Week 10: Engines and Interactivity

Spring 2017
Matthew Turk

Broadcasting

go.ischool.illinois.edu/meet2

Warm-Up Activity

1. What is the visualization trying to show?
2. What are its methods?
3. What are the strengths / weaknesses?
4. (Bonus) How was the data collected?

https://vimeo.com/239582792

Evaluating Visualization Systems

- Costs
- Functionality
- Aesthetics

Choices

- Can I get ahold of this software?
- Do I install it, or do I use it on a server?
- What’s the user interface like?
- Is it declarative or is it procedural?

License: Software

- What can you do with the software?
- Can you study the software?
- Who can you share it with?
- Who can you give your derivative works to?

License: Software

- Copyleft: share and share-alike
- Non-copyleft: share, but don’t necessarily need to share-alike
- https://choosealicense.com/

License: Data

- What can you do with the data?
- How do you credit that data?
- Can the data be redistributed, remixed, modified?
- http://opendefinition.org/guide/data/
- https://theodi.org/guides/publishers-guide-open-data-licensing

Accessibility

- Is the software installed locally on your machine?
- Is it hosted at a local or remote instance?
- Who owns the visualizations, and how is access to them controlled?

Interface

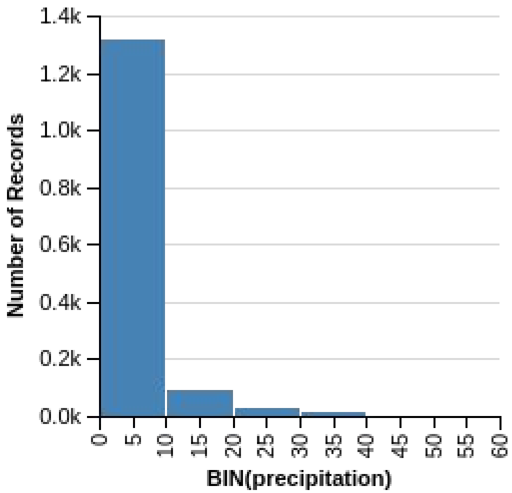
- How do you interact with the software?

Declarative vs Procedural

- Declarative: how do you want the plot to look?
- Procedural: what are the steps to make the plot look that way?

Declarative vs Procedural: Example Declarative

```
Chart(df).mark_bar().encode(
    X('precipitation', bin=True),
    Y('count(*):Q')
)
```



Altair documentation: https://altair-viz.github.io/tutorials/exploring-weather.html

Evaluation: Costs

- Costs
- Functionality
- Aesthetics

Evaluation: Aesthetics

- Costs
- Functionality
- Aesthetics

Visualization Engines

- Matplotlib
- Bokeh
- Plotly
- D3 / Vega / Lyra
- Altair

Visualization Engines: Matplotlib

- License: non-copyleft open source
- Construction of a *plot* is the focus
- Interactivity is possible
- Procedural
- Pros?
- Cons?

Visualization Engines: Bokeh

- License: non-copyleft open source
- Construction of a set of visualizations is focus
- Interactivity is fundamental
- Mix of declarative and procedural
- Pros?
- Cons?

Visualization Engines: Plotly

- License: non-copyleft open source & commercial
- Construction of a set of visualizations or dashboards is focus
- Interactivity is possible
- Web UI promotes interactivity
- Pros?
- Cons?

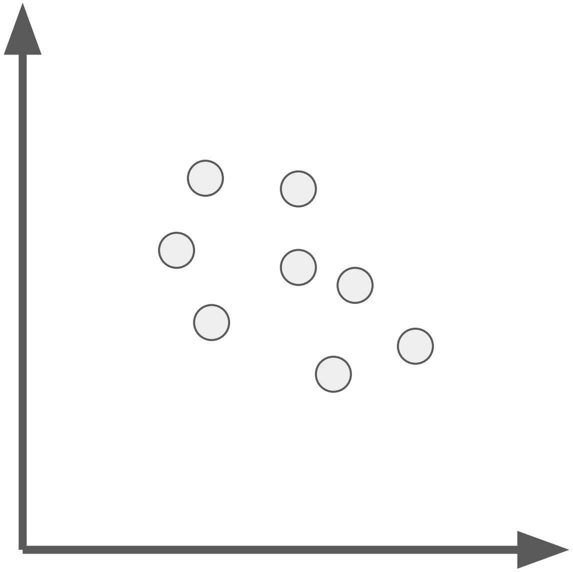
Visualization Engines: D3 / Vega / Lyra

- License: non-copyleft open source
- Construction of a set of visualizations or dashboards is focus
- Interactivity is possible
- Web UI promotes interactivity
- Pros?
- Cons?

Visualization Engines: Altair

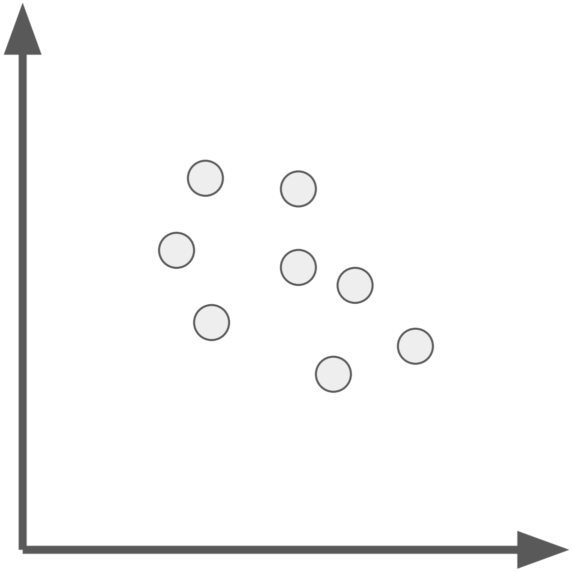
- License: non-copyleft open source
- Construction of charts and visualizations
- Interactivity is coming
- Domain-specific language for reductions and transformations
- Pros?
- Cons?

Basic Principles of Interactivity



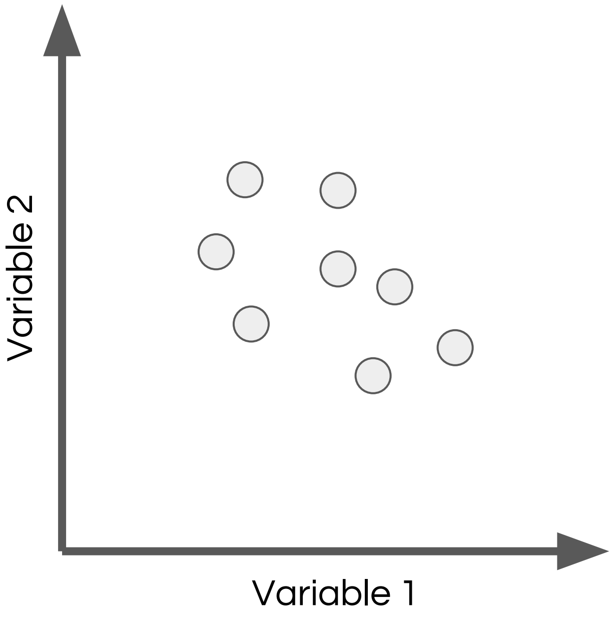
- Point characteristics
- Axis limits/bounds
- Transform/scale of axes

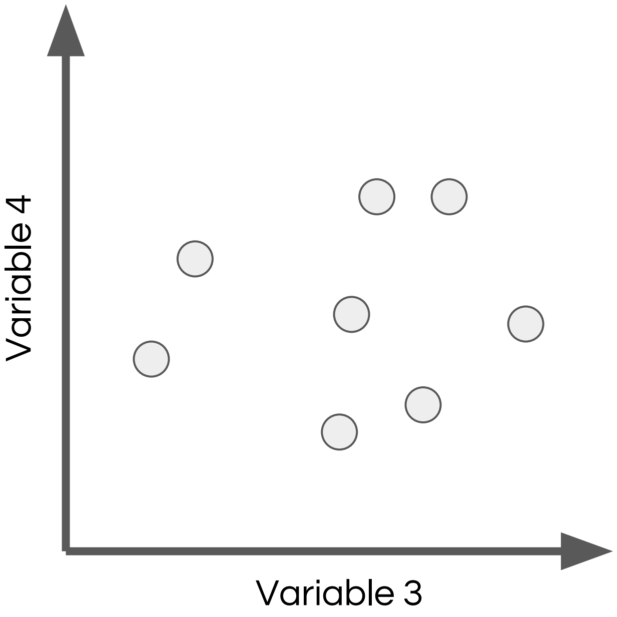
Basic Principles of Interactivity



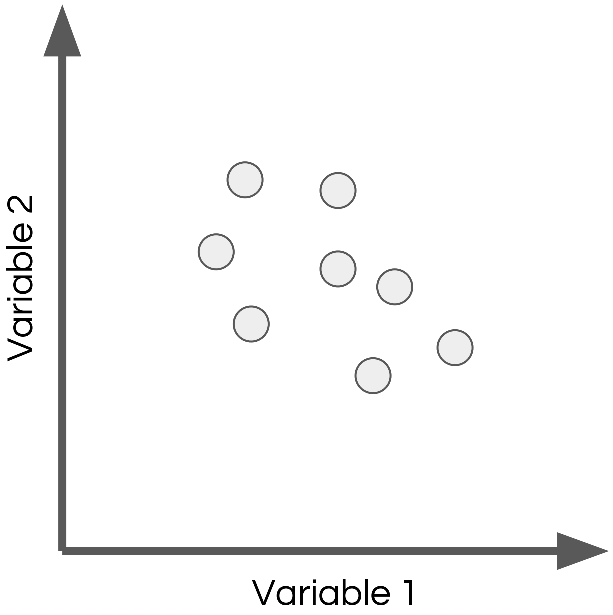
- Click-and-drag
- Rectangle zoom
- Adjustment

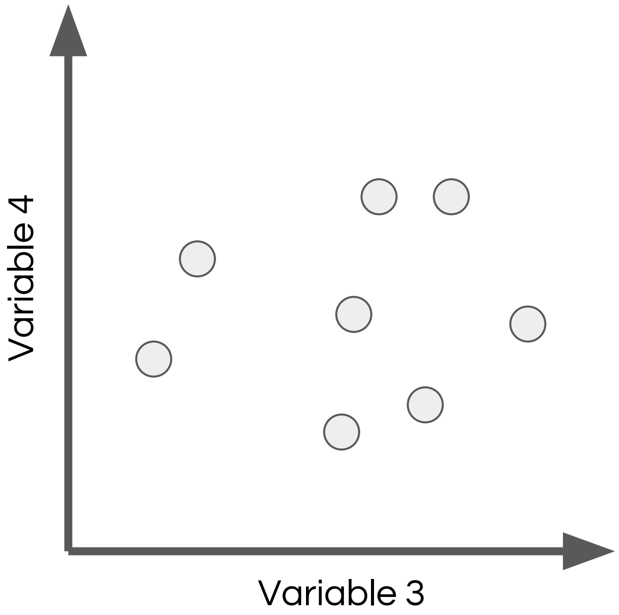
Linking and Brushing





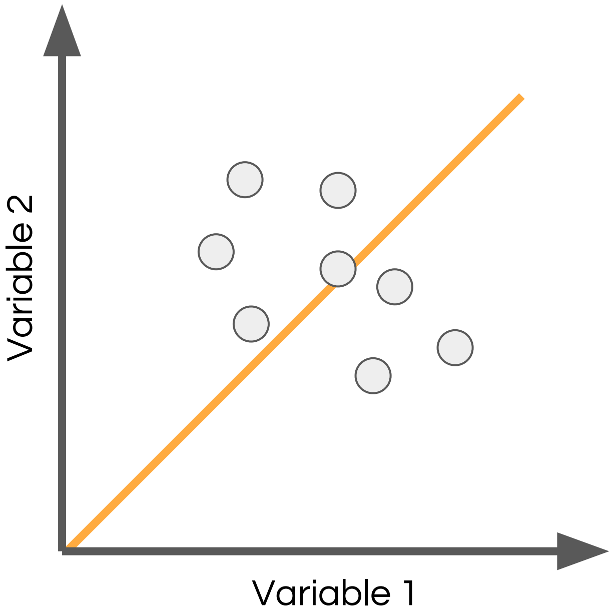
Linking and Brushing

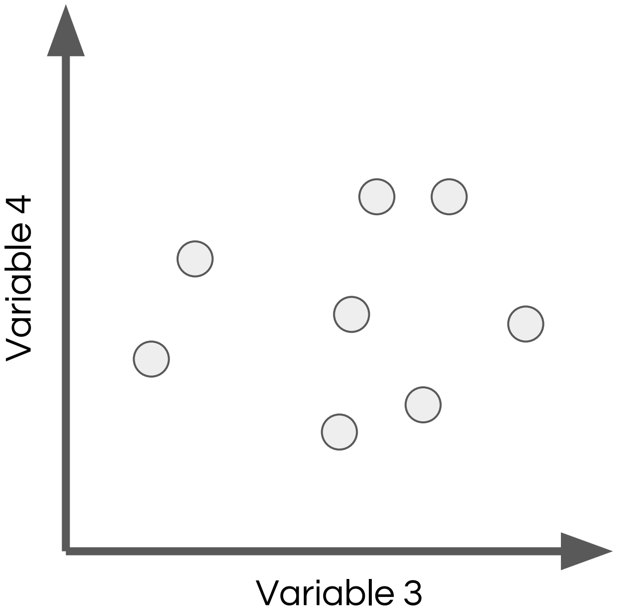




```
filter( variable2 > variable1 )
```

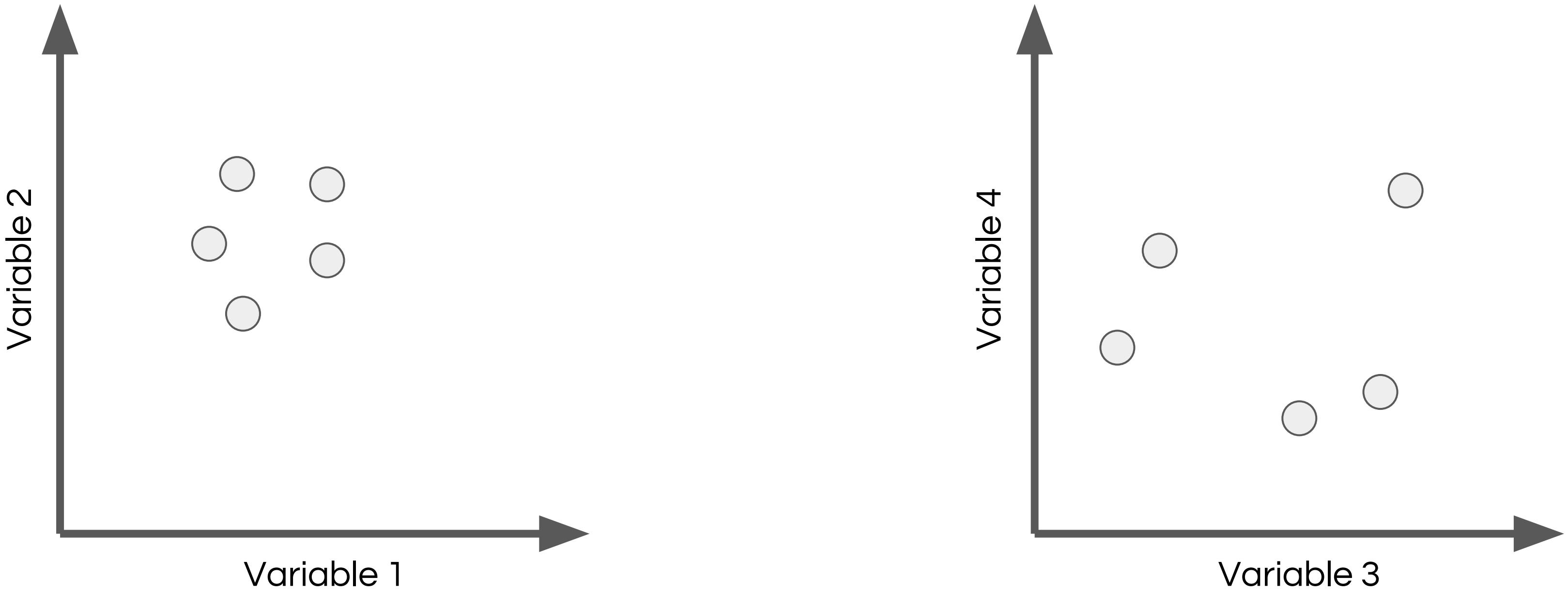
Linking and Brushing





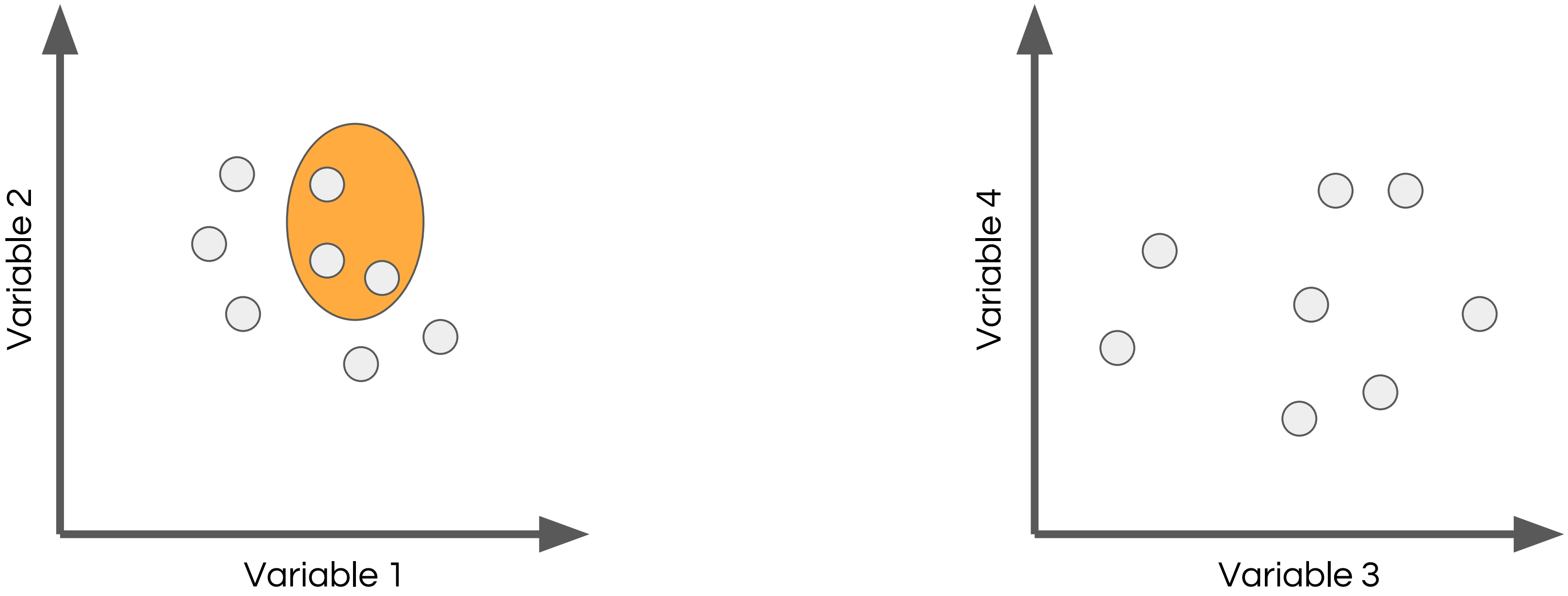
```
filter( variable2 > variable1 )
```

Linking and Brushing

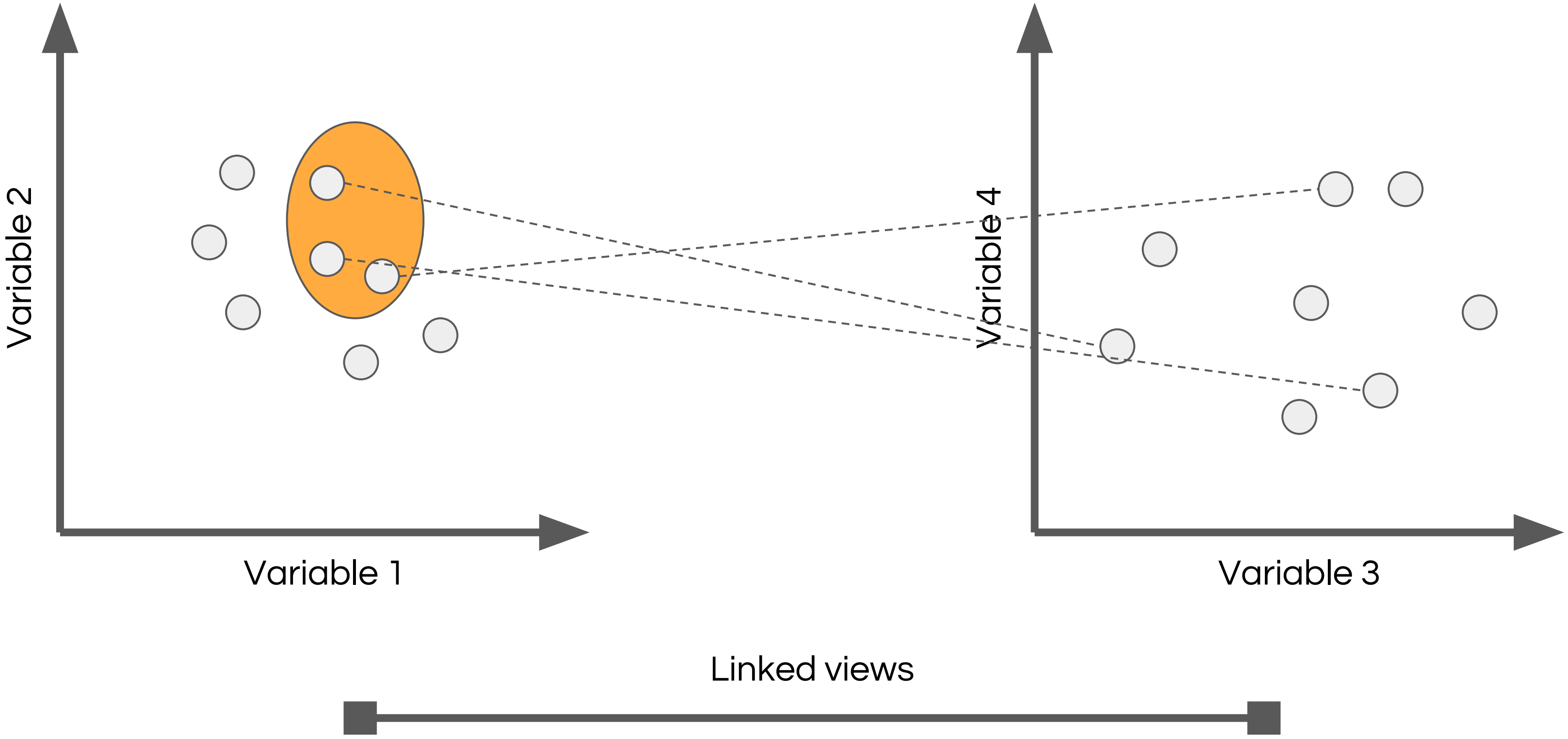


```
filter( variable2 > variable1 )
```

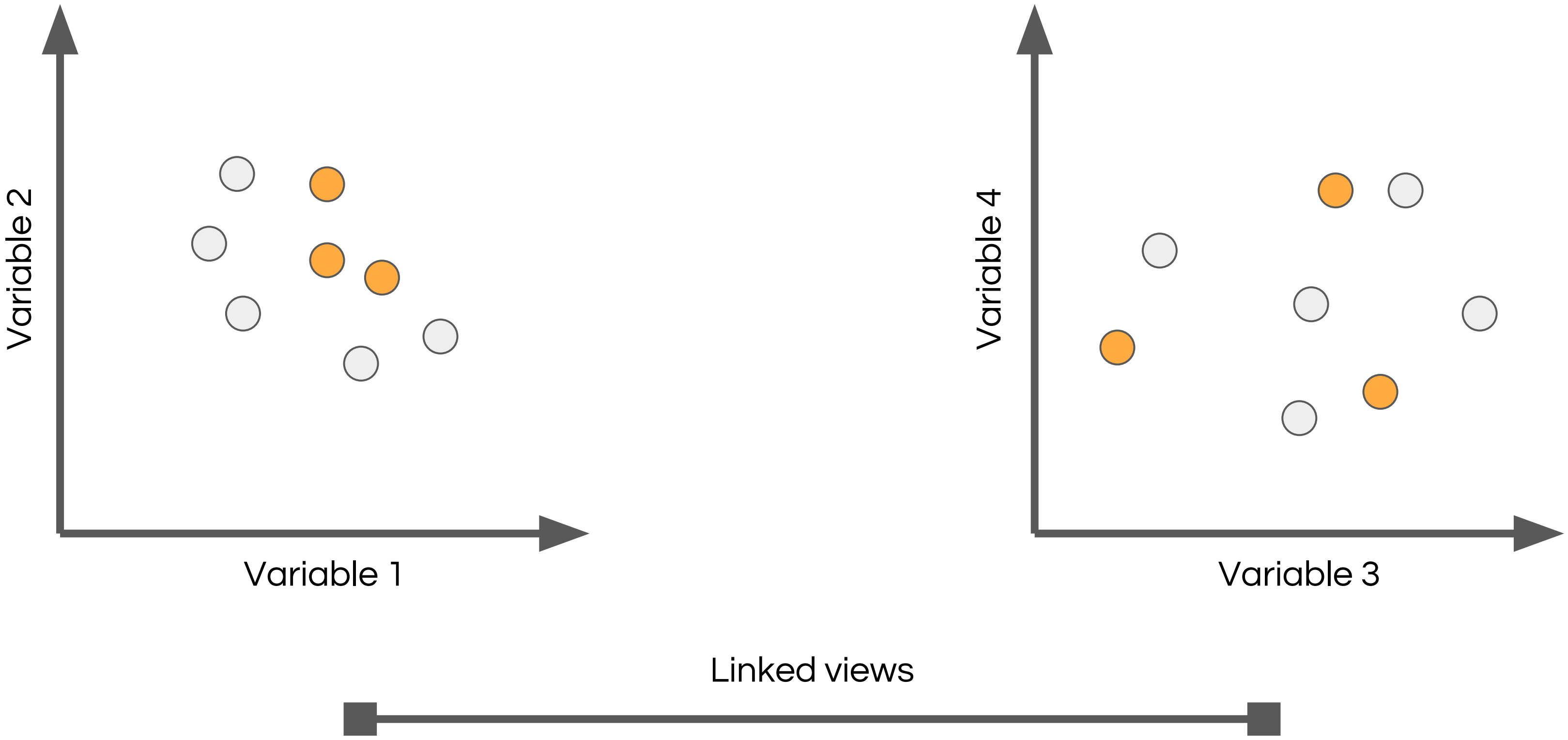
Linking and Brushing



Linking and Brushing



Linking and Brushing



π°
(12 GeV/c)
K°
X =
X₁ π°
X₂ K°, π°
X₃ K°
X₄

Basic Interactivity

- IPython Widgets
  - Declarative setting of parameters
  - Interactive, stylable, groupable
- Output recording
- `ipywidgets` module
- Useful for parameter studies
- Require considerable effort for brushing

Demos today

- Glue (glueviz.org)
- vega-lite
- IPython Widgets for linking
- Loading and examining Transportable Array data

Project

- Groups of 3 or 4 (please choose people you don’t know that well)
- Four visualizations or visualization systems
- Projects due December 8
  a. Report (written)
  b. Full source code (must be runnable)
  c. Example outputs
  d. Oral presentation to class

Project Part 1: Transportable Array

- Use mediaspace.illinois.edu to demonstrate visualization
- Time-varying (i.e., movie)
- Augment with other information you find on your own