Declarative Python Visualization with Altair

Annette Greiner

NERSC Data Day, October 27, 2022
Declarative Versus Imperative

**Imperative**

- draw axes
- draw ticks
- #layer one
- for item in data:
  - set color
  - draw item
- #layer two
- for item in data:
  - ...  

**Declarative**

- chart(data).points.encode(
  - column1 as x position,
  - column2 as y position,
  - column3 as color
  )
Declarative Versus Imperative

**Imperative**
- draw axes
- draw ticks
- #layer one
- for item in data:
  - set color
  - draw item
- #layer two
- for item in data:
  - ...

**Declarative**
- chart(data).points.encode(
  - column1 as x position,
  - column2 as y position,
  - column3 as color
)

**How**

**What**
Declarative Versus Imperative

Declarative visualization lets you think about data and relationships rather than control flow.
What is Altair?

A python library based on Vega-Lite
Vega-Lite is a high-level grammar of graphics based on Vega
Vega is a declarative grammar with a runtime built on D3
D3 is a low-level Javascript library for data visualization

Altair
Vega-Lite
Vega
D3
Python

https://altair-viz.github.io
D3 JavaScript Library

D3js.org

Web outputs!
interaction!
Vega Spec for an Area Graph

```json
{
  "$schema": "https://vega.github.io/schema/vega/v5.json",
  "description": "A basic area chart example.",
  "width": 500,
  "height": 200,
  "padding": 5,
  "signals": [
    {
      "name": "interpolate",
      "value": "monotone",
      "bind": {
        "input": "select",
        "options": [
          "basis",
          "cardinal",
          "catmull-rom",
          "linear",
          "monotone",
          "natural",
          "step",
          "step-after",
          "step-before"
        ]
      }
    }
  ],
  "data": [
    {
      "name": "table",
      "values": [
        {"u": 1,  "v": 28}, {"u": 2,  "v": 55},
        {"u": 3,  "v": 43}, {"u": 4,  "v": 91},
        {"u": 5,  "v": 81}, {"u": 6,  "v": 53},
        {"u": 7,  "v": 19}, {"u": 8,  "v": 87},
        {"u": 9,  "v": 52}, {"u": 10, "v": 48},
        {"u": 11, "v": 24}, {"u": 12, "v": 49},
        {"u": 13, "v": 87}, {"u": 14, "v": 66},
        {"u": 15, "v": 17}, {"u": 16, "v": 27},
        {"u": 17, "v": 68}, {"u": 18, "v": 16},
        {"u": 19, "v": 49}, {"u": 20, "v": 15}
      ]
    }
  ],
  "scales": [
    {
      "name": "xscale",
      "type": "linear",
      "range": "width",
      "zero": false,
      "domain": {"data": "table", "field": "u"}
    },
    {
      "name": "yscale",
      "type": "linear",
      "range": "height",
      "nice": true,
      "zero": true,
      "domain": {"data": "table", "field": "v"}
    }
  ],
  "axes": [
    {"orient": "bottom", "scale": "xscale", "tickCount": 20},
    {"orient": "left", "scale": "yscale"}
  ],
  "marks": [
    {
      "type": "area",
      "from": {"data": "table"},
      "encode": {
        "enter": {
          "x": {"scale": "xscale", "field": "u"},
          "y": {"scale": "yscale", "field": "v"},
          "y2": {"scale": "yscale", "value": 0},
          "fill": {"value": "steelblue"}
        },
        "update": {
          "interpolate": {"signal": "interpolate"},
          "fillOpacity": {"value": 1}
        },
        "hover": {
          "fillOpacity": {"value": 0.5}
        }
      }
    }
  ]
}
```
Vega-Lite Spec for an Area Graph

```json
{
  "$schema": "https://vega.github.io/schema/vega-lite/v5.json",
  "width": 300,
  "height": 200,
  "data": {"url": "data/unemployment-across-industries.json"},
  "mark": "area",
  "encoding": {
    "x": {
      "timeUnit": "yearmonth", "field": "date",
      "axis": {"format": "%Y"}
    },
    "y": {
      "aggregate": "sum", "field": "count",
      "title": "count"
    }
  }
}
```
An Area Graph in Altair

From vega_datasets import data
source = data.stocks()

alt.Chart(source).transform_filter(
    'datum.symbol==="GOOG"'
).mark_area(
    line={'color':'darkblue'}).encode(
    alt.X('date:T'),
    alt.Y('price:Q')
)
Gallery

Simple Charts

- Simple Bar Chart
- Simple Heatmap
- Simple Histogram
- Simple Line Chart
- Simple Scatter Plot with Toothpicks
- Simple Stacked Area Chart
- Simple Stacked Bar Chart

Bar Charts

- Bar Chart with Highlighted Bar
- Bar Chart with Labels
- Bar Chart with Line at Mean
- Bar Chart with Line on Dual Axes
- Bar Chart with Negative Values
- Bar Chart with Rolling Mean
- Bar Chart with rounded edges
- Bar and Tick Chart
- Calculating Percentage of Total
- Compact Trellis Grid of Bar Charts
- Diverging Stacked Bar Chart
- Grouped Bar Chart
- Line Chart with Layered Aggregates
- Line Chart with Logarithmic Scale
- Line Chart with Percent area
- Line Chart with Sequence Generator
- Line Chart with Varying Size
- Line Chart with datum
- Line Chart with datum for color

Line Charts

- Bump Chart
- Filled Step Chart
- Line Chart with Confidence Interval Band
- Line Chart with Cumulative Sum
- Streamgraph
- Trellis Area Chart
- Trellis Area Sort Chart

Area Charts

- Area Chart with Gradient
- Cumulative Count Chart
- Faceted Density Estimates
- Horizon Graph
- Interval Selection Example
- Layered Area Chart
- Normalized Stacked Area Chart
- Stacked Density Estimates

Circular Plots

- Donut Chart
- Pacman Chart
- Pie Chart
- Pie Chart with Labels
- Radial Chart
What Can Altair Take as Input?

- Dataframes
- CSV
- TSV
- JSON
- URLs
- Geo data (e.g., GeoDataFrame, GeoJSON)
What Can Altair Output?

JSON
HTML
PDF
SVG
PNG
Create a Conda env to use as a Jupyter kernel

```
agreiner@cori06> module load python
agreiner@cori06> conda create -n altair python=3.9 ipykernel altair
agreiner@cori06> conda activate altair
(altair) agreiner@cori06> python -m ipykernel install --user --name altair
--display-name Altair
Installed kernelspec altair in
/global/u1/a/agreiner/.local/share/jupyter/kernels/altair
```

Grab some handy datasets to play with

```
(altair) agreiner@cori06> conda install -c conda-forge vega_datasets
https://github.com/altair-viz/vega_datasets
```
Altair in Jupyter at NERSC
Thank You