Search

Publish...

Xinyu Jiang

You Edited Feb 25 Fork of Observable Plot

Observable Plot

Observable Plot is a free, open-source JavaScript library to help you quickly visualize tabular data. It has a concise and (hopefully) memorable API to foster fluency — and plenty of examples to learn from and copy-paste.

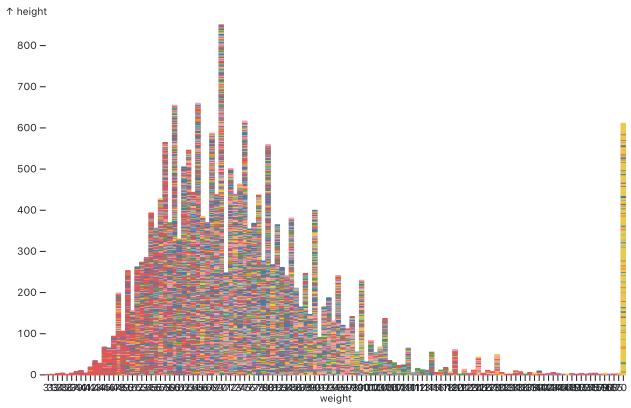
In the spirit of *show don't tell*, below is a scatterplot of the height and weight of Olympic athletes (sourced from Matt Riggott), constructed using a dot mark. We assign columns of data (such as *weight*) to visual properties (such as the dot's *x*), and Plot infers the rest. You can configure much more, if needed, but Plot's goal is to help you get a meaningful visualization quickly.

```
athletes = ▸ Array(11538) [Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object,
```

```
athletes = FileAttachment("athletes.csv").csv({typed: true})
```

aquatics archery athletics badminton basketball boxing canoe cycling equestrian fencing football golf gymnastics handball hockey judo modern pentathlon rowing rugby sevens sailing shooting table tennis taekwondo tennis triathlon volleyball weightlifting wrestling

```
dotplot.legend("color")
```



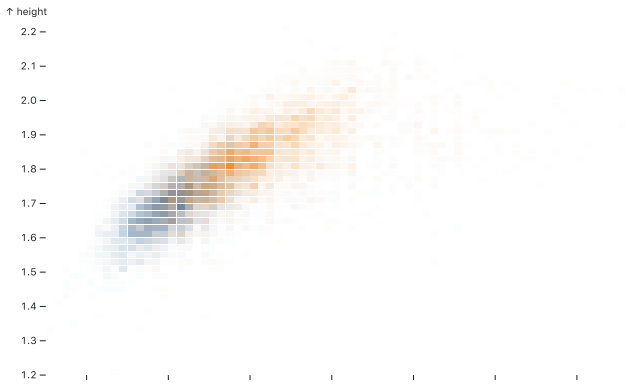

```
// new_athletes=[]
// for(int i=0;i<athletes.length;i++){
//   if(athletes[i]["gold"]>0){
//     new_athletes.append(athletes[i])
//   }
// }
```

```
dotplot = Plot.barY(athletes, {x: "weight", y: "height", stroke: "sport"}).plot()
```

id	name	nationality	sex	date_of_birth	height	weight	sport	gold	silver	bronze
702,606,719	Abdoul Khadre Mbaye Niane	SEN	male	1988-08-20	1.9	90	aquatics	0	0	0
259,675,127	Abdoulkarim Fawziya	CMR	female	1989-03-01	1.8	67	volleyball	0	0	0
469,953,606	Abdoullah Bamoussa	ITA	male	1986-06-08			athletics	0	0	0
962,468,808	Abdoulrazak Issoufou Alfaga	NIG	male	1994-12-26	2.07	98	taekwondo	0	1	0
958,967,643	Abdul Khalili	SWE	male	1992-06-07	1.81	71	football	0	0	0
325,809,293	Abdul Omar	GHA	male	1993-10-03			boxing	0	0	0
152,408,417	Abdul Wahab Zahiri	AFG	male	1992-05-27	1.75	68	athletics	0	0	0
262,868,423	Abdulaziz Alshatti	IOA	male	1990-10-30			fencing	0	0	0
101,781,750	Abdulkadir Abdullayev	AZE	male	1988-07-17	1.88		boxing	0	0	0
5,763,609	Abdullah Abkar Mohammed	KSA	male	1997-01-01	1.72	73	athletics	0	0	0
969,824,503	Abdullah Alrashidi	IOA	male	1963-08-21	1.83	84	shooting	0	0	1

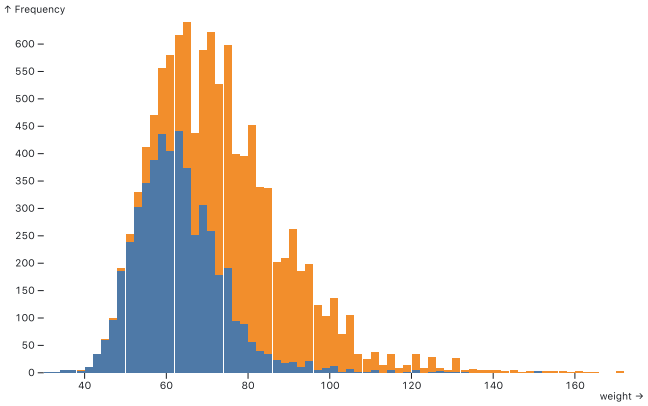
```
Inputs.table(athletes)
```

This scatterplot suffers from overplotting: many dots are drawn in the same spot, so it's hard to perceive density. We can fix this by binning athletes of similar height and weight (and sex), then using opacity to encode the number of athletes in the bin.



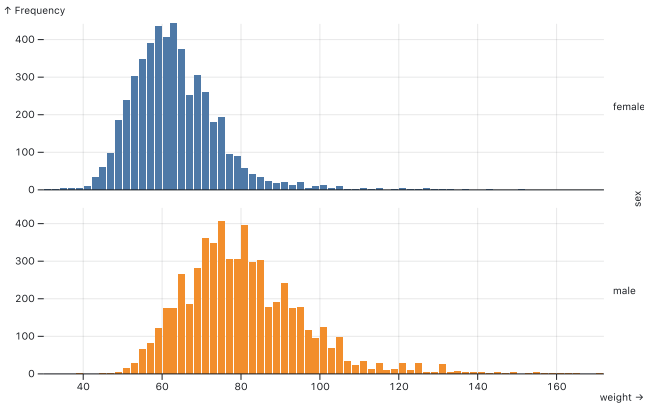

```
Plot.rect(athletes, Plot.bin({fillOpacity: "count"}, {x: "weight", y: "height", fill: "sex"})).plot()
```

A simpler take on this data is to focus on one dimension: weight. We can use the bin transform again to make a histogram with weight on the *x*-axis and frequency on the *y*-axis. This plot uses a rect mark and an implicit stack transform.




```
Plot.rectY(athletes, Plot.binX({y: "count"}, {x: "weight", fill: "sex"})).plot()
```

Or if we'd prefer to show the two distributions separately as small multiples, we can facet the data along *y* (keeping the *fill* encoding for consistency, and adding grid lines and a rule at *y* = 0 to improve readability).




```
Plot.plot({
  grid: true,
  facet: {
    data: athletes,
    y: "sex"
  },
  marks: [
    Plot.rectY(athletes, Plot.binX({y: "count"}, {x: "weight", fill: "sex"})),
    Plot.ruleY([0])
  ]
})
```

Plot employs a layered grammar of graphics inspired by Vega-Lite, ggplot2, Wilkinson's *Grammar of Graphics*, and Bertin's *Semiology of Graphics*. Plot rejects a chart typology in favor of marks, scales, and transforms. Plot can be readily extended in JavaScript, whether to define a channel, a transform, or even a custom mark. And Plot is compatible with Observable dataflow: use inputs to control charts, and views to read chart selections. (This last part is coming soon!)

We created Plot to better support exploratory data analysis in reactive, JavaScript notebooks like Observable. We continue to support D3 for bespoke explanatory visualization and recommend Vega-Lite for imperative, polyglot environments such as Jupyter. Plot lets you see your ideas quickly, supports interaction with minimal fuss, is flexible with respect to data, and can be readily extended by the community. We believe people will be more successful finding and sharing insight if there's less wrestling with the intricacies of programming and more "using vision to think."

To learn more, start with Plot: Marks or dive into the docs below.

Or for a quick overview, see the interactive Plot Cheatsheets or download the PDF.

Concepts

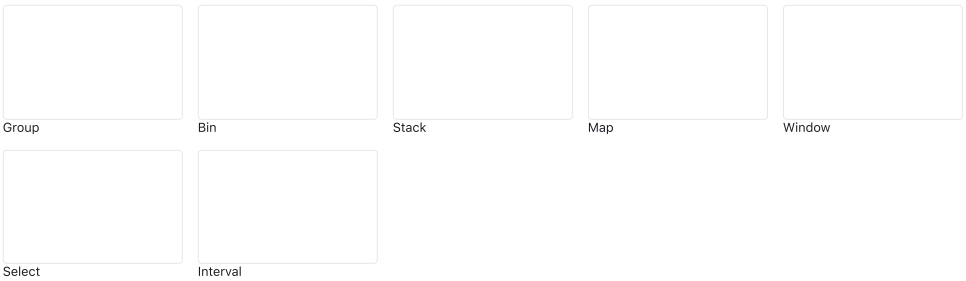
These core concepts provide Plot's foundation.

Marks

Plot provides built-in graphical marks for common charts.

Transforms

Plot provides built-in transforms for deriving data.



And more!

Got the basics? Here are even more resources:

- API Reference
- Cheatsheets
- Exploration: Penguins
- Exploration: Weather
- … and more coming soon!

Observable Plot is released under the ISC license and depends on D3. It works great on Observable but does not depend on Observable; use it anywhere! To contribute to Plot's development or to report an issue, please see our repository. For recent changes, please see our release notes.

Appendix

```
function previews(notebooks, options) {
  return htl.html`<div style="display: grid; grid-gap: .875rem; grid-template-columns: repeat(auto-fill,
minmax(160px, 5fr));">${notebooks.map(notebook => preview(notebook, options))}</div>`;
}
```

```
function preview({path, title, author, thumbnail}, {target = "_blank"} = {}) {
  return htl.html`<a href=${`/${path}`} target=${target} title="${title}${author ? `by ${author}`: ""}"
style="display: inline-flex; flex-direction: column; align-items: start; font: 400 .75rem var(--sans-serif);
color: #1b1e23; width: 100%;" onmouseover=${event => event.currentTarget.firstElementChild.style.borderColor =
"#1b1e23"} onmouseout=${event => event.currentTarget.firstElementChild.style.borderColor = "#e8e8e8"}>
  <div style="border: solid 1px #e8e8e8; border-radius: 4px; box-sizing: border-box; width: 100%; padding-top:
62.5%; background-size: cover; background-image:
url(https://static.observableusercontent.com/thumbnail/${encodeURI(thumbnail)}.jpg);"></div>
  <div style="width: 100%; white-space: nowrap; text-overflow: ellipsis; overflow: hidden;">${title}</div>
</a>`;
}
```