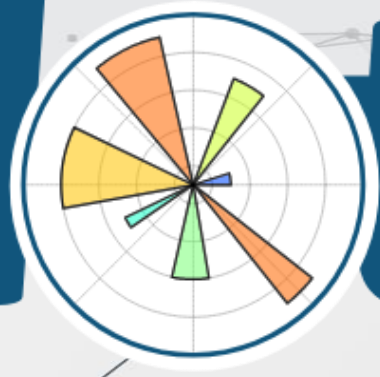


matplotlib





Line plot

Scatter plot

Histogram

Bar plot

Pie plot

What is Matplotlib?

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

Create publication quality plots.

Make interactive figures that can zoom, pan, update.

Customize visual style and layout.

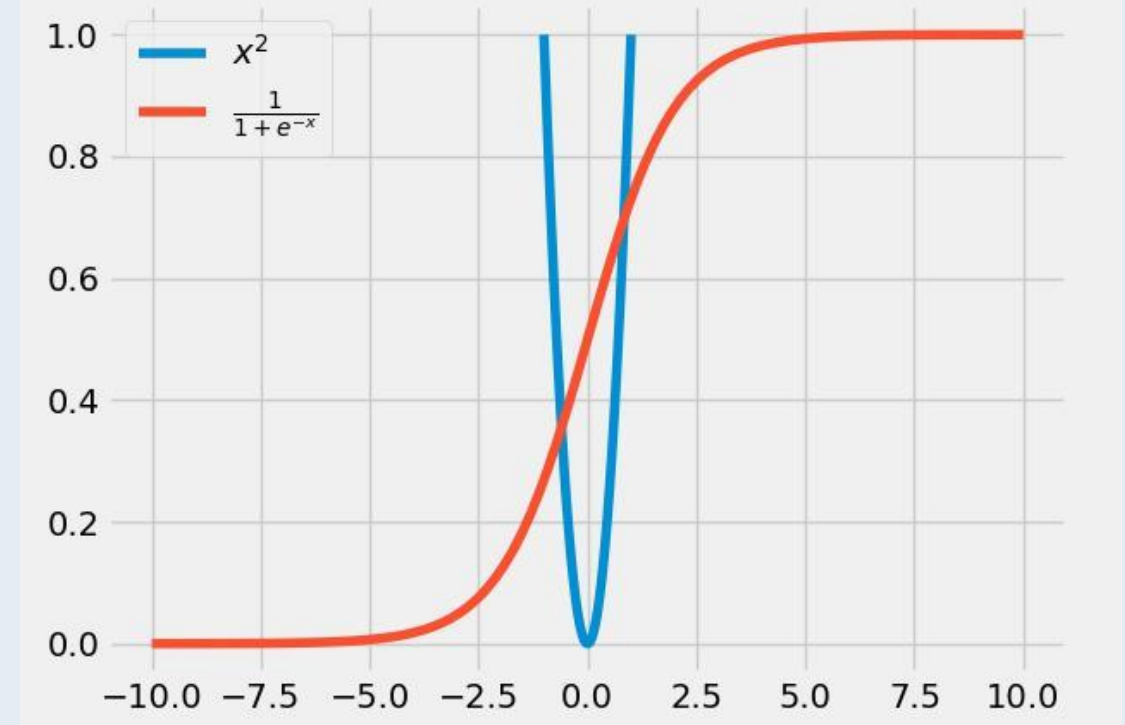
Export to many file formats.

Embed in JupyterLab and Graphical User Interfaces.

Use a rich array of third-party packages built on Matplotlib.

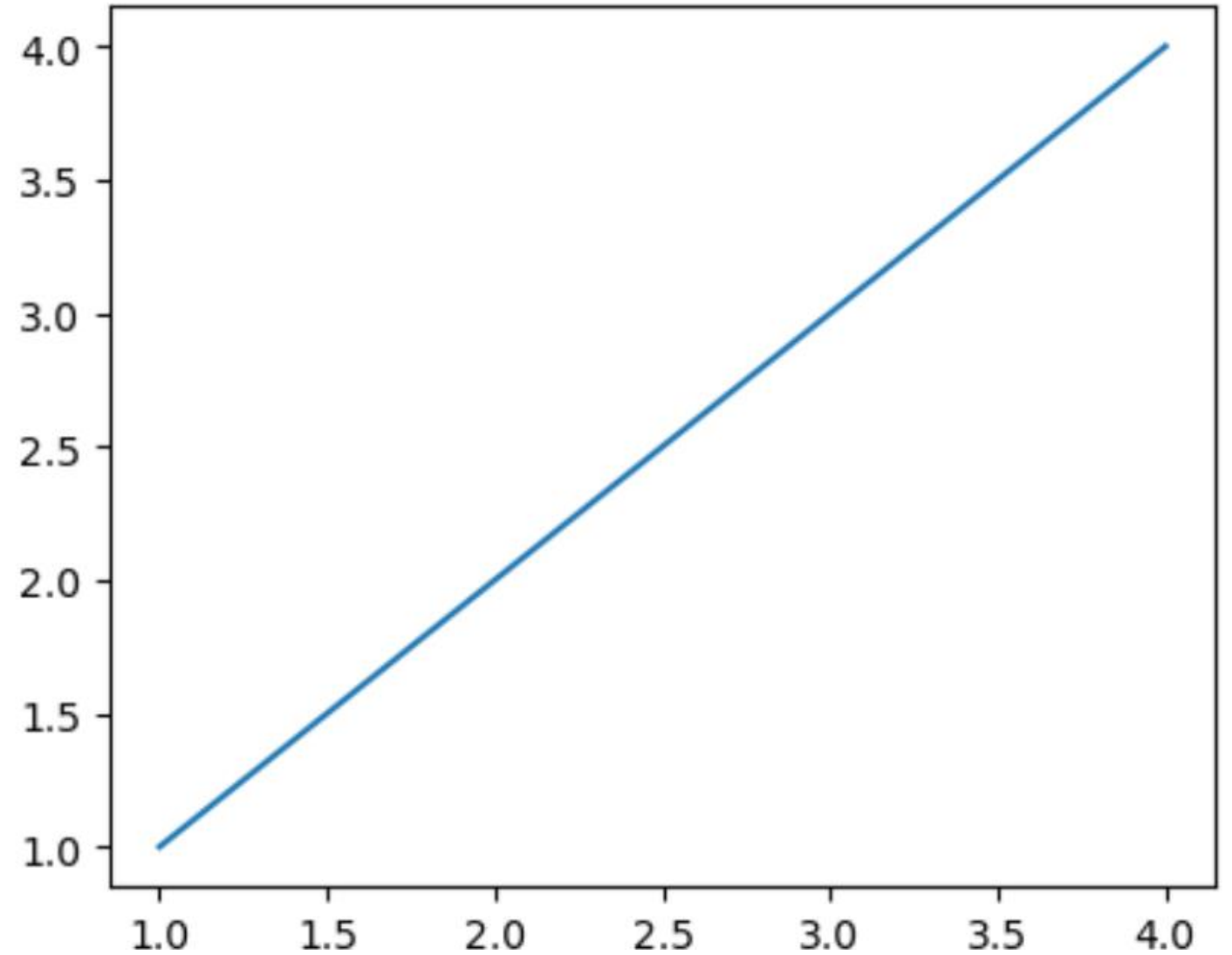
Line plot

- Best for visualizing trends or changes over a continuous interval.
- Example: Plot a stock price over time.



Line Plot

```
x = [1,2,3,4]  
y = [1,2,3,4]  
plt.plot(x,y)
```



Line Plot

The sigmoid function is defined as

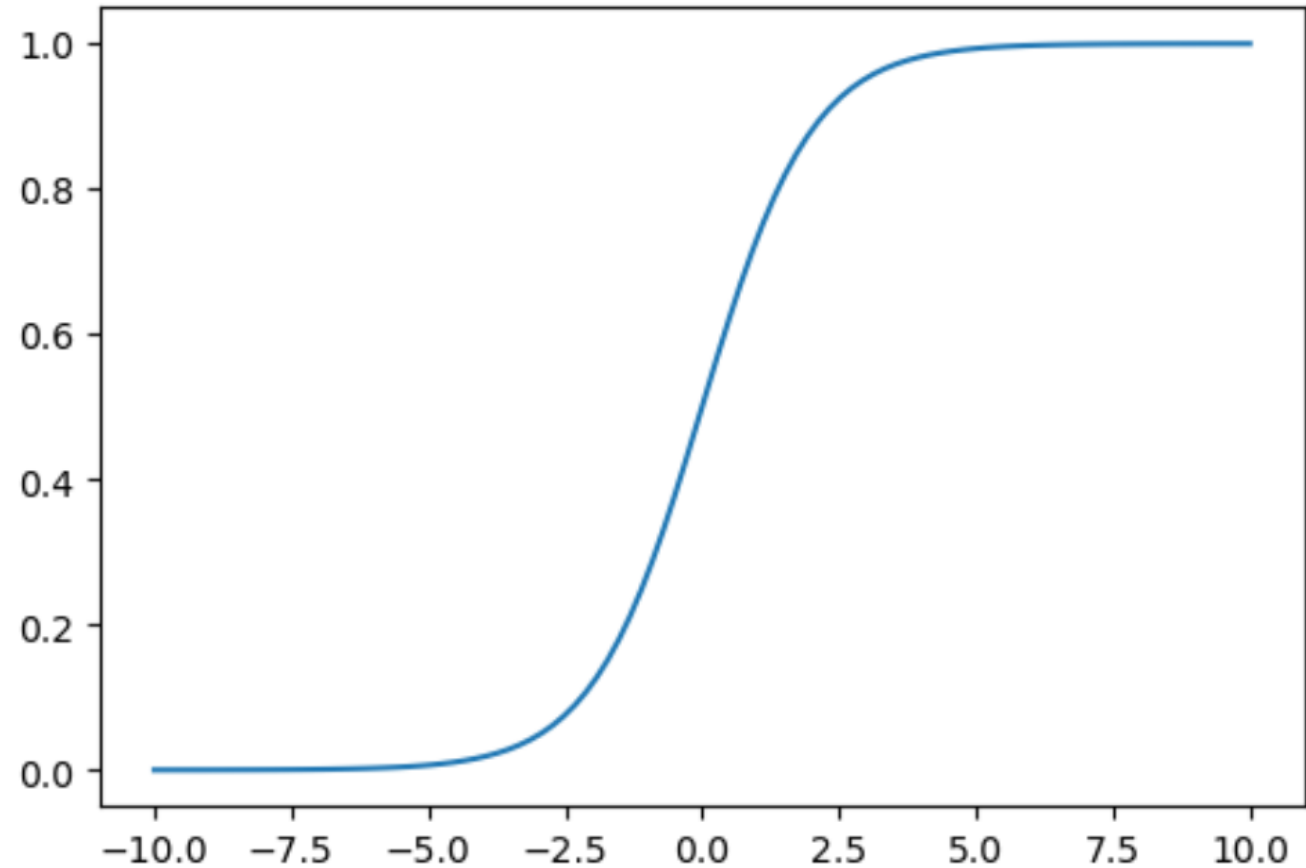
$$f(x) = \frac{1}{1+e^{-x}}$$

Behavior:

For $x \gg 0$, $e^{-x} \rightarrow 0$, so $f(x) \rightarrow 1$.

For $x \ll 0$, $e^{-x} \rightarrow \infty$, so $f(x) \rightarrow 0$.

At $x = 0$, $f(x) = 0.5$.



Line Plot - customizing

```
plt.plot(x,y, c = 'red',  
         marker='.',  
         linestyle = '--',  
         linewidth = 2)
```

Specify the color of the line

How to sign the spots












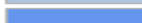
























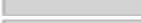



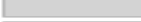



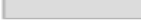



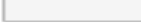




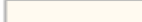
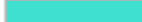

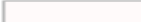
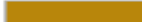







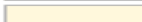
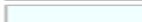






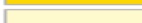



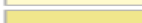
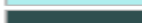





















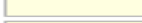



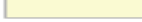






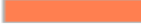




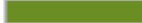










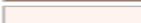









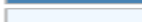



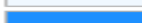





The look of the line

Line thiknes

Online help

colors:

https://matplotlib.org/stable/gallery/color/named_colors.html

	black		bisque		forestgreen		slategrey
	dimgray		darkorange		limegreen		lightsteelblue
	dimgrey		burlywood		darkgreen		cornflowerblue
	gray		antiquewhite		green		royalblue
	grey		tan		lime		ghostwhite
	darkgray		navajowhite		seagreen		lavender
	darkgrey		blanchedalmond		mediumseagreen		midnightblue
	silver		papayawhip		springgreen		navy
	lightgray		moccasin		mediumspringgreen		darkblue
	lightgrey		orange		mintcream		mediumblue
	gainsboro		wheat		mediumaquamarine		blue
	whitesmoke		oldlace		aquamarine		slateblue
	white		floralwhite		turquoise		darkslateblue
	snow		darkgoldenrod		lightseagreen		mediumslateblue
	rosybrown		goldenrod		mediumturquoise		mediumpurple
	lightcoral		cornsilk		azure		rebeccapurple
	indianred		gold		lightcyan		blueviolet
	brown		lemonchiffon		paleturquoise		indigo
	firebrick		khaki		darkslategray		darkorchid
	maroon		palegoldenrod		darkslategrey		darkviolet
	darkred		darkkhaki		teal		mediumorchid
	red		ivory		darkcyan		thistle
	mistyrose		beige		aqua		plum
	salmon		lightyellow		cyan		violet
	tomato		lightgoldenrodyellow		darkturquoise		purple
	darksalmon		olive		cadetblue		darkmagenta
	coral		yellow		powderblue		fuchsia
	orangered		olivedrab		lightblue		magenta
	lightsalmon		yellowgreen		deepskyblue		orchid
	sienna		darkolivegreen		skyblue		mediumvioletred
	seashell		greenyellow		lightskyblue		deeppink
	chocolate		chartreuse		steelblue		hotpink
	saddlebrown		lawngreen		aliceblue		lavenderblush
	sandybrown		honeydew		dodgerblue		palevioletred
	peachpuff		darkseagreen		lightslategrey		crimson
	peru		palegreen		lightslategrey		pink
	linen		lightgreen		slategrey		lightpink

Online help








- Linestyle

https://matplotlib.org/stable/api/_as_gen/matplotlib.lines.Line2D.html#matplotlib.lines.Line2D.set_linestyle

- Markers

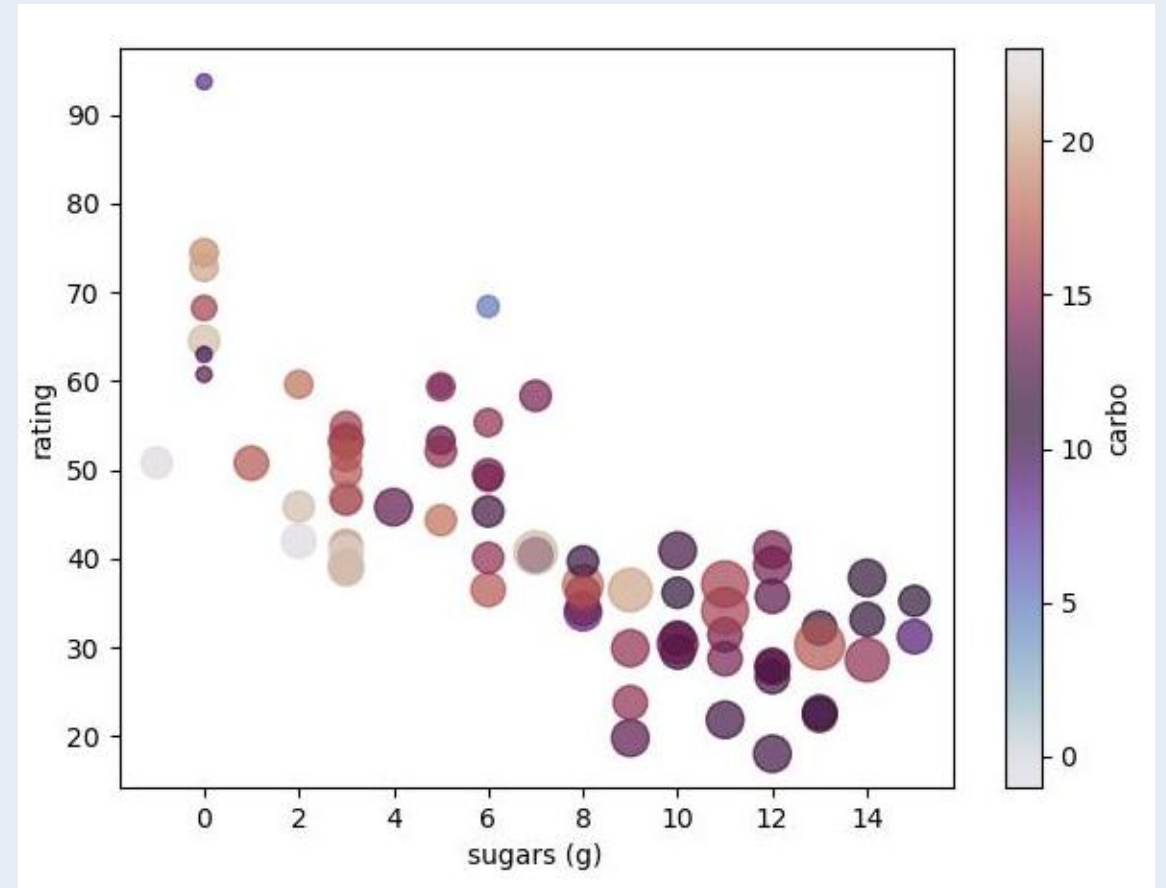
https://matplotlib.org/stable/api/markers_api.html

linestyle	description
'-' or 'solid'	solid line
'--' or 'dashed'	dashed line
'-.' or 'dashdot'	dash-dotted line
':' or 'dotted'	dotted line
'none', 'None', '', or ''	draw nothing

marker	symbol	description
"."		point
","		pixel
"o"		circle
"v"		triangle_down
"^"		triangle_up
"<"		triangle_left
">"		triangle_right

Scatter plot

- Ideal for exploring relationships or correlations between two variables.
- Example: Plot height vs. weight of individuals.



Scatter plot

```
x = df['sugars']  
y = df['rating']
```

alpha : float, default: None
between 0 (transparent) and (opaque).

```
plt.figure(figsize=(7, 5))  
plt.scatter(x,y, alpha =0.5,
```

```
s = 100,
```

s : float or array-like, shape (n,), optional The marker size

```
c =df['carbo'],
```

c : array-like or list of colors or color

```
cmap = 'twilight' )
```

cmap -The Colormap instance or registered colormap name used to map scalar data to colors

```
#Labels
```

```
plt.colorbar().set_label('carbo')
```

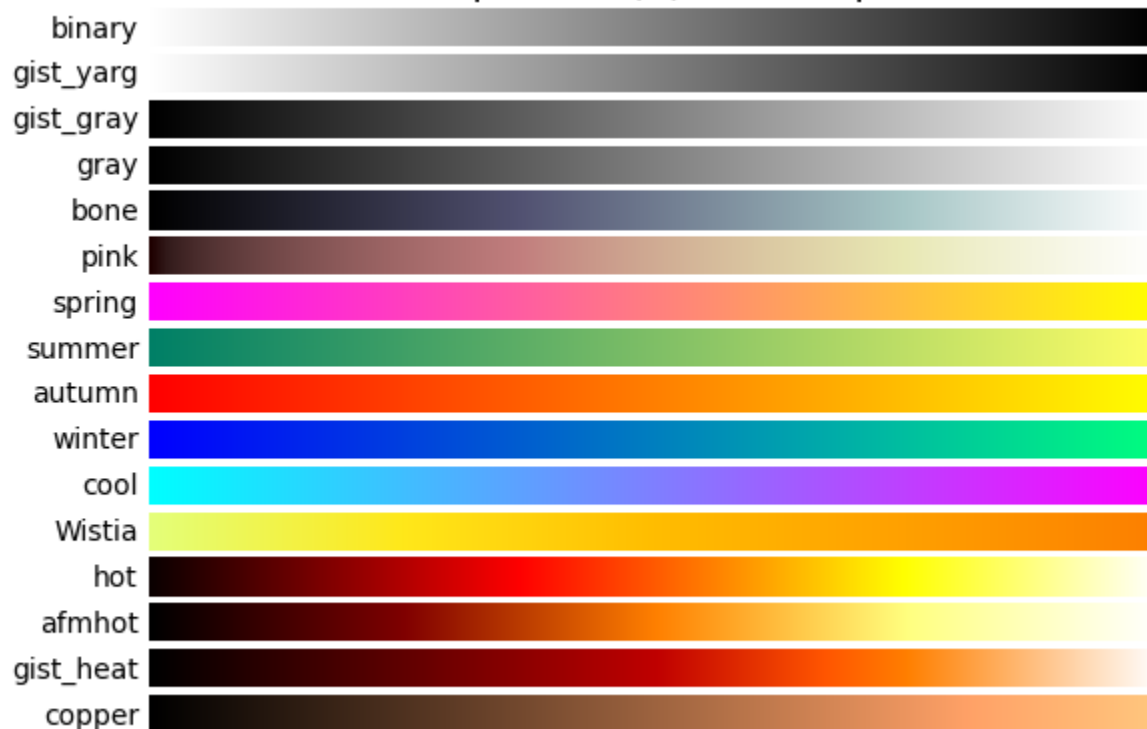
```
plt.xlabel('sugars (g)')
```

```
plt.ylabel('rating')
```

Cmap options

- <https://matplotlib.org/stable/users/explained/colors/colormaps.html>

Sequential (2) colormaps



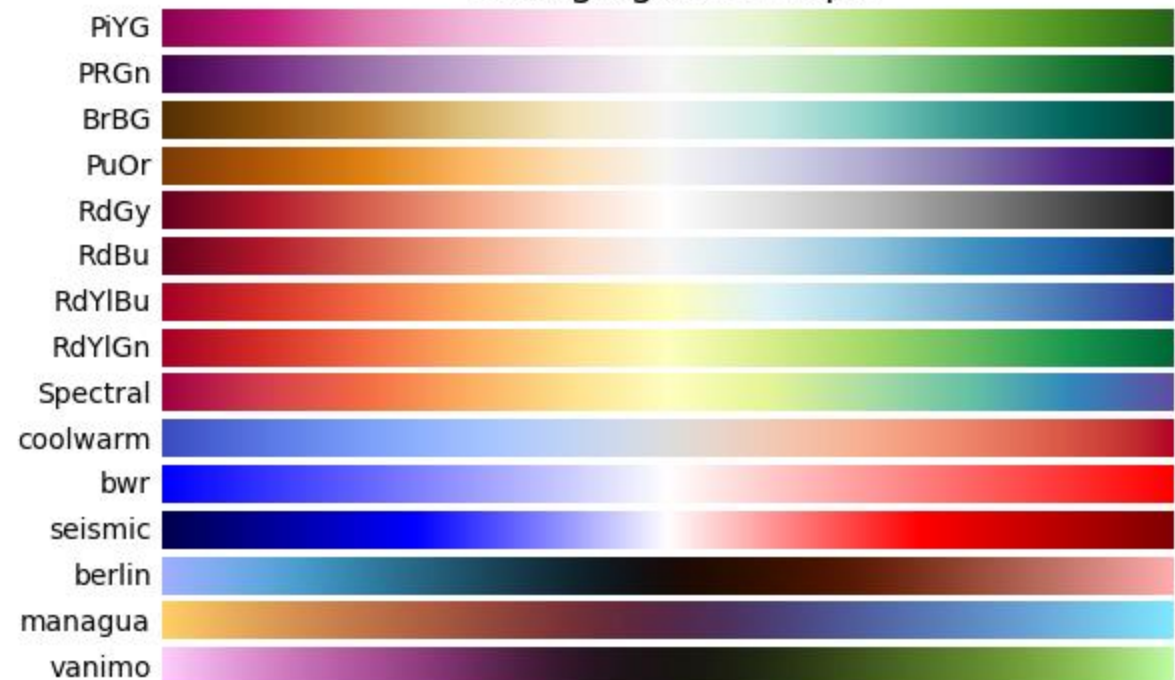
Perceptually Uniform Sequential colormaps



Cyclic colormaps



Diverging colormaps



Practice - Iris data set:

```
iris.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 150 entries, 0 to 149
```

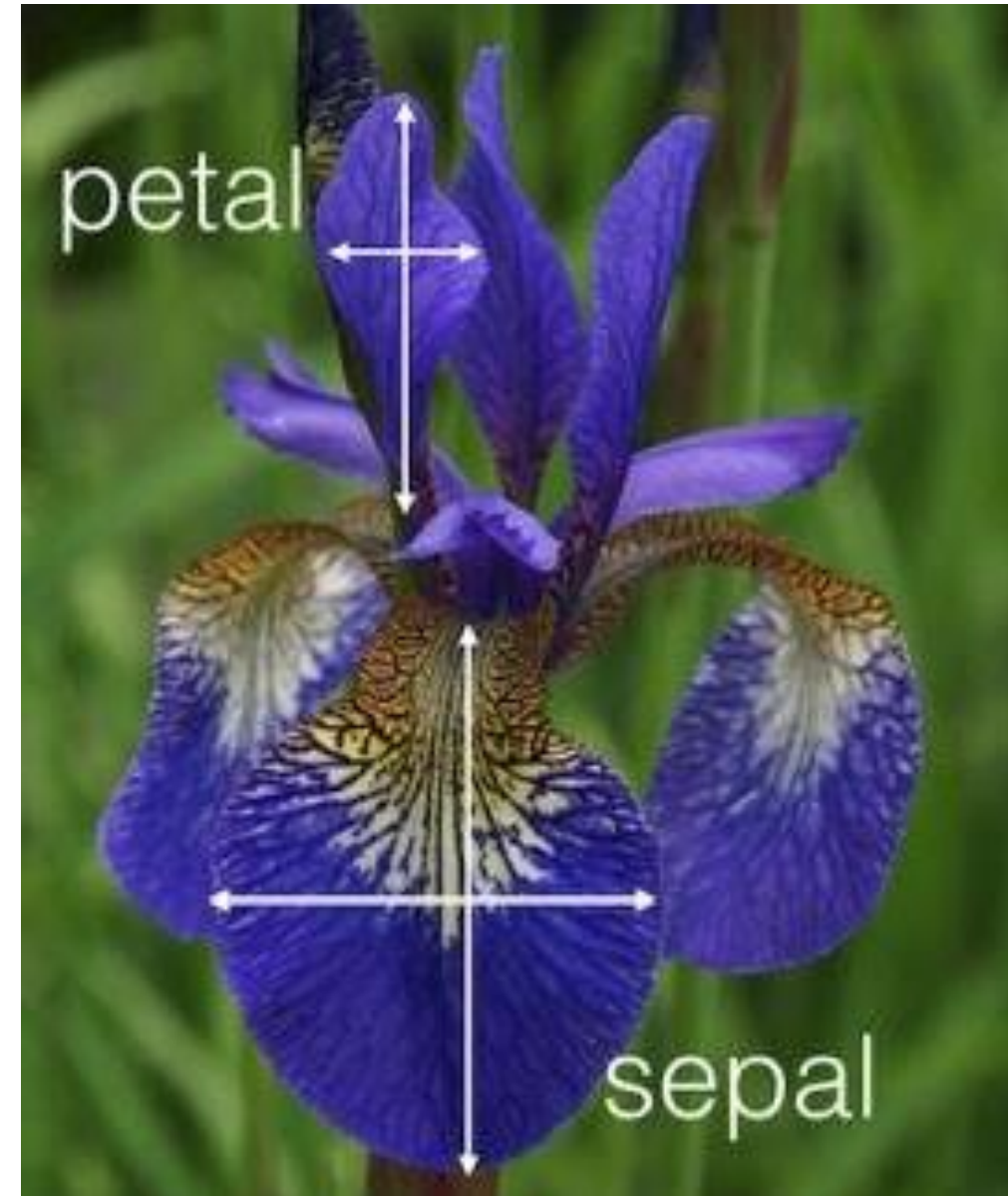
```
Data columns (total 6 columns):
```

#	Column	Non-Null Count	Dtype
0	Id	150 non-null	int64
1	SepalLengthCm	150 non-null	float64
2	SepalWidthCm	150 non-null	float64
3	PetalLengthCm	150 non-null	float64
4	PetalWidthCm	150 non-null	float64
5	Species	150 non-null	object

```
dtypes: float64(4), int64(1), object(1)
```

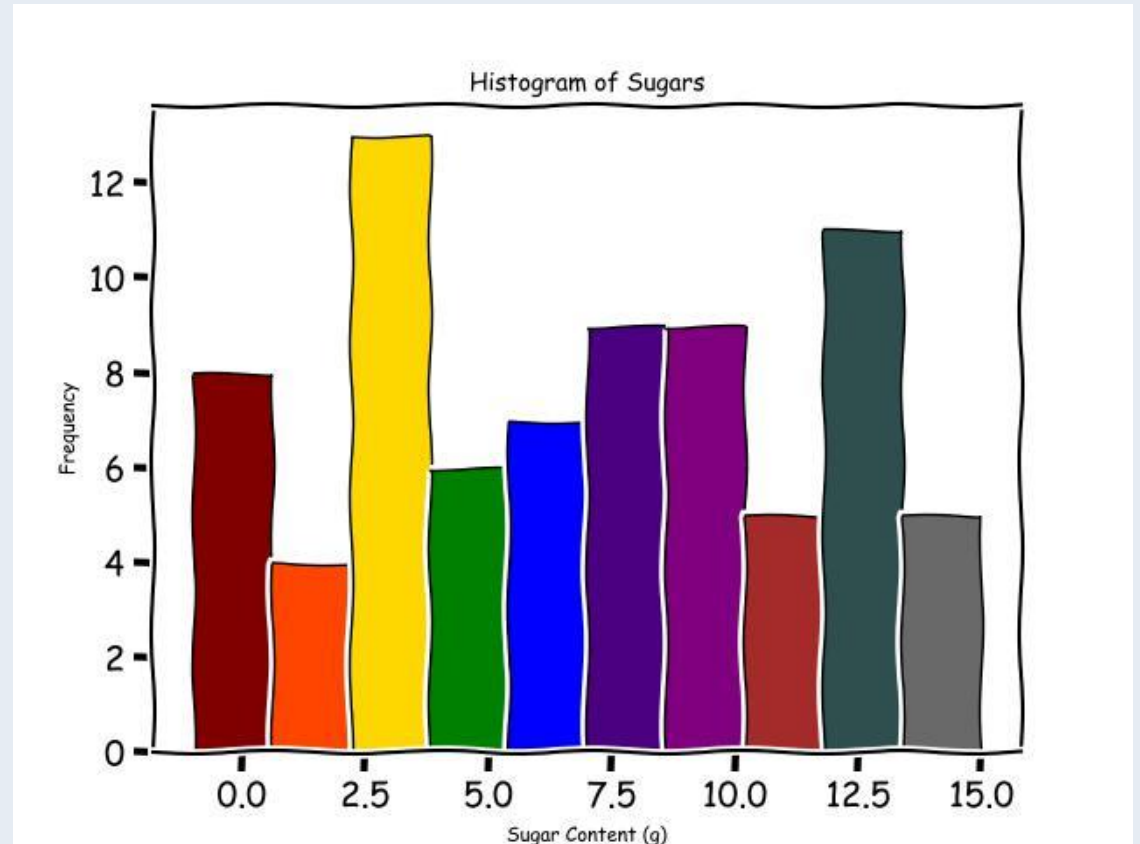
```
memory usage: 7.2+ KB
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
60	61	5.0	2.0	3.5	1.0	Iris-versicolor
62	63	6.0	2.2	4.0	1.0	Iris-versicolor
119	120	6.0	2.2	5.0	1.5	Iris-virginica
68	69	6.2	2.2	4.5	1.5	Iris-versicolor
41	42	4.5	2.3	1.3	0.3	Iris-setosa



Histogram

- Useful for showing the distribution of a dataset.
- Example: Display the frequency of age ranges in a population.



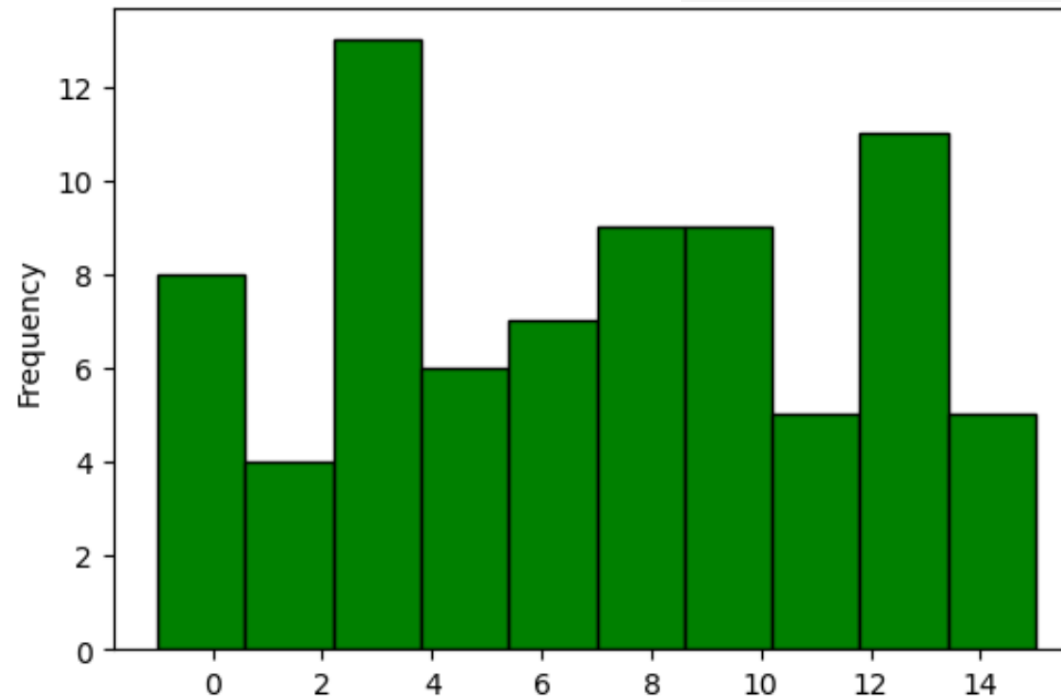
Histogram: both ways can give the same result

```
df['sugars'].hist()
```

```
df['sugars'].hist(bins = 10,  
                  color = 'green',  
                  edgecolor = 'black',  
                  grid = False)  
  
plt.show()
```

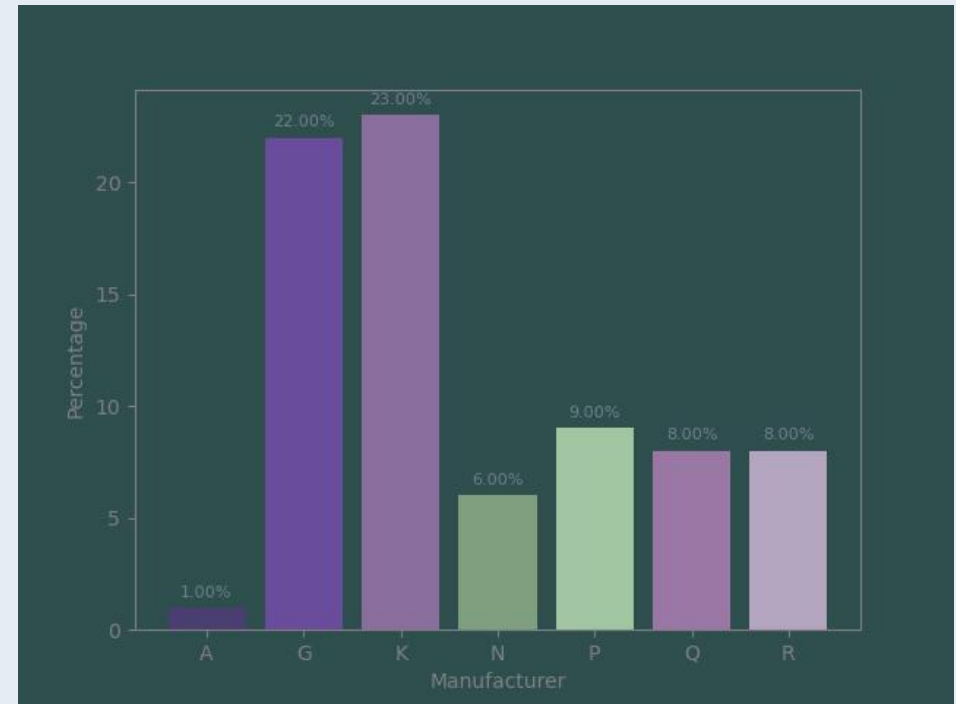
```
df['sugars'].plot(kind = 'hist')
```

```
df['sugars'].plot(kind = 'hist',  
                  bins = 10,  
                  color = 'green',  
                  edgecolor = 'black',  
                  )  
  
plt.show()
```



Bar Chart

- Suitable for comparing categories or discrete variables.
- Example: Compare sales of different product categories.



Bar Chart

```
mfrs = df.groupby('mfr')  
mfr_ratings = mfrs['rating'].mean()
```

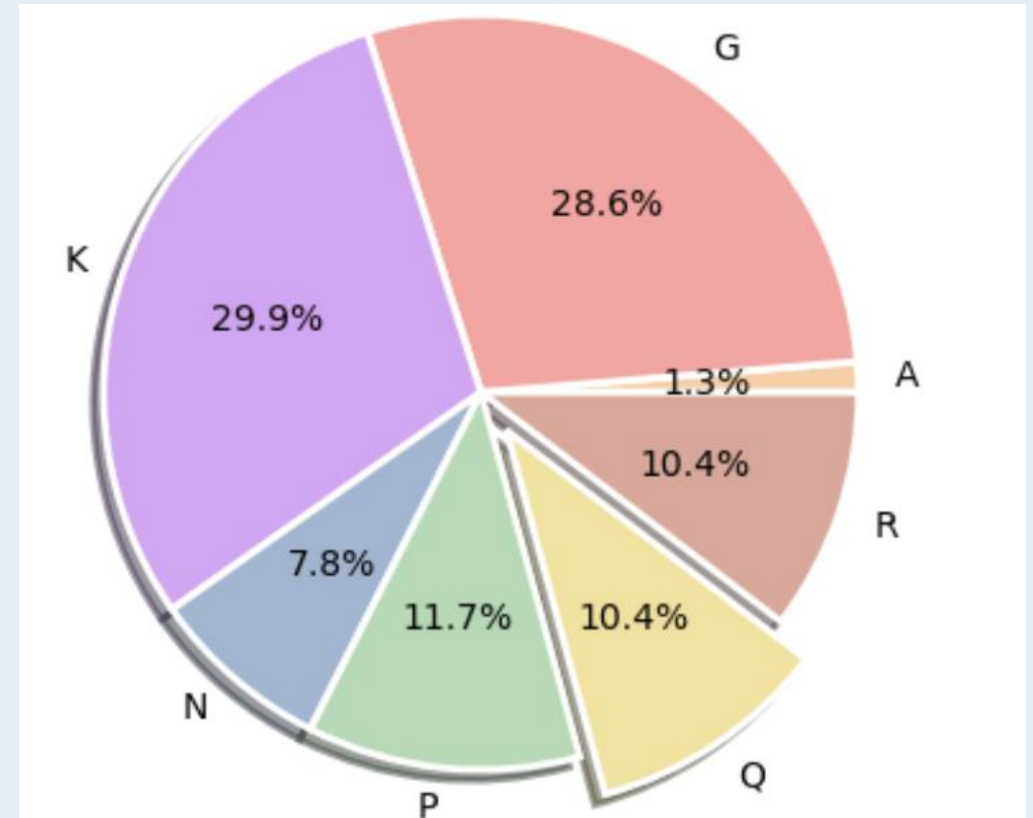
```
mfr_ratings.plot(kind = 'bar')
```

```
plt.bar(mfr_ratings.index, mfr_ratings)
```

Different syntax,
similar result

Pie Chart:

- Effective for visualizing proportions within a whole.
- Example: Show the market share of different companies in an industry.



Pie plot

```
percentage.plot( kind = 'pie')
```

```
plt.pie(percentage)
```

Different syntax,
similar result

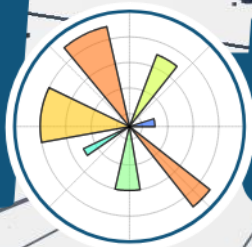
```
percentage = mfrs.size()
plt.pie(percentage, labels = percentage.index,
        wedgeprops = {'edgecolor': 'k',
                       'linewidth': 0.5},
        explode = [0, 0, 0, 0, 0, 0.1, 0],
        shadow = True,
        colors = ['red', 'orange', 'yellow', 'green', 'blue', 'purple', 'pink'])
plt.show()
```

Color and width of separating line

Creates an effect of a slice coming out

Colors for the
slices of pies

matplotlib



seaborn

The Plot Thickens

