DSFA

Spring 2019

Lecture 4

Data Types, Arithmetic, Tables and Arrays

Announcements

- URL for website:
- What if you just added?
- HW 01 due Today (bonus point for early submission)
- Lab 02
- HW 02 posted later today
- iClicker/Reef polling trial next week

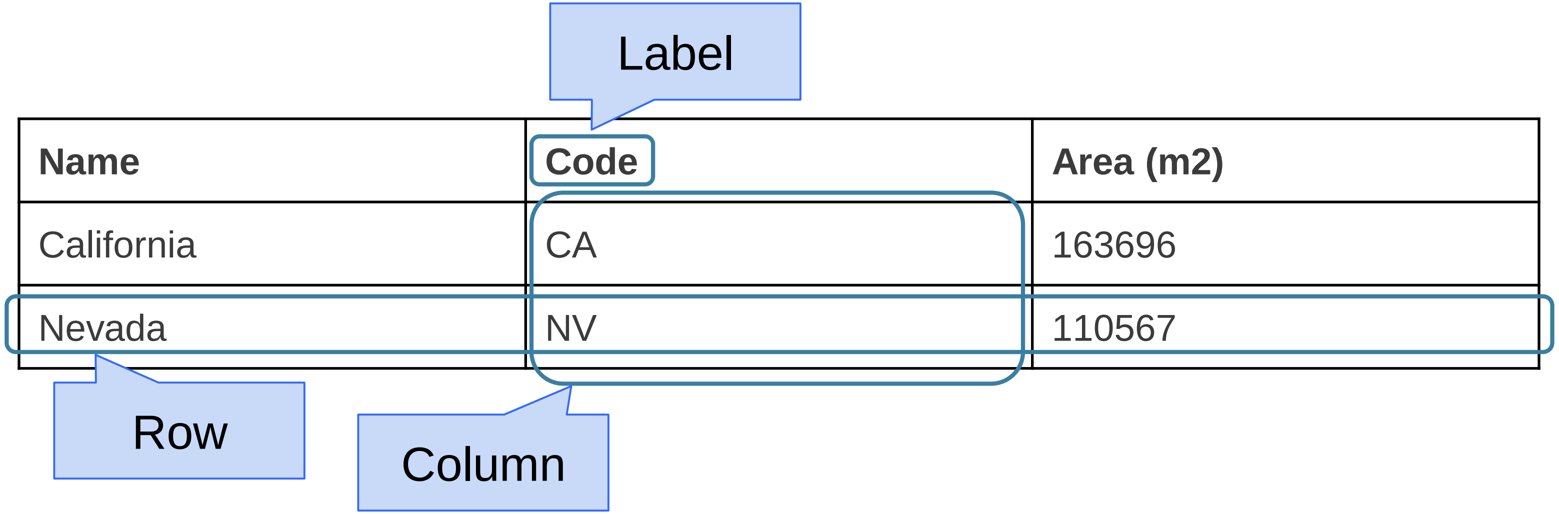
Announcements

- Please:
    - Office hours and Piazza is your first point of contact for questions about the course
    - Your section TA is your first point of contact for questions about your personal logistics

Tables

Table Structure

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



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

Table Methods

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

Manipulating Rows

- **`t.sort(column)`** sorts the rows in increasing 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
- **`t.with_row`** makes a new table that has another row

(Demo)

Discussion Questions

The table `nba` has columns `NAME`, `POSITION`, and `SALARY`.

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

```
nba.where(1, 'PG').where(2, are.above(15)).column(0)
```

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

```
nba.with_row(['Samosa', 'Mascot', 100])
nba.where('NAME', are.containing('Samo'))
```

Arithmetic

Arithmetic Operators

Operation	Operator	Example	Value
Addition	+	2 + 3	5
Subtraction	-	2 - 3	-1
Multiplication	*	2 * 3	6
Division	/	7 / 3	2.66667
Remainder	%	7 % 3	1
Exponentiation	**	2 ** 0.5	1.41421

(Demo)

Ints and Floats

Python has two numeric types

- **`int`**: an integer of any size
- **`float`:** a number with an optional fractional part

An **`int`** never has a decimal point; a **float** always does

A **`float`** might be printed using scientific notation

Three limitations of float values:

- They have limited size (but the limit is huge)
- They have limited precision of 15-16 decimal places
- After arithmetic, the final few decimal places can be wrong

Strings

Text and Strings

A string value is a snippet of text of any length

- `'a'`
- `'word’`
- `“There can be 2 sentences. Here's the second!"`

Strings that contain numbers can be converted to numbers

- `int('12')`
- `float('1.2')`

Any value can be converted to a string

- `str(5)`

(Demo)

Discussion Question

Assume you have run the following statements

```
x = 3
y = '4'
z = '5.6'
```

What's the source of the error in each example?

A. `x + y`

B. `x + int(y + z)`

C. `str(x) + int(y)`

D. `str(x, y) + z`

Arrays and Ranges

Arrays

An array contains a sequence of values

- All elements of an array should have the same type
- Arithmetic is applied to each element individually
- When two arrays are added, they must have the same size; corresponding elements are added in the result
- A column of a table is an array

(Demo)

Ranges

A range is an array of consecutive numbers

- **`np.arange(end)`**:
  An array of increasing integers from 0 up to **`end`**
- **`np.arange(start, end)`**:
  An array of increasing integers from **`start`** up to **`end`**
- **`np.arange(start, end, step)`**:
  A range with **`step`** between consecutive values

The range always includes **`start`** but excludes **`end`**

Ways to create a table

- `Table.read_table(filename)` - reads a table from a spreadsheet
- `Table()` - an empty table
- and...

Arrays → Tables

- `Table().with_column(label, data)` - creates a table with a single column; `data` is an array
- `Table().with_columns(label1, data1, ...)` - creates a table, with an array of data for each column

Table Methods

- Creating and extending tables:
  - `Table().with_columns` and `Table.read_table`
- Finding the size: `num_rows` and `num_columns`
- Referring to columns: labels, relabeling, and indices
  - `labels` and `relabeled`; 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`

Rows

Take Rows, Select Columns

The `select` method returns a table with only some columns

The `take` method returns a table with only some rows

- Rows are numbered, starting at 0
- Taking a single number returns a one-row table
- Taking a list of numbers returns a table as well

(Demo)

The where method

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

(Demo)

Manipulating Rows

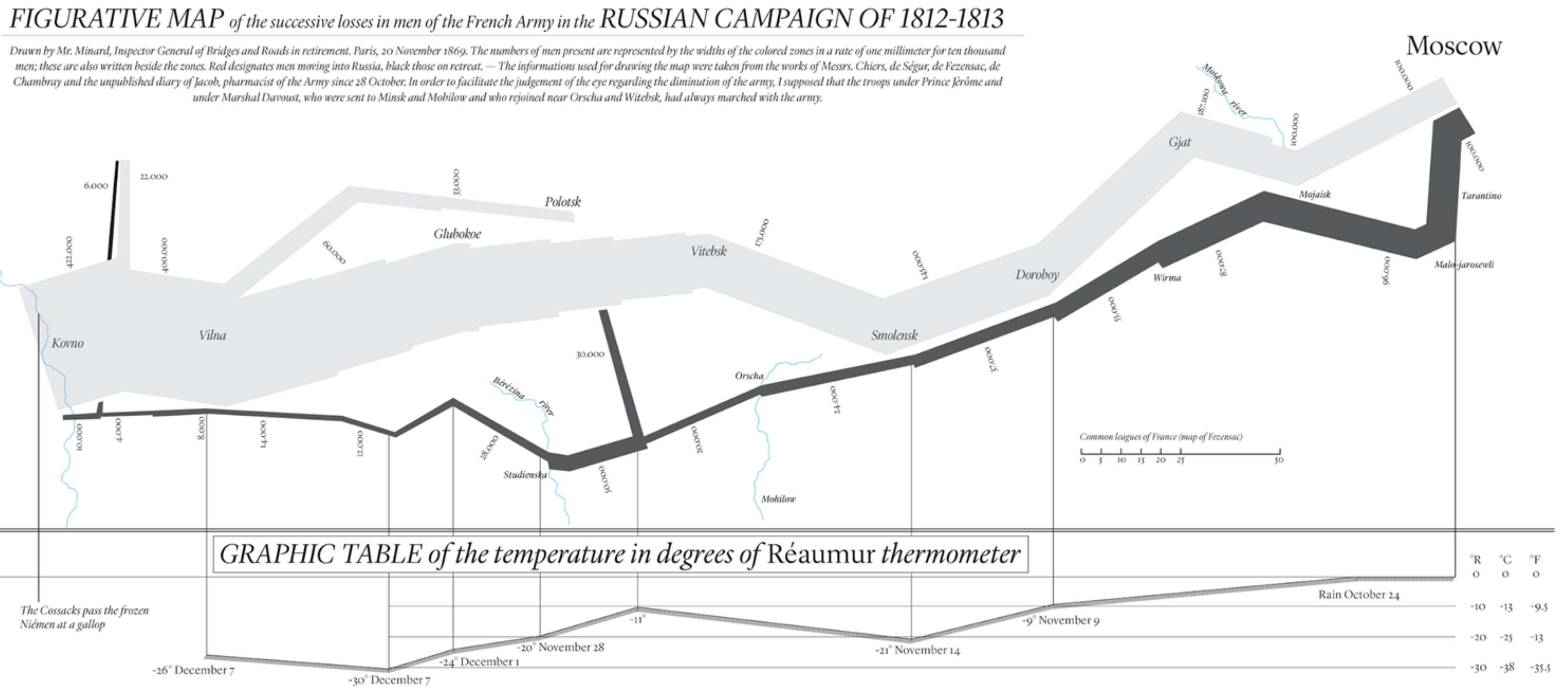
- `t.sort(column)` sorts the rows in increasing order
- `t.take(row_numbers)` keeps the numbered rows
  - Each `row` has an index, starting at 0
- `t.where(`*`column`*`, are.`*`condition`*`)` keeps all rows for which a column's value satisfies a condition
- `t.where(`*`column`*`, value)` keeps all rows containing a certain value in a column

Minard's Map

Charles Joseph Minard, 1781-1870

- French civil engineer who created one of the greatest graphs of all time
- Visualized Napoleon's 1812 invasion of Russia, including
    - the number of soldiers
    - the direction of the march
    - the latitude and longitude of each city
    - the temperature on the return journey
    - Dates in November and December

Visualization of 1812 March

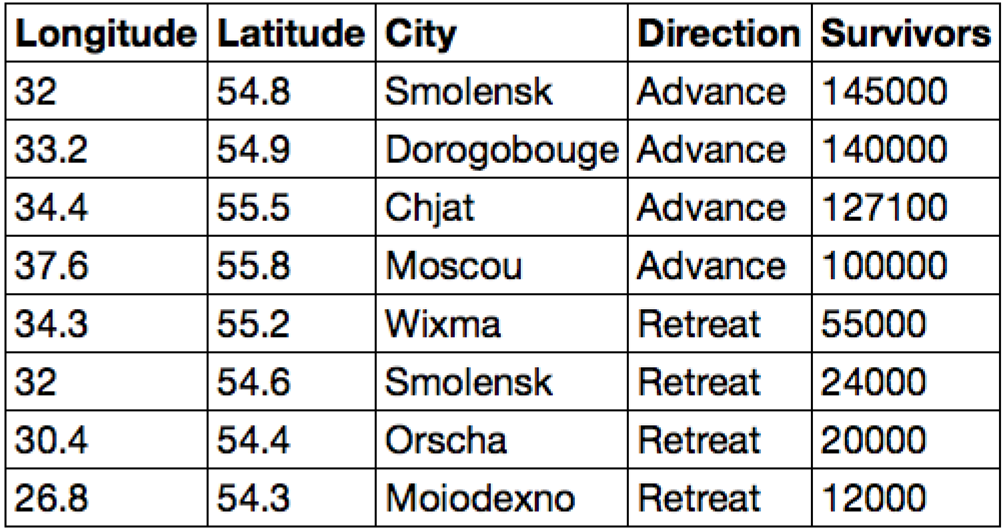


FIGURATIVE MAP *of the successive losses in men of the French Army in the* RUSSIAN CAMPAIGN OF 1812-1813

Drawn by Mr. Minard, Inspector General of Bridges and Roads in retirement. Paris, 20 November 1869. The numbers of men present are represented by the widths of the colored zones in a rate of one millimeter for ten thousand men; these are also written beside the zones. Red designates men moving into Russia, black those on retreat. — The informations used for drawing the map were taken from the works of Messrs. Chiers, de Ségur, de Fezensac, de Chambray and the unpublished diary of Jacob, pharmacist of the Army since 28 October. In order to facilitate the judgement of the eye regarding the diminution of the army, I supposed that the troops under Prince Jérôme and under Marshal Davoust, who were sent to Minsk and Mobilow and who rejoined near Orscha and Witebsk, had always marched with the army.

GRAPHIC TABLE *of the temperature in degrees of* Réaumur *thermometer*

Different types of data



Longitude	Latitude	City	Direction	Survivors
32	54.8	Smolensk	Advance	145000
33.2	54.9	Dorogobouge	Advance	140000
34.4	55.5	Chjat	Advance	127100
37.6	55.8	Moscou	Advance	100000
34.3	55.2	Wixma	Retreat	55000
32	54.6	Smolensk	Retreat	24000
30.4	54.4	Orscha	Retreat	20000
26.8	54.3	Moiodexno	Retreat	12000

float:
decimal number

string:
text

int:
integer

Lists

Lists are Generic Sequences

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

$$\texttt{[}\underline{\texttt{2+3}}\texttt{, }\underline{\texttt{'four'}}\texttt{, }\underline{\texttt{Table().with\_column('K', }\underline{\texttt{[3, 4]}}\texttt{)}}\texttt{]}$$

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

(Demo)

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 …”

Analyzing Census Data

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

(Demo)

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