Week 03

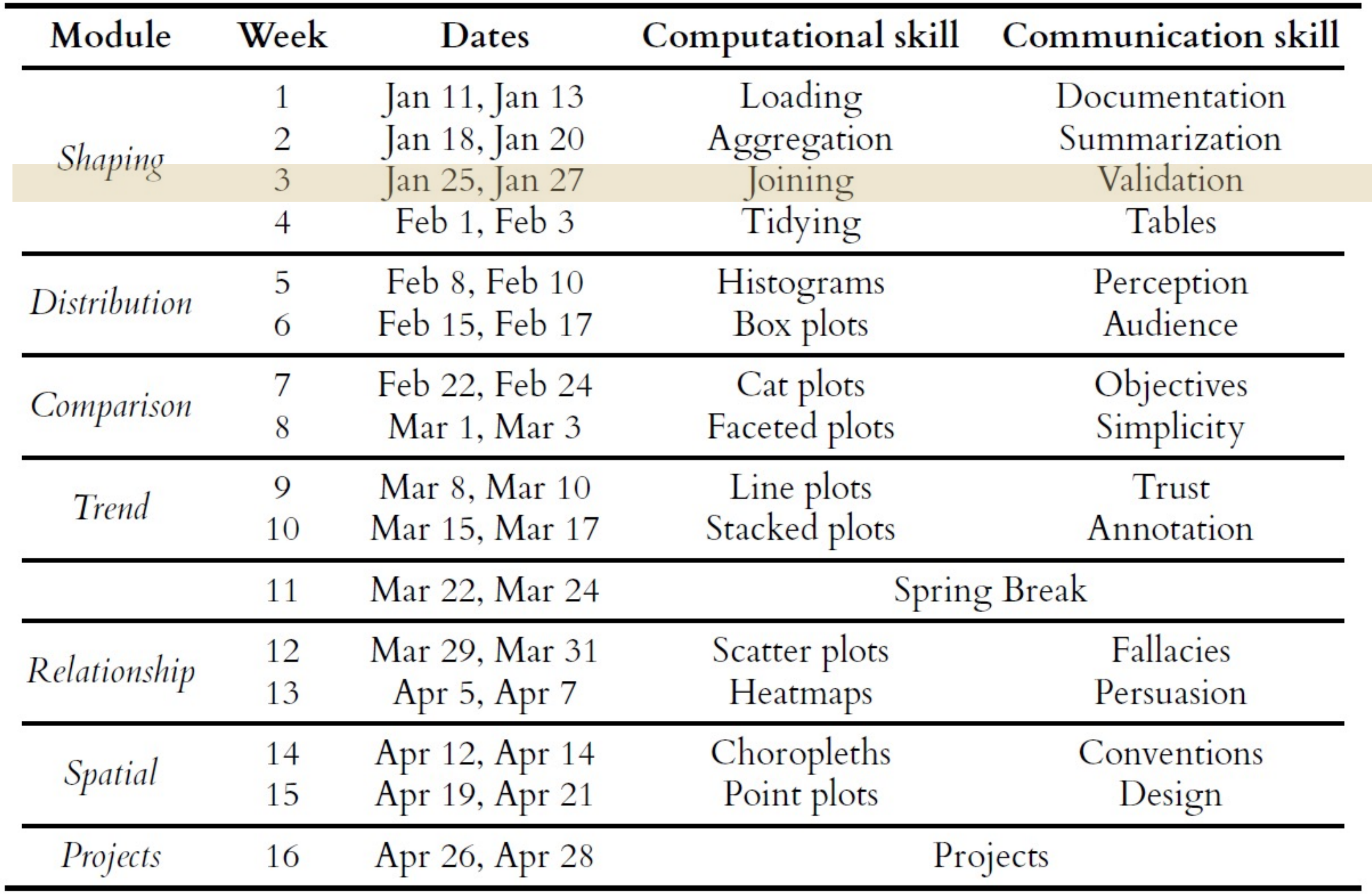
Combining & Validation

INFO 3402: Information Exposition

Brian Keegan, Ph.D.
Assistant Professor, Information Science
brian.keegan@colorado.edu

University of Colorado
Boulder

Course Overview



Module	Week	Dates	Computational skill	Communication skill
Shaping	1	Jan 11, Jan 13	Loading	Documentation
	2	Jan 18, Jan 20	Aggregation	Summarization
	3	Jan 25, Jan 27	Joining	Validation
	4	Feb 1, Feb 3	Tidying	Tables
Distribution	5	Feb 8, Feb 10	Histograms	Perception
	6	Feb 15, Feb 17	Box plots	Audience
Comparison	7	Feb 22, Feb 24	Cat plots	Objectives
	8	Mar 1, Mar 3	Faceted plots	Simplicity
Trend	9	Mar 8, Mar 10	Line plots	Trust
	10	Mar 15, Mar 17	Stacked plots	Annotation
	11	Mar 22, Mar 24	Spring Break	
Relationship	12	Mar 29, Mar 31	Scatter plots	Fallacies
	13	Apr 5, Apr 7	Heatmaps	Persuasion
Spatial	14	Apr 12, Apr 14	Choropleths	Conventions
	15	Apr 19, Apr 21	Point plots	Design
Projects	16	Apr 26, Apr 28	Projects	

Recap – Module 01

Week	Skills	Datasets
01	Data science mindset; loading data; documentation	Census; Boulder+Broomfield weather
02	Types of data; groupby-aggregation; pivot tables	CDC deaths; time use
03	Concatenation; joining/merging	Baseball; county cannabis, crime, population, & COVID
04	TBD	TBD
Module Assignment 01 due on Tuesday, February 08 by 11am		

Weekly overview

- Notebook exercises (ungraded) – Assigned Tuesdays and reviewed Thursdays
- Add a Visualization Critique (ungraded) – Discussed in class on Thursdays
- Weekly Assignment (graded, 2%) – Assigned Tuesdays and due Fridays
- Weekly Quiz (graded, 1%) – End of class on Thursdays (12pm)

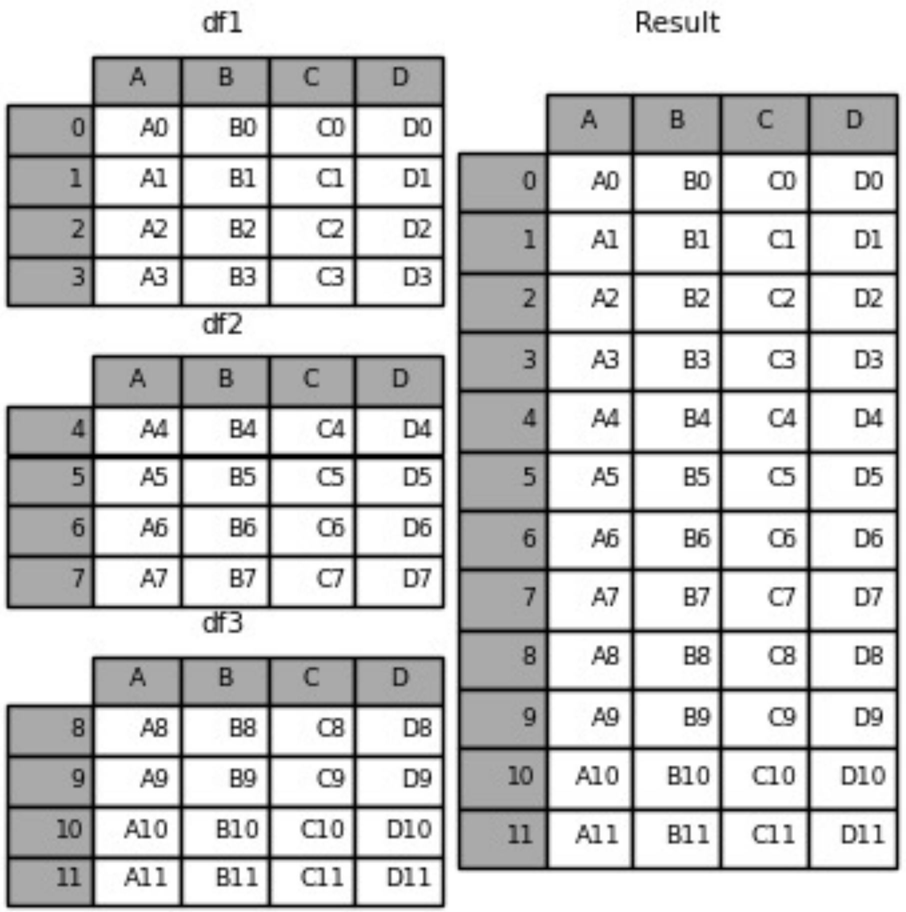
Combining Data

Concatenating

- DataFrames with similar columns but different rows can be combined with `concat`
  - Like “stacking” DataFrames *on top of* each other!
    - Similar(ish) columns and many more rows
  - Data from different times, locations, *etc*.

- Put each DataFrame to be concatenated into a container like a `list` or `dict` and pass the container to `concat`

- Careful with the indices on the resulting DataFrame! Concatenated DF preserves parents’ indices
  - The “ignore_index” parameter is your friend



df1

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

df2

	A	B	C	D
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7

df3

	A	B	C	D
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

Result

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

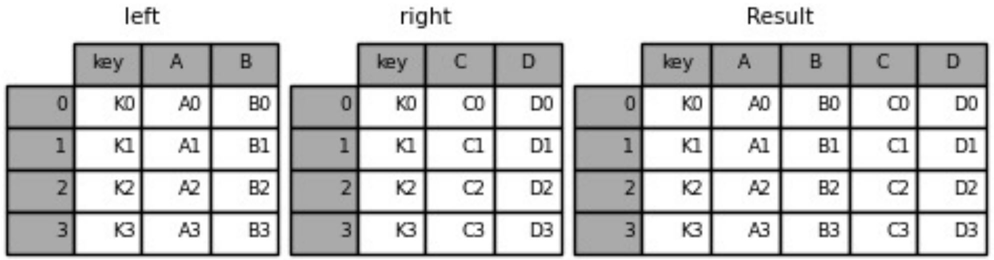
https://pandas.pydata.org/docs/user_guide/merging.html#concatenating-objects

Merging

- DataFrames with at least one column in common can be combined with `merge` or `join`
  - Like stacking DataFrames *next to* each other
    - Similar(ish) rows and many more columns
  - Different data with elements in commons

- Combine DataFrames a pair at a time: left & right
  - Identify the column (or index) in each DataFrame with the values to combine in the other DataFrame
    - Can be multiple columns!

- Biggest (and hardest!) decision is *how* to join
  - left, right, inner, outer
  - ”indicator” parameter can be your friend!



left

	key	A	B
0	K0	A0	B0
1	K1	A1	B1
2	K2	A2	B2
3	K3	A3	B3

right

	key	C	D
0	K0	C0	D0
1	K1	C1	D1
2	K2	C2	D2
3	K3	C3	D3

Result

	key	A	B	C	D
0	K0	A0	B0	C0	D0
1	K1	A1	B1	C1	D1
2	K2	A2	B2	C2	D2
3	K3	A3	B3	C3	D3

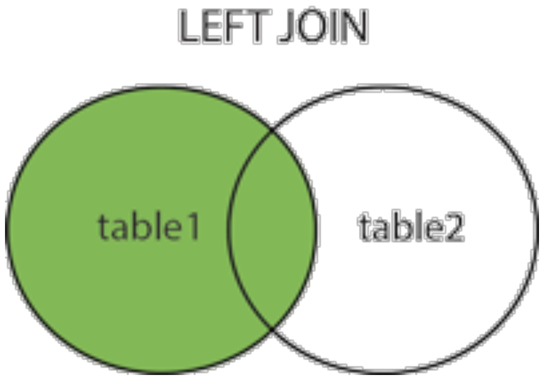
https://pandas.pydata.org/docs/user_guide/merging.html#concatenating-objects

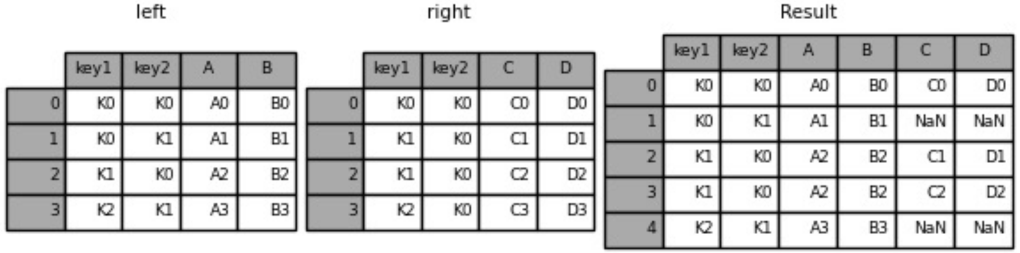
Left join

- Preserve the keys in the left DataFrame
  - Drop the rows on right if keys not present in left
  - Add NaNs to rows on left if not present on right

- left = { (K0,K0), (K0,K1), (K1,K0), (K2,K1) }
- right = { (K0,K0), (K1,K0), (K1, K0), (K2, K0) }

- Left join
  - Left loses nothing
  - Right loses (K2, K0)
  - NaNs for right columns’ (K0,K1) and (K2,K1) values





left

	key1	key2	A	B
0	K0	K0	A0	B0
1	K0	K1	A1	B1
2	K1	K0	A2	B2
3	K2	K1	A3	B3

right

	key1	key2	C	D
0	K0	K0	C0	D0
1	K1	K0	C1	D1
2	K1	K0	C2	D2
3	K2	K0	C3	D3

Result

	key1	key2	A	B	C	D
0	K0	K0	A0	B0	C0	D0
1	K0	K1	A1	B1	NaN	NaN
2	K1	K0	A2	B2	C1	D1
3	K1	K0	A2	B2	C2	D2
4	K2	K1	A3	B3	NaN	NaN

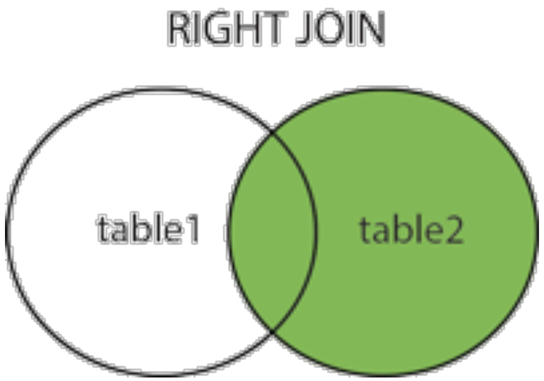
https://pandas.pydata.org/docs/user_guide/merging.html#brief-primer-on-merge-methods-relational-algebra

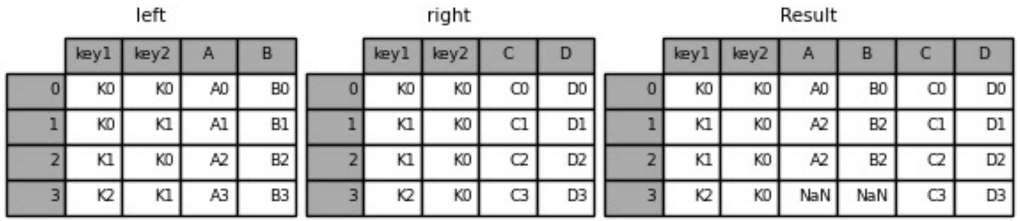
Right join

- Preserve the keys in the right DataFrame
  - Drop the rows on left if keys not present in right
  - Add NaNs to rows on right if not present on left

- left = { (K0,K0), (K0,K1), (K1,K0), (K2,K1) }
- right = { (K0,K0), (K1,K0), (K1, K0), (K2, K0) }

- Right join
  - Right loses nothing
  - Left loses (K0, K1) and (K2,K1)
  - NaNs for left columns' (K2,K0) values




left

	key1	key2	A	B
0	K0	K0	A0	B0
1	K0	K1	A1	B1
2	K1	K0	A2	B2
3	K2	K1	A3	B3

right

	key1	key2	C	D
0	K0	K0	C0	D0
1	K1	K0	C1	D1
2	K1	K0	C2	D2
3	K2	K0	C3	D3

Result

	key1	key2	A	B	C	D
0	K0	K0	A0	B0	C0	D0
1	K1	K0	A2	B2	C1	D1
2	K1	K0	A2	B2	C2	D2
3	K2	K0	NaN	NaN	C3	D3

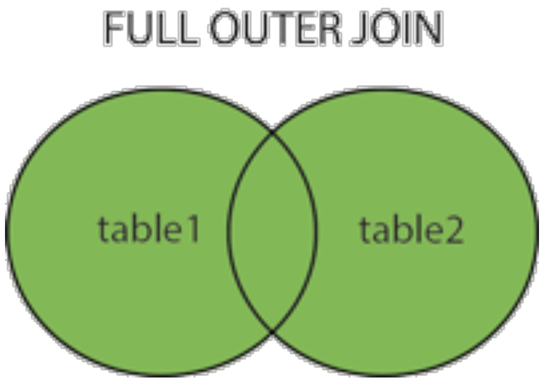
https://pandas.pydata.org/docs/user_guide/merging.html#brief-primer-on-merge-methods-relational-algebra

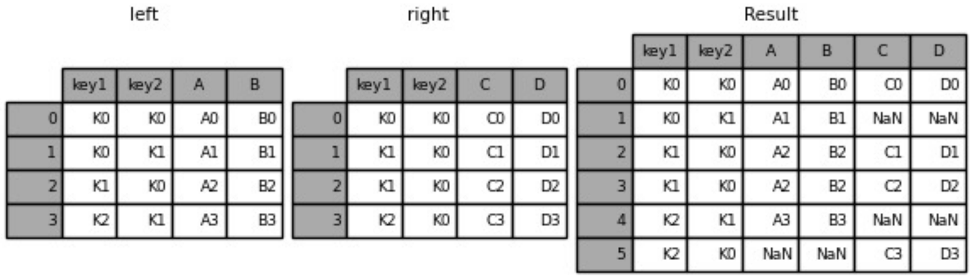
Outer join

- Preserve **all** keys in the right DataFrame
  - No rows dropped but lots of NaNs added

- left = { (K0,K0), (K0,K1), (K1,K0), (K2,K1) }
- right = { (K0,K0), (K1,K0), (K1, K0), (K2, K0) }

- Outer join
  - Nothing lost
  - NaNs for left columns’ (K2,K0) values
  - NaNs for right columns’ (K0,K1) and (K2,K1) values





left

	key1	key2	A	B
0	K0	K0	A0	B0
1	K0	K1	A1	B1
2	K1	K0	A2	B2
3	K2	K1	A3	B3

right

	key1	key2	C	D
0	K0	K0	C0	D0
1	K1	K0	C1	D1
2	K1	K0	C2	D2
3	K2	K0	C3	D3

Result

	key1	key2	A	B	C	D
0	K0	K0	A0	B0	C0	D0
1	K0	K1	A1	B1	NaN	NaN
2	K1	K0	A2	B2	C1	D1
3	K1	K0	A2	B2	C2	D2
4	K2	K1	A3	B3	NaN	NaN
5	K2	K0	NaN	NaN	C3	D3

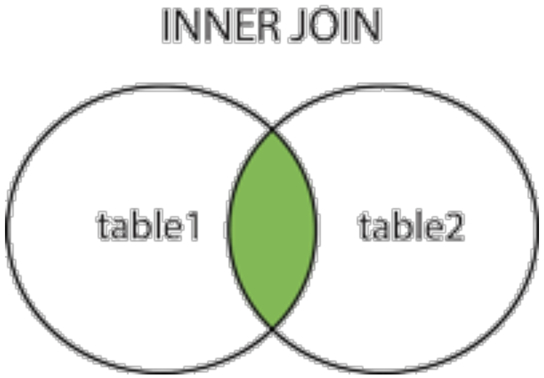
https://pandas.pydata.org/docs/user_guide/merging.html#brief-primer-on-merge-methods-relational-algebra

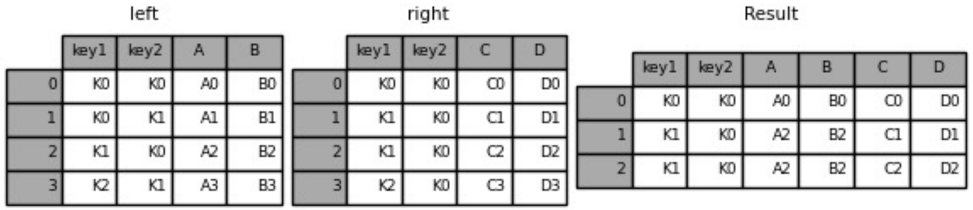
Inner join

- Preserve **only** keys in both DataFrames
  - Lots of rows dropped but no NaNs added

- left = { (K0,K0), (K0,K1), (K1,K0), (K2,K1) }
- right = { (K0,K0), (K1,K0), (K1, K0), (K2, K0) }

- Inner join
  - (K2,K0), (K0,K1), (K2,K1) lost
  - Only (K0,K0) and (K1,K0) preserved





left

	key1	key2	A	B
0	K0	K0	A0	B0
1	K0	K1	A1	B1
2	K1	K0	A2	B2
3	K2	K1	A3	B3

right

	key1	key2	C	D
0	K0	K0	C0	D0
1	K1	K0	C1	D1
2	K1	K0	C2	D2
3	K2	K0	C3	D3

Result

	key1	key2	A	B	C	D
0	K0	K0	A0	B0	C0	D0
1	K1	K0	A2	B2	C1	D1
2	K1	K0	A2	B2	C2	D2

https://pandas.pydata.org/docs/user_guide/merging.html#brief-primer-on-merge-methods-relational-algebra

Weekly Assignments

Feedback

- `.loc[ index_name, column_name ]` to return DataFrame values in a particular spot
  - https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#selection-by-label
- Multilevel indexing: `.loc[ (index_1, index_2), column_name ]`
  - https://pandas.pydata.org/pandas-docs/stable/user_guide/advanced.html#advanced-indexing-with-hierarchical-index
- Boolean indexing: create a series of True/False values + pass back to DF to get only True rows
  - https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#boolean-indexing
- Adding a new column: assign column name as string in brackets to DF and then operation
  - https://pandas.pydata.org/docs/getting_started/intro_tutorials/05_add_columns.html
- `.max()` returns the maximum value itself; `.idxmax()` returns the location of the maximum value
  - Powerful in combination with .loc!
  - https://pandas.pydata.org/pandas-docs/stable/user_guide/basics.html#index-of-min-max-values

Feedback

- Return the specific value we're looking for → not just a series of values containing it
- "Find the maximum value of X in Y"
  - **BAD**: sorted DF/Series/array of Xs with the highest value at the top we have to look up
  - **GOOD**: sorting, Boolean indexing, idxmax-ing, *etc*. and printing/returning a single largest value
  - We've been giving partial or full credit so far, but won't in future!
- Homeworks are released on Tuesdays, get started early!
  - Get questions in over email or in Thursday lecture
  - Samantha and I are may not be available on Thursday nights to help with last-minute debugging
- Join the class Discord server for collaborating, questions, *etc*.
  - https://discord.gg/Jmzq3dHQv4

Module Assignments

Module Assignment 01

- Module Assignment 01 will be due Tuesday, February 08 by 11am
  - Tag assignment with “INFO3402S22MA1”
- Write up an exploratory data analysis on any of the datasets we’ve used in Weeks 01-04
  - Census; Weather; CDC deaths; Time use; Baseball; County cannabis/crime/COVID
- Should: (1) be a good question; (2) follow the EDA checklist; (3) include at least one numeric table and one visualization; and (4) communicate for general audience

- Rubric
  - ≦ **A**: New/novel data source; good motivation for question; high-quality visualizations; excellent communication
  - ≦ **B**: Trivial or previously-analyzed data; mediocre question; off-the-shelf visualizations; average communication
  - ≦ **C**: Duplicates existing tutorials/docs; trivial EDA with poor question; no or poor visualizations; poor communication

Using Medium

- We will be using Medium, a popular blogging platform launched in 2012
- A WYSIWYG interface like WordPress, upvoting like Reddit, tags like Instagram/Tumblr/TikTok
- Create account using your Colorado.edu email address and no need to purchase membership
- Posts will be submitted to our class publication and tag
  - https://medium.com/information-expositions-s2022 and Info3402s22
- Use the Medium Help Center or Google around for tips and advice on writing Medium posts

- Submit your Medium user name here

Medium check-in

- Can you access Medium.com?
- Have you created an account?
- Have you followed some writers, publications, or keywords that interest you?
- **WITHOUT PUBLISHING**, draft a 250-word story on what you hope to accomplish this semester
- Can you find this draft under your account?
- Can you add this draft to the “Information Expositions S2022” class publication?
- Have I added your username to the class publication as a writer?

Characteristics of a good question

- Question should be of interest to audience
- Question should not already been answered
- Question should be plausible
- Question should be answerable
- Question should be specific

From Peng & Matsui (2018). *The Art of Data Science.*

Exploratory data analysis checklist

1. **Formulate your question** → see “Characteristics of a good question”
2. **Read in your data** → Is it properly formatted? Perform cleanup activities
3. **Check the packaging** → Make sure there are the right number of rows & columns, formats, *etc*.
4. **Look at the top and bottom of data** → Confirm that all observations are there
5. **Check the “n”s** → Identify “landmark” values and to check expectations (number of states, *etc*.)
6. **Validate against an external data source** → Right order of magnitude, expected distribution, *etc*.
7. **Make a plot** → Checking and creating expectations about the shape of data and appropriate analyses
8. **Try an easy solution** → What is the simplest test for your question?

From Peng & Matsui (2018). *The Art of Data Science*.

Communicating results of an analysis

- **Audience**: Who cares about your question and results? Why? Do they share your expertise?
- **Content**: What is appropriate background? What are your objectives? Breadth *vs*. depth?
- **Style**: Formality of venue? Jargon? When to do discussion?
- **Attitude**: What kind of feedback do you want? What actions do you want audience to take?

From Peng & Matsui (2018). *The Art of Data Science*.

Weekly Assignment & Next Class

Readings

- Questions for Friday’s Weekly Quiz 03 will be drawn from these readings

- pandas Getting Started - How to combine data from multiple tables?
- pandas User Guide - Database-style DataFrame joining/merging
- Chan, L. (2021). "Python Tricks: How to Check Table Merging with Pandas." TDS.

Weekly Assignment 03

- **Skills**: Concatenating and merging data
- **Data**: County-level longitudinal data about cannabis sales, crime, and COVID

- Due Friday before midnight on Canvas
  - Save an HTML version of your notebook with all output present
    - File > Download as > HTML (.html)
  - Upload the HTML file to Canvas

Next Class

- Review concepts and exercises from last class
- Critique a data narrative or visualization
  - Post a link and a few sentences about a data visualization on Canvas discussion (ungraded/optional)
- Time to work on Weekly Assignment
  - Weekly Assignment due on Friday by submitting HTML notebook to Canvas before midnight
- Weekly quiz at the end of class (12:00–12:30)