Refresher on Linear Algebra

GU MSDSPP

Winter 2023

Acknowledgments

Prof. Johnson adapted these materials from materials she developed in summer of 2016 for Princeton Sociology's Math Camp for incoming PhD students. These final slides are based on adaptations of those original slides by subsequent cohorts of instructors, discussed here: https://pusocmethodscamp.org

Outline: vectors

- **Vectors**
  - Basic notation
  - Multiplying by a scalar
  - Addition and subtraction (sidenote on conformability)
  - Two forms of vector multiplication:
    1. Dot/inner product
    2. Cross product (including review of determinants)
  - Norms

Outline: matrices

- **Matrices**
  - Typology of matrix types
  - Basic matrix algebra: addition/subtraction, scalar multiplication
  - Matrix multiplication
  - Matrix transpose and inversion
  - Matrix rank

Vectors: basic notation

Data source for vectors and matrix: senators' co-sponsorship of bills during 2004 session

Source for data: James H. Fowler: Connecting the Congress: A Study of Cosponsorship Networks , *Political Analysis* 14 (4): 456-487 (Fall 2006) and Legislative Cosponsorship Networks in the U.S. House and Senate, *Social Networks* 28 (4): 454-465 (October 2006). Cleaned senate network data provided as part of Skyler Cranmer, ICPSR 2016 Network Analysis workshop.

Why vectors?

- Compact way of storing information, instead of:

- Use:

	Gregg	Alexander	Isaakson	Lieberman
Burr	1	1	1	0

Vector notation

- **Example**: two vectors: John McCain's versus Russ Feingold's cosponsorship of bills with Hillary Clinton, Lincoln Chaffee, Joseph Lieberman, and Strom Thurmond, stored in that order in the vector
- Let:

$$JohnMcCain's cosponsorship = \mathbf{u} = \begin{bmatrix} HRC & LC & JL & ST \end{bmatrix} = \begin{bmatrix} 1 & 1 & 10 & 5 \end{bmatrix}$$

$$Russ\,Feingold's\,cosponsorship = \mathbf{v} = \begin{bmatrix} HRC & LC & JL & ST \end{bmatrix} = \begin{bmatrix} 2 & 2 & 8 & 1 \end{bmatrix}$$

- **Bold** = entire vector; non-bold = element of vector. For instance:
  - $\mathbf{u}$: all of John McCain's sponsorship information
  - u_3: John McCain's sponsorship with Joe Lieberman (ten bills)
  - v_4: Russ Feingold's sponsorship with Strom Thurmond (one bill)

Vector notation

Can arrange data in multiple ways depending on purpose:

- Row vector:
$$\mathbf{u} = \begin{bmatrix} 1 & 1 & 10 & 5 \end{bmatrix}$$

- Column vector (in this case):
$$\mathbf{u}^T = \begin{bmatrix} 1 \\ 1 \\ 10 \\ 5 \end{bmatrix}$$

- Or vice versa (so if $\mathbf{u}$ were a column vector, $\mathbf{u}^T$ would be a row vector)

Important concept: vectors, visualized

Vectors contain data, and have direction and magnitude.

Example one: visualizing cosponsorship

Example two: visualizing relative cosponsorship

Vectors: operations

Basic operations with vectors

Two types of operations:

1. Vector *operation* scalar
2. Vector *operation* other vector (can include the vector itself)

Scalar operations

- **Example**: right now, the co-sponsorship is coded as a continuous variable ranging from 0 to 10. What if we want to rescale so that each element is between 0 and 1 to more easily compare John McCain and Russ Feingold's sponsorship patterns?
- Two potential ways to rescale:
  1. Multiply each cosponsorship vector by the maximum cosponsorship value *across all* senators: $\frac{1}{max(\mathbf{u}, \mathbf{v})}$
  2. Multiply each cosponsorship vector by the maximum cosponsorship value *within each* senator: $\frac{1}{max(\mathbf{u})}, \frac{1}{max(\mathbf{v})}$

Scalar operations

1. Rescaling one (maximum cosponsorship across all senators).
   Let $s = \frac{1}{max(\mathbf{u},\mathbf{v})}$:

$$\mathbf{u}_{scaledallmax} = s\mathbf{u} = \frac{1}{10}\begin{bmatrix}1 & 1 & 10 & 5\end{bmatrix} = \begin{bmatrix}\frac{1}{10}\times 1 & \frac{1}{10}\times 1 & \frac{1}{10}\times 10 & \frac{1}{10}\times 5\end{bmatrix}$$
$$= \begin{bmatrix}0.1 & 0.1 & 1 & 0.5\end{bmatrix}$$

$$\mathbf{v}_{scaledallmax} = s\mathbf{v} = \frac{1}{10}\begin{bmatrix}2 & 2 & 8 & 1\end{bmatrix}$$
$$= \begin{bmatrix}0.2 & 0.2 & 0.8 & 0.1\end{bmatrix}$$

2. Rescaling two (maximum cosponsorship within each senator).
   Let $s_1 = \frac{1}{max(\mathbf{u})} \implies s_1 = \frac{1}{10}$, $s_2 = \frac{1}{max(\mathbf{v})} \implies s_2 = \frac{1}{8}$:

$$\mathbf{u}_{scaledwithinmax} = s_1\mathbf{u} = \begin{bmatrix}0.1 & 0.1 & 1 & 0.5\end{bmatrix}$$
$$\mathbf{v}_{scaledallmax} = s_2\mathbf{v} = \begin{bmatrix}0.25 & 0.25 & 1 & 0.125\end{bmatrix}$$

Key takeaway: in a scalar operation, we just apply the scalar to each element in the vector in order.

Scalar operations

Table 1: John McCain's cosponsorship

	Hillary	Lincoln	Joe	Strom
Original	1	1	10	5
Max across all	0.10	0.10	1.00	0.50
Max within senator	0.10	0.10	1.00	0.50

Table 2: Russ Feingold's cosponsorship

	Hillary	Lincoln	Joe	Strom
Original	2	2	8	1
Max across all	0.20	0.20	0.80	0.10
Max within senator	0.25	0.25	1.00	0.12

So what does it *mean?*

The answer to the pressing question that has been keeping you up at night (who is closer to Joe Lieberman, John McCain or Russ Feingold?) depends on whether you adjust for senator-specific levels of co-sponsorship activity or not.

Operations with other vectors

Sometimes we want to use operations with other vectors. Why?

- We used scalar multiplication to compare two vectors indirectly, but what if we want to compare directly?
- For instance, which pair of senators exhibits a higher degree of similarity in terms of their cosponsorship patterns?:
  1. John McCain and Russ Feingold
  2. John McCain and Rick Santorum
- Answering this question requires performing operations on pairs of vectors

Conformability

When we do scalar operations, we apply one scalar to each element of a vector.

$$\underset{1\times 1}{\begin{bmatrix}\frac{1}{10}\end{bmatrix}} \underset{1\times 4}{\begin{bmatrix}1 & 1 & 10 & 5\end{bmatrix}} = \underset{1\times 4}{\begin{bmatrix}0.1 & 0.1 & 1 & 0.5\end{bmatrix}}$$

But when we perform operations with two vectors, we nneed to make sure they have compatible dimentions. The vectors need to be **conformable.**

Conformable is a term for when the dimensions of a vector or matrix allow us to perform some operation. Conformability is always in the context of *some operation* (e.g., addition versus multiplication).

Vector addition and subtraction

For addition and subtraction, two vectors are conformable if they have the same number of elements. When you have two conformable vectors, you perform vector addition or subtraction by adding or subtracting the matching elements of each vector, yielding a new vector as your answer.

Let's look at an example...

Vector addition and subtraction: example

- **Example**: finding the residual, where y = vector of observed values for the outcome variable and $\hat{y}$ = vector of fitted values for the outcome variable ($\hat{e}$ is sometimes denoted $\hat{u}$):

$$\hat{e} = y - \hat{y}$$

- **Example from cosponsorship data**: fit a linear regression that regresses a senator's total number of cosponsorships against a measure of how liberal versus conservative they are on economic issues . Residuals are:

$$\hat{e} = \text{observed cosponsorship count - predicted (fitted) cosponsorship count}$$

Vector addition and subtraction: example

Say we're interested in five senators:

$$\underset{5\times 1}{\overset{\textit{Senators}}{\begin{bmatrix} \textit{John McCain} \\ \textit{Joe Lieberman} \\ \textit{Rick Santorum} \\ \textit{Joe Biden} \\ \textit{John Edwards} \end{bmatrix}}}$$

And we have the following observed and predicted cosponsorship counts for each:

$$\underset{5\times 1}{\overset{\textit{Observed cosponsorships}}{\begin{bmatrix} 176 \\ 351 \\ 158 \\ 247 \\ 203 \end{bmatrix}}} \quad , \quad \underset{5\times 1}{\overset{\textit{Fitted cosponsorships}}{\begin{bmatrix} 215.21 \\ 300.92 \\ 211.23 \\ 320.14 \\ 304.06 \end{bmatrix}}}$$

Vector addition and subtraction: example

Because these two vectors are conformable (have the same number of elements), we can find our residuals:

$$\underset{5\times1}{\overset{\color{red}{\textit{Senators}}}{\begin{bmatrix} \textit{John McCain} \\ \textit{Joe Lieberman} \\ \textit{Rick Santorum} \\ \textit{Joe Biden} \\ \textit{John Edwards} \end{bmatrix}}} : \underset{5\times1}{\overset{\color{red}{\textit{Observed cosponsorships}}}{\begin{bmatrix} 176 \\ 351 \\ 158 \\ 247 \\ 203 \end{bmatrix}}} - \underset{5\times1}{\overset{\color{red}{\textit{Fitted cosponsorships}}}{\begin{bmatrix} 215.21 \\ 300.92 \\ 211.23 \\ 320.14 \\ 304.06 \end{bmatrix}}} = \underset{5\times1}{\overset{\color{red}{\textit{Residuals}}}{\begin{bmatrix} -39.21 \\ 50.08 \\ -53.23 \\ -73.14 \\ -101.06 \end{bmatrix}}}$$

But if we throw in Barack Obama, we run into problems.

$$\underset{6\times1}{\overset{\color{red}{\textit{Senators}}}{\begin{bmatrix} \textit{John McCain} \\ \textit{Joe Lieberman} \\ \textit{Rick Santorum} \\ \textit{Joe Biden} \\ \textit{John Edwards} \\ \textit{Barack Obama} \end{bmatrix}}} : \underset{5\times1}{\overset{\color{red}{\textit{Observed cosponsorships}}}{\begin{bmatrix} 176 \\ 351 \\ 158 \\ 247 \\ 203 \end{bmatrix}}} - \underset{6\times1}{\overset{\color{red}{\textit{Fitted cosponsorships}}}{\begin{bmatrix} 215.21 \\ 300.92 \\ 211.23 \\ 320.14 \\ 304.06 \\ 332.74 \end{bmatrix}}}$$

Vector multiplication

So we've covered vector addition and subtraction, with the example of residuals. For multiplication, we're going to use a different example goal.

- **(Stylized) example**: we want to assess the possibility of two senators cosponsoring a bill in the future using their patterns of collaboration with the three senators who have the highest cosponsorship counts: Mary Landrieu (conservative Dem), Tim Johnson (centrist Dem), and Jon Corzine (liberal Dem)
- Let's visualize this:

$$
\begin{aligned}
&= \begin{bmatrix} \textit{Mary L.} & \textit{Tim J.} & \textit{Jon C.} \end{bmatrix} \\
\textit{Paul Wellstone} = \mathbf{u} &= \begin{bmatrix} 0 & 8 & 2 \end{bmatrix} \\
\textit{Joe Lieberman} = \mathbf{v} &= \begin{bmatrix} 4 & 2 & 6 \end{bmatrix} \\
\textit{Dianne Feinstein} = \mathbf{z} &= \begin{bmatrix} 3 & 1 & 1 \end{bmatrix}
\end{aligned}
$$

Vector multiplication

We will use this example to review two forms of vector multiplication:

1. Dot product: $\mathbf{u} \bullet \mathbf{v}$
2. Cross product: $\mathbf{u} \times \mathbf{v}$

Vector multiplication: dot product

Let's assume that we think cosponsorship with similar people increases the likelihood of two senators cosponsoring a bill together.

$$
\begin{array}{rl}
 & = \begin{bmatrix} \textit{Mary L.} & \textit{Tim J.} & \textit{Jon C.} \end{bmatrix} \\
\textit{Paul Wellstone} = \mathbf{u} = & \begin{bmatrix} 0 & 8 & 2 \end{bmatrix} \\
\textit{Joe Lieberman} = \mathbf{v} = & \begin{bmatrix} 4 & 2 & 6 \end{bmatrix} \\
\textit{Dianne Feinstein} = \mathbf{z} = & \begin{bmatrix} 3 & 1 & 1 \end{bmatrix}
\end{array}
$$

- Therefore, in our vectors above, any non-zero cosponsorship with one of our baseline senators will boost the likelihood that a pair of our senators of interest will work together. That boost is lost if either of the senators of interest has sponzored no bills with a baseline senator.
- For instance, Joe Lieberman's high degree of collaboration with Mary Landrieu does not provide a boost towards collaboration with Paul Wellstone, since Paul Wellstone has zero

Vector multiplication: dot product

This is the intuition behind dot product: a single value that represents the likelihood of two senators of interest cosponsoring a bill, based on what we assume matters for their likelihood of cosponsorship.

As with addition and subtraction, conformability for dot products requires that two vectors have the same number of elements.

- Let n = number of elements in the vector:

$$\mathbf{u} \bullet \mathbf{v} = u_1 v_1 + u_2 v_2 ... u_n v_n = \sum_{i=1}^{n} u_i * v_i$$

- Example with Paul Wellstone and Joe Lieberman:

$$\textit{Paul Wellstone} = \mathbf{u} = \begin{bmatrix} 0 & 8 & 2 \end{bmatrix}$$
$$\textit{Joe Lieberman} = \mathbf{v} = \begin{bmatrix} 4 & 2 & 6 \end{bmatrix}$$

Results to previous practice problem

$$\mathbf{u} \bullet \mathbf{v} = 0 * 4 + 8 * 2 + 2 * 6 = 28$$

Vector multiplication: dot product practice

$$\begin{aligned}
&= \begin{bmatrix} \textit{Mary L.} & \textit{Tim J.} & \textit{Jon C.} \end{bmatrix} \\
\textit{Paul Wellstone} = \mathbf{u} &= \begin{bmatrix} 0 & 8 & 2 \end{bmatrix} \\
\textit{Joe Lieberman} = \mathbf{v} &= \begin{bmatrix} 4 & 2 & 6 \end{bmatrix} \\
\textit{Dianne Feinstein} = \mathbf{z} &= \begin{bmatrix} 3 & 1 & 1 \end{bmatrix}
\end{aligned}$$

In problem 1.1 in your Python notebook, find the dot product for the other two combinations: Wellstone and Feinstein, Lieberman and Feinstein. First find it "by hand" (see worksheet for example) and then by using the `dot` function in numpy

Vector multiplication: dot product practice solutions

$$
\begin{aligned}
&= \begin{bmatrix} \textit{Mary L.} & \textit{Tim J.} & \textit{Jon C.} \end{bmatrix} \\
\textit{Paul Wellstone} = \mathbf{u} &= \begin{bmatrix} 0 & 8 & 2 \end{bmatrix} \\
\textit{Joe Lieberman} = \mathbf{v} &= \begin{bmatrix} 4 & 2 & 6 \end{bmatrix} \\
\textit{Dianne Feinstein} = \mathbf{z} &= \begin{bmatrix} 3 & 1 & 1 \end{bmatrix}
\end{aligned}
$$

- **Results**:
  1. Paul Wellstone and Joe Lieberman (28)
  2. Joe Lieberman and Dianne Feinstein (20)
  3. Paul Wellstone and Dianne Feinstein (10)
- **Interpretation**: the fact that Dianne Feinstein's highest collaborator is Mary L. hurts her potential-for-collaboration score with Paul Wellstone, since his zero collaborations with Mary L. makes her high score count for nothing towards their potential

Vector multiplication: dot product notation

See this video for more discussion: click here

For now, worth highlighting:

- Alternate way to write the dot product: $\mathbf{u} \bullet \mathbf{v} = \mathbf{u'v}$
- Can write Lieberman and Wellstone vectors as:

$$\mathbf{u'} = \underset{1\times3}{\begin{bmatrix} 0 & 8 & 2 \end{bmatrix}},\ \mathbf{v} = \underset{3\times1}{\begin{bmatrix} 4 \\ 2 \\ 6 \end{bmatrix}} :\ \mathbf{u}^T\mathbf{v} = 1 \times 1$$

$$\mathbf{u'v} = 0 * 4 + 8 * 2 + 6 * 2 = 28 = \mathbf{u} \bullet \mathbf{v}$$

Vector multiplication: dot product and orthogonality

- We've used the dot product as one measure of the similarity of a given pair of senators' co-sponsorship patterns, but more broadly, we can use the dot product to assess the *orientation* of two vectors

- When the dot product equals zero, the vectors are **orthogonal**. Formally, this means they are perpendicular to each other. Intuitively, this means there's no overlap between them, no similarity. Imagine the following senator for whom we're trying to find a collaboration score with Wellstone ($\mathbf{u}$):

$$\mathbf{u} = \begin{bmatrix} 0 & 8 & 2 \end{bmatrix}, \ \mathbf{w} = \begin{bmatrix} 4 & 0 & 0 \end{bmatrix}$$

$$\mathbf{u} \bullet \mathbf{w} = 0$$

- Intuitively, we can see that this senator of interest does not co-sponsor bills with senators that Wellstone co-sponsors bills with. Formally, this means their records are orthogonal.

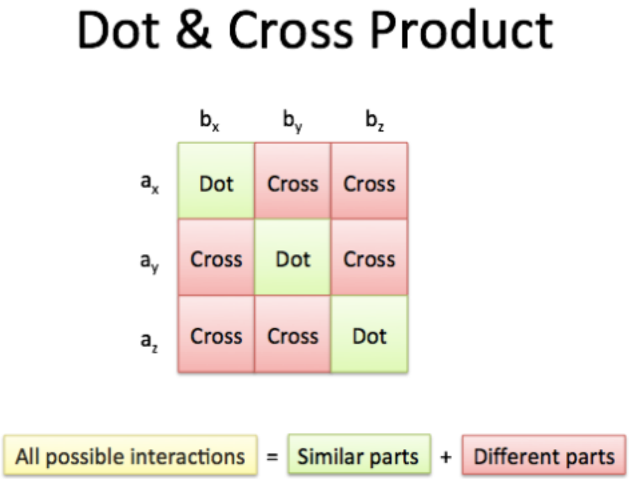
Vector multiplication: cross product

- With the **dot product**, a senator pair's "potential for collaboration" score increases when they share a *common cosponsor*; the resulting score is a single value (a scalar)
  - If Paul Wellstone sponsors a bill with Jon Corzine and Dianne Feinstein sponsors a bill with Tim Johnson, the product of those cosponsors *does not increase* Paul and Dianne's potential for collaboration score
- Basically, we made a specific assumption about what increases the likelihood of cosponsorship. But what if our assumption is different? What if we assume that senators aren't only interested in cosponsoring with like minds? What if we assume they're interested in bridging disparate cliques in the senate?

Vector multiplication: cross product

- With the **cross product**, we can form a different "potential for collaboration" measure that increases not if the senators share a *common cosponsor*, but instead if the senators share a *dissimilar cosponsor*(e.g., want to accumulate diverse cosponsors to help bridge disparate cliques in the senate); the resulting score is a vector composed of each interaction
    - If Paul Wellstone sponsors a bill with Jon Corzine and Dianne Feinstein sponsors a bill with Tim Johnson, the product of those cosponsors *does* appear in their potential for collaboration vector

Vector multiplication: cross product



Source: Betterexplained.com

Vector multiplication: cross product

1. Stack the vectors on top of each into a matrix:

$$\mathbf{A} = \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \textit{Joe Lieberman} \\ \textit{Dianne Feinstein} \end{bmatrix} = \begin{bmatrix} 4 & 2 & 6 \\ 3 & 1 & 1 \end{bmatrix}$$

2. Extract all 2×2 sub-matrices from that matrix in the following order:

$$\mathbf{A}[1,2;\ 2,3] = \begin{bmatrix} 2 & 6 \\ 1 & 1 \end{bmatrix} \ \mathbf{A}[1,2;3,1] = \begin{bmatrix} 6 & 4 \\ 1 & 3 \end{bmatrix} \ \mathbf{A}[1,2;1,2] = \begin{bmatrix} 4 & 2 \\ 3 & 1 \end{bmatrix}$$

3. Find the determinant of each sub-matrix and arrange into a vector:

$$\mathbf{u} \times \mathbf{v} = \begin{bmatrix} -4 & 14 & -2 \end{bmatrix}$$

Vector multiplication: determinant

The determinant uses all of the values of a square matrix (more on that in a bit) to provide a summary of structure.

Let $\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$,

$\det(\mathbf{A}) = ad - bc$

Vector multiplication: cross product

The magnitude of the cross product can be interpreted as the size of the area between the two vectors if we plot them in the (x, y, z) plane. Some stylized examples:

1. Senators with very different co-sponsorship patterns: large-magnitude cross product

$$\mathbf{A} = \begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix} = \begin{bmatrix}Sen1\\ Sen2\end{bmatrix} = \begin{array}{c} [\textit{Mary L.} \quad \textit{Tim J.} \quad \textit{Jon C.}] \\ \begin{bmatrix}8 & 0 & 2\\ 1 & 9 & 7\end{bmatrix}\end{array}$$

$$\mathbf{u} \times \mathbf{v} = \begin{bmatrix}-18 & -54 & 72\end{bmatrix}$$

2. Senators with very similar co-sponsorship patterns (vectors are almost overlapping, can see that the zero element stems from the leftmost sub-matrix where determinant $= 0$):

$$\mathbf{A} = \begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix} = \begin{bmatrix}Sen1\\ Sen2\end{bmatrix} = \begin{array}{c} [\textit{Mary L.} \quad \textit{Tim J.} \quad \textit{Jon C.}] \\ \begin{bmatrix}8 & 8 & 2\\ 7 & 7 & 3\end{bmatrix}\end{array}$$

$$\mathbf{u} \times \mathbf{v} = \begin{bmatrix}10 & -10 & 0\end{bmatrix}$$

Vector multiplication: cross product practice

In problem 1.2 in your Python notebook, find the cross product for the following two senate voting vectors using the `cross` function in numpy; you can also find the answer manually using the sub-matrix approach reviewed previously

$$\mathbf{A} = \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} Sen1 \\ Sen2 \end{bmatrix} = \begin{array}{c} [\textit{Mary L.} \quad \textit{Tim J.} \quad \textit{Jon C.}] \\ \begin{bmatrix} 2 & 4 & 1 \\ 6 & 12 & 3 \end{bmatrix} \end{array}$$

Vector multiplication: cross product practice solutions

$$\mathbf{A} = \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} Sen1 \\ Sen2 \end{bmatrix} = \begin{array}{c} [\textit{Mary L.} \quad \textit{Tim J.} \quad \textit{Jon C.}] \\ \begin{bmatrix} 2 & 4 & 1 \\ 6 & 12 & 3 \end{bmatrix} \end{array}$$

- Cross product is 0, which results from the fact that Senator 1's cosponsorship can be written as a linear combination of senator 2's cosponsorship vector ($Sen\,2 = 3 * Sen\,1$), more formally: $\mathbf{v} = 3\mathbf{u}$)
- Less formally, the vectors fully overlap; more formally, they are *linearly dependent*

Vector multiplication: when the answer is 0

Different implications:

- $\mathbf{u} \bullet \mathbf{v} = 0$: vectors are perpendicular/orthogonal
- $\mathbf{u} \times \mathbf{v} = 0$: vectors are linearly dependent, which in geometric terms, means the vectors are *parallel*

Vectors: length and distance between

- Thus far, we've been constructing our own "potential for collaboration" score as either a scalar measuring the extent to which two senators' cosponsorship vectors overlap (dot product) or as a vector measuring the extent to which two senators' help "bridge" disparate parts of the cosponsorship space (a large area between them, the cross product)
- More general/common uses of the dot and cross product are to calculate:
  - The length of a vector, more formally known as the vector's **norm**
  - The distance between two vectors

Vectors: why length and distance between?

- Returning to our previous example:

$$\begin{array}{rl} & = \begin{bmatrix} \textit{Mary L.} & \textit{Tim J.} & \textit{Jon C.} \end{bmatrix} \\ \textit{Paul Wellstone} = \mathbf{u} = & \begin{bmatrix} 0 & 8 & 2 \end{bmatrix} \\ \textit{Joe Lieberman} = \mathbf{v} = & \begin{bmatrix} 4 & 2 & 6 \end{bmatrix} \\ \textit{Dianne Feinstein} = \mathbf{w} = & \begin{bmatrix} 3 & 1 & 1 \end{bmatrix} \end{array}$$

- For this example, it is easy to see which senators have higher magnitude of co-sponsorship because all elements of the vectors have positive values
- But what if we were dealing with, for instance, how far away the senator's *observed* cosponsorship count with another senator was from his/her *predicted* cosponsorship count based on some model. And we want underestimates to count for the same as overestimates:

$$\begin{array}{rl} & = \begin{bmatrix} \textit{Mary L.} & \textit{Tim J.} & \textit{Jon C.} \end{bmatrix} \\ \textit{Paul Wellstone} = \mathbf{u} = & \begin{bmatrix} 2 & -3 & 5 \end{bmatrix} \\ \textit{Joe Lieberman} = \mathbf{v} = & \begin{bmatrix} -2 & -3 & -4 \end{bmatrix} \end{array}$$

Vectors: length and distance between

- Squaring to the rescue! To help both negative and positive prediction errors (and elements more generally) contribute the same magnitude to the vector's overall size
- We could depict that squaring in a lengthy way, e.g. if we had Wellstone's vector **u**, we could begin to find its length using some function of:

$$u_1^2 + u_2^2 .... u_n^2$$

- But what's a more compact way to write this...?

Vectors: length and distance between

- **Euclidean norm**: (one option for measuring vector length):

$$\|\mathbf{u}\|^2 = \mathbf{u} \bullet \mathbf{u} = \mathbf{u}^T \mathbf{u}$$
$$\|\mathbf{u}\| = \sqrt{\mathbf{u} \bullet \mathbf{u}}$$

- **Difference norm**: one measure of the distance between two vectors:

  1. Start with:
  $$\|\mathbf{u} - \mathbf{v}\|^2 = \|\mathbf{u} - \mathbf{v}\| \|\mathbf{u} - \mathbf{v}\|$$
  2. Expand/distribute:
  $$\|\mathbf{u}\|^2 - 2(\mathbf{u} \bullet \mathbf{v}) + \|\mathbf{v}\|^2$$
  3. Can also rewrite using dot product or vector-transpose notation:
  $$= \mathbf{u} \bullet \mathbf{u} - 2(\mathbf{u} \bullet \mathbf{v}) + \mathbf{v} \bullet \mathbf{v}$$
  $$= \mathbf{u}^T \mathbf{u} - 2(\mathbf{u}^T \mathbf{v}) + \mathbf{v}^T \mathbf{v}$$

Matrices

Why matrices?

$$
\begin{array}{r}
= \begin{bmatrix} \textit{Mary L.} & \textit{Tim J.} & \textit{Jon C.} \end{bmatrix} \\
\textit{Paul Wellstone} = \mathbf{u} = \begin{bmatrix} 0 & 8 & 2 \end{bmatrix} \\
\textit{Joe Lieberman} = \mathbf{v} = \begin{bmatrix} 4 & 2 & 6 \end{bmatrix} \\
\textit{Dianne Feinstein} = \mathbf{w} = \begin{bmatrix} 3 & 1 & 1 \end{bmatrix}
\end{array}
$$

- Thus far, to evaluate the structure of these co-sponsorship patterns, we've been pulling out each pair of vectors
- We'll keep doing that, but we can also treat the stacked vectors as *matrices* and discern new information about cosponsorship patterns

And with that somewhat forced transition...

Where we're going:

- Types of matrices
- Matrix algebra:
  - Basic: addition, subtraction, scalar multiplication
  - More complex: matrix multiplication and transposition
- Ways to use matrices
  - Preview of role matrices play in linear regression
  - Invertible matrices
  - What is the rank of the matrix?

Matrices: types

- We've been drawing out vectors from the senate cosponsorship data, but now let's view the structure of the entire matrix (or more precisely, a matrix with more obscure senators cruelly culled to help it fit on the slide):

$$\mathbf{A} = \begin{array}{c|ccccc} & \textit{Hillary Clinton} & \textit{Rick Santorum} & \textit{Joe Lieberman} & \textit{John McCain} & \textit{Joe Biden} \\ \hline \textit{Hillary Clinton} & 0 & 3 & 9 & 7 & 12 \\ \textit{Rick Santorum} & 0 & 0 & 6 & 0 & 4 \\ \textit{Joe Lieberman} & 9 & 3 & 0 & 20 & 9 \\ \textit{John McCain} & 1 & 1 & 10 & 0 & 3 \\ \textit{Joe Biden} & 8 & 0 & 2 & 9 & 0 \end{array}$$

- This is a 5×5 matrix
- In matrix notation, which element represents John McCain's cosponsorship of Hillary Clinton's bills? How about Joe Biden's cosponsorship of McCain's bills?
  - Answer: $a_{4,1}$ for McCain-Clinton, $a_{5,4}$ for Biden-McCain

Matrices: types (square and rectangular)

- If we denote matrix dimensions as $m \times n$:
  - Square: $m = n$, which for whatever reason, we often describe as a $k \times k$ matrix. Another way to describe is that it is *order* k. What order is the matrix from the previous slide?
  - Rectangular: $m \neq n$
- Typical square matrices in social science: matrices to summarize pairwise measures (e.g., our co-sponsorship data; a correlation matrix summarizing correlations between any two variables in a dataset; etc.)
- Typical rectangular matrices in social science: rows = observations, columns = predictor variables (unless your number of observations happens to equal number of covariates)

Matrices: types (square and rectangular)

Sometimes it's possible to go from a rectangular matrix to a square one.

- **Example**: in addition to the cosponsorship data, we have a rectangular 101×2 matrix summarizing each senator's ideology along two dimensions (based on their vote record)
- Viewing this rectangular matrix for our five senators of interest, we have:

	ideology1	ideology2
Biden Joseph R. Jr.	-0.335	0.016
Clinton Hillary Rodham	-0.344	0.017
Lieberman Joseph I.	-0.219	-0.130
McCain John	0.298	-0.445
Santorum Rick	0.322	-0.263

- We can transform into a 5×5 square matrix of ideological distance for each senator pair by going pair by using our measure of euclidean distance on each pair of senators:

$$\|\mathbf{u} - \mathbf{v}\| = \sqrt{\mathbf{u} \bullet \mathbf{u} - 2(\mathbf{u} \bullet \mathbf{v}) + \mathbf{v} \bullet \mathbf{v}}$$

Matrices: types (square and rectangular)

So for each pair of senators, we calculate a value of ideological distance

1. Let $McCain = \mathbf{u} = \begin{bmatrix} 0.298 & -0.445 \end{bmatrix}$, and $Biden = \mathbf{v} = \begin{bmatrix} -0.335 & 0.016 \end{bmatrix}$
2. Separate $\|\mathbf{u} - \mathbf{v}\| = \sqrt{\mathbf{u} \bullet \mathbf{u} - 2(\mathbf{u} \bullet \mathbf{v}) + \mathbf{v} \bullet \mathbf{v}}$ into sub-components and calculate each:

$$\mathbf{u} \bullet \mathbf{u} = 0.298^2 + (-0.445)^2 = 0.286829$$

$$\mathbf{u} \bullet \mathbf{v} = 0.298 * -0.335 + (-0.455 * 0.016) = -0.10695$$

$$\mathbf{v} \bullet \mathbf{v} = -0.335^2 + 0.016^2 = 0.112481$$

3. Combine and take the square root:

$$\|\mathbf{u} - \mathbf{v}\| = \sqrt{0.286829 + 2 * -0.10695 + 0.112481} = 0.783$$

Matrices: types (square and rectangular)

	Hillary Clinton	*Rick Santorum*	*Joe Lieberman*	*John McCain*	*Joe Biden*
Hillary Clinton	0.000	0.722	0.193	0.791	0.009
Rick Santorum	0.722	0.000	0.557	0.184	0.714
Joe Lieberman	0.193	0.557	0.000	0.605	0.186
John McCain	0.791	0.184	0.605	0.000	0.783
Joe Biden	0.009	0.714	0.186	0.783	0.000

Matrices: types (symmetric)

- Focusing on the ideology matrix, the way we defined distance meant that $distance_{McCain,Biden} = distance_{Biden,McCain}$
- Since we defined distance similarly for every pair, the matrix is symmetric ($a_{ij} = a_{ji}$ for all i, j), which informally means that if you split the matrix in two along the diagonal (bolded below), the two halves are mirror images (also means $\mathbf{X}^T = \mathbf{X}$):

$$\mathbf{X} = \begin{array}{cccccc} & \textit{Hillary Clinton} & \textit{Rick Santorum} & \textit{Joe Lieberman} & \textit{John McCain} & \textit{Joe Biden} \\ \textit{Hillary Clinton} & \mathbf{0.000} & 0.722 & 0.193 & 0.791 & 0.009 \\ \textit{Rick Santorum} & 0.722 & \mathbf{0.000} & 0.557 & 0.184 & 0.714 \\ \textit{Joe Lieberman} & 0.193 & 0.557 & \mathbf{0.000} & 0.605 & 0.186 \\ \textit{John McCain} & 0.791 & 0.184 & 0.605 & \mathbf{0.000} & 0.783 \\ \textit{Joe Biden} & 0.009 & 0.714 & 0.186 & 0.783 & \mathbf{0.000} \end{array}$$

Matrices: types (diagonal)

- If a matrix is symmetric *and* square, it may also be a diagonal matrix, where $a_{ij} = 0$ for all $i \neq j$
- For instance, if we were to form a square matrix that simply represented each senator's age (so a non pairwise measure), it might look like (ages are made up!):

$$
\mathbf{A} = \begin{array}{c|ccccc}
 & \textit{Hillary Clinton} & \textit{Rick Santorum} & \textit{Joe Lieberman} & \textit{John McCain} & \textit{Joe Biden} \\
\textit{Hillary Clinton} & 68 & 0 & 0 & 0 & 0 \\
\textit{Rick Santorum} & 0 & 50 & 0 & 0 & 0 \\
\textit{Joe Lieberman} & 0 & 0 & 70 & 0 & 0 \\
\textit{John McCain} & 0 & 0 & 0 & 73 & 0 \\
\textit{Joe Biden} & 0 & 0 & 0 & 0 & 69
\end{array}
$$

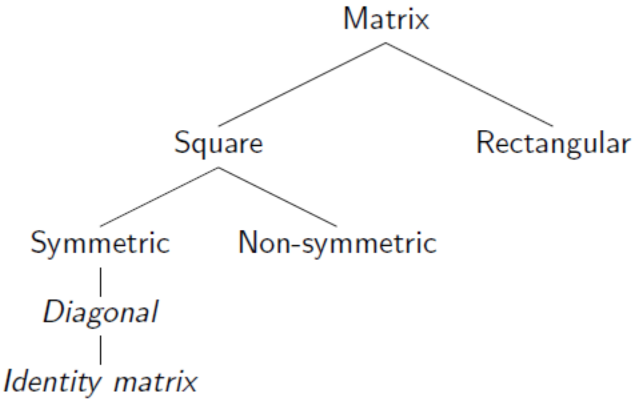
Matrices: types (identity matrix)

- If a matrix is symmetric *and* square *and* diagonal, it may also be an identity matrix, where $a_{ij} = 0$ for all $i \neq j$ and $a_{ij} = 1$ for all $i = j$
- What makes the identity matrix special is it acts like a one in matrix multiplication. Specifically, if two matrices multiply to get the identity matrix, they are the inverse of each other.
- More formally, the notation for the identity matrix is $\mathbf{I}_n$
- In this context, $\mathbf{I}_5$:

$$\mathbf{A} = \begin{array}{rccccc} & \textit{Hillary Clinton} & \textit{Rick Santorum} & \textit{Joe Lieberman} & \textit{John McCain} & \textit{Joe Biden} \\ \textit{Hillary Clinton} & 1 & 0 & 0 & 0 & 0 \\ \textit{Rick Santorum} & 0 & 1 & 0 & 0 & 0 \\ \textit{Joe Lieberman} & 0 & 0 & 1 & 0 & 0 \\ \textit{John McCain} & 0 & 0 & 0 & 1 & 0 \\ \textit{Joe Biden} & 0 & 0 & 0 & 0 & 1 \end{array}$$

Matrices: types (summing up)

Note: *italicized* nodes on the tree represent special cases of the matrix in question rather than an exhaustive set of cases. So, for example. all square matrices are either symmetric or non-symmetric, but there are symmetric matrices that are *not* diagonal matrices.



Matrices: types (summing up)

- We're reviewed a taxonomy of special matrix types: is a matrix square? If so, is it symmetric? If so, is it a diagonal matrix? If so, is it an identity matrix?
- These might seem abstract for now, but they come up, for instance, in matrix-based approach to linear regression and in matrix decomposition (how to represent matrix $\mathbf{A}$ as the product of other matrices)
- In addition, the matrix algebra and summary operations we review next are mechanically easier for certain special matrices
  - For instance, the determinant of a diagonal matrix is just the product of the diagonal elements

Matrices: operations

- Basic operations: addition/subtraction, scalar multiplication
- More complex operations: multiplication, transpose
- For each, we'll cover:
  - Motivation/preview of applications
  - Mechanics: what counts as conformable matrices for the purposes of the operation and how to perform

Matrices: addition and subtraction

- **Example**: You want to conduct a time-series analysis of Senate cosponsorship patterns investigating factors that explain a senator's *deviation* from his or her average cosponsorship patterns. For example, does a senator's cosponsoring pattern change relative to his/her average pattern after a close election?
- Represent deviation as $\tilde{\mathbf{A}}$, where $\tilde{\mathbf{A}} = \mathbf{A}_{closeelect} - \bar{\mathbf{A}}$
- Fake data:

$$\mathbf{A}_{closeelect} = \begin{array}{lccc} & \textit{Hillary Clinton} & \textit{Rick Santorum} & \textit{Joe Lieberman} \\ \textit{Hillary Clinton} & 0 & 3 & 9 \\ \textit{Rick Santorum} & 0 & 0 & 6 \\ \textit{Joe Lieberman} & 9 & 3 & 0 \end{array}$$

$$\bar{\mathbf{A}} = \begin{array}{lccc} & \textit{Hillary Clinton} & \textit{Rick Santorum} & \textit{Joe Lieberman} \\ \textit{Hillary Clinton} & 0 & 2 & 12 \\ \textit{Rick Santorum} & 0 & 0 & 3 \\ \textit{Joe Lieberman} & 7 & 5 & 0 \end{array}$$

Matrices: addition and subtraction

- **Conformable in this case**: matrices have exact same dimensions (in this case, both 3×3)
- How to do: element by element addition/subtraction

$$\tilde{\mathbf{A}} = \mathbf{A}_{closeelect} - \bar{\mathbf{A}} = \begin{array}{cccc} & \textit{Hillary Clinton} & \textit{Rick Santorum} & \textit{Joe Lieberman} \\ \textit{Hillary Clinton} & 0 & 3-2 & 9-12 \\ \textit{Rick Santorum} & 0 & 0 & 6-3 \\ \textit{Joe Lieberman} & 9-7 & 3-5 & 0 \end{array}$$

Matrices: scalar multiplication

- Similar motivation as in vector case: can simultaneously rescale all the elements by some constant
- **Conformable in this case**: since scalar is applied to every element of the matrix, works for a matrix of any dimension
- Example: rescale cosponsorship by maximum cosponsorship value (multiply by $\frac{1}{max(\mathbf{A})} = \frac{1}{20}$) to constrain to be between 0 and 1, mechanics shown for first line:

	Hillary Clinton	*Rick Santorum*	*Joe Lieberman*	*John McCain*	*Joe Biden*
Hillary Clinton	0.00	$\frac{1}{20} * 3 = 0.15$	$\frac{1}{20} * 9 = 0.45$	$\frac{1}{20} * 7 = 0.35$	$\frac{1}{20} * 12 = 0.60$
Rick Santorum	0.00	0.00	0.30	0.00	0.20
Joe Lieberman	0.45	0.15	0.00	1.00	0.45
John McCain	0.05	0.05	0.50	0.00	0.15
Joe Biden	0.40	0.00	0.10	0.45	0.00

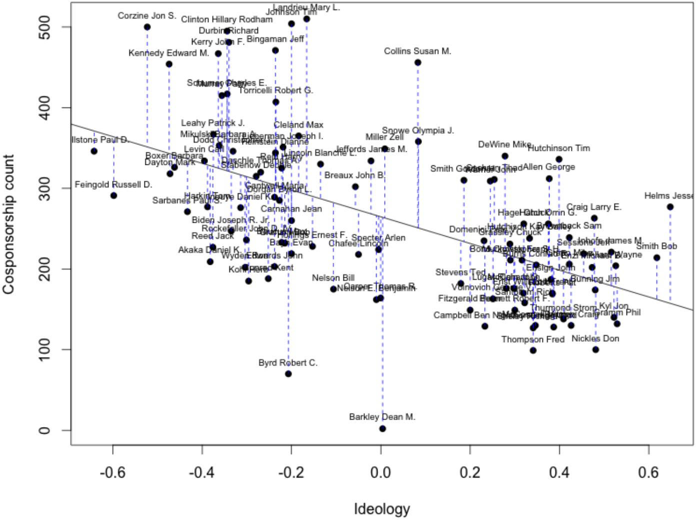
Matrices: matrix multiplication

- Many applications- an important one is linear regression, where we can begin with typical way of writing the regression equation, and rewrite using matrices and vectors:

$$Y = \beta_0 + \beta_1 X_1 \ldots \beta_n X_n$$

- You might be familiar (if not, you'll learn this year) with how to adjudicate between these options in the univariate case by using a "best fit" line when we have just one predictor. In the single variable case, we're just trying to find the best values for a intercept and slope of a linear equation.

Matrices: matrix multiplication (what is regression?)



Matrices: matrix multiplication (what is regression?)

- What happens when we have more than one predictor?
- Outcome variable (Y): number of bills a senator cosponsors
- We have intuitions about what might explain variation between senators in Y:

$$Y = \beta_0 + \textit{ideology} * \beta_1 + \textit{tenure} * \beta_2 + \textit{donations} * \beta_3$$

- Linear regression: which set of weights gets us closest to the observed cosponsorship count? Where the weights can be any set of linear combinations in $\mathbb{R}^n$, so they might be:

  Option one: Option two: Option three:

- $$\begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \\ \hat{\beta}_3 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 0.5 \\ 0.8 \\ 0.9 \end{bmatrix} \qquad \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \\ \hat{\beta}_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 0.1 \\ 0.001 \\ 5 \end{bmatrix} \qquad \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \\ \hat{\beta}_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 0.4 \\ 2.3 \\ 4.7 \end{bmatrix}$$

Matrices: matrix multiplication

- You'll learn how to actually pick β values in a stats class. For now, we're just trying to see why matrix multiplication might be useful.
- Let's see how we can rewrite our original equation with vectors and matrices:
- $$Y = \beta_0 + ideology * \beta_1 + tenure * \beta_2 + donations * \beta_3$$
- Matrix representation (the vector of 1's represents β_0)

$$\mathbf{1} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \mathbf{ideology} = \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \\ x_{41} \end{bmatrix}, \mathbf{tenure} = \begin{bmatrix} x_{12} \\ x_{22} \\ x_{32} \\ x_{42} \end{bmatrix}, \mathbf{donations} = \begin{bmatrix} x_{13} \\ x_{23} \\ x_{33} \\ x_{43} \end{bmatrix}$$

Matrices: matrix multiplication

- We can squish these together into a matrix.

$$\begin{bmatrix} 1 & x_{1,1} & x_{1,2} & x_{1,3} \\ 1 & x_{2,1} & x_{2,2} & x_{2,3} \\ 1 & x_{3,1} & x_{3,2} & x_{3,3} \\ 1 & x_{4,1} & x_{4,2} & x_{4,3} \end{bmatrix}$$

  But where are our betas?

- Our beta values are in their own vector of unknowns:

$$\begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}$$

Matrices: matrix multiplication

- This makes our whole equation:

$$y = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} * \begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} \\ 1 & x_{21} & x_{22} & x_{23} \\ 1 & x_{31} & x_{32} & x_{33} \\ 1 & x_{41} & x_{42} & x_{43} \end{bmatrix}$$

  Which can also be written as:

$$Y = X\beta$$

- So if we have values for our beta vector and X matrix, how would we actually multiply the two together to spit out the vector of y that we want?

Matrices: matrix multiplication

Conformable for multiplication: number of *columns* in first matrix must equal number of *rows* in second matrix, so:

- Not conformable:

$$\beta = \underset{4\times 1}{\begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}}, \quad \mathbf{X} = \underset{4\times 4}{\begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} \\ 1 & x_{21} & x_{22} & x_{23} \\ 1 & x_{31} & x_{32} & x_{33} \\ 1 & x_{41} & x_{42} & x_{43} \end{bmatrix}}$$

- Conformable:

$$\mathbf{X}\beta = \underset{4\times 4}{\begin{bmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ 1 & x_{31} & x_{32} \\ 1 & x_{41} & x_{42} \end{bmatrix}} \underset{4\times 1}{\begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}}$$

- Dimensions of resulting matrix: same number of *rows* as first matrix and same number of *columns* as the second matrix (in this case, 4×1)

Matrices: matrix multiplication

- Another example: here, result will be a 4×2 matrix:

$$\mathbf{X}\beta = \underset{4\times 3}{\begin{bmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ 1 & x_{31} & x_{32} \\ 1 & x_{41} & x_{42} \end{bmatrix}} \underset{3\times 2}{\begin{bmatrix} \beta_0 & \gamma_0 \\ \beta_1 & \gamma_1 \\ \beta_2 & \gamma_2 \end{bmatrix}}$$

- Formally, we take the dot product of each row vector from the first matrix and column vector from the second matrix (red = changes from row to row of results):

$$\mathbf{X}\beta = \underset{4\times 2}{\begin{bmatrix} (1 \quad x_{11} \quad x_{12}) \bullet (\beta_0 \quad \beta_1 \quad \beta_2) & (1 \quad x_{11} \quad x_{12}) \bullet (\gamma_0 \quad \gamma_1 \quad \gamma_2) \\ (1 \quad x_{21} \quad x_{22}) \bullet (\beta_0 \quad \beta_1 \quad \beta_2) & (1 \quad x_{21} \quad x_{22}) \bullet (\gamma_0 \quad \gamma_1 \quad \gamma_2) \\ (1 \quad x_{31} \quad x_{32}) \bullet (\beta_0 \quad \beta_1 \quad \beta_2) & (1 \quad x_{31} \quad x_{32}) \bullet (\gamma_0 \quad \gamma_1 \quad \gamma_2) \\ (1 \quad x_{41} \quad x_{42}) \bullet (\beta_0 \quad \beta_1 \quad \beta_2) & (1 \quad x_{41} \quad x_{42}) \bullet (\gamma_0 \quad \gamma_1 \quad \gamma_2) \end{bmatrix}}$$

- Informally, we can 1) draw out a shell matrix with correct dimensions for results; 2) circle rows in first, columns in second and proceed

Matrices: matrix multiplication practice

In problem 2.1 in your Python notebook, you can practice with conformability and multiplication. For the following matrices:

$$\mathbf{Y} = \begin{bmatrix} 3 & 1 & -2 \\ 6 & 3 & 4 \end{bmatrix}, \ \mathbf{X} = \begin{bmatrix} 4 & 2 \\ 3 & 0 \\ 1 & 2 \end{bmatrix}$$

1. Use the `shape` command to find out the dimensions of each
2. Arrange multiplication in a way that makes matrices conformable to multiply *and* that results in a 3×3 matrix
3. Multiply either more manually using the rules discussed on previous slide or through the `dot` command in numpy

Matrices: matrix multiplication practice solutions

$$\mathbf{Y} = \begin{bmatrix} 3 & 1 & -2 \\ 6 & 3 & 4 \end{bmatrix}, \ \mathbf{X} = \begin{bmatrix} 4 & 2 \\ 3 & 0 \\ 1 & 2 \end{bmatrix}$$

1. Write out dimensions of each: $\mathbf{X} = 3 \times 2$; $\mathbf{Y} = 2 \times 3$
2. Arrange multiplication in a way that makes matrices conformable to multiply: order that makes conformable, and will result in a 3 x 3 matrix:

$$\mathbf{X} = \begin{bmatrix} 4 & 2 \\ 3 & 0 \\ 1 & 2 \end{bmatrix} \mathbf{Y} = \begin{bmatrix} 3 & 1 & -2 \\ 6 & 3 & 4 \end{bmatrix},$$

3. Multiply by hand:

$$\begin{bmatrix} 12+12 & 4+6 & -8+8 \\ 9 & 3 & -6 \\ 3+12 & 1+6 & -2+8 \end{bmatrix}$$

Matrices: other tools for multiplication

- We've emphasized the importance of checking to make sure matrices are conformable before matrix multiplication
- One way we created conformable conditions: switching around order of matrices before multiplication
- But what about the following case (which,incidentally, is the total count of bills a senator cosponsored ($\mathbf{Y}$) and the two measures of senator ideology along with an intercept term ($\mathbf{X}$)):

$$\mathbf{Y} = \begin{bmatrix} HRC \\ RS \\ JL \\ JM \\ JB \end{bmatrix} = \underset{5\times 1}{\begin{bmatrix} 31 \\ 10 \\ 41 \\ 15 \\ 19 \end{bmatrix}}, \quad \mathbf{X} = \underset{5\times 3}{\begin{bmatrix} 1 & -0.34 & 0.02 \\ 1 & -0.34 & 0.02 \\ 1 & -0.22 & -0.13 \\ 1 & 0.30 & -0.45 \\ 1 & 0.32 & -0.26 \end{bmatrix}}$$

- $Y_{5\times 1} X_{5\times 3}$ is not conformable
- $X_{5\times 3} Y_{5\times 1}$ is also not conformable

Matrices: transpose

- **Transpose**: we've already reviewed with vectors, but with matrices, written as $\mathbf{A}^T$ or $\mathbf{A}'$ and just means switching the rows and columns
- We've run into the situation where we want to multiply two matrices:
  1. $\mathbf{X}$: 5×3
  2. $\mathbf{Y}$: 5×1
- Can use transpose to 1) get the inner dimensions to match in a way that makes the matrices conformable; 2) produces a results matrix with dimensions that are appropriate for the question at hand
- Dimensions after transposing:
  1. $\mathbf{X}^T : 3 \times 5$
  2. $\mathbf{Y} : 5 \times 1$
  3. Check conformability– we're good!: $(3 \times 5)(5 \times 1)$
  4. Dimensions after multiplying: $\mathbf{X}^T\mathbf{Y} : 3 \times 1$

Matrices: transpose

First column becomes first row, second column becomes second row, and so on. Visual depiction of $\mathbf{X}^T$:

$$\underset{5\times 3}{\mathbf{X} = \begin{bmatrix} 1 & -0.34 & 0.02 \\ 1 & -0.34 & 0.02 \\ 1 & -0.22 & -0.13 \\ 1 & 0.30 & -0.45 \\ 1 & 0.32 & -0.26 \end{bmatrix}}$$

$$\underset{3\times 5}{\mathbf{X}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -0.34 & -0.34 & -0.22 & 0.30 & 0.32 \\ 0.02 & 0.02 & -0.13 & -0.45 & -0.26 \end{bmatrix}}$$

To do in Python: np.transpose(matrix).

Matrices: transpose properties

These are invaluable for matrix-based proofs of regression properties:

- Invertibility (or: if you transpose a transposed matrix, you get back the original): $(\mathbf{X}^T)^T = \mathbf{X}$
- Additive (or: can distribute a transpose to addition without switching order since matrix addition is commutative): $(\mathbf{X} + \mathbf{Y})^T = \mathbf{X}^T + \mathbf{Y}^T = \mathbf{Y}^T + \mathbf{X}^T$
- Multiplicative (or: can distribute a transpose to matrix multiplication but need to switch the order): $(\mathbf{XYZ})^T = \mathbf{Z}^T\mathbf{Y}^T\mathbf{X}^T$
- For a symmetric matrix (e.g., our ideological distance one): $\mathbf{X}^T = \mathbf{X}$

Matrices: transpose practice

In problem 2.2 in your Python notebook, for the matrices with the dimensions below, write

1. The dimensions of the $Y - X\beta$. Hint: what are the dimensions of $X\beta$ and then what are the dimensions of Y minus that result?
2. Given those dimensions, how would you would use transpose to make the following multiplication 1) conformable, 2) produce a 1×1 result?: $(Y - X\beta)(Y - X\beta)$
3. After step two, if it involves transposing one or both of the $Y - X\beta$, how would those transposes be distributed using the properties on the previous slide (we can flip back),

Matrices:

$$\mathbf{X} = \underset{5\times 3}{\begin{bmatrix} x_{11} & \cdots & x_{31} \\ \vdots & & \vdots \\ x_{51} & \cdots & x_{53} \end{bmatrix}}, \quad \mathbf{Y} = \underset{5\times 1}{\begin{bmatrix} y_{11} \\ \vdots \\ y_{51} \end{bmatrix}}, \quad \beta = \underset{3\times 1}{\begin{bmatrix} \beta_{11} \\ \beta_{21} \\ \beta_{31} \end{bmatrix}}$$

Matrices: transpose practice solutions

1. Since $X\beta = (5 \times 3)(3 \times 1) = 5 \times 1$ (outer dimensions), we can do $Y - X\beta$, and since subtraction doesn't change dimensions, result is 5×1
2. We want to be able to multiply: $(5 \times 1)(5 \times 1)$. If we transpose the first matrix, we multiply: $(1 \times 5)(5 \times 1) = (1 \times 1)$. If we transpose the second matrix, we multiply: $(5 \times 1)(1 \times 5) = (5 \times 5)$. Therefore, we want to transpose the first matrix to get the 1×1 results we want.
3. $(Y - X\beta)^T$. Can break into steps:
   3.1 Distribute the transpose across subtraction (no reordering needed):
   $$Y^T - (X\beta)^T$$
   3.2 Distribute the transpose across multiplication (need to reorder):
   $$Y^T - \beta^T X^T$$
   3.3 You might notice that the result is not conformable for that expression alone, but it would be if we then multiply by $Y - X\beta$ and distribute (we can try as a group if enough time)

Matrices: inversion

- What if need to divide matricies? For example, what we want to solve for $\hat{\beta}$ in this expression:

$$X^T(Y - X\hat{\beta}) = 0$$

- Steps
  1. $X^TY - X^TX\hat{\beta} = 0$
  2. $X^TY = X^TX\hat{\beta}$
  3. How do we divide X^TY by X^TX?
  4. With matrices, can't just write:

$$\frac{X^TY}{X^TX}$$

  5. Need to write:

$$(X^TY)(X^TX)^{-1} = \hat{\beta}$$

Matrices: inversion

Two separate questions:

1. Can we invert a matrix? (invertible/nonsingular)
    - *Minimal condition*: a matrix must be square
    - If a matrix is square, it still may not be invertible
2. If yes to 1, what is the matrix's inverse?
3. For a 2×2 matrix $\mathbf{A}$, where $\mathbf{A}$ is represented as:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

We can invert using the following formula, where det(A) = *ad*- *bc*:

$$\frac{1}{det(A)} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Matrices: inversion practice

In problem 2.3 in your python notebook, find the inverse of the following matrix more manually

$$\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}$$

Matrices: inversion practice

- Matrix inversion by hand can be tedious and hard.
- Some matrices are invertible – no solution exists
- Luckily, very easy to do in Python with the `np.linalg.inv()` function

Matrices: matrix rank

- Basic idea: how much unique information a matrix holds
- Imagine two versions of the senator cosponsorship matrix, the original:

$$\mathbf{A} = \begin{array}{r|ccccc} & \textit{Hillary Clinton} & \textit{Rick Santorum} & \textit{Joe Lieberman} & \textit{John McCain} & \textit{Joe Biden} \\ \hline \textit{Hillary Clinton} & 0 & 3 & 9 & 7 & 12 \\ \textit{Rick Santorum} & 0 & 0 & 6 & 0 & 4 \\ \textit{Joe Lieberman} & 9 & 3 & 0 & 20 & 9 \\ \textit{John McCain} & 1 & 1 & 10 & 0 & 3 \\ \textit{Joe Biden} & 8 & 0 & 2 & 9 & 0 \end{array}$$

- And a modified version, where Joe Biden, hoping to curry favor with Hillary Clinton, decides to take any senator who she cosponsors a bill with and cosponsor with them twice as many times (even absurdly cosponsoring with himself)

$$\mathbf{A}_{shortrank} = \begin{array}{r|ccccc} & \textit{Hillary Clinton} & \textit{Rick Santorum} & \textit{Joe Lieberman} & \textit{John McCain} & \textit{Joe Biden} \\ \hline \textit{Hillary Clinton} & 0 & 3 & 9 & 7 & 12 \\ \textit{Rick Santorum} & 0 & 0 & 6 & 0 & 4 \\ \textit{Joe Lieberman} & 9 & 3 & 0 & 20 & 9 \\ \textit{John McCain} & 1 & 1 & 10 & 0 & 3 \\ \textit{Joe Biden} & 0 & 6 & 18 & 14 & 24 \end{array}$$

Matrices: matrix rank

- The original matrix is what we call **full rank**: none of its rows or columns are linearly dependent upon another column
- The modified matrix is what we call not full rank or short rank, since we can take the Hillary row, multiply it by a scalar (in this case 2) and end up with the Biden row of the matrix (aka, the two rows are linearly dependent)
- If *either* any rows *or* any columns are linearly dependent, the matrix is not full rank. We get excited because based on its dimensions in this case, we think the cosponsorship matrix will provide five rows worth of unique information, when in reality, it's only providing four rows of unique information

Matrices: matrix rank

- Why is having full rank important? If we don't have full rank, we might have a non-unique solution for $\hat{\beta}$

- In applied contexts, we call this multicollinearity: e.g., if GDP and $\frac{GDP}{1{,}000{,}000}$ are included in the same regression, linear model commands return an error since at least two of the columns in the covariate matrix are linearly dependent. If you lose a column of information, your matrices are no longer conformable.

- Additional things to consider:
  - Why can we include GDP and $log(GDP)$, or GDP and GDP^2, in the same regression but not include GDP and $\frac{GDP}{1{,}000{,}000}$? Answer is that squaring and log are not *linear* transformations while multiplying by a scalar is; and we are only concerned with transformations that induce linear dependence

  - Are linear independence and statistical independence the same thing? Answer is no, for instance GDP and $log(GDP)$ are linearly independent, but they are not statistically independent (because if we know log(GDP), we have more information...in fact perfect information...to know GDP)