THE UNIVERSITY OF CHICAGO

MASTERS IN COMPUTATIONAL SOCIAL SCIENCE
THE UNIVERSITY OF CHICAGO

MACS 30111

M8: LAST DAY!!!!

Announcements

- Switch to accelerated track: need MPCS exam
  - https://masters.cs.uchicago.edu/page/placement-exams-0
  - Register, take the exam
- Final exam schedule
  - Dec 5th, 10-12pm
- Review
  - Textbook
  - Team tutorial
  - Short exercise
  - Programming assignment
  - Extra exercise

Common Summary Statistics

`DataFrame.count`

Count number of non-NA/null observations.

`DataFrame.max`

Maximum of the values in the object.

`DataFrame.min`

Minimum of the values in the object.

`DataFrame.mean`

Mean of the values.

`DataFrame.std`

Standard deviation of the observations.

`DataFrame.select_dtypes`

Subset of a DataFrame including/excluding columns based on their dtype.

```
s_trees.count()
```

```
spc_common    652169
status        683788
health        652172
borough       683788
latitude      683788
longitude     683788
dtype: int64
```

```
s_trees.status.value_counts()
```

```
Alive    652173
Stump     17654
Dead      13961
Name: status, dtype: int64
```

Summary statistics for each column: *df.describe()*

Summary of categorical values: *df.value_counts()*

Pairwise correlation of columns: *df.corr()*

Reference: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.describe.html

Applying Functions to Data

`Series.apply`

For applying more complex functions on a Series.

`DataFrame.apply`

Apply a function row-/column-wise.

`DataFrame.applymap`

Apply a function elementwise on a whole DataFrame.

```
trees["tree_dbh"].apply(np.sqrt)
```

Reference: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.apply.html

Topics:

- Introduction to Pandas
- Creating DataFrames and Series
- Working with DataFrames and Series
- Applying functions to data
- **Group, combine, and pivoting**
- Visualizing DataFrames and Series

Grouping

```
s_trees.groupby("borough").size()
```

```
borough
Bronx              85203
Brooklyn          177293
Manhattan          65423
Queens            250551
Staten Island     105318
dtype: int64
```

```
s_trees.groupby(["borough", "status"]).size()
```

```
borough        status
Bronx          Alive      80585
               Dead        2530
               Stump       2088
Brooklyn       Alive     169744
               Dead        3319
               Stump       4230
Manhattan      Alive      62427
               Dead        1802
               Stump       1194
Queens         Alive     237974
               Dead        4440
               Stump       8137
Staten Island  Alive     101443
               Dead        1870
               Stump       2005
dtype: int64
```

Unstack: converts series with hierarchical index into a DataFrame

Coding practice: 4.3.5

Combining DataFrames

- **concat( )**
  - perform concatenation along an axis
  - while performing set logic of the indexes on other axes
  - make a full copy of the data

- **merge( )**
  - Standard database join operations between DataFrame or Series objects

- **join( )**
  - join on index
  - combine columns of two differently indexed DataFrames into a single one

Reference: https://pandas.pydata.org/docs/user_guide/merging.html

Pivoting

Convert a long and thin DataFrame into a short and wide DataFrame

- A column for index
- A column to supply column names for the new DataFrame
- A column to fill the new DataFrame

Coding practice: 4.3.6

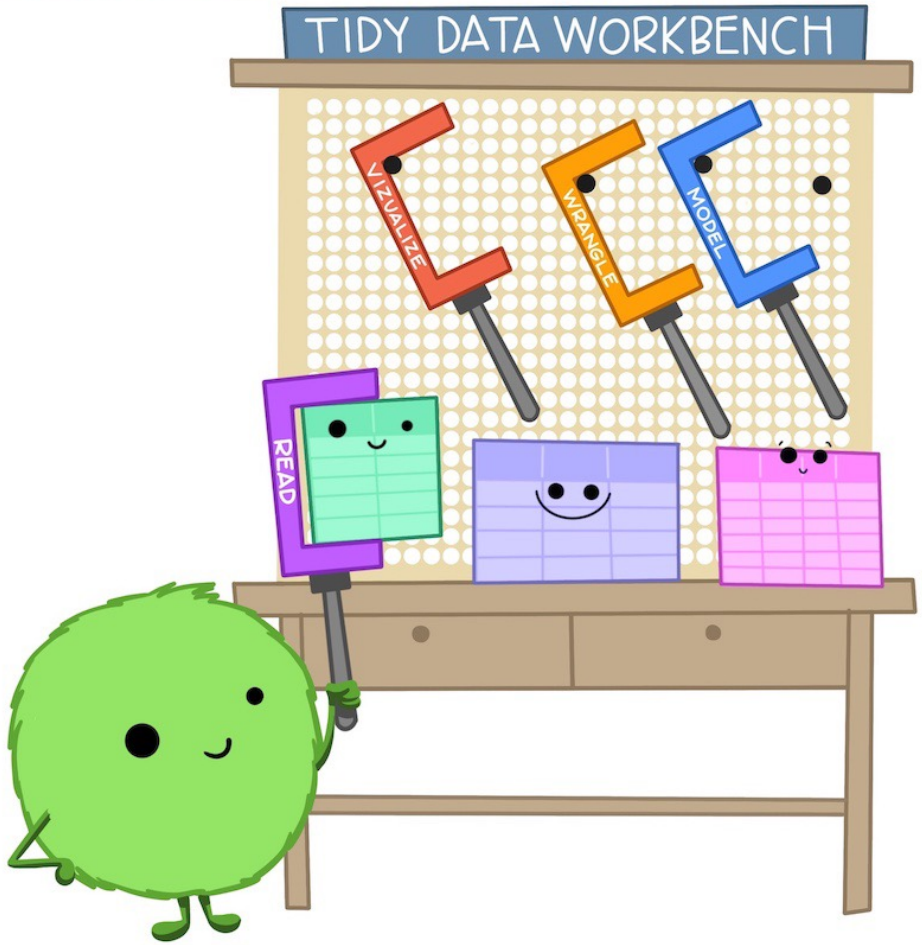
Principles of Tidy Data

- Tidy data is a standard way of mapping the meaning of a dataset to its structure. A dataset is messy or tidy depending on how rows, columns and tables are matched up with observations, variables and types.
  In **tidy data**:

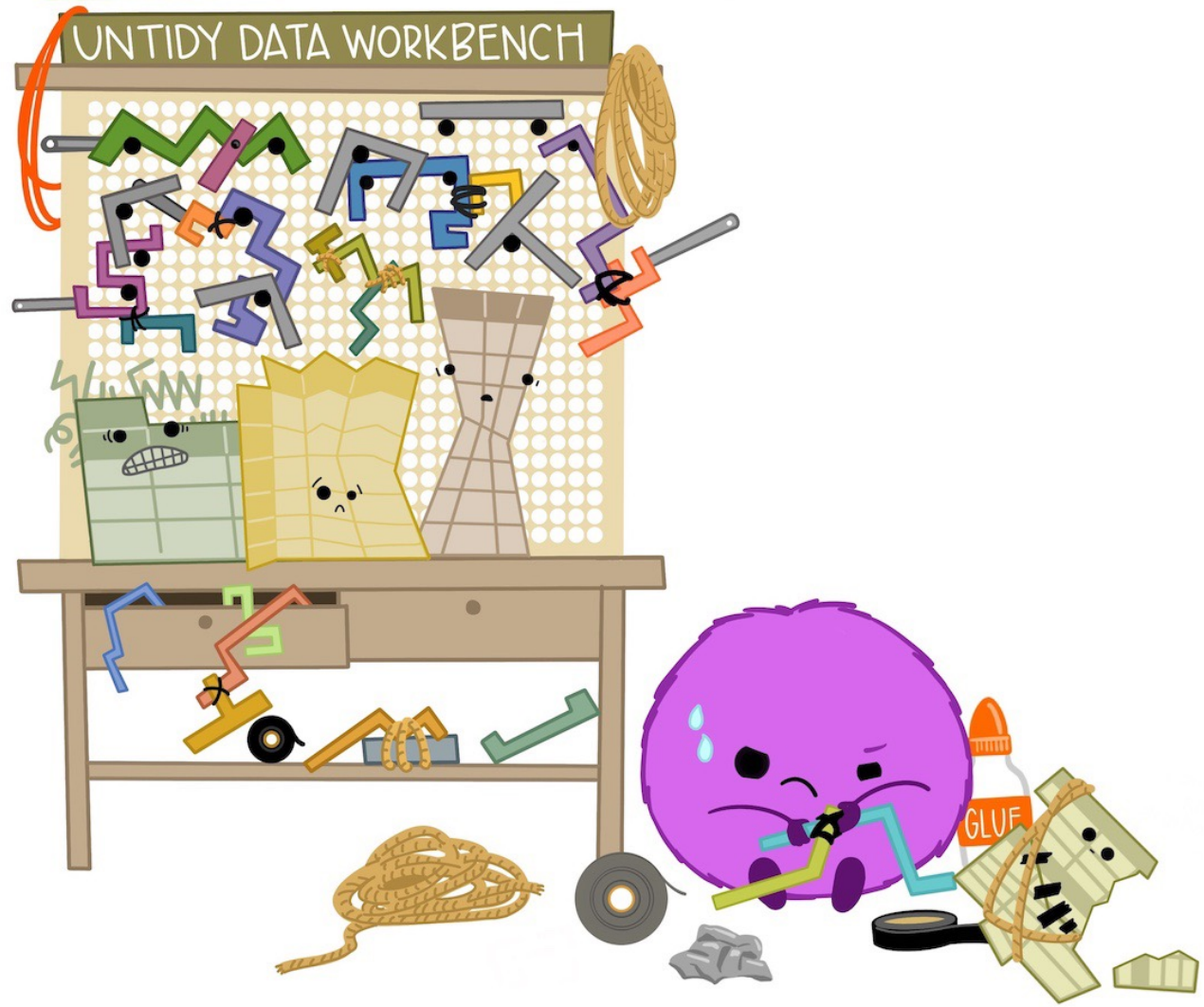
1. Every column is a variable.
2. Every row is an observation.
3. Every cell is a single value.

https://tidyr.tidyverse.org/articles/tidy-data.html

When working with tidy data, we can use the same tools in similar ways for different datasets...



...but working with untidy data often means reinventing the wheel with one-time approaches that are hard to iterate or reuse.



wide

id	x	y	z
1	a	c	e
2	b	d	f

long

id	key	val
1	x	a
2	x	b
1	y	c
2	y	d
1	z	e
2	z	f

https://bookdown.org/martin_monkman/DataAnalyticsCodingFundamentals/tidy-data.html

Pivoting

Convert a long and thin DataFrame into a short and wide DataFrame

- A column for index
- A column to supply column names for the new DataFrame
- A column to fill the new DataFrame

Coding practice: 4.3.6

CLASS REVIEW TIME

- What did we cover, what matters?
- Collaborative doc: https://docs.google.com/drawings/d/19H9LWFdsqJlVWxFnT9vxC_lvsgeu2tMAhQFZoymsUPM/edit?usp=sharing

For another time!

Topics:

- Introduction to Pandas
- Creating DataFrames and Series
- Working with DataFrames and Series
- Applying functions to data
- Group, combine, and pivoting
- **Visualizing DataFrames and Series**

Plotting with Pandas

Using the plot method:

df.plot(kind='line') # produces a line plot by default
s.plot()

Reference: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.html

Topics:

- Introduction to Pandas
- Creating DataFrames and Series
- Working with DataFrames and Series
- Applying functions to data
- Group, combine, and pivoting
- Visualizing DataFrames and Series

More details: https://pandas.pydata.org/docs/user_guide/

Memory lane

Course Description

- This course is the first in a three-quarter sequence that teaches computational thinking and skills to students in the sciences, mathematics, economics, etc. The course will cover abstraction and decomposition, simple modeling, basic algorithms, and programming in Python. Applications from a wide variety of fields serve both as examples in lectures and as the basis for programming assignments. In recent offerings, students have written programs to simulate a model of housing segregation, determine the number of machines needed at a polling place, and analyze tweets from presidential debates.

Assignments and Skills

- Consider your bootcamp exam and your first assignment – how it felt re: time and difficulty
- Look at your most recent PA – consider how that felt re: time and difficulty!
- Tremendous progress and thoughtfulness over the term in assignments and reflections

How we did it

- **Weekly assignments** balanced between:
  - TT – answers provided for easy comparison
  - SE – intermediate-level for slightly more challenge
  - PAs – more/most challenging but smaller number required
- **Exams**:
  - In-class to assess recall / facility with concepts (mirrors interview questions, etc.)
  - Take-home to assess ability to work on more complex projects
- **Class sessions**:
  - Applied hands-on lessons where we interact and grapple with code and problems
  - Use of Ed discussion to provide channels for communication
  - Regular and expanded OH around tricky assignments and important deadlines

Evaluations: purpose and role

- Help us understand how your learning connected with / came from the course
- Additional (instructor-designed) questions:
  - Which element of the course was the most helpful for your learning and why? (e.g. assignments (PAs, SEs, TTs), course lecture, Ed discussion board, office hours, etc.)
  - Which element of the course would you like to see changed and how? (e.g. assignments (PAs, SEs, TTs), course lecture, Ed discussion board, office hours, etc.) Please be specific.
  - Are there any additional topics, concepts, or assignments you would like to see in the future? Please explain.

Where we go from here

- EXAM (!!!)
- Final course reflection + evaluation
  - Reflection = your own growth and progress
  - Evaluation = course design and structure
- KEEP IN TOUCH!
  - I want to hear how your future projects go, what you do, the courses you take, etc!
  - Feedback: are there skills you later use that are/not helpful? In what way(s)? How can this course be improved to help you in your later coursework?