DS 100 – Intro to Data Science

Lecture 6 – Visualizations & Functions

02/06/2025

Adam Poliak

BRYN MAWR
COLLEGE

Announcements

HW01:

expect grades back this weekend

Lab02 (Data Types and Arrays) due Friday

HW02 - Table Manipulation & Visualization:

- Due Wednesday (02/12)

Office Hours

TA office hours in Park 230

Patrick Monday OH at HC (Lutnick Library 232)

Allison’s Tuesday time TBD

Adam’s OH:
Thursday @ Dalton 300
Friday @ Park 200C

Adam	Thursday 2:30-3:30 Friday 11:30 – 1
Allison	Sunday 6-8 Tuesday TBD
Patrick	Monday 2-4 (HC) Wednesday 4-6
Candy	Wednesday 5-7 Thursday 5-7

Course Outline

Exploration **Week 1 - 5**

- Discover patterns in data
- Articulate insights (visualizations)

Inference **Week 6-10**

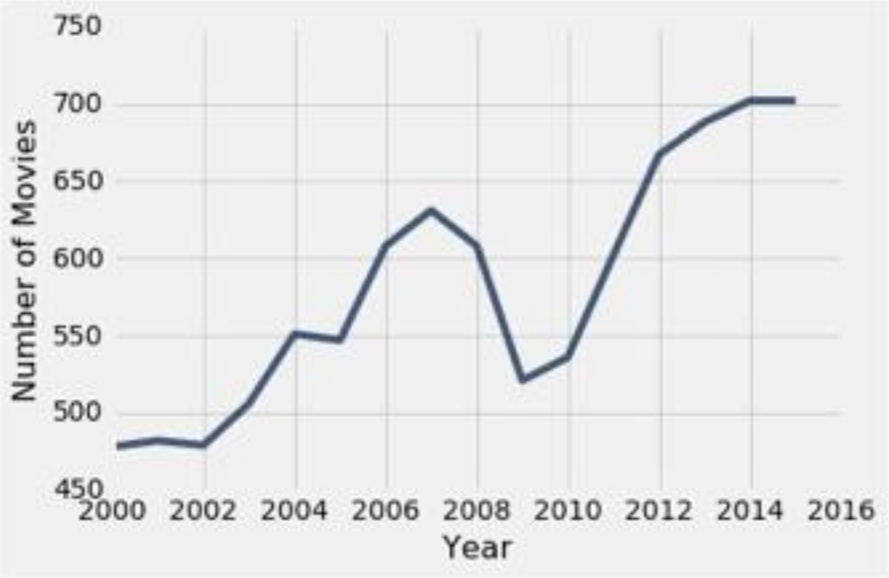
- Make reliable conclusions about the world
- Statistics is useful

Prediction **Week 11-14**

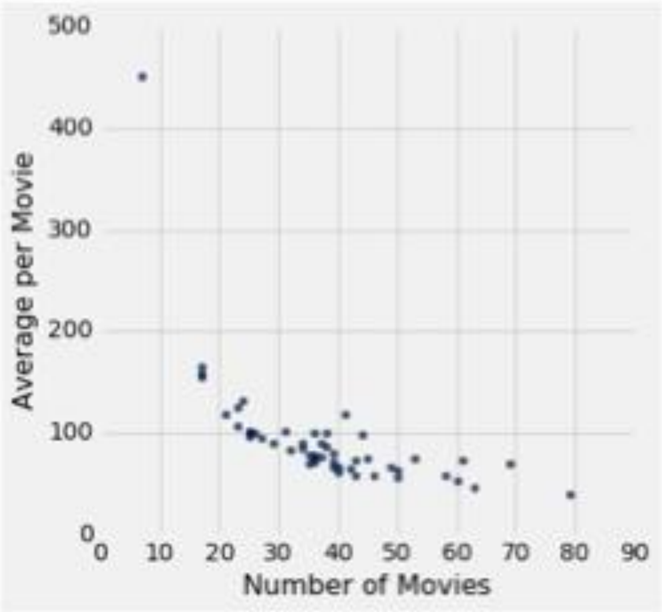
- Informed guesses about unseen data

Plotting Numerical data

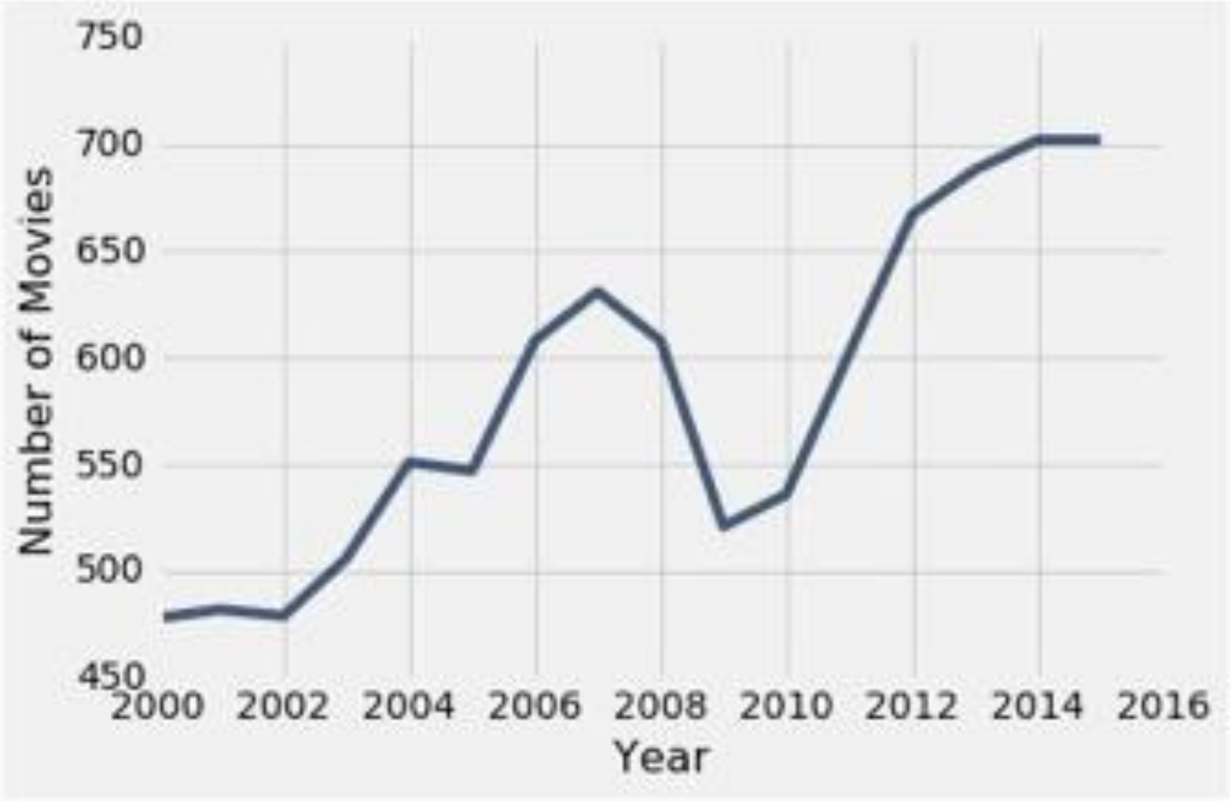
Line graph plot



Scatter plot scatter



x-axis and y-axis



Which is the x-axis?

- Year

Which is the y-axis?

- Number of Movies

Line vs Scatter Plot – when to use which?

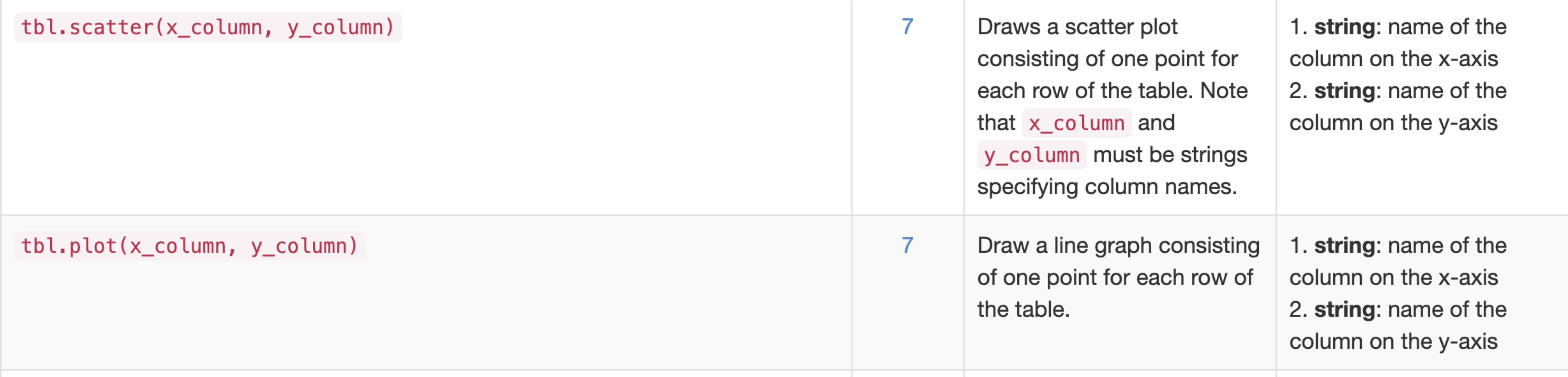
Use **line plots** for sequential data if

- x-axis has an order
- sequential differences in y values are meaningful
- there's only one y-value for each x-value
- usually: x-axis is time or distance

Use **scatter plots**

- when looking for associations between numerical attributes

Plotting in datascience



`tbl.scatter(x_column, y_column)`	7	Draws a scatter plot consisting of one point for each row of the table. Note that `x_column` and `y_column` must be strings specifying column names.	1. **string**: name of the column on the x-axis 2. **string**: name of the column on the y-axis
`tbl.plot(x_column, y_column)`	7	Draw a line graph consisting of one point for each row of the table.	1. **string**: name of the column on the x-axis 2. **string**: name of the column on the y-axis

Visualizations

Line plots

- Sequential data

Scatter plots

- Finding associations between numerical attributes

Bar plots

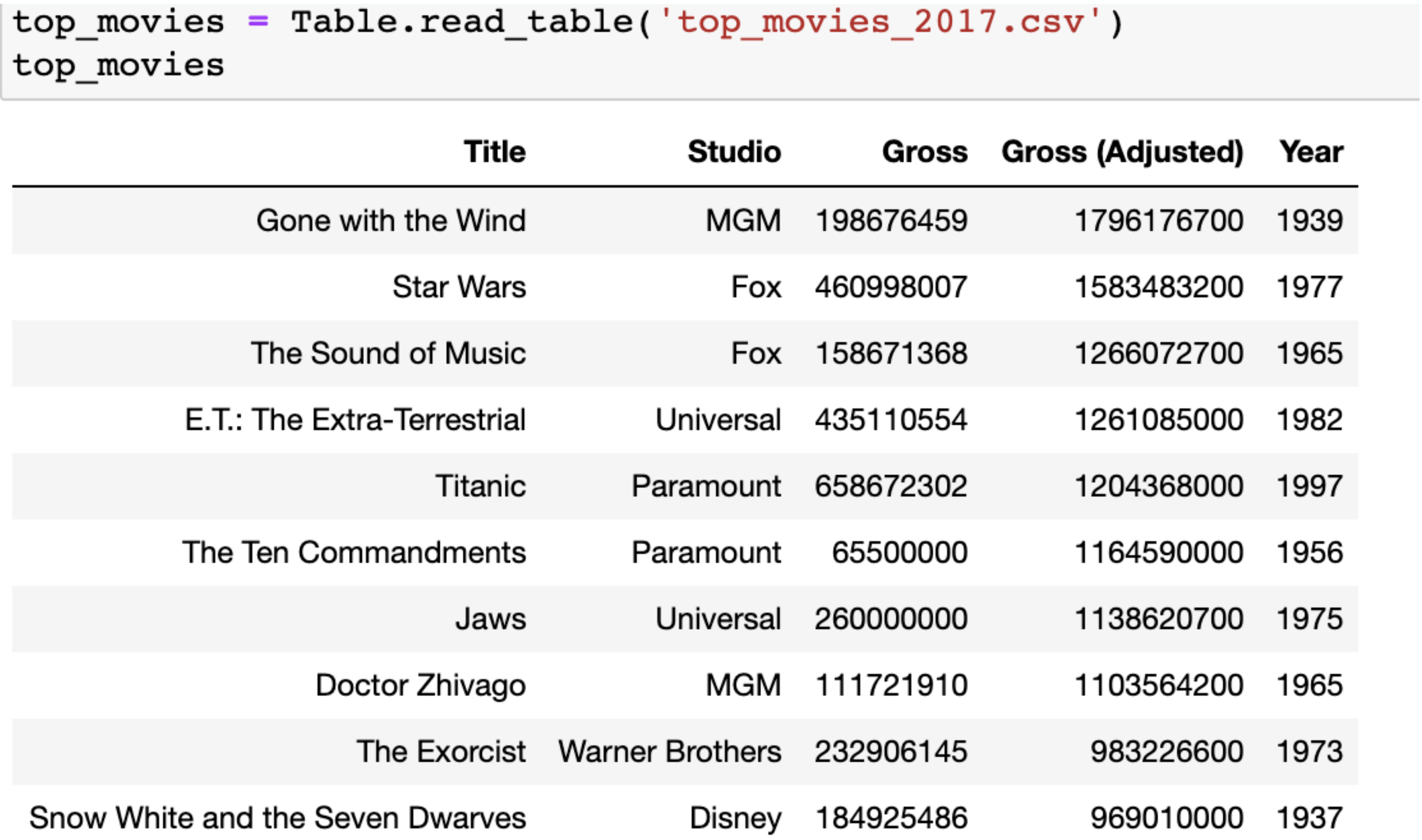
- Categorical distributions

Bar Plots

Display relationship between categorical variable and a numerical variable

Display a categorical distribution

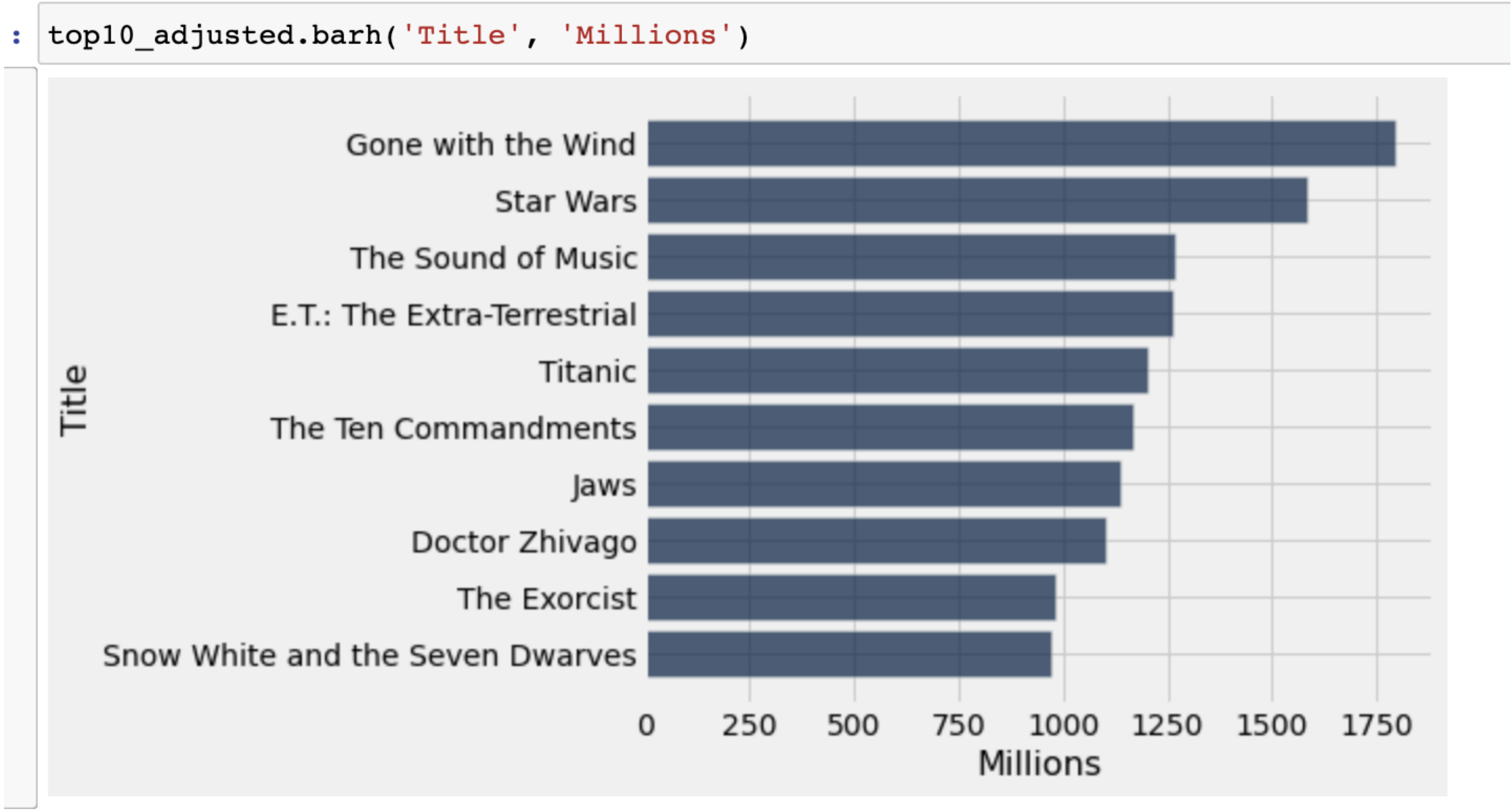
Bar Charts



```
top_movies = Table.read_table('top_movies_2017.csv')
top_movies
```

Title	Studio	Gross	Gross (Adjusted)	Year
Gone with the Wind	MGM	198676459	1796176700	1939
Star Wars	Fox	460998007	1583483200	1977
The Sound of Music	Fox	158671368	1266072700	1965
E.T.: The Extra-Terrestrial	Universal	435110554	1261085000	1982
Titanic	Paramount	658672302	1204368000	1997
The Ten Commandments	Paramount	65500000	1164590000	1956
Jaws	Universal	260000000	1138620700	1975
Doctor Zhivago	MGM	111721910	1103564200	1965
The Exorcist	Warner Brothers	232906145	983226600	1973
Snow White and the Seven Dwarves	Disney	184925486	969010000	1937

Bar Plot



```
top10_adjusted.barh('Title', 'Millions')
```

Displaying a categorical distribution

Distribution of a variable describes the frequencies of the values

The group method counts the number of values in the column

Bar chart displays the distribution of categorical variable:

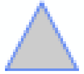
- One bar per category/value
- Length of bar is the count of individuals in that category

Area Principle

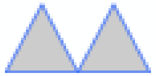
Area Principle

Areas should be proportional to values they represent

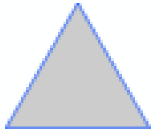
If you represent 20% by



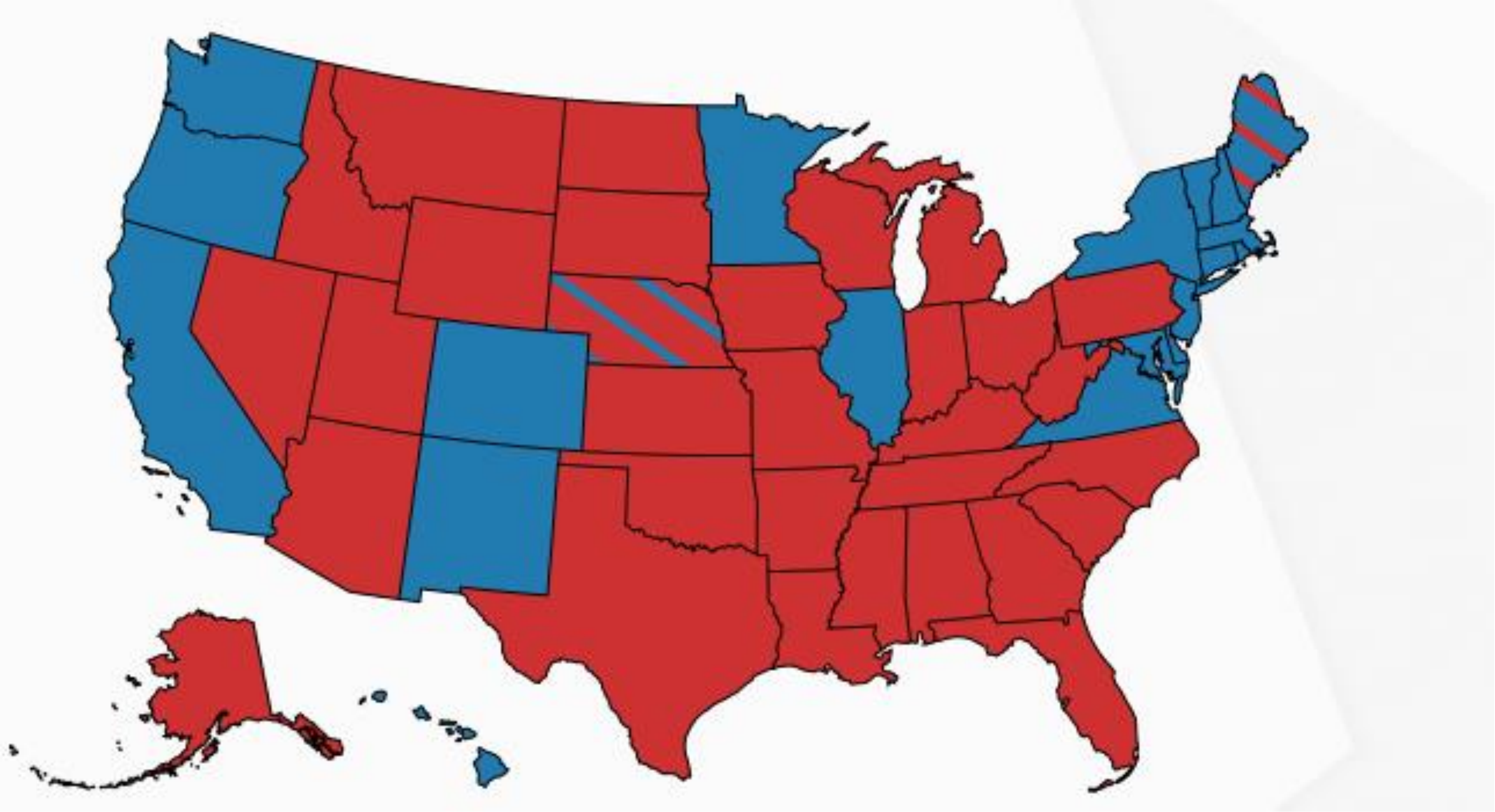
40% should be represented by



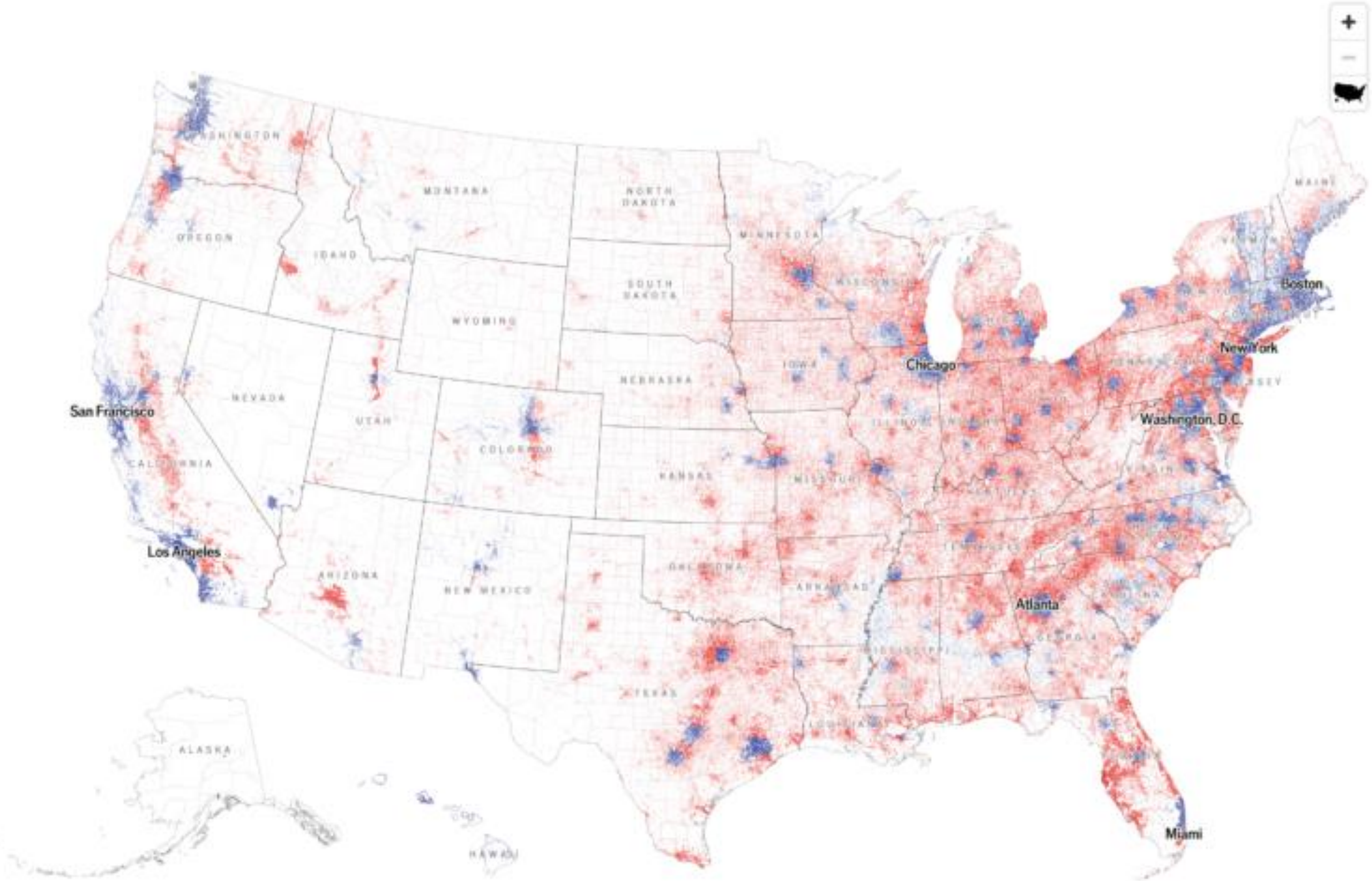
and not by



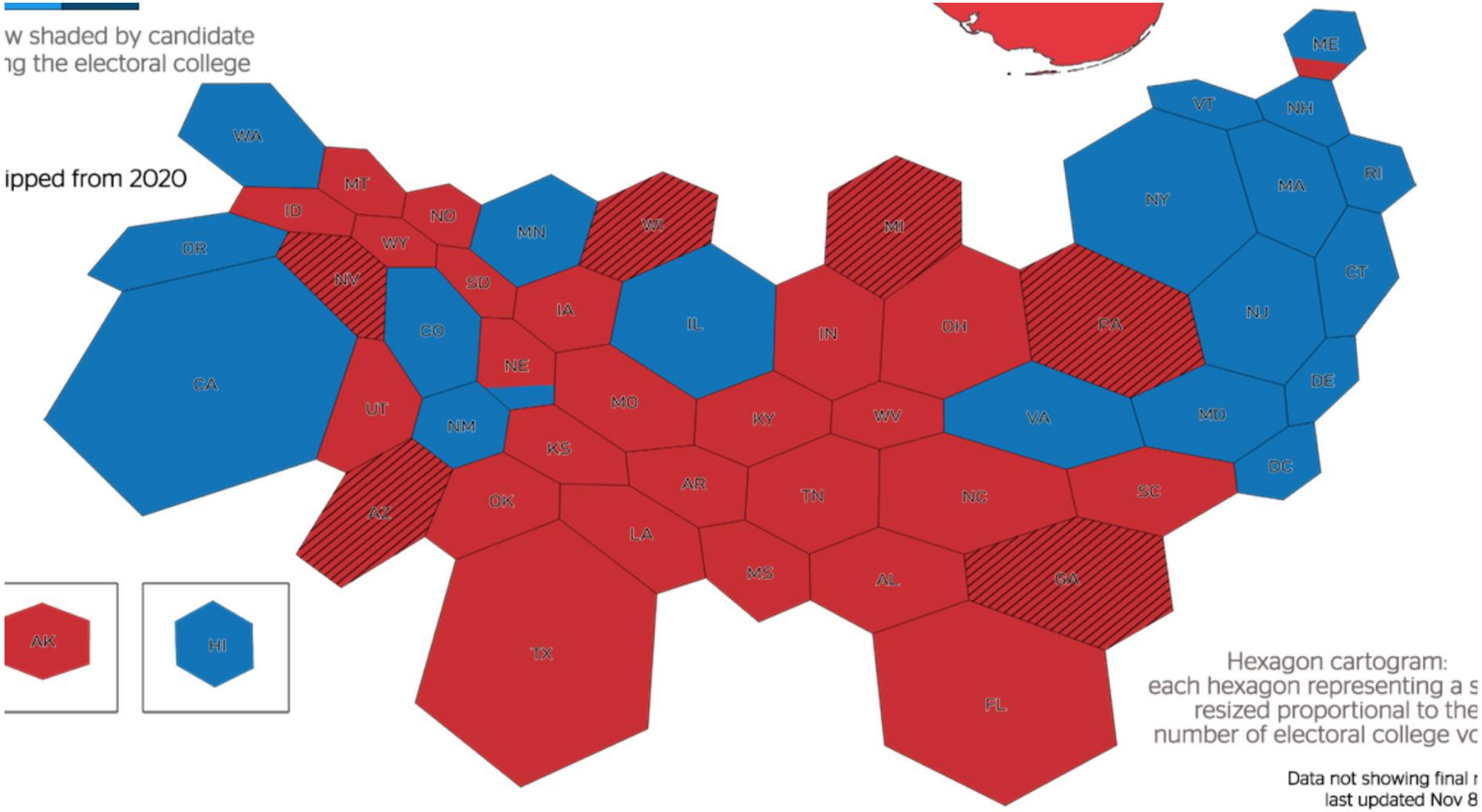
Area Principle – 2024 Election Map



Area Principle – 2024 Election Map



Area Principle – 2024 Election Map



Hexagon cartogram:
each hexagon representing a s
resized proportional to the
number of electoral college vo

Data not showing final r
last updated Nov 8

Plotting Numerical Distributions

Binning coverts a numerical distribution to a categorical distribution

Binning counts the number of numerical values that lie within a range, aka a bin

Bins contain:

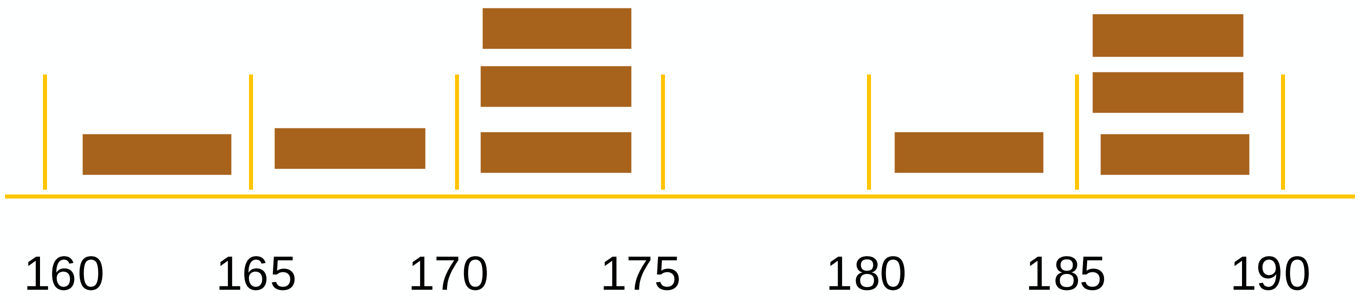
- A lower bound (inclusive)
- An upper bound (exclusive)

Bins - Example

Bins contain:

- A lower bound (inclusive)
- An upper bound (exclusive)

188, 170, 189, 163, 183, 171, 185, 168, 173, ...



Histogram

Chart that displays the distribution of a numerical variable

Uses bins; there is one bar corresponding to each bin

Uses the area principle:

- The ***area*** of each bar is the percent of individuals in the corresponding bin

Understanding Histograms

Axes

Height

Area

Histogram Axes

By default, hist uses a scale (normed=True) that ensures the area of the chart sums to 100%

The **area** of each bar is a percentage of the whole

The horizontal (x-) axis is a number line (e.g., years), and the bins sizes don’t have to be equal to each other

The vertical axis is a rate (e.g., percent per year)

Histogram Height (of a bin)

$$\text{Height} = \frac{\%\text{ in bin}}{\text{width of bin}}$$

Height measures density

the percent of data in the bin ***relative to the amount of space in the bin***

Units: percent per unit on the horizontal axis

Histogram Area (of a bar)

Area tells us what percent of all data is in a bin

Area of a bar = Height times width of a bin

- *“How many individuals in the bin?”* Use area.
- *“How crowded is the bin?”* Use height

Bar Chart or Histogram?

Bar Chart

- Distribution of categorical variable
- Bars have arbitrary (but equal) widths and spacings
- **height (or length)** and **area** of bars proportional to the percent of individuals

Histogram

- Distribution of numerical variable
- Horizontal axis is numerical: to scale, no gaps, bins can be unequal
- **Area** of bars proportional to the percent of individuals; **height** measures density

Functions

Anatomy of a Function

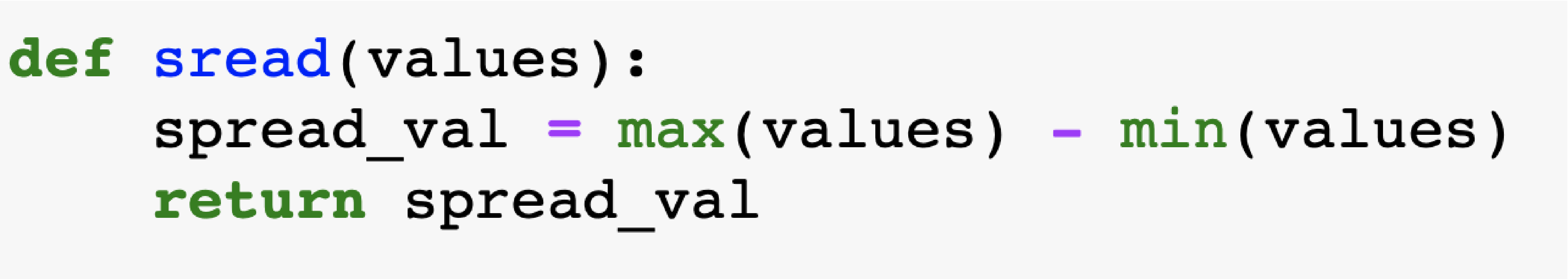
Name

Parameters / Argument Names

Body

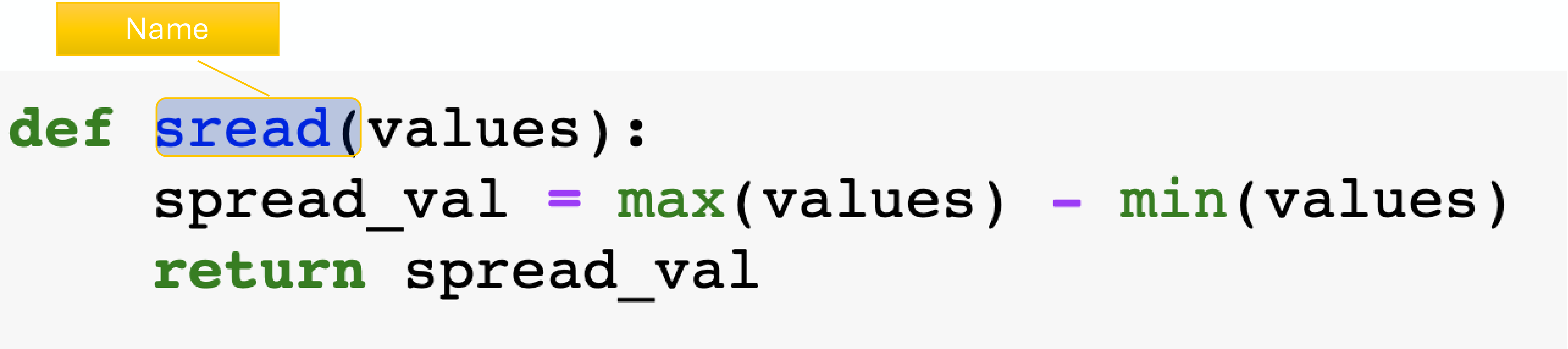
Return Expression

Example Function



```
def sread(values):
    spread_val = max(values) - min(values)
    return spread_val
```

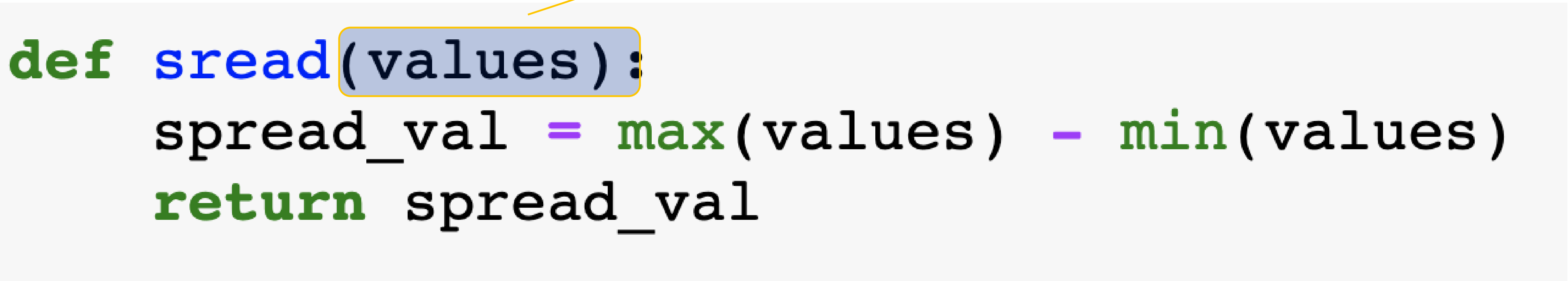
Example Function



Name

```
def sread(values):
    spread_val = max(values) - min(values)
    return spread_val
```

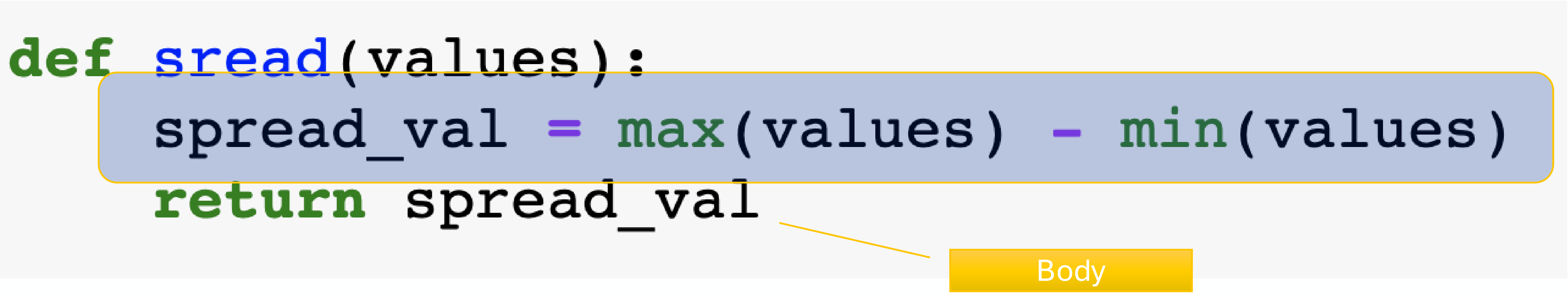
Example Function



Argument Names / Parameters

```
def sread(values):
    spread_val = max(values) - min(values)
    return spread_val
```

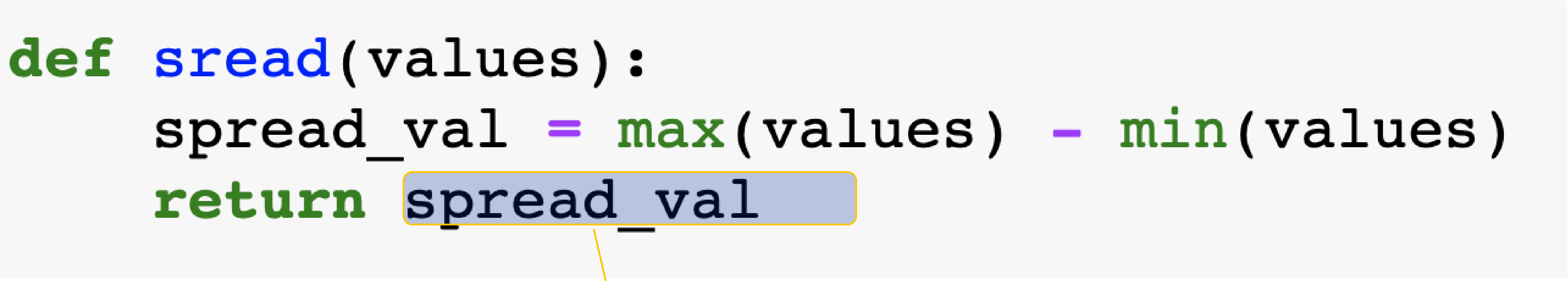
Example Function



```python
def sread(values):
    spread_val = max(values) - min(values)
    return spread_val
```

Body

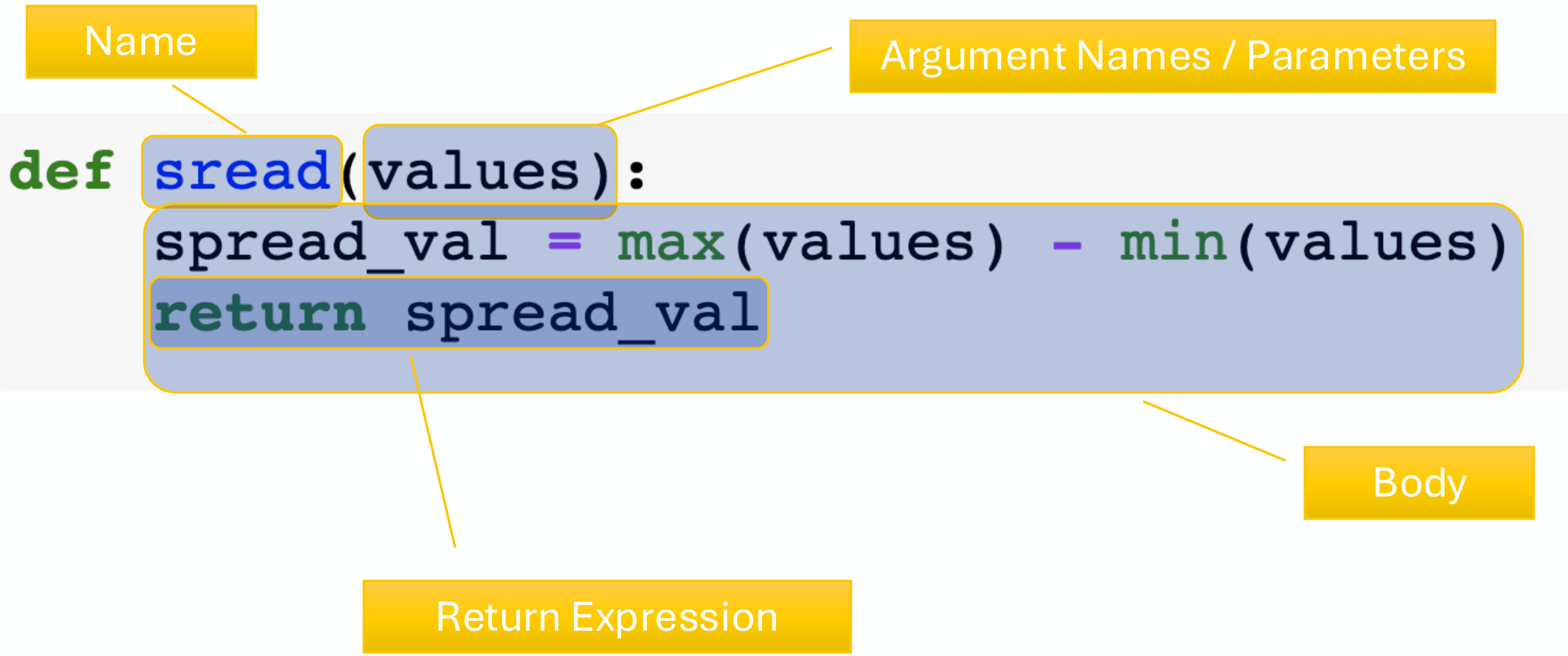
Example Function



```
def sread(values):
    spread_val = max(values) - min(values)
    return spread_val
```

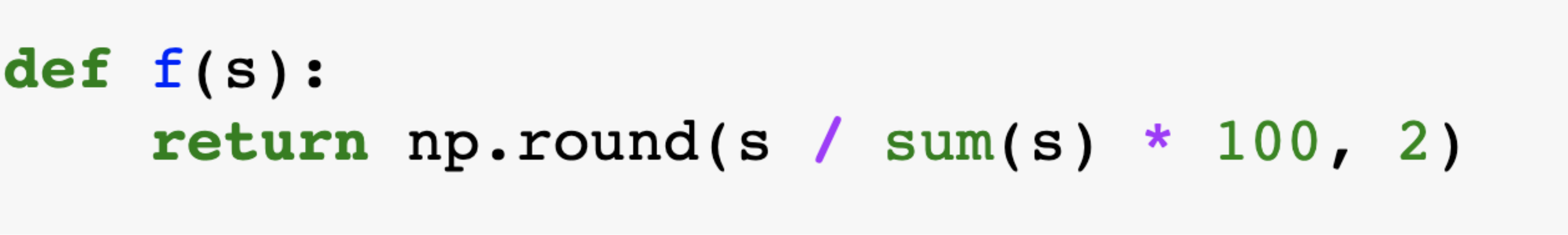
Return Expression

Example Function



```
def sread(values):
    spread_val = max(values) - min(values)
    return spread_val
```

What does this function do?



```
def f(s):
    return np.round(s / sum(s) * 100, 2)
```

What kind of input does it take?
What output will it give?
What's a reasonable name?