Origin Country of Origin

Binding

- Interval selection을 사용하면 scale과 bind를 하여 손쉽게 panning과 zooming을 구현하실 수 있습니다.

```
selection = alt.selection_interval(bind='scales')
color = alt.condition(selection,
                        alt.Color('Origin:N', legend=None),
                        alt.value('lightgray'))
```

```
alt.Chart(cars).mark_point().encode(
    x='Horsepower:Q',
    y="Miles_per_Gallon:Q",
    color=color,
    tooltip="Name:N"
).add_selection(
    selection
)
```

