Transforms

Augment your data for plotting

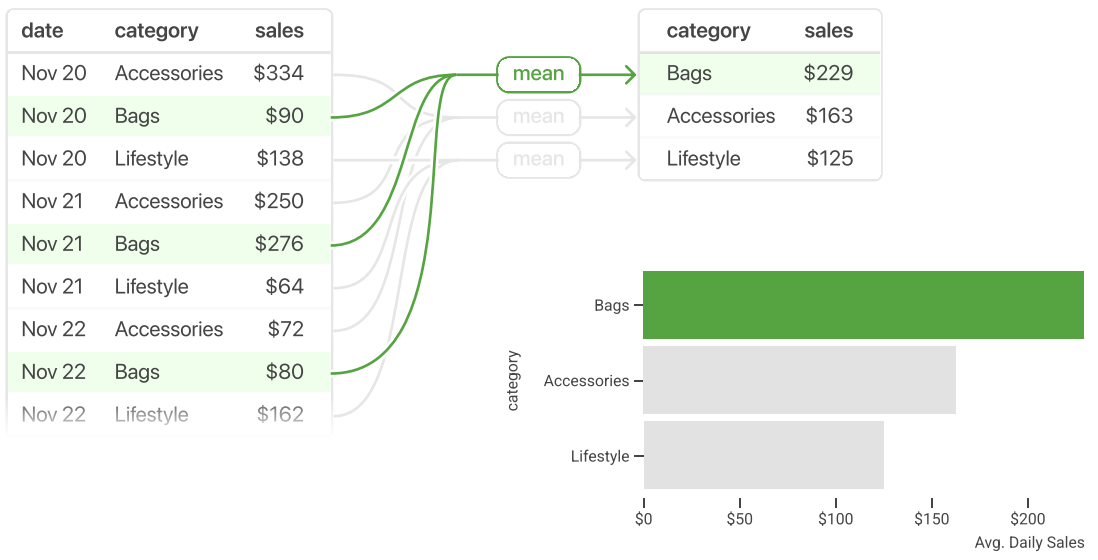
Observable Plot
version 0.4.0

Group to categorize data

`Plot.group`, `Plot.groupX`, `Plot.groupY`, `Plot.groupZ`

Compute the mean sales for each category:

```
Plot.groupY({ x: "mean" }, { x: "sales", y: "category" })
```

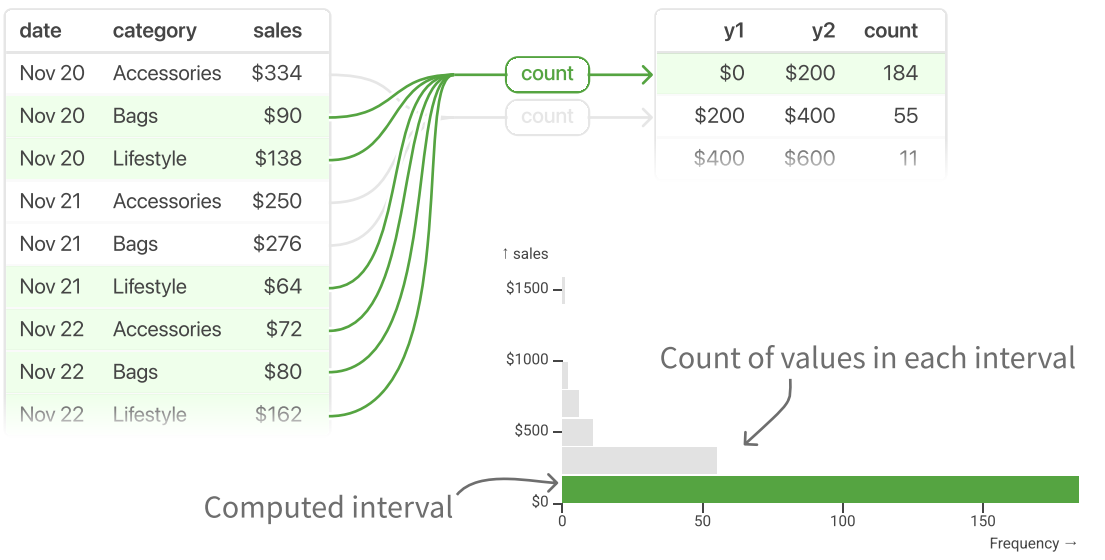


Bin to count data

`Plot.bin`, `Plot.binX`, `Plot.binY`

Count observations in each interval, created based on sales:

```
Plot.binY({ x: "count" }, { y: "sales" })
```

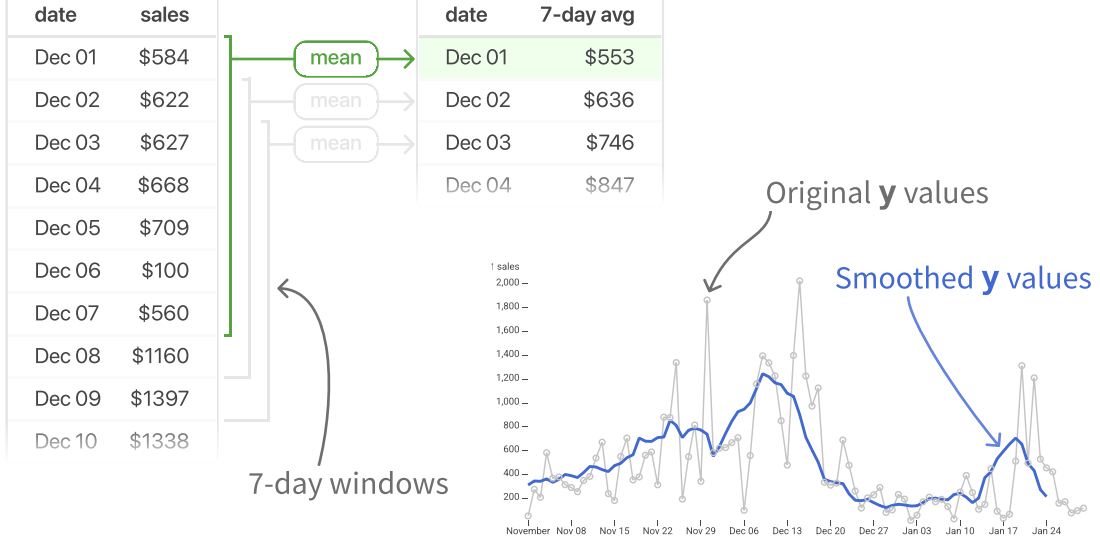


Window to smooth values

`Plot.window`, `Plot.windowX`, `Plot.windowY`

Compute the 7-day moving average of sales:

```
Plot.windowY({ reduce: "mean", k: 7 }, { x: "date", y: "sales" })
```

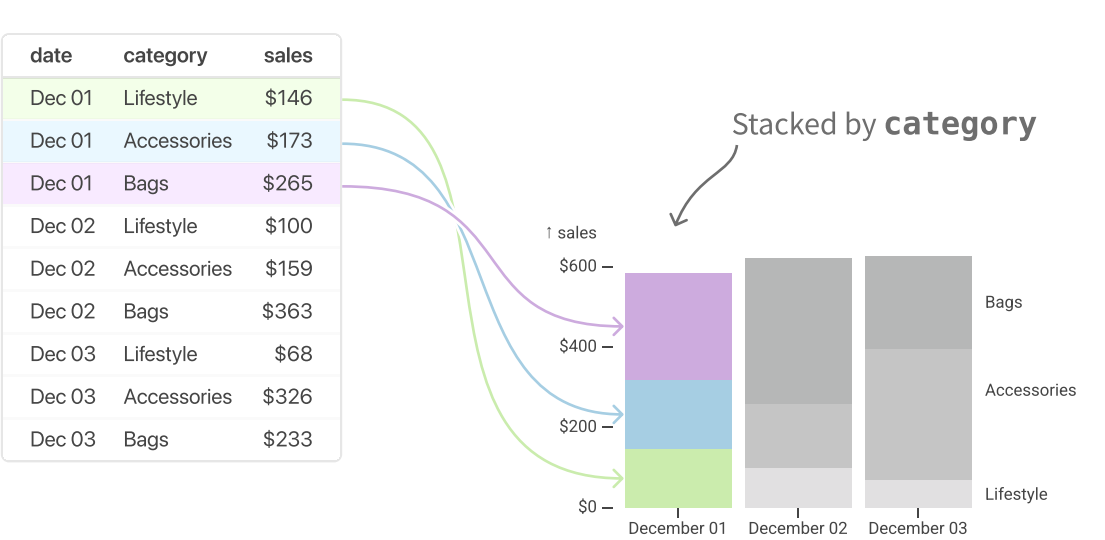


Stack to layer values

`Plot.stackX`, `Plot.stackX1`, `Plot.stackX2`, `Plot.stackY`, `Plot.stackY1`, `Plot.stackY2`, `Plot.barX`, `Plot.barY`, `Plot.areaX`, `Plot.areaY`

Stack a bar chart of sales by category:

```
Plot.barY(data, { x: "date", y: "sales", fill: "category" })
```

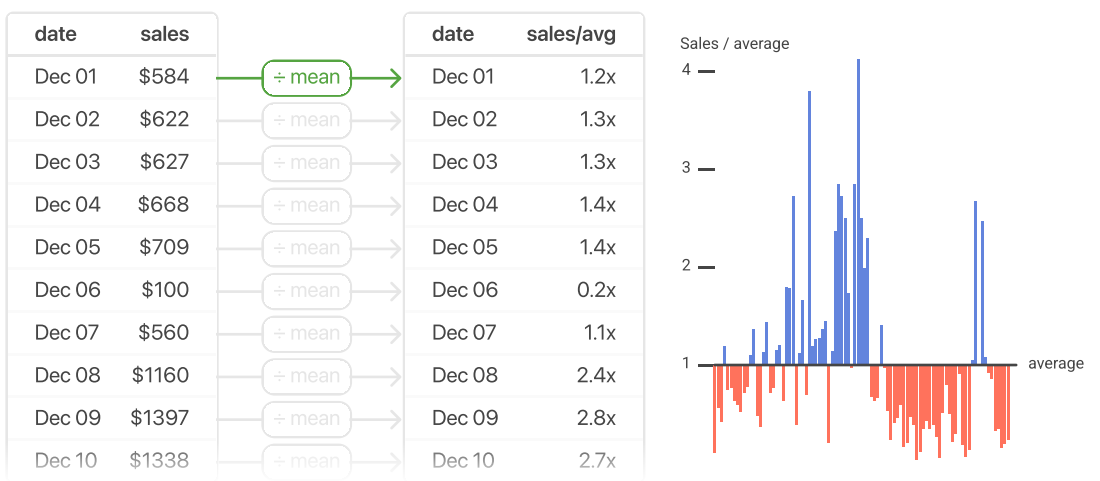


Normalize to see deviations

`Plot.normalize`, `Plot.normalizeX`, `Plot.normalizeY`

Divide each sale by the mean of all sales:

```
Plot.normalizeY({ basis: "mean", x: "date", y: "sales" })
```



Select to pick specific values

`Plot.selectFirst`, `Plot.selectLast`, `Plot.selectMaxX`, `Plot.selectMaxY`, `Plot.selectMinX`, `Plot.selectMinY`

Select the observation with the highest sales:

```
Plot.selectMaxY({ x: "date", y: "sales" })
```