

Customizing Visualizations



Douglas Starnes

SOFTWARE ENGINEER / CONFERENCE SPEAKER / TECH AUTHOR

@poweredbyaltnet <http://douglasstarnes.com>



Representing Color



Name

- Corresponds to the list of CSS colors
- 'blue'
- 'mediumslateblue'
- 'blanchedalmond'

Abbreviations

- Blue, green, red, cyan, magenta, yellow
black, white
- First letter of the name
- Black is 'k' to avoid conflict with blue



Representing Color



Hexadecimal codes

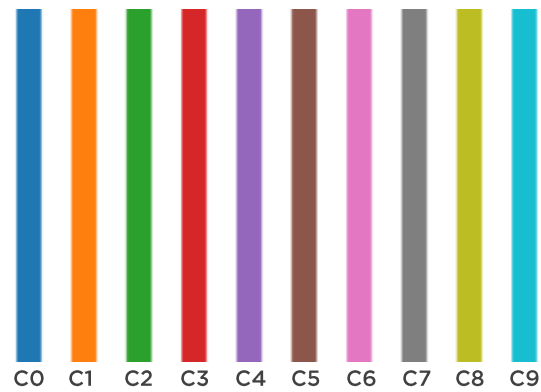
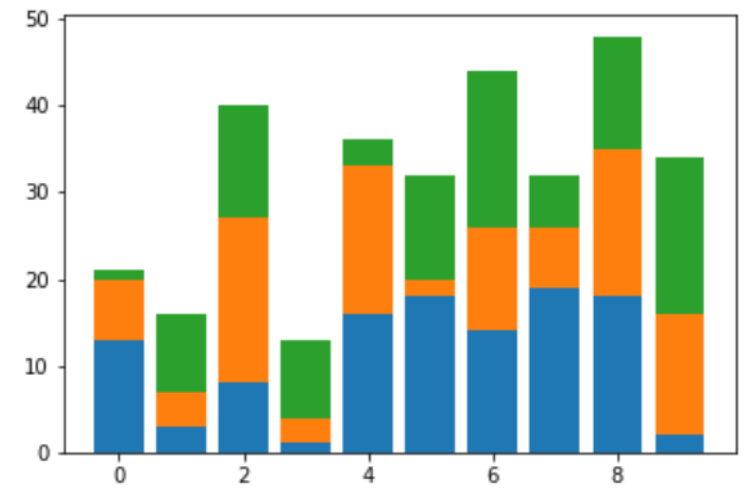
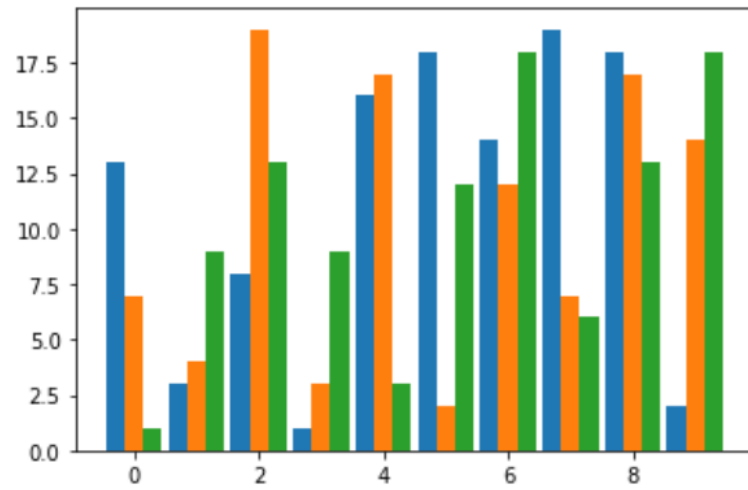
Most specific

Six digit string preceded by a hashtag or
pound sign

- i.e. '#a1b4c5'



Color Cycle



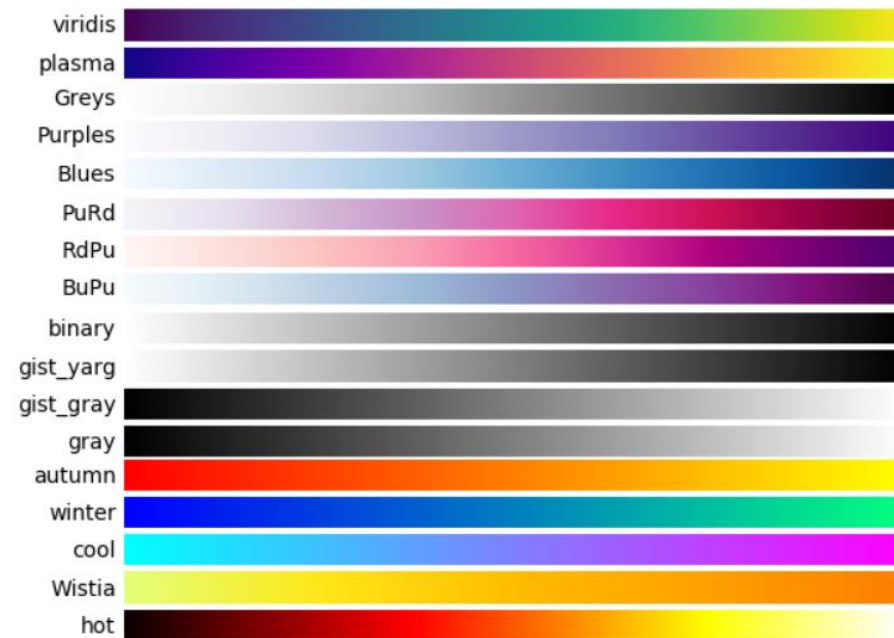
Color Maps



Spectrum of color

Appropriate for continuous ranges

Gradient transition between more than one color or shades of the same color



Line Styles



Lines created with the `plot()` function

Use the `linestyle=` keyword argument

Line styles have names and abbreviations

- Name: 'solid', abbreviation: '-'
- Name: 'dashed', abbreviation: '--'
- Name: 'dashdot', abbreviation: '-.'
- Name: 'dotted', abbreviation: ':'

The 'None' style

- Removes the line completely
- Abbreviation is single space or empty string
- We'll see this used later in the module



Custom Line Styles



Tuple with two elements

- Ex: (0, (1, 3, 2, 3))
- First element is an offset
- Second element is the 'on-off sequence'

On-off sequence

- Tuple of even number of non-negative values
- ({draw}, {space}, {draw}, {space}, ...)
- Measured in points
- Sequence repeats for the length of the line

Offset

- Number of points to offset before sequence begins



More Styles



Dashes

The `dash_capstyle=` keyword argument

- Butt (default)
- Round
- Projecting

The `dash_joinstyle=` keyword argument

- Round (default)
- Miter
- Bevel

Line Width



The linewidth= keyword argument

- Non-negative floating point
- Larger the number, the wider the line
- And that's it!

Aliases



Aliased keyword arguments and functions expect the same values

More than just styles, also applies to the `color=` keyword argument

Markers



The marker= keyword argument

- Lowercase 'o' / circle
- Period '.' / point
- Lowercase 's' / square
- Lowercase 'v' / triangle down
- Caret '^' / triangle up
- Less than '<' / triangle left
- Greater than '>' / triangle right
- Asterisk '*' / five sided star
- Plus '+' / plus with thin lines
- Capital 'P' / plus with thick lines
- Lowercase 'p' / pentagon
- Eight '8' / octagon
- And more!

Custom Markers



A tuple with three elements

- Number of sides
 - 0 / straight sides
 - 1 / star like
 - 2 / lines, no fill
- Angle of rotation

Upside down star

- (5, 1, 180)

Plus sign (thin)

- (4, 2, 0)

Triangle pointing southeast

- (3, 0, 330)



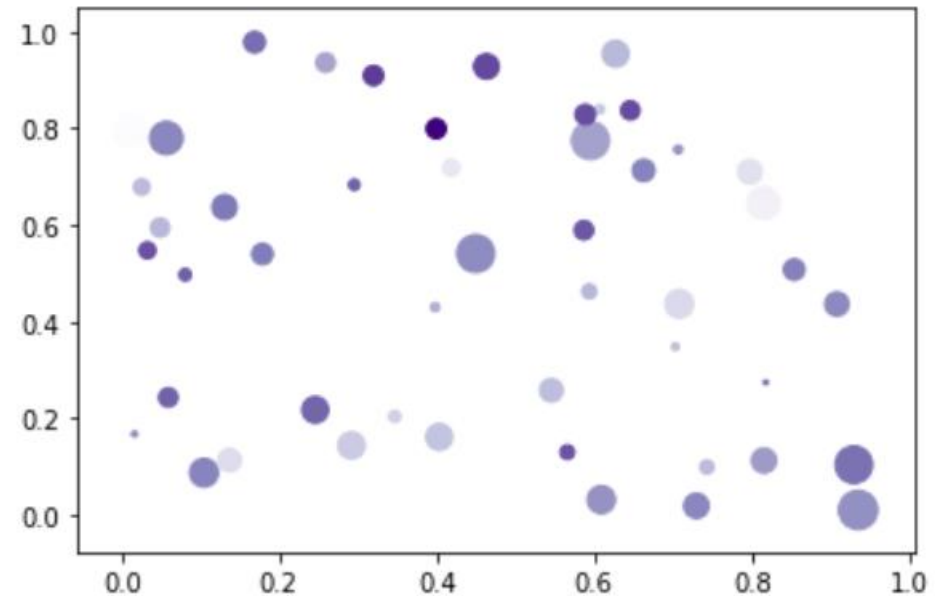
Marker Size



Just like with color, size can be used to represent a third data dimension

The `s=` keyword argument

- Iterable with one value per data point

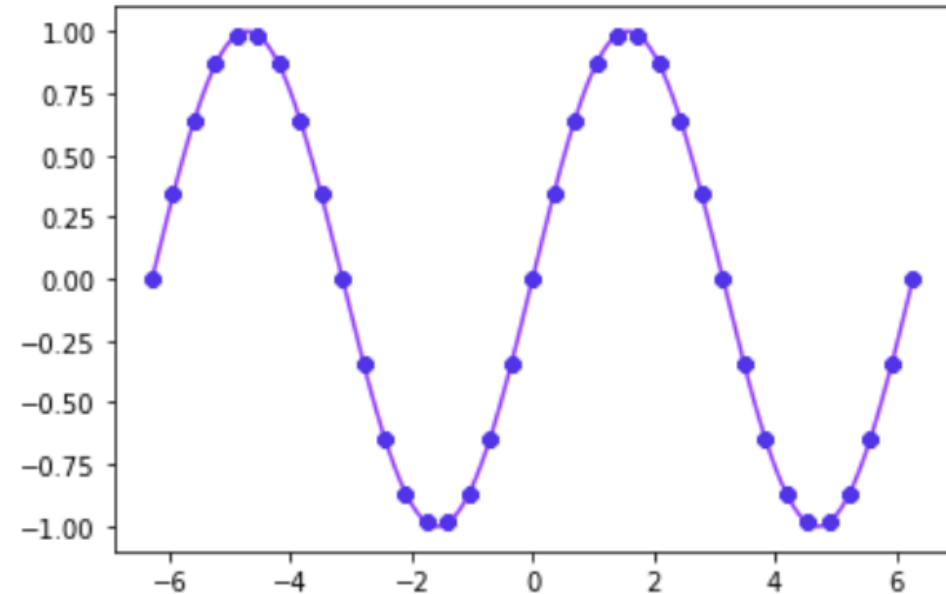


Markers and Line Charts

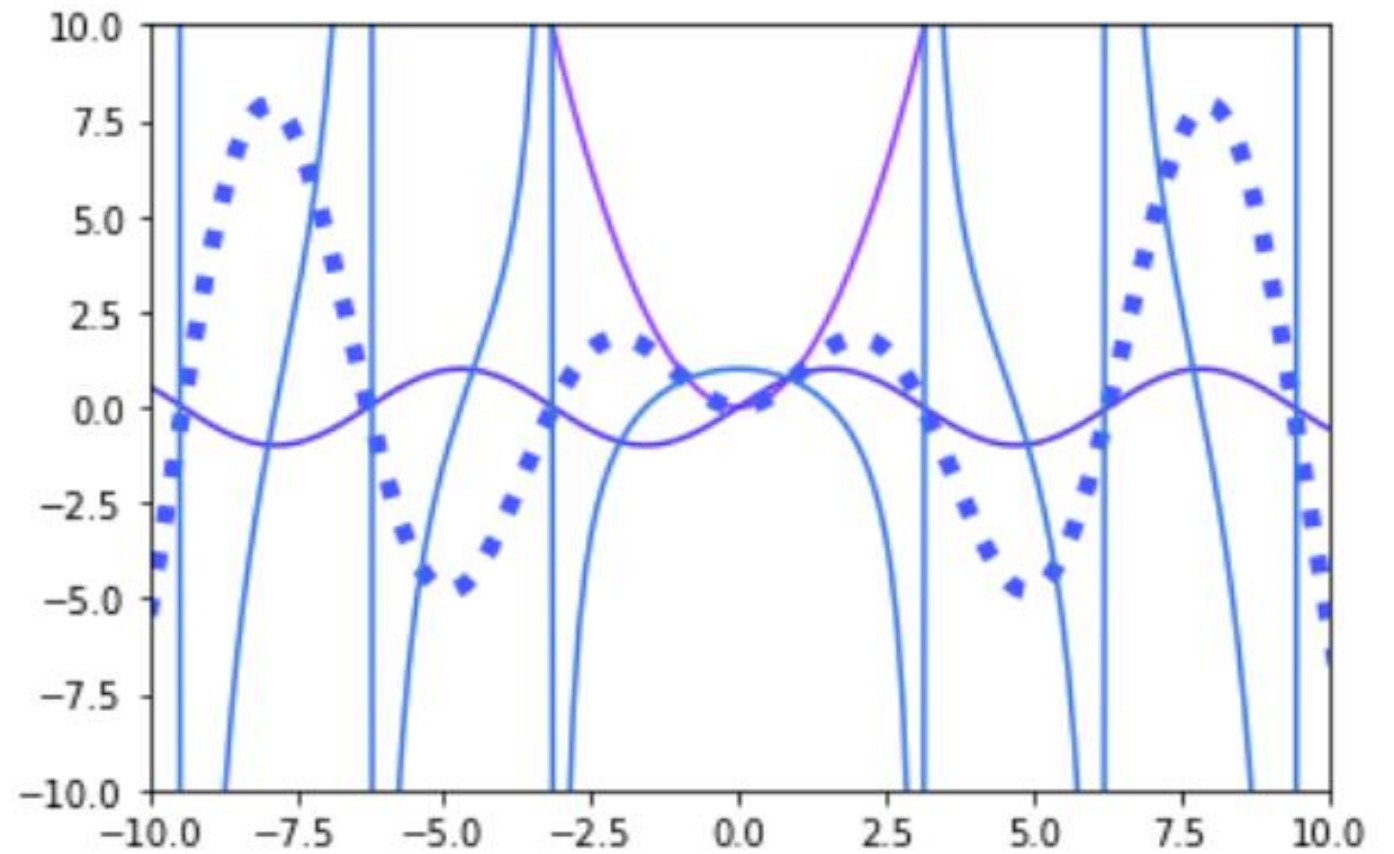


Just add the `marker=` keyword argument

- Uses the same symbols



Legends



Legends



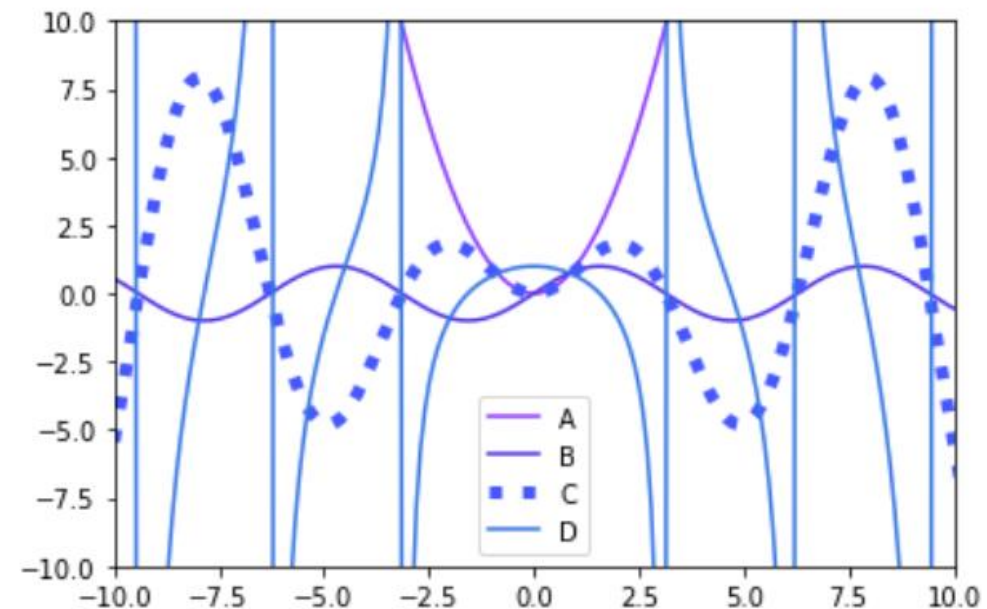
We need a reference to label the lines

A legend is a list of samples of styles and associated labels

Call the `legend()` function

Legends are built using labeled lines

- The `label=` keyword argument



Legends: Alternative Method



Save the handle to each line

Pass the lines and labels to the legend() function

Doesn't require the label= keyword argument

Specify the order of the lines and label

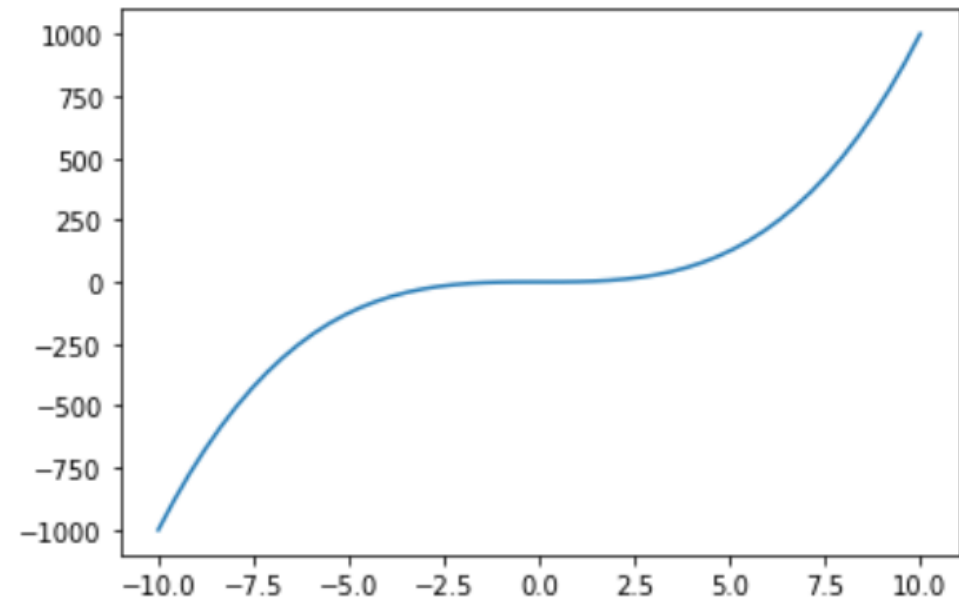
Legend Location



Will be inferred by default to a position that does not obscure the visualization

The `loc=` keyword argument

- Explicit legend location
- Can be faster for complex visualizations



Legend Appearance



The ncol= keyword arguments

- Default is 1

Background color

Font options

Border

Spacing