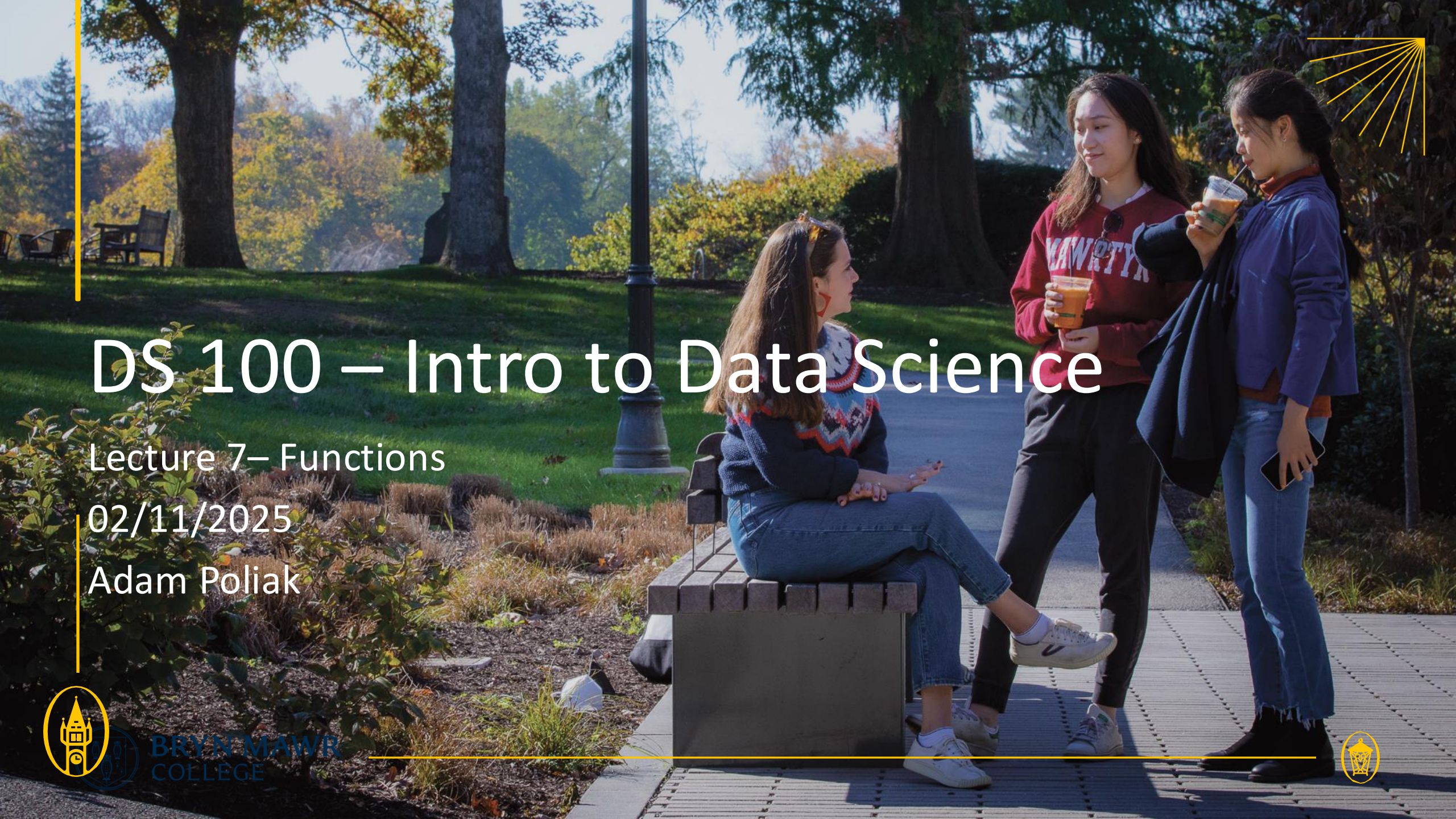


DS 100 – Intro to Data Science

Lecture 7– Functions

02/11/2025

Adam Poliak

BRYN MAWR COLLEGE

Announcements

Lab03 (Functions & Visualizations) due Friday

HW02 - Table Manipulation & Visualization:

- Due Wednesday (02/12)

HW03 – Functions, Histograms, and Groups:

- Due Wednesday (02/19)

Checkpoint/Project 1:

- Paired assignment that covers the previous section of the course material
- Released today
- Due Wednesday 02/28

Projects – Paired Assignment

3 Projects

Exploration project

- Released today
- Due Friday 02/28
- HW3 & HW4 are on the shorter side

Histograms

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:

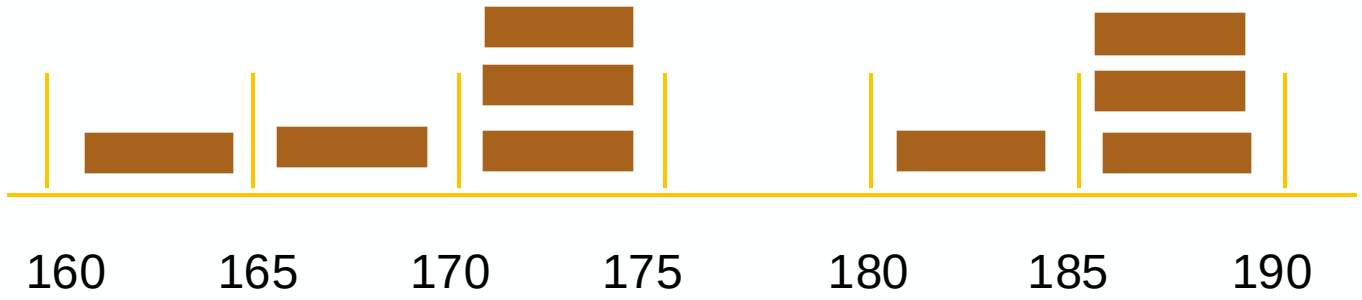
- A lower bound (inclusive)
- An upper bound (exclusive)

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

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

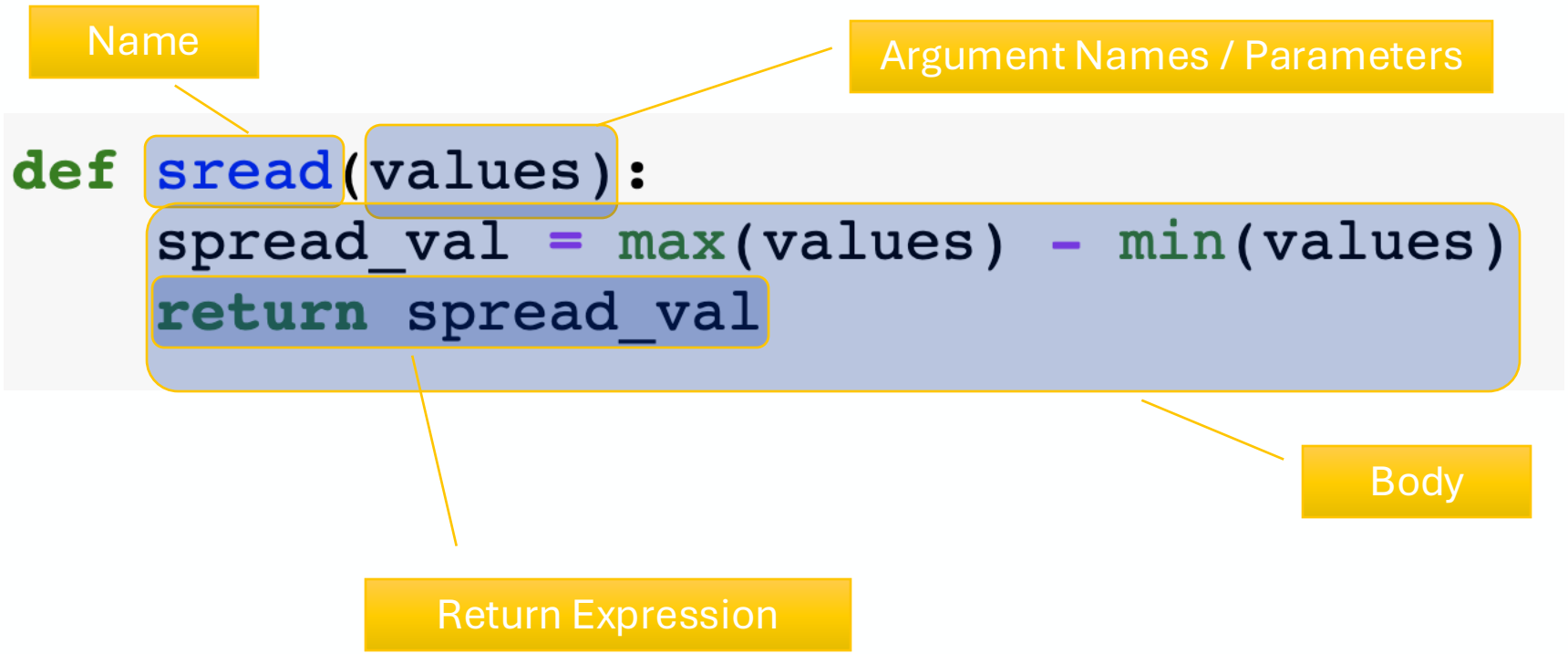
Body

Example Function

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

Return Expression

Example Function



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?

Applying Functions to Columns

The apply method creates an array by calling a function on every element in input column(s)

- First argument: Function to apply
- Other arguments: The input column(s)

table_name.apply(function_name, 'column_label')

Grouping by One Column

The **group** method aggregates all rows with the same value for a column into a single row in the resulting table.

- First argument: Which column to group by
- Second argument: (Optional) How to combine values

len — number of grouped values (default)
list — list of all grouped values
sum — total of all grouped values

Lists as Generic Sequences

A list is a sequence of values (just like an array), but the values can all have different types

```
[2+3, 'four', Table().with_column('K', [3, 4])]
```

Lists can be used to create table rows.

If you create a table column from a list, it will be converted to an array automatically

Grouping by Multiple Columns

The **group** method can also aggregate all rows that share the combination of values in multiple columns

- First argument: A list of which columns to group by
- Second argument: (Optional) How to combine values

Pivot Tables

Cross-classifies according to two categorical variables
Produces a grid of counts or aggregated values
Two required arguments:

- First: variable that forms column labels of grid
- Second: variable that forms row labels of grid

Two optional arguments (include **both** or **neither**)
values='column_label_to_aggregate'
collect=function_to_aggregate_with

Group vs Pivot

Pivot

- One combo of grouping variables **per entry**
- **Two** grouping variables: columns and rows
- Aggregate values of **values column**
- Missing combos **= 0 (or empty string)**

Group

- One combo of grouping variables **per row**
- **Any number** of grouping variables
- Aggregate values of **all other columns** in table
- Missing combos **absent**

Joining Two Tables

tblA.join(colA, tblB, colB)

tblA.join(colA, tblB)