MATH1401

Fall 2021

Lecture 6

Tables (Again)

Class Checklist

- **Lab 2 – Due Date** : Thursday 9/9 – 5 PM
  - Graded Questions : 1.1, 2.1,2.1.1,3.1-3.6, 4.1-4.2
- **HW 2** – Tuesday: 9/14

- **Quiz 4** – Tuesday: 9/7 – Covers Chapter 5
- **Quiz 5** – Thursday: 9/9 – Covers Chapter 6

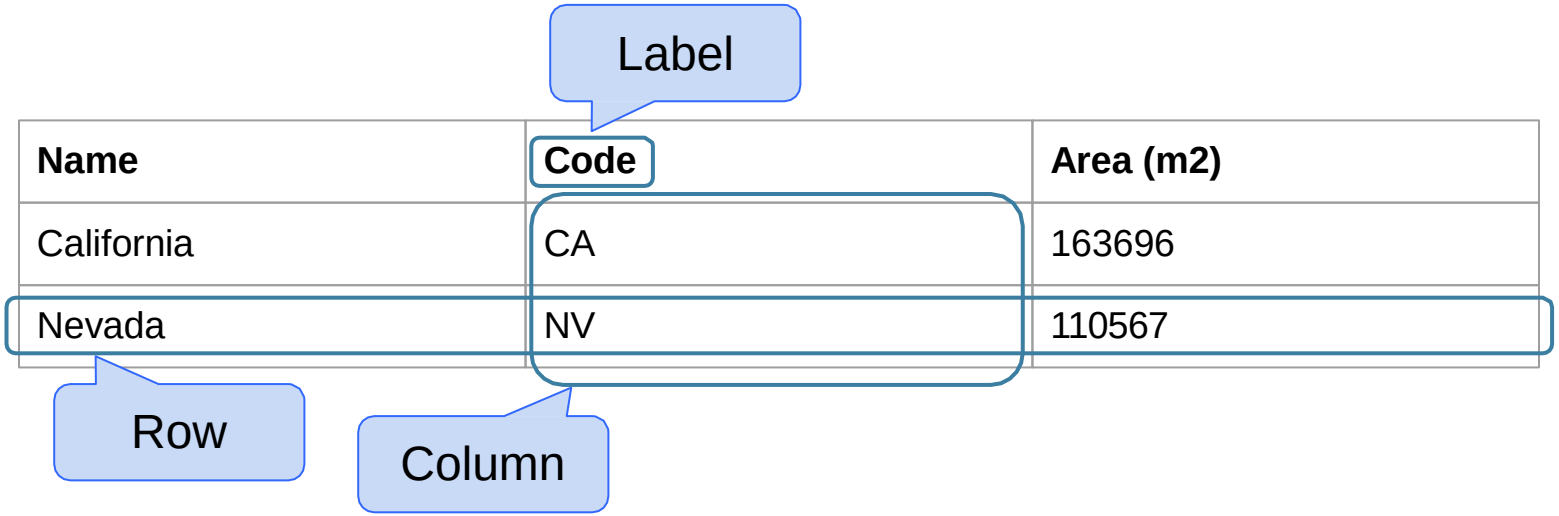
Review

Lecture 5 Review

- Declare an Array
- Array Operations - +,-,*,**
- Use np.range()
- Read a table
  - `Table.read_table(filename)` - reads a table from a spreadsheet

Table Structure

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



Create a table

- `Table()`
  - Creates an empty table

- `Table().with_column('Name',arraydata)`
  - Creates a table with column ‘Name’ and entries given by arraydata

More Table Methods

- `t.num_rows:` finds number of rows
- `t.num_columns:` finds number of columns
- `t.labels:` returns array of labels
- `t.relabeled('oldlabel','newlabel'):` replaces old label with new label
- `t.column('columnName')-` returns column as an array

(Demo)

Tables

Lecture 6 Checklist

- Use **table.where(column,are.condition)**
- Numerical Data
- Categorical Data

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

Manipulating Rows

- `t.where(column, value)` keeps all rows for which a column's value equals some particular value
- `t.where(column, are.condition)` keeps all rows for which a column's value satisfies a condition

Manipulating Rows

Predicate	Description
`are.equal_to(Z)`	Equal to `Z`
`are.above(x)`	Greater than `x`
`are.above_or_equal_to(x)`	Greater than or equal to `x`
`are.below(x)`	Less than `x`
`are.below_or_equal_to(x)`	Less than or equal to `x`
`are.between(x, y)`	Greater than or equal to `x`, and less than `y`
`are.strictly_between(x, y)`	Greater than `x` and less than `y`
`are.between_or_equal_to(x, y)`	Greater than or equal to `x`, and less than or equal to `y`
`are.containing(S)`	Contains the string `S`

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)