CompSci 190: Tables & Graphs

Jeff Forbes

September 10, 2018

Sit in groups of 4!

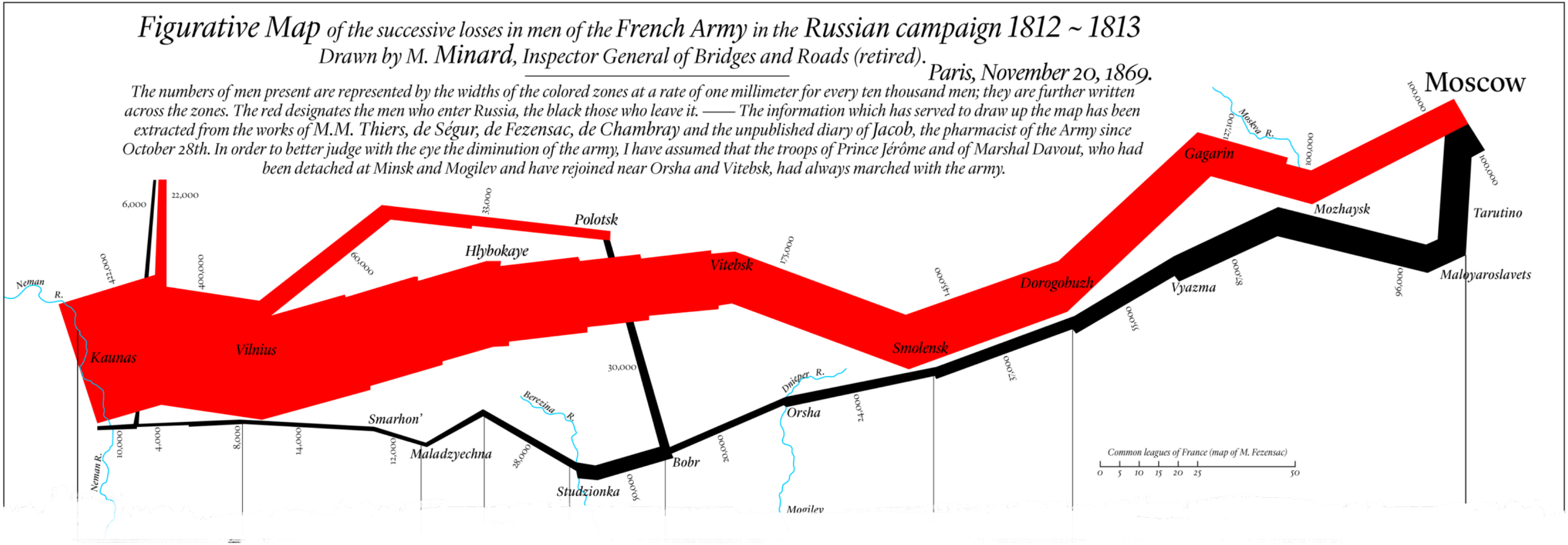
Plan For The Week (PFTW)

- Do Homework 1
- Create tables from datafiles and other sources
- Consider different methods for visualizations of data

Where are you from?

- Compute distance from your hometown* to Duke
  - https://www.distancecalculator.net
  - Your hometown is where you lived immediately prior to coming to Duke

- Go to http://okpy.org and sign up for this class
  - Choose partner for homework within your group
  - Recommend *pair programming*
    - One partner is driver – controls keyboard
    - One partner is navigator - provides **directions** and **support**
    - **Switch roles often**



Minard's Visualization

- Napoleon's 1812 invasion of Russia
  - # of soldiers
  - direction of the march
  - latitude and longitude of each city
  - temperature on the return journey
  - dates

Data in a tabular form

- What information does this visualization hide? What is more clear?

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

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

Decennial Census

- Count how many people are in the US
- U.S. Constitution. Article I, Section 2

> "Representatives and direct Taxes shall be apportioned among the several States which may be included within this Union, according to their respective Numbers . . . The actual Enumeration shall be made within three Years after the first Meeting of the Congress of the United States, and within every subsequent Term of ***ten Years***, in such Manner as they shall by Law direct."

- Census Bureau estimates the population in intervening years

Census Table

- 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. Why?
- 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

What's next?

- Read Chapters 7 of *Computational and Inferential Thinking*

- Start working on Homework 1