

# YData: Introduction to Data Science



Lecture 06: Tables continued

# Overview

Review and continuation of Tables

Lists

Census Data

# Announcements

**Homework 2** [has been posted](#), It is **due on Sunday February 13<sup>th</sup> at 11pm**

**Practice 2 exercises** have also been posted

- **These are not turned in** but will be useful to complete to gain more Python practice

Any questions about anything?



# Review of Tables

# Review: Table Methods

`tb.select(label)` - constructs a new table with just the specified columns

`tb.drop(label)` - constructs a new table in which the specified columns are omitted

`tb.sort(label)` - constructs a new table with rows sorted by the specified column

`tb.where(label, condition)` - constructs a new table with just the rows that match the condition

The diagram shows a table with three columns: 'Name', 'Code', and 'Area (m2)'. The 'Code' column is highlighted with a green dashed border and a green callout box labeled 'Label'. The 'Nevada' row is highlighted with a blue dashed border and a blue callout box labeled 'Row'. The 'CA' cell is highlighted with a red solid border and a red callout box labeled 'Column'.

Name	Code	Area (m2)
California	CA	163696
Nevada	NV	110567

# Additional Table Methods and Properties

`tb.num_rows` – returns the number of rows

`tb.num_columns` – returns the number of columns

`relabel('column_name', 'new_name')` - constructs a new table where 'column\_name' has been renamed to 'new\_name'

`tb.take(numbers)` - constructs a new table with just the row numbers specified

A diagram illustrating a table structure with three columns: Name, Code, and Area (m2). The table contains two rows: California and Nevada. Annotations include a green callout labeled 'Label' pointing to the 'Code' header, a blue callout labeled 'Row' pointing to the 'Nevada' row, and a red callout labeled 'Column' pointing to the 'Code' column. The 'Code' column is highlighted with a red border, and the 'Nevada' row is highlighted with a blue dashed border.

Name	Code	Area (m2)
California	CA	163696
Nevada	NV	110567

[See Berkeley's documentation](#)

Let's explore this in Jupyter!

# Lists

# Lists are Generic Sequences

Lists are one of the most widely used data types in Python

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

```
[3, 21.3, "unicorn"]
```

Lists can be used to create table rows, and for many other things!

Let's explore this in Jupyter!



# Constructing Tables

# Creating Tables

We can create tables by:

- Constructing a new Table using the Table() construction
- Adding columns using the method
- `tb.with_columns('name', array)`

Let's construct a Table specifying how far different streets are from our classroom



# Census Data

# Census Data

Every ten years, the Census Bureau counts how many people there are in the U.S.

- [Rich data set](#) that can be used to explore demographic trends

Information about the data [from the codebook](#):

- The SEX column:
  - 1 is Male
  - 2 is Female
  - 0 is Total (Male + Female)
- The AGE column
  - 999 is total of all ages
  - 100 is total of people 100 years or older
- The POPESTIMATE2014 column: 7/1/2014 estimate



Let's explore this in Jupyter!