

Untitled

November 5, 2025

```
[1]: import altair as alt
import pandas as pd
cars = 'https://cdn.jsdelivr.net/npm/vega-datasets@1/data/cars.json'
movies = 'https://cdn.jsdelivr.net/npm/vega-datasets@1/data/movies.json'
stocks = 'https://cdn.jsdelivr.net/npm/vega-datasets@1/data/stocks.csv'
stateinfo = 'https://cdn.jsdelivr.net/npm/vega-datasets@v1.29.0/data/
↳population_engineers_hurricanes.csv'
usshapes = 'https://cdn.jsdelivr.net/npm/vega-datasets@v1.29.0/data/us-10m.json'
states = alt.topo_feature(usshapes, 'states')
```

```
[31]: original = alt.Chart(movies).mark_circle(size=40, opacity=0.4).encode(
    alt.X('IMDB_Votes:Q',
        scale=alt.Scale(type='log'),
        axis=alt.Axis(tickCount=10),
        title='IMDB Votes (log10 scale)'),
    alt.Y('IMDB_Rating:Q',
        scale=alt.Scale(type='log'),
        axis=alt.Axis(tickCount=10),
        title='IMDB Rating'),
    alt.Color('Major_Genre:N',
        scale=alt.Scale(scheme='tableau20'))
).properties(title='Original',width=300, height=300)
updated = alt.Chart(movies).mark_circle(size=40, opacity=0.4).add_selection(
    alt.selection_interval(bind='scales')
).encode(
    alt.X('IMDB_Votes:Q',
        scale=alt.Scale(type='log'),
        axis=alt.Axis(tickCount=10),
        title='IMDB Votes (log10 scale)'),
    alt.Y('IMDB_Rating:Q',
        scale=alt.Scale(type='log'),
        axis=alt.Axis(tickCount=10),
        title='IMDB Rating'),
    alt.Color('Major_Genre:N',
        scale=alt.Scale(scheme='tableau20'))
).properties(title='Updated',width=300, height=300)
original | updated
```

```
[31]: alt.HConcatChart(...)
```

The basic encodings in this scatter plot map IMDB Votes on a log scale to the x-axis and IMDB Rating to the y-axis. Using color to represent Major Genre helps distinguish categories while maintaining perceptual balance. The addition of panning and zooming directly supports exploration across a wide vote range, where film popularity spans several orders of magnitude, making static viewing insufficient. This interaction allows viewers to zoom into dense clusters or pan toward outliers, enabling finer inspection of genre specific patterns and relationships that might otherwise remain hidden, thus enhancing both interpretability and user engagement.

```
[48]: original = alt.Chart(cars).mark_circle().encode(
    alt.X('Horsepower:Q'),
    alt.Y('Miles_per_Gallon:Q'),
    alt.Color('Origin:N'),
).properties(
    height=300,
    title='Original'
)

# Selections
hover = alt.selection_single(on='mouseover', nearest=True, empty='none')
click = alt.selection_multi(empty='none')
pick = alt.selection_multi(fields=['Origin'], bind='legend') # legend-driven

# Base encodings (no transform_filter here => legend shows all Origins)
points = alt.Chart(cars).mark_circle().encode(
    x=alt.X('Horsepower:Q', title='Horsepower'),
    y=alt.Y('Miles_per_Gallon:Q', title='Miles per Gallon'),
    color=alt.Color('Origin:N', legend=alt.Legend(title='Origin')),
    opacity=alt.condition(pick, alt.value(0.9), alt.value(0.12)),
    tooltip=['Name:N', 'Origin:N', 'Horsepower:Q', 'Miles_per_Gallon:Q']
).add_selection(
    hover, click, pick
)

# Layers that react to hover/click (these CAN be filtered by pick)
selected = points.transform_filter(hover | click).transform_filter(pick)
updated = alt.layer(
    points, # drives the legend (unfiltered domain)
    selected.mark_point(size=120, stroke='black', strokeWidth=1),
    selected.mark_text(dx=6, dy=-8, align='left', stroke='white',
        ↪strokeWidth=2).encode(text='Name:N'),
    selected.mark_text(dx=6, dy=-8, align='left').encode(text='Name:N'),
).properties(
    width=300, height=300, title='Updated'
)

(original | updated).resolve_scale(color='independent')
```

[48]: alt.HConcatChart(...)

The plot encodes Horsepower (x) against Miles per Gallon (y) with color for Origin, which cleanly shows the inverse power efficiency trend while separating regional patterns (Europe/Japan/USA). This suits the interactive need because dense clusters and overlapping categories benefit from point-level inspection and category-focused views. The interactions work together: hover reveals the nearest car with a readable label for quick lookup; click pins multiple cars so we can compare multiple models; and the pick selection bound to the legend helps to focus by Origin without collapsing and the legend's domain—non-picked categories are de-emphasized via opacity while picked ones stay prominent. Together, these selections support targeted comparisons (e.g., Japanese vs. US models at 80–120 HP), and make outliers or notable exemplars easy to find and explain.

[]:

