DATA 8

Spring 2020

Lecture 6

Census

Announcements

- HW 2 due on Thursday, 2/6
  - Turn in on Wednesday for a bonus point!
- **Tutoring sections start today**
  - We'll be adding more later this week
- Swupnil's office hours are live!
  - Sundays 5-7PM at Caffe Strada

Weekly Goals

- **Today**
  - Table review
  - Working with Census data
- Wednesday
  - Visualizing data
  - Distributions
- Friday
  - Visualizing two kinds of distributions
  - Proportions as areas

Table Review

Table Structure

- A Table is a sequence of labeled columns
- Labels are strings
- Columns are arrays, all with the same length

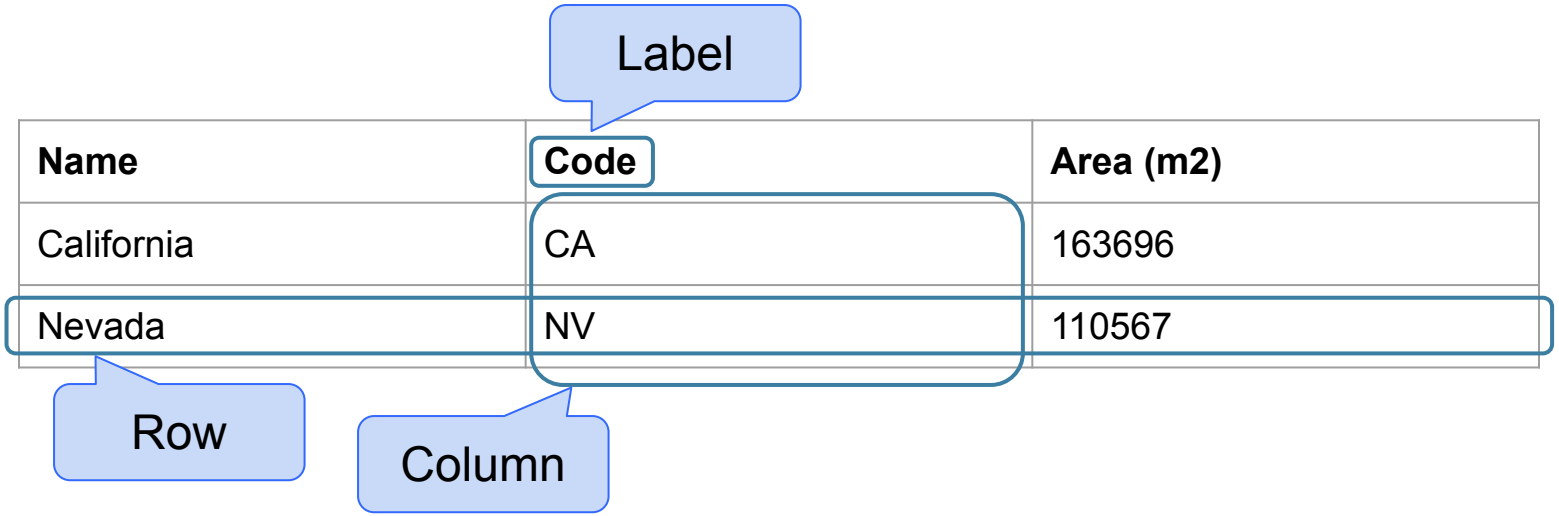

Table Methods

- Creating and extending tables:
  - `Table().with_column` and `Table.read_table`
- Finding the size: `num_rows` and `num_columns`
- Referring to columns: indices
  - column indices start at 0
- Accessing data in a column
  - `column` takes a label or index and returns an array
- Using array methods to work with data in columns
  - `item`, `sum`, `min`, `max`, and so on
- Creating new tables containing some of the original columns:
  - `select`, `drop`

(Demo)

Manipulating Rows

- `t.sort(column)` sorts the rows in increasing order
- `t.sort(column, descending=True)` sorts the rows in decreasing order
- `t.take(row_numbers)` keeps the numbered rows
  - Each row has an index, starting at 0
- `t.where(column, are.condition)` keeps all rows for which a column's value satisfies a condition
- `t.where(column, value)` keeps all rows for which a column's value equals some particular value
  - Same as `t.where(column, are.equal_to(value))`

Discussion Questions

The table **`nba`** has columns **`PLAYER`**, **`POSITION`**, and **`SALARY`**.

a) Create an array containing the names of all point guards (**`PG`**) who make more than $15M/year

```
guards = nba.where('POSITION', 'PG')
guards.where('SALARY', are.above(15)).column('PLAYER')
```

b) After evaluating these two expressions in order, what's the result of the second one?

```
nba.drop('POSITION')
nba.num_columns
```

(Demo)

Attribute Types

Types of Attributes

All values in a column of a table should be both the same type **and** be comparable to each other in some way

- **Numerical** — Each value is from a numerical scale
  - Numerical measurements are ordered
  - Differences are meaningful
- **Categorical** — Each value is from a fixed inventory
  - May or may not have an ordering
  - Categories are the same or different

“Numerical” Attributes

Just because the values are numbers, doesn’t mean the variable is numerical

- Census example has numerical `SEX` code (0, 1, and 2)
- It doesn’t make sense to perform arithmetic on these “numbers”, e.g. 1 - 0 or (0+1+2)/3 are meaningless
- The variable `SEX` is still categorical, even though numbers were used for the categories

Census Data

The Decennial Census

- Every ten years, the Census Bureau counts how many people there are in the U.S.
- In between censuses, the Bureau estimates how many people there are each year.
- Article 1, Section 2 of the Constitution:
  - “Representatives and direct Taxes shall be apportioned among the several States … according to their respective Numbers …”

Census Table Description

- Values have column-dependent interpretations
  - The SEX column: 1 is *Male*, 2 is *Female*
  - The POPESTIMATE2010 column: *7/1/2010 estimate*
- In this table, some rows are sums of other rows
  - The SEX column: 0 is *Total* (of *Male* + *Female*)
  - The AGE column: 999 is *Total* of all ages
- Numeric codes are often used for storage efficiency
- Values in a column have the same type, but are not necessarily comparable (AGE 12 vs AGE 999)

http://www2.census.gov/programs-surveys/popest/datasets/2010-2015/national/asrh/nc-est2015-agesex-res.pdf

Analyzing Census Data

Leads to the discovery of interesting features and trends in the population

(Demo)