“I Got 99 Computational Problems and a Bit ain’t one”
— Jay Z
FIRST: SOME REMINDERS

1. Start bringing your laptops to class (for “Lab Time” session). If you don’t have a laptop, CIS can lend you one. Talk to us for details.

2. When e-mailing the course staff, use our emails (unless it’s a question for a specific individual, like a grading issue):
   
   1. cs0030tas@lists.brown.edu
   
   2. cs0030headtas@lists.brown.edu

3. Join Piazza for course announcements / questions (link available for our course website)
Lecture 2

- Problem Solving Workflow
- Our Data
Lecture 2

- Problem Solving Workflow
- Our Data
Worked Example

Where do your legislators exist on a political scale?

Sheldon Whitehouse  Jack Reed

Conservative  Liberal
Problem Solving Workflow

1. Define question
2. Select appropriate dataset
3. Design a valid method
4. Analyze your results
5. Communicate your findings
Problem Solving Workflow

1. Define question
2. Select appropriate dataset
3. Design a valid method
4. Analyze your results
5. Communicate your findings
Problem Solving Workflow: Define Question

- Tradeoffs exist between data availability and questions you might want to answer.

- Ask yourself, what data is available to answer your question(s)?

Sheldon Whitehouse  
Jack Reed

Conservative  
Liberal
Conservative/Liberal scales

- **ontheissues.org**: Manually inspect politicians votes and statements to find their opinion on key issues

- **thatsmycongress.com**: Label bills and amendments liberal or conservative, then look at voting patterns

- **Esquire’s Warren-Cruz scale**: Label issues liberal or conservative, then survey people
### Conservative/Liberal scales

**On Social Issues**
- Abortion is a woman’s unrestricted right: Hillary Clinton strongly favors, Donald Trump opposes
- Legally require hiring women & minorities: Hillary Clinton strongly favors, Donald Trump opposes
- Comfortable with same-sex marriage: Hillary Clinton strongly favors, Donald Trump opposes
- Keep God in the public sphere: Hillary Clinton strongly favors, Donald Trump opposes
- Vouchers for school choice: Hillary Clinton strongly opposes, Donald Trump supports

**On Domestic Issues**
- Expand ObamaCare: Hillary Clinton strongly favors, Donald Trump opposes
- EPA regulations are too restrictive: Hillary Clinton strongly opposes, Donald Trump supports
- Stricter punishment reduces crime: Hillary Clinton opposes, Donald Trump supports
- Absolute right to gun ownership: Hillary Clinton opposes, Donald Trump supports
- Marijuana is a gateway drug: Hillary Clinton supports, Donald Trump opposes

**On Economic Issues**
- Privatize Social Security: Hillary Clinton strongly opposes, Donald Trump supports
- Higher taxes on the wealthy: Hillary Clinton strongly favors, Donald Trump supports
- Stricter limits on political campaign funds: Hillary Clinton strongly favors, Donald Trump supports
- Stimulus better than market-led recovery: Hillary Clinton strongly favors, Donald Trump supports
- Prioritize green energy: Hillary Clinton supports, Donald Trump opposes

**On International Issues**
- Pathway to citizenship for illegal aliens: Hillary Clinton favors, Donald Trump opposes
- Support & expand free trade: Hillary Clinton opposes, Donald Trump favors
- Support American Exceptionalism: Hillary Clinton mixed opinion, Donald Trump favors
- Expand the military: Hillary Clinton opposes, Donald Trump supports
- Avoid foreign entanglements: Hillary Clinton supports, Donald Trump supports
Conservative/Liberal scales

- **ontheissues.org**: Manually inspect politicians votes and statements to find their opinion on key issues.

- **thatsmycongress.com**: Label bills and amendments liberal or conservative, then look at voting patterns.

- **Esquire’s Warren-Cruz scale**: Label issues liberal or conservative, then survey people.
Problem Solving Workflow: Define Question

Detail the Specifics

- Choose a legislative body. We’ll pick U.S. Senate
- What behavior should we evaluate? e.g.
  - Voting
  - Sponsorship/ co-sponsorship
  - Public speeches
  - Supporting donors
- Look for loaded words that are difficult to be precise about
  - Issues arise in defining how to measure conservative vs liberal
  - Labels are heavily context dependent; may hold diff stances on diff issues.
Where do your legislators exist on a political scale?

- Sheldon Whitehouse
- Jack Reed
- Jim Inhofe
- Elizabeth Warren
Problem Solving Workflow: Define Question

1. Define question
2. Select appropriate dataset
3. Design a valid method
4. Analyze your results
5. Communicate your findings

Problem Solving Workflow

Our Data
Problem Solving Workflow: Select Dataset

Problem Solving Workflow

- Define question
- Select appropriate dataset
- Design a valid method
- Analyze your results
- Communicate your findings

Our Data
Problem Solving Workflow: Select Dataset

- News articles
- Stated policies on the Senator’s webpages
- Look at Congressional records:
  - voting
  - sponsorship
  - co-sponsorship
- Speeches during Senate sessions
Problem Solving Workflow: Select Dataset

News Articles

• Report information directly attained from elsewhere; there might be a better source

• Are heavily focused on the most popular politicians

• Sussing out political leanings requires a complex understanding of the issues

• Nunberg’s analysis found that conservative/liberal labels close to names helped identify on which side they fell, but difficult to understand the degree
• Report information directly attained from elsewhere; there

In the six years when Ms. Abrams led the Democrats in the State House, though, she proved to be a much more complex political character than the portraits now painted by either her most outspoken adversaries or her most fervent fans.

New York Times, August 19, 2018

• Nunberg’s analysis found that conservative/liberal labels close to names helped identify on which side they fell, but difficult to understand the degree
Problem Solving Workflow: Select Dataset

Congressional Records

• Available at www.congress.gov

• Each bill, passed or not, is available with sponsors and cosponsors.

• Voting:

   Issues put to a vote are limited by the issues considered by the majority party and the House

• Sponsorship:

   Very few senators sponsor bills that get very far in the process. Difficult to compare alignment

• Co-sponsorship:

   Like speeches, not all senators sponsor bills at the same frequency.
### Problem Solving Workflow: Select Dataset

#### Congressional Records

**Grouped By Vote Position**

<table>
<thead>
<tr>
<th>YEAs ---81</th>
<th>NAYS ---17</th>
<th>Not Voting - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barrasso (R-WY)</td>
<td>Flake (R-AZ)</td>
<td>Leahy (D-VT)</td>
</tr>
<tr>
<td>Bennet (D-CO)</td>
<td>Franken (D-MN)</td>
<td>Markey (D-MA)</td>
</tr>
<tr>
<td>Blunt (R-MO)</td>
<td>Gardner (R-CO)</td>
<td>Merkley (D-OR)</td>
</tr>
<tr>
<td>Boozman (R-AR)</td>
<td>Graham (R-SC)</td>
<td>Murphy (D-CT)</td>
</tr>
<tr>
<td>Brown (D-OH)</td>
<td>Grassley (R-IA)</td>
<td>Murray (D-WA)</td>
</tr>
<tr>
<td>Burr (R-NC)</td>
<td>Harris (D-CA)</td>
<td>Sanders (I-VT)</td>
</tr>
<tr>
<td>Cantwell (D-WA)</td>
<td>Hassan (D-NH)</td>
<td></td>
</tr>
<tr>
<td>Capito (R-WV)</td>
<td>Hatch (R-UT)</td>
<td></td>
</tr>
<tr>
<td>Cardin (D-MD)</td>
<td>Heinrich (D-NM)</td>
<td></td>
</tr>
<tr>
<td>Carper (D-DE)</td>
<td>Heitkamp (D-ND)</td>
<td></td>
</tr>
<tr>
<td>Casey (D-PA)</td>
<td>Heller (R-NV)</td>
<td></td>
</tr>
<tr>
<td>Cassidy (R-LA)</td>
<td>Hirono (D-HI)</td>
<td></td>
</tr>
<tr>
<td>Cochran (R-MS)</td>
<td>Hoeven (R-ND)</td>
<td></td>
</tr>
<tr>
<td>Collins (R-ME)</td>
<td>Inhofe (R-OK)</td>
<td></td>
</tr>
<tr>
<td>Coons (D-DE)</td>
<td>Isakson (R-GA)</td>
<td></td>
</tr>
<tr>
<td>Corker (R-TN)</td>
<td>Johnson (R-WI)</td>
<td></td>
</tr>
<tr>
<td>Cornyn (R-TX)</td>
<td>Kaine (D-VA)</td>
<td></td>
</tr>
<tr>
<td>Cortez Masto (D-NV)</td>
<td>Kennedy (R-LA)</td>
<td></td>
</tr>
<tr>
<td>Cotton (R-AR)</td>
<td>King (I-ME)</td>
<td></td>
</tr>
<tr>
<td>Crapo (R-ID)</td>
<td>Klobuchar (D-MN)</td>
<td></td>
</tr>
<tr>
<td>Cruz (R-TX)</td>
<td>Lankford (R-OK)</td>
<td></td>
</tr>
<tr>
<td>Daines (R-MT)</td>
<td>Lee (R-UT)</td>
<td></td>
</tr>
<tr>
<td>Donnelly (D-IN)</td>
<td>Manchin (D-WV)</td>
<td></td>
</tr>
<tr>
<td>Enzi (R-WY)</td>
<td>McCain (R-AZ)</td>
<td></td>
</tr>
<tr>
<td>Ernst (R-IA)</td>
<td>McCaskill (D-MO)</td>
<td></td>
</tr>
<tr>
<td>Feinsteint (D-CA)</td>
<td>McConnell (R-KY)</td>
<td></td>
</tr>
<tr>
<td>Fischer (R-NE)</td>
<td>Menendez (D-NJ)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Young (R-IA)</td>
<td></td>
</tr>
</tbody>
</table>

YEAs: Yea Votes, NAYs: Nay Votes, Not Voting: Absent from Vote
Problem Solving Workflow: Select Dataset

Speeches

• Speech transcripts are available at www.congress.gov/

• Not all senators get equal speech time on the floor, so there is a bias toward the more verbose

• We can analyze text using techniques we’ll learn later in the semester

• However, text can be tricky even for humans, much less computers.
However, text can be tricky even for humans, much less computers.

Can you discern if Senator Durbin is a Republican or Democrat from the following text?
Speeches

• However, text can be tricky even for humans, much less computers.

Sen. DURBIN. "If we stick to the basic principles of the Affordable Care Act, or ObamaCare, we run into some problems in a hurry. The first basic principle accepted by President-Elect Trump is that we want to make sure that no health insurance company can ever discriminate against you or your family because of a preexisting condition—a baby born with cancer, a child with diabetes, a spouse who survives a cancer scare. In the old days before ObamaCare, that meant that you either were disqualified from insurance for your family or you couldn’t afford it. So we said as part of the Affordable Care Act: No more—they cannot discriminate against those who are less than perfect when it comes to health because so many of us are less than perfect. OK, my friends in the Grand Old Party, how are you going to deal with that? How are we going to make sure that every family is protected with their health insurance plan? We haven’t heard a word"
Let’s choose to look at the voting record

Our question then becomes: which senators agree or disagree with Senator Warren, based on their voting record?
Problem Solving Workflow: Select Dataset

1. Define question
2. Select appropriate dataset
3. Design a valid method
4. Analyze your results
5. Communicate your findings
Problem Solving Workflow: Design Method

- Problem Solving Workflow
- Our Data

1. Define question
2. Select appropriate dataset
3. Design a valid method
4. Analyze your results
5. Communicate your findings
Redefined Question and Dataset

• Let’s choose to look at the voting record

• Our question then becomes: which senators agree or disagree with Senator Warren, based on their voting record?
Problem Solving Workflow: Design Method

- To compute our spectrum, we choose one senator to align to complete disagreement and complete agreement.
Problem Solving Workflow: Design Method

- To compute our spectrum, we choose one senator to align to

- For each other senator, we want to compute a score quantifying their voting alignment
Problem Solving Workflow: Design Method

• To compute our spectrum, we choose one senator to align to

• For each other senator, we want to compute a score quantifying their voting alignment
Problem Solving Workflow: Design Method

• To compute our spectrum, we choose one senator to align to

• For each other senator, we want to compute a score quantifying their voting alignment

Sheldon Whitehouse

Elizabeth Warren

complete disagreement

complete agreement
Problem Solving Workflow: Design Method

- To compute our spectrum, we choose one senator to align to

- For each other senator, we want to compute a score quantifying their voting alignment

Our Data

<table>
<thead>
<tr>
<th></th>
<th>Bill #1</th>
<th>Resolution #1</th>
<th>Bill #2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Warren</td>
<td>Yay</td>
<td>Nay</td>
<td>Yay</td>
</tr>
<tr>
<td>Whitehouse</td>
<td>Nay</td>
<td>Nay</td>
<td>Nay</td>
</tr>
<tr>
<td>Reed</td>
<td>Nay</td>
<td>Nay</td>
<td>Yay</td>
</tr>
</tbody>
</table>
Problem Solving Workflow: Design Method

- To compute our spectrum, we choose one senator to align to

- For each other senator, we want to compute a score quantifying their voting alignment

- Computation always operates on inputs and produces an output
Problem Solving Workflow: Design Method

- To compute our spectrum, we choose one senator to align to
- For each other senator, we want to compute a score quantifying their voting alignment
- Computation always operates on inputs and produces an output

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voting record</td>
<td>Alignment Score</td>
</tr>
<tr>
<td>of two senators</td>
<td>Score</td>
</tr>
</tbody>
</table>
Problem Solving Workflow: Design Method

- To compute our spectrum, we choose one senator to align to

- For each other senator, we want to compute a score quantifying their voting alignment

- Computation always operates on inputs and produces an output

Voting record of two senators → ? → Alignment Score

Input

Output
Problem Solving Workflow: Design Method

- To compute our spectrum, we choose one senator to align to.
- For each other senator, we want to compute a score quantifying their voting alignment.
- Computation always operates on inputs and produces an output.

Voting record of two senators \[\rightarrow\] Function \[\rightarrow\] Alignment Score

**Input** \[\rightarrow\] **Function** \[\rightarrow\] **Output**
Problem Solving Workflow: Design Method

- Determine your input(s): aka the Domain
  - Yays or Nays for each vote

- Determine your output(s): aka the Range
  - What value should we use for complete disagreement?
    Complete agreement? No overlapping voting info?
**Problem Solving Workflow: Design Method**

1. **Determine your input(s): aka the Domain**
   - Yays or Nays for each vote

2. **Determine your output(s): aka the Range**
   - What value should we use for complete disagreement?
     - Complete agreement? No overlapping voting info?

---

### Voting record of two senators

<table>
<thead>
<tr>
<th></th>
<th>Whitehouse</th>
<th>Warren</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bill 1</strong></td>
<td>Yay</td>
<td>Yay</td>
</tr>
<tr>
<td><strong>Res1</strong></td>
<td>Nay</td>
<td>Nay</td>
</tr>
<tr>
<td><strong>Bill 2</strong></td>
<td>Nay</td>
<td>Yay</td>
</tr>
<tr>
<td><strong>Bill 3</strong></td>
<td>Yay</td>
<td>Yay</td>
</tr>
</tbody>
</table>

---

**Input**

**Function**

**Output**

**Alignment Score**
Problem Solving Workflow: Design Method

• Determine your input(s): aka the Domain
  - Yays or Nays for each vote

• Determine your output(s): aka the Range
  - What value should we use for complete disagreement?
    Complete agreement? No overlapping voting info?

Voting record of two senators  →  Alignment Score
Input → Function → Output
Problem Solving Workflow: Design Method

- Determine your input(s): aka the Domain
  - Yays or Nays for each vote

- Determine your output(s): aka the Range
  - What value should we use for complete disagreement?
  - Complete agreement? No overlapping votes?

Voting record of two senators

Alignment Score

value between -1 and 1
Problem Solving Workflow: Design Method

- Determine your input(s): aka the Domain
  - Yays or Nays for each vote

- Determine your output(s): aka the Range
  - What value should we use for complete disagreement?
    Complete agreement? No overlapping voting info?

Voting record of two senators  Function  Alignment Score
Input  Output
Problem Solving Workflow: Design Method

- Determine your input(s): aka the Domain
  - Yays or Nays for each vote

- Determine your output(s): aka the Range
  - What value should we use for complete disagreement?

  Complete agreement? No overlapping voting info?

Just like in Algebra:

\[ f(x) = 2x + 3 \]
Problem Solving Workflow: Design Method

- Determine your input(s): aka the Domain
  - Yays or Nays for each vote

- Determine your output(s): aka the Range
  - What value should we use for complete disagreement?
  
    Complete agreement? No overlapping voting info?

Just like in Algebra:

\[ f(x) = 2x + 3 \]

Given input, such as 1, produce a value, such as 5
Problem Solving Workflow: Design Method

- Determine your input(s): aka the Domain
  - Yays or Nays for each vote

- Determine your output(s): aka the Range
  - What value should we use for complete disagreement?
  - Complete agreement? No overlapping voting info?

Just like in Algebra:

\[ f(x) = 2x + 3 \]

Given input, such as 1, produce a value, such as 5

Voting record of two senators

Input → Function → Alignment Score → Output
Problem Solving Workflow: Design Method

- Determine your input(s): aka the Domain
  - Yays or Nays for each vote

- Determine your output(s): aka the Range
  - What value should we use for complete disagreement?
    - Complete agreement? No overlapping voting info?

Just like in Algebra:
\[ f(x) = 2x + 3 \]

Given input, such as 1, produce a value, such as 5

This is the crux of the entire class.

Voting record of two senators

Input

Function

Alignment Score

Output
Problem Solving Workflow: Design Method

- Determine your input(s): aka the Domain
  - Yays or Nays for each vote

- Determine your output(s): aka the Range
  - What value should we use for complete disagreement?
    - Complete agreement? No overlapping voting info?

Just like in Algebra:
\[ f(x) = 2x + 3 \]

Given input, such as 1, produce a value, such as 5

This is the crux of the entire class.

You get to make your own magic!
Problem Solving Workflow: Design Method

• Determine your input(s): aka the Domain
  - Yays or Nays for each vote

• Determine your output(s): aka the Range
  - What value should we use for complete disagreement?
    Complete agreement? No overlapping voting info?
Converting Yays and Nays to Values

- Want to convert a non-numerical value into a number, what can we do?
- Our problem stresses agreement and disagreement, we need to convert yays/nays into a score based on agreement
- We can calculate the number of agreements and disagreements
Converting Yays and Nays to Values

• How should the output value change with respect to changes of your inputs?

\[ f(A, D) \rightarrow [-1, 1] \]
Problem Solving Workflow: Design Method

Converting Yays and Nays to Values

• How should the output value change with respect to changes of your inputs?

  • As agreements increase, the output value should *increase*.

  \[
  f(A, D) \rightarrow [-1, 1]
  \]

  \[f(100, 0) > f(50, 50)\]
Converting Yays and Nays to Values

- How should the output value change with respect to changes of your inputs?
- As agreements increase, the output value should increase
  - Agreements should be positive in our function

\[ f(A, D) \rightarrow [-1, 1] \]
\[ f(100, 0) > f(50, 50) \]
\[ f(A, D) = +A \cdot \cdot \cdot \]
Problem Solving Workflow: Design Method

Converting Yays and Nays to Values

• How should the output value change with respect to changes of your inputs?
  • As agreements increase, the output value should increase
  • Agreements should be positive in our function
  • As disagreements increase, the output value should decrease

\[
f(A, D) \rightarrow [-1, 1]
\]

\[
f(100, 0) > f(50, 50)
\]

\[
f(A, D) = +A \cdots
\]

\[
f(0, 100) < f(50, 50)
\]
Converting Yays and Nays to Values

• How should the output value change with respect to changes of your inputs?
  • As agreements increase, the output value should increase
    • Agreements should be positive in our function
  • As disagreements increase, the output value should decrease
    • Disagreements should be negative

\[ f(A, D) \rightarrow [-1, 1] \]
\[ f(100, 0) > f(50, 50) \]
\[ f(A, D) = +A \cdots \]
\[ f(0, 100) < f(50, 50) \]
\[ f(A, D) = -D \cdots \]
Problem Solving Workflow: Design Method

Converting Yays and Nays to Values

• How should the output value change with respect to changes of your inputs?

  • As agreements increase, the output value should increase
    • Agreements should be positive in our function
  • As disagreements increase, the output value should decrease
    • Disagreements should be negative
  • Equal agreements/disagreements -> 0.0

\[ f(A, D) \rightarrow [-1, 1] \]
\[ f(100, 0) > f(50, 50) \]
\[ f(A, D) = +A \cdots \]
\[ f(0, 100) < f(50, 50) \]
\[ f(A, D) = -D \cdots \]
\[ f(50, 50) = 0.0 \]
Converting Yays and Nays to Values

• How should the output value change with respect to changes of your inputs?
  
  • As agreements increase, the output value should **increase**
    
    • Agreements should be **positive** in our function
  
  • As disagreements increase, the output value should **decrease**
    
    • Disagreements should be **negative**
  
  • Equal agreements/disagreements -> 0.0
    
    • Agreements and Disagreements should be equally weighted

\[
f(A, D) \rightarrow [-1, 1]
\]

\[
f(100, 0) > f(50, 50)
\]

\[
f(A, D) = +A \cdots
\]

\[
f(0, 100) < f(50, 50)
\]

\[
f(A, D) = -D \cdots
\]

\[
f(50, 50) = 0.0
\]

\[
f(A, D) = -f(D, A)
\]
Problem Solving Workflow: Design Method

Scaling the Output

- Our function is bounded by 1 and -1
  - 100% agreements \(\rightarrow 1.0\)
    - Divide by positive agreements
  - 100% disagreements \(\rightarrow -1.0\)
    - Divide by positive disagreements
- Re-evaluate sample inputs and decide whether the function does what you want

\[
f(A, D) = \frac{A - D}{\ldots}
\]

\[
f(100, 0) = 1.0
\]

\[
f(A, D) = \frac{A - D}{A\ldots}
\]

\[
f(0, 100) = -1.0
\]

\[
f(A, D) = \frac{A - D}{D\ldots}
\]
Problem Solving Workflow: Design Method

Scaling the Output

• Our function is bounded by 1 and -1
  • 100% agreements -> 1.0
    • Divide by positive agreements
  • 100% disagreements -> -1.0
    • Divide by positive disagreements
  • Re-evaluate sample inputs and decide whether the function does what you want

\[ f(A, D) = \frac{A - D}{\ldots} \]

\[ f(100, 0) = 1.0 \]

\[ f(A, D) = \frac{A - D}{A \ldots} \]

\[ f(0, 100) = -1.0 \]

\[ f(A, D) = \frac{A - D}{D \ldots} \]
How to Code our Function

• Generally, there are usually many ways to code the same calculation/function
Problem Solving Workflow: Design Method

How to Code our Function

• Generally, there are usually many ways to code the same calculation/function
• Coding (aka Programming) is simply writing a list of instructions to be executing
Generally, there are usually many ways to code the same calculation/function

Coding (aka Programming) is simply writing a list of instructions to be executing

Akin to writing instructions for a food recipe, whereby a robot (aka a computer) will execute the recipe.

**Problem Solving Workflow: Design Method**

**How to Code our Function**

- Step 1: add 1 tablespoon of salt
- Step 2: add 1/2 tablespoon of cayenne

**RECIPE #1**

- Step 1: add 1.5 teaspoons of cayenne
- Step 2: add 3 teaspoons of salt

**RECIPE #2**

- Step 1: add 1/2 lbs of butter
- Step 2: add 3 lbs of cheese

**RECIPE #3**
Problem Solving Workflow: Design Method

How to Code our Function

Let’s make our input a specific pair of votes from 2 senators
### How to Code our Function

Let's make our input a specific pair of votes from 2 senators.

<table>
<thead>
<tr>
<th>White-house</th>
<th>Warren</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bill 1</strong></td>
<td>Yay</td>
</tr>
<tr>
<td><strong>Res1</strong></td>
<td>Nay</td>
</tr>
<tr>
<td><strong>Bill 2</strong></td>
<td>Nay</td>
</tr>
<tr>
<td><strong>Bill 3</strong></td>
<td>Yay</td>
</tr>
</tbody>
</table>

The [alignment score](#) is a value between -1 and 1.
How to Code our Function

Let's make our input a specific pair of votes from 2 senators

<table>
<thead>
<tr>
<th></th>
<th>White-house</th>
<th>Warren</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bill 1</td>
<td>Yay</td>
<td>Yay</td>
</tr>
<tr>
<td>Res1</td>
<td>Nay</td>
<td>Nay</td>
</tr>
<tr>
<td>Bill 2</td>
<td>Nay</td>
<td>Yay</td>
</tr>
<tr>
<td>Bill 3</td>
<td>Yay</td>
<td>Yay</td>
</tr>
</tbody>
</table>

Voting record of two senators

Function

$$f(A, D) = \frac{A - D}{A + D}$$

Alignment Score

Output

value between -1 and 1
How to Code our Function

Let's make our input a specific pair of votes from 2 senators

<table>
<thead>
<tr>
<th></th>
<th>Whitehouse</th>
<th>Warren</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bill 1</td>
<td>Yay</td>
<td>Yay</td>
</tr>
<tr>
<td>Res1</td>
<td>Nay</td>
<td>Nay</td>
</tr>
<tr>
<td>Bill 2</td>
<td>Nay</td>
<td>Yay</td>
</tr>
<tr>
<td>Bill 3</td>
<td>Yay</td>
<td>Yay</td>
</tr>
</tbody>
</table>

Function:
\[ f(A, D) = \frac{A - D}{A + D} \]

Output:
Alignment Score
value between -1 and 1
How to Code our Function

Let's make our input a specific pair of votes from 2 senators

- Voting record of two senators
- White-house
- Warren
- Bill 1: Yay, Yay
- Res: Nay, Nay
- Bill 2: Nay, Yay
- Bill 3: Yay, Yay

$$f(A, D) = \frac{A - D}{A + D}$$

value between -1 and 1

Alignment Score

Function

Output
How to Code our Function

Let's make our input a specific pair of votes from 2 senators

<table>
<thead>
<tr>
<th>Whitehouse</th>
<th>Warren</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bill 1</td>
<td>Yay</td>
</tr>
<tr>
<td>Res1</td>
<td>Nay</td>
</tr>
<tr>
<td>Bill 2</td>
<td>Nay</td>
</tr>
<tr>
<td>Bill 3</td>
<td>Yay</td>
</tr>
</tbody>
</table>

Input: Voting record of two senators

Function: $f(A, D) = \frac{A - D}{A + D}$

Output: Alignment Score

Value between -1 and 1
Problem Solving Workflow: Design Method

How to Code our Function

Let’s make our input a specific pair of votes from 2 senators

<table>
<thead>
<tr>
<th></th>
<th>Whitehouse</th>
<th>Warren</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bill 1</td>
<td>Yay</td>
<td>Yay</td>
</tr>
<tr>
<td>Res 1</td>
<td>Nay</td>
<td>Nay</td>
</tr>
<tr>
<td>Bill 2</td>
<td>Nay</td>
<td>Yay</td>
</tr>
<tr>
<td>Bill 3</td>
<td>Yay</td>
<td>Yay</td>
</tr>
</tbody>
</table>
How to Code our Function

Let’s make our input a specific pair of votes from 2 senators

<table>
<thead>
<tr>
<th></th>
<th>White-house</th>
<th>Warren</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bill 1</td>
<td>Yay</td>
<td>Yay</td>
</tr>
<tr>
<td>Res 1</td>
<td>Nay</td>
<td>Nay</td>
</tr>
<tr>
<td>Bill 2</td>
<td>Nay</td>
<td>Yay</td>
</tr>
<tr>
<td>Bill 3</td>
<td>Yay</td>
<td>Yay</td>
</tr>
</tbody>
</table>

\[ f(A, D) = \frac{A - D}{A + D} \]

** not actually valid computer language; it’s pseudocode, as we’re just trying to convey the logic.
Problem Solving Workflow: Design Method

How to Code our Function

Let’s make our input a specific pair of votes from 2 senators

Think, how would you instruct a person to keep track of the number of agreements and disagreements?

\[ f(A, D) = \frac{A - D}{A + D} \]

**not actually valid computer language; it's pseudocode, as we're just trying to convey the logic.**
Problem Solving Workflow: Design Method

How to Code our Function

Let’s make our input a specific pair of votes from 2 senators

<table>
<thead>
<tr>
<th></th>
<th>White-house</th>
<th>Warren</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bill 1</td>
<td>Yay</td>
<td>Yay</td>
</tr>
<tr>
<td>Res1</td>
<td>Nay</td>
<td>Nay</td>
</tr>
<tr>
<td>Bill 2</td>
<td>Nay</td>
<td>Yay</td>
</tr>
<tr>
<td>Bill 3</td>
<td>Yay</td>
<td>Yay</td>
</tr>
</tbody>
</table>

** if vote1 is Yay and vote2 is Yay: 
increase agreement count by 1

** or vote1 is Nay and vote2 is Nay: 
increase agreement count by 1

\[ f(A, D) = \frac{A - D}{A + D} \]

** not actually valid computer language; it’s pseudocode, as we’re just trying to convey the logic.
Problem Solving Workflow: Design Method

How to Code our Function

Let’s make our input a specific pair of votes from 2 senators

<table>
<thead>
<tr>
<th>White-house</th>
<th>Warren</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bill 1</td>
<td>Yay</td>
</tr>
<tr>
<td>Res1</td>
<td>Nay</td>
</tr>
<tr>
<td>Bill 2</td>
<td>Nay</td>
</tr>
<tr>
<td>Bill 3</td>
<td>Yay</td>
</tr>
</tbody>
</table>

** if vote1 is Yay and vote2 is Yay:
increase agreement count by 1

** or vote1 is Nay and vote2 is Nay:
increase agreement count by 1

** or vote1 is Yay and vote2 is Nay:
increase disagreement count by 1

** or vote1 is Nay and vote2 is Yay:
increase disagreement count by 1

** not actually valid computer language; it’s pseudocode, as we’re just trying to convey the logic.

$$ f(A, D) = \frac{A - D}{A + D} $$

Function
How to Code our Function

Let’s make our input a specific pair of votes from 2 senators

<table>
<thead>
<tr>
<th>White-house</th>
<th>Warren</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bill 1</td>
<td>Yay</td>
</tr>
<tr>
<td>Res 1</td>
<td>Nay</td>
</tr>
<tr>
<td>Bill 2</td>
<td>Nay</td>
</tr>
<tr>
<td>Bill 3</td>
<td>Yay</td>
</tr>
</tbody>
</table>

go through all pairs of votes:  
- if vote1 is Yay and vote2 is Yay:  
  increase agreement count by 1
- or vote1 is Nay and vote2 is Nay:  
  increase agreement count by 1
- or vote1 is Yay and vote2 is Nay:  
  increase disagreement count by 1
- or vote1 is Nay and vote2 is Yay:  
  increase disagreement count by 1

\[ f(A, D) = \frac{A - D}{A + D} \]

**not actually valid computer language; it's pseudocode, as we’re just trying to convey the logic.**
Problem Solving Workflow: Design Method

How to Code our Function

Let’s make our input a specific pair of votes from 2 senators

<table>
<thead>
<tr>
<th></th>
<th>Whitehouse</th>
<th>Warren</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bill 1</td>
<td>Yay</td>
<td>Yay</td>
</tr>
<tr>
<td>Res1</td>
<td>Nay</td>
<td>Nay</td>
</tr>
<tr>
<td>Bill 2</td>
<td>Nay</td>
<td>Yay</td>
</tr>
<tr>
<td>Bill 3</td>
<td>Yay</td>
<td>Yay</td>
</tr>
</tbody>
</table>

go through all pairs of votes:
- if vote1 is Yay and vote2 is Yay:
  increase agreement count by 1
- or vote1 is Nay and vote2 is Nay:
  increase agreement count by 1
- or vote1 is Yay and vote2 is Nay:
  increase disagreement count by 1
- or vote1 is Nay and vote2 is Yay:
  increase disagreement count by 1

return (agreement count - disagreement count) / (agreement count + disagreement count)

\[ f(A, D) = \frac{A - D}{A + D} \]

** not actually valid computer language; it’s pseudocode, as we’re just trying to convey the logic.
Problem Solving Workflow: Design Method

1. Define question
2. Select appropriate dataset
3. Design a valid method
4. Analyze your results
5. Communicate your findings
Problem Solving Workflow: Analyze Results

- Problem Solving Workflow
- Our Data

1. Define question
2. Select appropriate dataset
3. Design a valid method
4. Analyze your results
5. Communicate your findings
Problem Solving Workflow: Analyze Results

- Sanity check your results — start with simple, small examples

- Do you have errors?

- Are there values that do not fit into your desired range?

- Look at intermediate values you produce, like agreements and disagreements, do they agree with the data you have?

- Look for the extreme cases to ensure your function seems to work properly: do Sanders and Warren have a high score? Do Warren and Inhofe have a low score?
About making mistakes

• Everyone makes bugs, quite often actually (even FB, gmail, etc have app updates regularly)

• “An ounce of prevention is worth a pound of cure”
  — Benjamin Franklin

• Debugging code can sometimes take a long time

• Deliberately changing correct code, tinkering, and exploring to see what happens is one of the best ways to learn.
Problem Solving Workflow: Analyze Results

1. Define question
2. Select appropriate dataset
3. Design a valid method
4. Analyze your results
5. Communicate your findings
Problem Solving Workflow: Analyze Results

- Problem Solving Workflow
- Our Data

1. Define question
2. Select appropriate dataset
3. Design a valid method
4. Analyze your results
5. Communicate your findings
In this example, it’s sufficient to just output a score for any two candidates.

One could imagine making a bar graph for a given candidate (Warren), where the X-axis is various other candidates (Whitehouse, Reed, Sanders, Inhofe, etc) and Y-axis is the score from -1 to 1.

One could imagine making a heat map which displays similarity between all senators.
Problem Solving Workflow: Analyze Results

1. Define question
2. Select appropriate dataset
3. Design a valid method
4. Analyze your results
5. Communicate your findings
Datasets

- Problem Solving Workflow
- Our Data
Remember from last lecture, data can come from a plethora of sources

Two highly common sources, especially for this course:

- downloadable pre-packaged, prepared data (in text file format)
- webpages
Datasets

URL: Universal Resource Locator

Our Senate Voting data:
Datasets

URL: Universal Resource Locator

Our Senate Voting data:

URL: https://www.senate.gov/legislative/LIS/roll_call_lists/roll_call_vote_cfm.cfm?congress=115&session=2&vote=00189
Datasets

URL: Universal Resource Locator

Our Senate Voting data:

URL:  https://www.senate.gov/legislative/LIS/roll_call_lists/roll_call_vote_cfm.cfm?congress=115&session=2&vote=00189

https://www.senate.gov: Homepage of the US Senate

/legislative: Information related to legislation (vs, people, history, etc)

/LIS: Stands for Legislative Information System

/roll_call_lists: list of roll votes for current session

/roll_call_vote_cfm.cfm: A specific roll call vote page in ColdFusion Markup (cfm)

?congress=115&session=2&vote=00189: URL parameters. Server processes parameters to provide specific information
Datasets

URL: Universal Resource Locator

Our Senate Voting data:

?congress=115&session=2&vote=00189

- Parameters allow a single webpage to dynamically provide content based on their values
- Webpage URL is followed by ? then the parameters
- Parameters are paired with their values with =
- Additional parameters are separated by & symbols
- Public documentation may not always be available, but sometimes parameters and values can be guessed
Datasets

URL: Universal Resource Locator

Our Senate Voting data:

[Image of Senate Voting data page]

- **Vote Summary**
  - Question: On the Amendment (Kennedy Amdt. No. 3703)
  - Vote Number: 189
  - Vote Date: August 21, 2013, 12:53 PM
  - Required For Majority: 1/2
  - Vote Result: Amendment Agreed to

- **State of Purpose**: To increase funding for the National Suicide Prevention Lifeline.

- **Vote Counts**:
  - Yea: 95
  - Nay: 0
  - Not Voting: 5

Information compiled through Senate.gov by the Senate Bill Clerk under the direction of the Secretary of the Senate.
Datasets

URL: Universal Resource Locator

Our Senate Voting data:

Oooo, look, you can download the data (an XML file) which is being displayed here!
Datasets

About XML

- **XML**: eXtensible Markup Language

- A markup “language” provides a system for annotating a document

- Markup languages are not programming languages!

- **Extensible** implies that these documents can have a flexible architecture, unlike HTML which has a specific set of elements

- Different pieces of information can be given labels called **elements**

- Elements can also have **attributes**, which provide additional context

```xml
<roll_call_vote>
  <congress>115</congress>
  <session>2</session>
  ...
  <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>
    </member>
  </members>
</roll_call_vote>

<myelement class='blue'>
  ...
</myelement>
```
Datasets

About XML

- **XML**: eXtensible Markup Language
- A markup “language” provides a system for annotating a document
- Markup languages are not programming languages!
- **Extensible** implies that these documents can have a flexible architecture, unlike HTML which has a specific set of elements
- Different pieces of information can be given labels called **elements**
- Elements can also have **attributes**, which provide additional context

```xml
<roll_call_vote>
  <congress>115</congress>
  <session>2</session>
  ...
  <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>
      ...
    </member>
  </members>
</roll_call_vote>

<myelement class='blue'>
  ...
</myelement>
```
Datasets

About XML

• **XML**: eXtensible Markup Language

• A markup “language” provides a system for annotating a document

• Markup languages are not programming languages!

• **Extensible** implies that these documents can have a flexible architecture, unlike HTML which has a specific set of elements

• Different pieces of information can be given labels called **elements**

• Elements can also have **attributes**, which provide additional context

```xml
<roll_call_vote>
  <congress>115</congress>
  <session>2</session>
  ...
  <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>
      ...
    </member>
  </members>
</roll_call_vote>

<myelement class='blue'>
  ...
</myelement>
```
Datasets

About XML

- **XML**: eXtensible Markup Language

- A markup “language” provides a system for annotating a document

- Markup languages are not programming languages!

- **Extensible** implies that these documents can have a flexible architecture, unlike HTML which has a specific set of elements

- Different pieces of information can be given labels called **elements**

- Elements can also have **attributes**, which provide additional context

---

```xml
<roll_call_vote>
  <congress>115</congress>
  <session>2</session>
  ...
  <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>
      ...
    </member>
  </members>
</roll_call_vote>

<myelement class='blue'>
  ...
</myelement>
```
Datasets

About XML

- **XML**: eXtensible Markup Language
- A markup “language” provides a system for annotating a document
- Markup languages are not programming languages!
- **Extensible** implies that these documents can have a flexible architecture, unlike HTML which has a specific set of elements
- Different pieces of information can be given labels called **elements**
- Elements can also have **attributes**, which provide additional context

```xml
<roll_call_vote>
  <congress>115</congress>
  <session>2</session>
  ...
  <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>
    </member>
    ...
  </members>
  ...
</roll_call_vote>
```

```xml
<myelement class='blue'>
  ...
</myelement>
```
Lecture 2: Homework

Install an editor, Python, and start playing around.
With practice, the problem solving workflow will not seem like a foreign, strict formula, but will flow naturally.