Sorting Senators, Collecting Data

Feb 4 2016

Warmup

- Get Google Spreadsheet from last class open!

Plan

- Pick liberal senator, Senator L
- Compare others to Senator L to determine liberalness

Problem

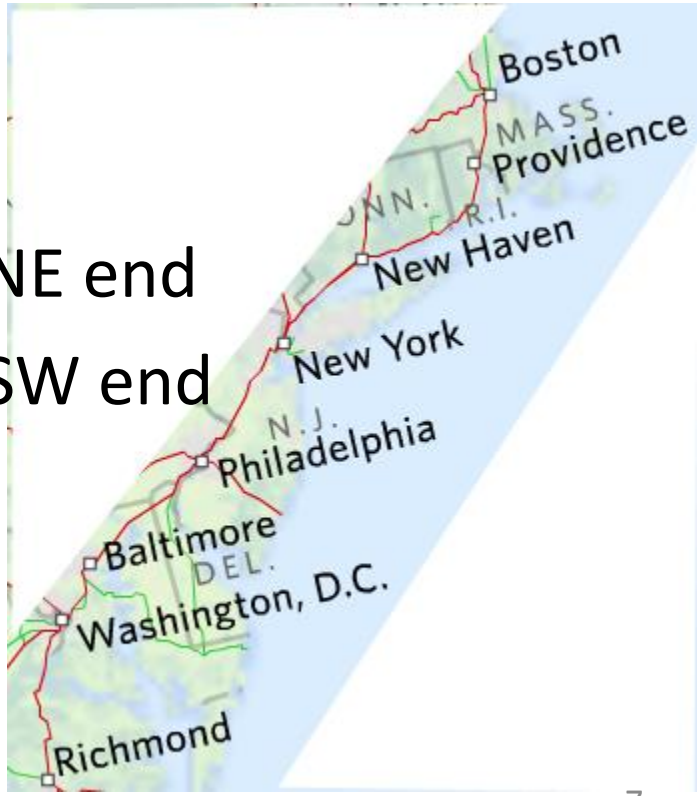
- What if senator L isn't the most liberal?
  - Even those more liberal will be rated as some distance from senator L, and hence appear more conservative!

Slight improvement

- Pick liberal Senator L, and conservative Senator C.
- Compare other senators to both of these
- Now a senator more liberal than L will not only be distant from L, but *more* distant from C than L is

Analogous problem

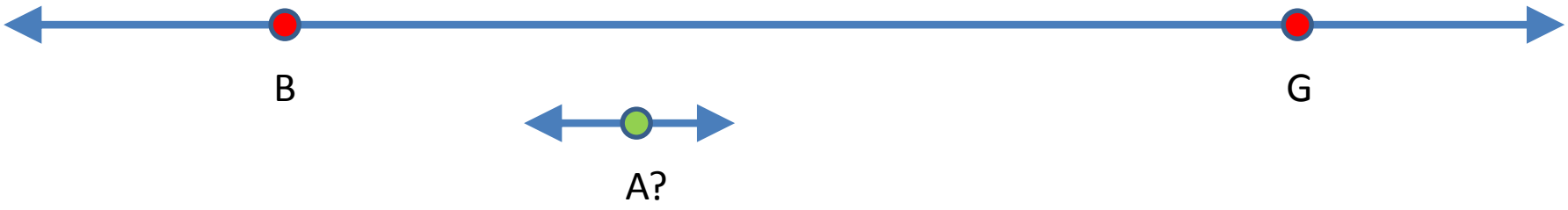
- Put the stations on Amtrak's Northeast Corridor in order
- You're given only
  - Distances between stations
  - An example station near the NE end
  - An example station near the SW end



Distance table

	A	B (SW)	C	D	E	F	G (NE)	H
A	0	180	300	220	200	80	165	100
B		0	120	40	380	260	345	80
C			0	80	500	380	465	200
D				0	420	300	385	120
E					0	120	35	300
F						0	85	180
G							0	265
H								0

	A	B (SW)	C	D	E	F	G (NE)	H
A	0	180	300	220	200	80	165	100
B		0	120	40	380	260	345	80
C			0	80	500	380	465	200
D				0	420	300	385	120
E					0	120	35	300
F						0	85	180
G							0	265
H								0



	A	B (SW)	C	D	E	F	G (NE)	H
A	0	180	300	220	200	80	165	100
B		0	120	40	380	260	345	80
C			0	80	500	380	465	200
D				0	420	300	385	120
E					0	120	35	300
F						0	85	180
G							0	265
H								0

Because BG distance is 345, BA = 180, and AG=165, A must be between them!

What about D?

Spend a minute trying to figure that out

	A	B (SW)	C	D	E	F	G (NE)	H
A (NY)	0	180	300	220	200	80	165	100
B (Balt)		0	120	40	380	260	345	80
C (Richmond)			0	80	500	380	465	200
D (DC)				0	420	300	385	120
E (BOS)					0	120	35	300
F (New Haven)						0	85	180
G (Prov)							0	265
H (Phila.)								0

Conclusion?

- Example suggests that we need not pick the **most** liberal or **most** conservative senator to do our ranking
- We can use comparisons to find senators "further out"

- Tonight's homework will suggest otherwise ☹
  - Don't worry: we need to compare them anyway!

Collecting Data

- Last class we showed you XML file structure
- Talked briefly about CSV ("comma separated values") file structure
- Had you load a CSV file
- Let's have some further info about loading XML

Getting at the contents of an XML file

Structure:

```
<roll_call_vote>
  <congress>113</congress>
  <session>2</session>
  <congress_year>2014</congress_year>
  <vote_number>8</vote_number>
  <vote_date>January 14, 2014, 03:22 PM</vote_date>
  <modify_date>January 14, 2014, 04:01 PM</modify_date>
  <vote_question_text>On the Motion to Table S. 1845</vote_question_text>
  <vote_document_text>
A bill to provide for the extension of certain unemployment benefits, and for
other purposes.
  </vote_document_text>
  <vote_result_text>Motion to Table Failed (45-55)</vote_result_text>
  <question>On the Motion to Table</question>
  <vote_title>
Motion to Table the Motion to Commit S. 1845 to the Committee on Finance with
Instructions
  </vote_title>
  <majority_requirement>1/2</majority_requirement>
  <vote_result>Motion to Table Failed</vote_result>
 ...
```

Interpretation: A "roll call vote" contains a "congress", a "session", a "vote number", and many other entities

```
<roll_call_vote>
  ...
  <count>
    <yeas>45</yeas>
    <nays>55</nays>
    <present/>
    <absent/>
  </count>
  <tie_breaker>
    <by_whom/>
    <tie_breaker_vote/>
  </tie_breaker>
  <members>
    <member>
      <member_full>Alexander (R-TN)</member_full>
      <last_name>Alexander</last_name>
      <first_name>Lamar</first_name>
      <party>R</party>
      <state>TN</state>
      <vote_cast>Yea</vote_cast>
       ...
```

Interpretation: A "roll call vote" contains a "members", which is itself a container, containing many "member"s. A "path" to senator Alexander's first name could be written

```
roll_call_vote/members/member/first_name
```

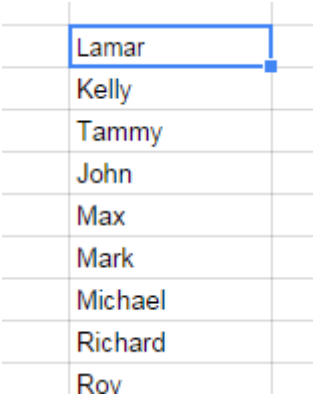
...but this would also be a path to any other senator's first name

XPath

- Listing “tags” separated by slashes is an instance of an “Xpath”, which is a standard for describing locations of data in an XML file.
- Google’s importXML uses this.

Example of importXML

=importXML("http://www.senate.gov/legislative/LIS/roll_call_votes/vote1132/vote_113_2_00008.xml", "//members/member/first_name")

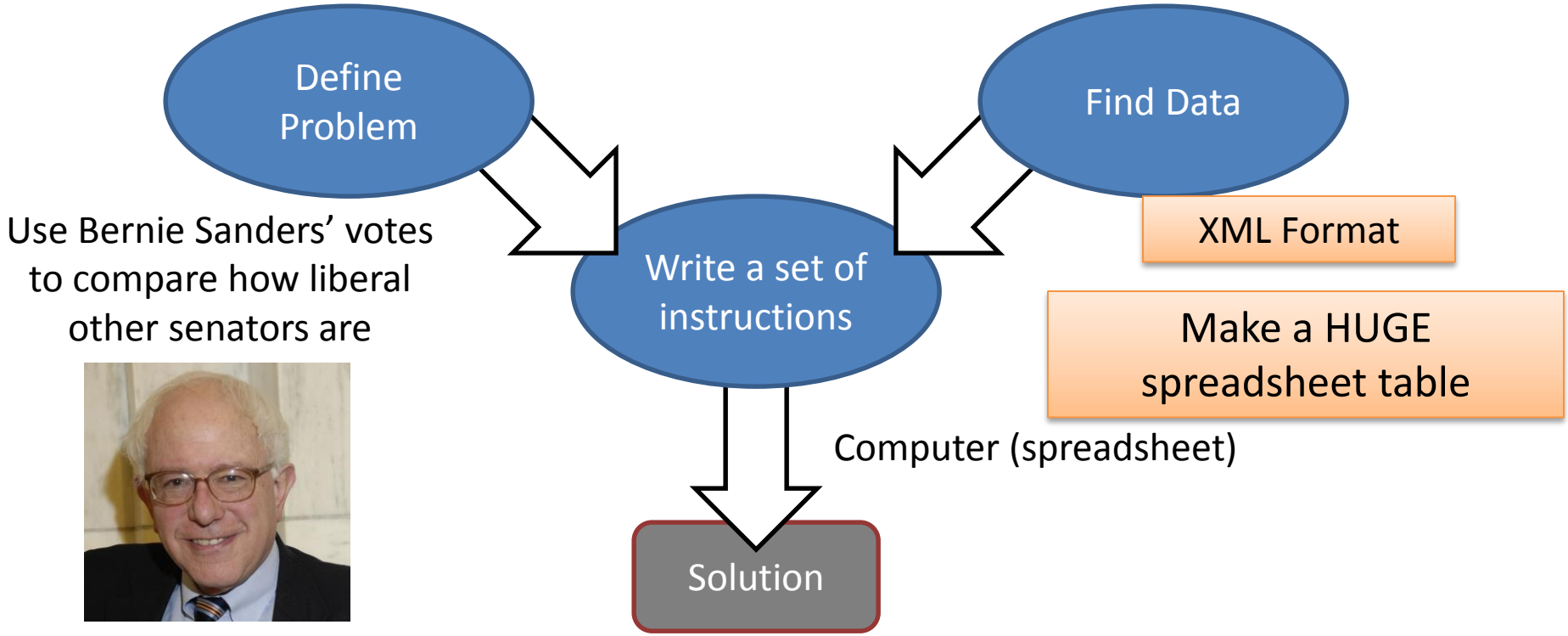


- The URL for the XML file: http://www.senate.gov/...008.xml
- The Xpath search string: "//members/member/first_name"

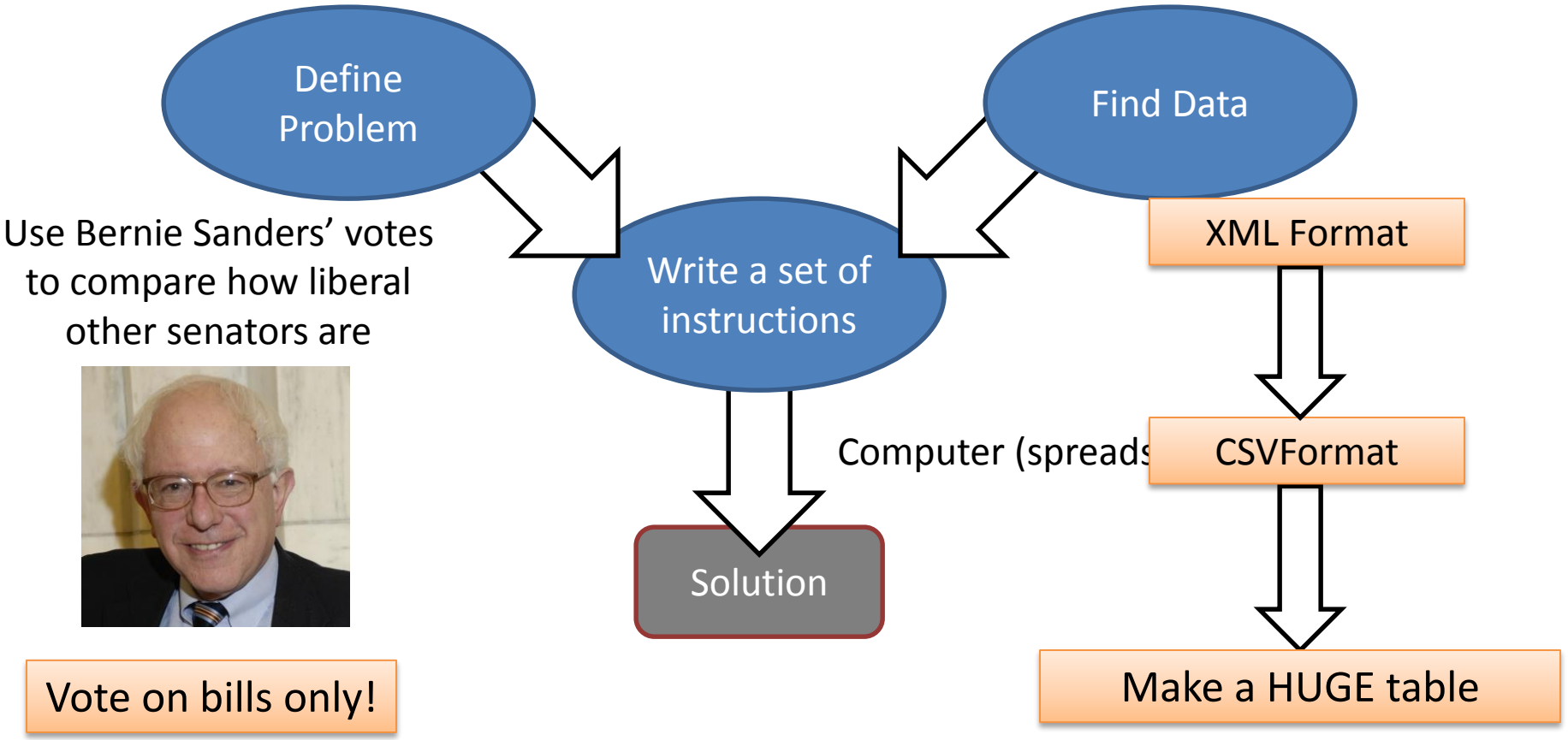
Meaning of Xpath String

- "//members/member/first_name”
- "//members/member/first_name”
  - Means “any path at all can go here”
  - Full path would be “/roll_call_vote/members/member/first_name”
  - Alternative short form that works for this doc: "//first_name”
- Different form: “/roll_call_vote/*/*/first_name”
- Any “first_name” that’s a great-grandchild of the roll_call_vote. (“*” means “replace with any one item”)
- Many fancier forms available...if you need them.

So Far


Define Problem
Find Data
Use Bernie Sanders' votes to compare how liberal other senators are
Write a set of instructions
XML Format
Make a HUGE spreadsheet table
Computer (spreadsheet)
Solution

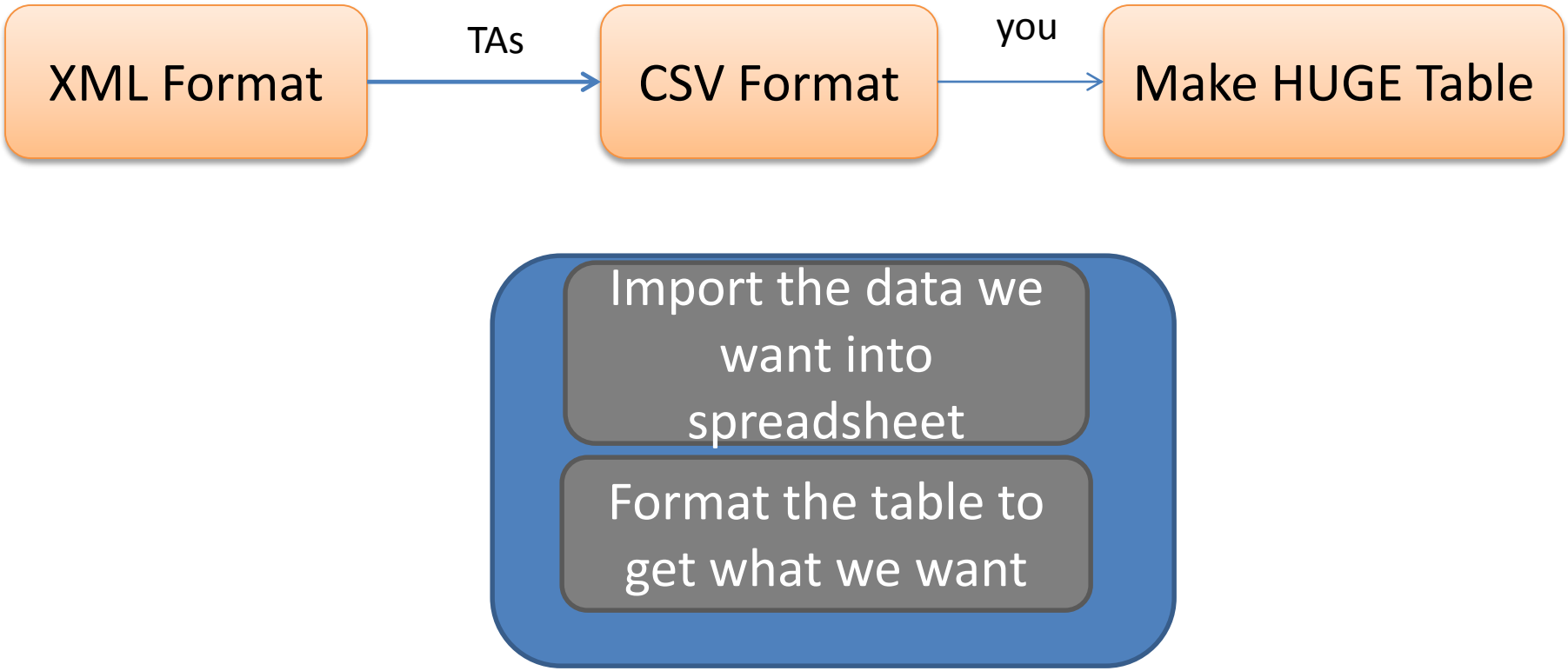
So Far


Define Problem
Find Data
Use Bernie Sanders' votes to compare how liberal other senators are
Write a set of instructions
XML Format
Computer (spreads
CSVFormat
Solution
Vote on bills only!
Make a HUGE table

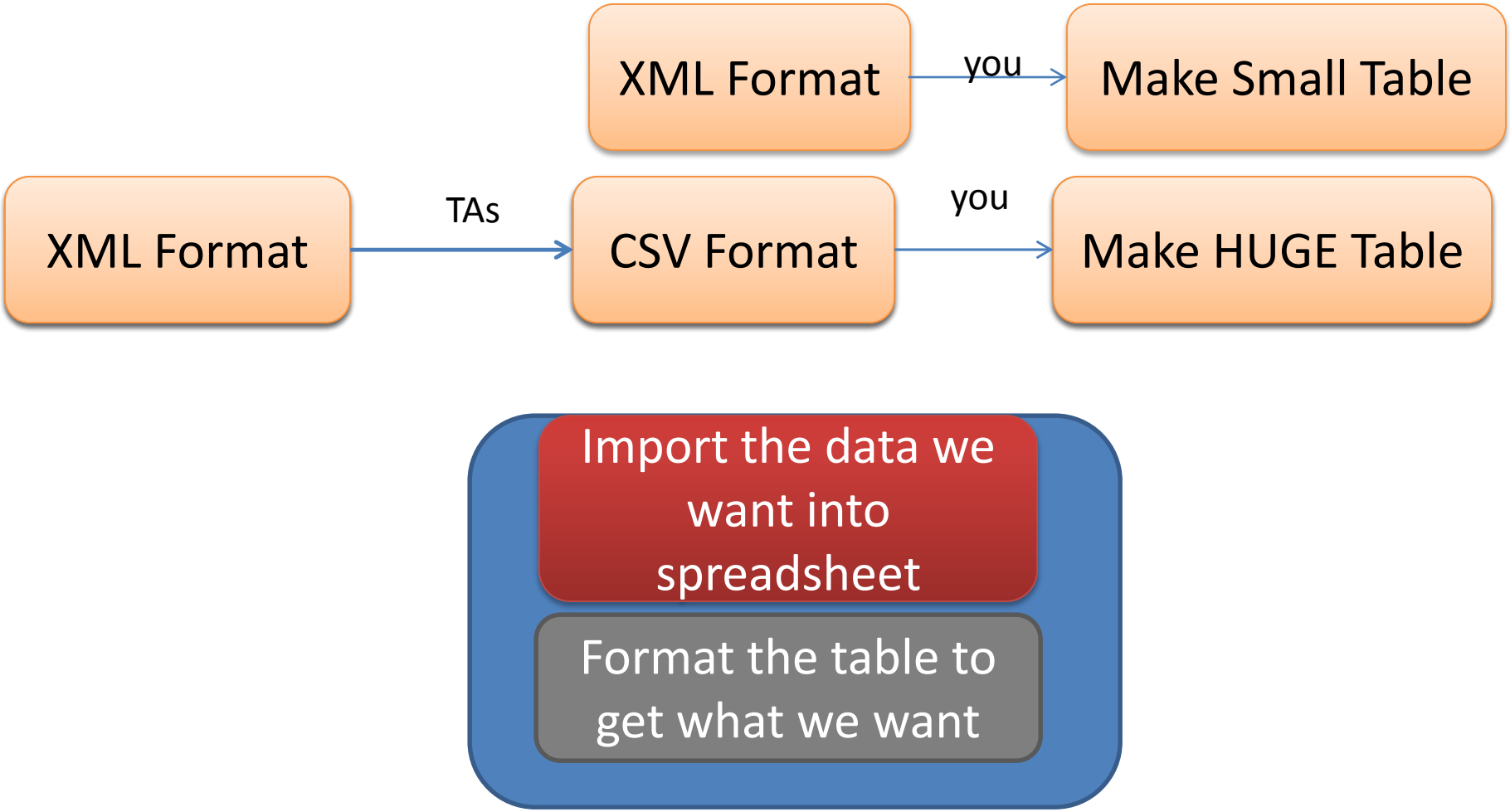
This is going to be a lab day

- Ask for help/clarification *at any point*.

Soon you'll have a big (but not so big) table of votes



So far, we've done this...



A note on names

- We call files by their file *extension*: an XML file ends with `.xml`, a CSV file ends with `.csv`, etc.

Why?

Why?

We’re learning how to gather data off the web, then format into something we can work with.

`Ctrl (Command on Mac)` is your friend

Bottommost Cell: `Ctrl` and ↓

Topmost Cell: `Ctrl` and ↑

Rightmost Cell: `Ctrl` and →

Leftmost Cell: `Ctrl` and ←

Pressing `Ctrl` selects each cell you click

`Shift` is your friend too: Pressing `Shift` selects all cells between clicks.

Pressing `Shift` and using arrow keys selects blocks

Activity 1-1

- Proceed from wherever you got to during the last class
- Hint for task 3 (formatting data)

Tip: Press ‘Ctrl’ and an arrow to go ALL THE WAY to the beginning/end of a row/column.

Tip: To get back to original sort order: Sort by both ‘session’ and then by ‘vote_number’

Look at your spreadsheet

1. Open the spreadsheet.
2. You should have 2,401 rows (Task 3.7)
3. You should have columns through E.

Task 3.8

- We want a unique identifier for the vote of each bill in this congress.
  - Which two columns together make a unique key?

Task 3.9

- Add another column to the table by entering a `vote_id` column in cell F1.
- Write a formula to output `session:vote_number` values for this row.
- Use fill down or copy/paste, if necessary, to apply this formula to all the other rows.

Task 3.10

- Add a `numerical_vote` column in cell G1.
- Write a formula to output:
  - `1` if the senator voted `Nay`
  - `2` if the senator voted `Yea`
  - `0` otherwise