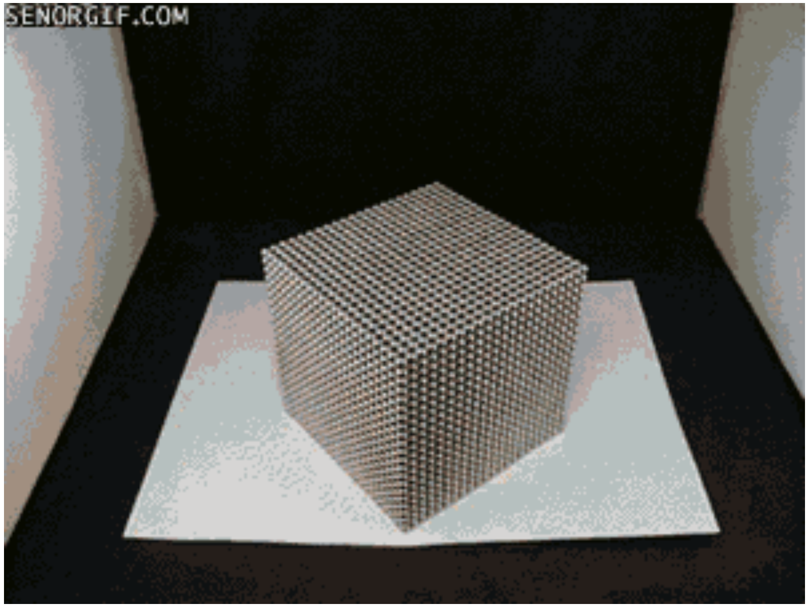


Managing Research Data

ELIZABETH WICKES, DATA CURATION SPECIALST

RESEARCH DATA SERVICE

@ELLIEWIX WICKES1@ILLINOIS.EDU

RESEARCH DATA SERVICE

Who am I: Professional Stuff

Curator with the Research Data Service

- We’re in the main library
- But we go all over campus!
  - Data doesn’t understand this ‘green street’ concept
- Have a poster in the poster session

(soon to be…) GSLIS alum

Former Wolfram|Alpha curator

Sociology/Psychology undergrad

- Technology & Society

Who am I: Personal Stuff

Co-organizer of the CU Python User Group (Py-CU)

Datasets I’ve made:

- Fanfiction
  - Checkout the poster in the poster session!
- Jeopardy players
- Python & Ruby conference talks and speaker genders

Things I’ve done:

- Webscraping
- Topic modeling
- Text mining
- GIS

What is RDS?

Data policies, best practices, campus resources, archiving, & preservation

Data Management Plan reviews

Illinois Data Bank (coming soon!)

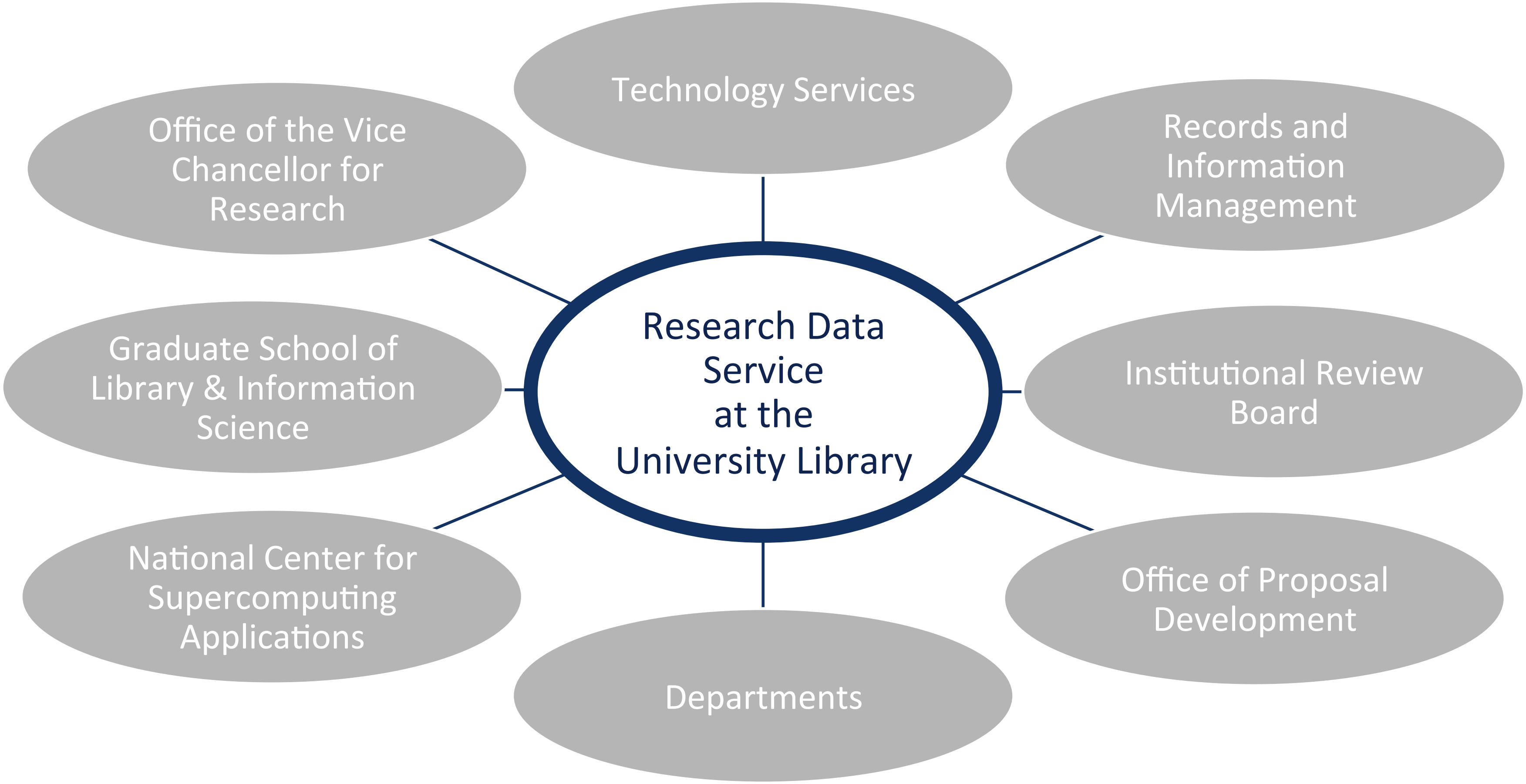
Workshop series on data management, documentation, and data publishing

- February 16, February 23, & March 1 (Tuesdays)
- 10:00-11:00am in Library 314

Open hours:

- Tuesdays 3-5pm in Scholarly Commons
- Piloting Thurs from 12-2 in Grainger 404

What is RDS?



Research Data Service at the University Library
Technology Services
Office of the Vice Chancellor for Research
Records and Information Management
Graduate School of Library & Information Science
Institutional Review Board
National Center for Supercomputing Applications
Office of Proposal Development
Departments

Why should you care?

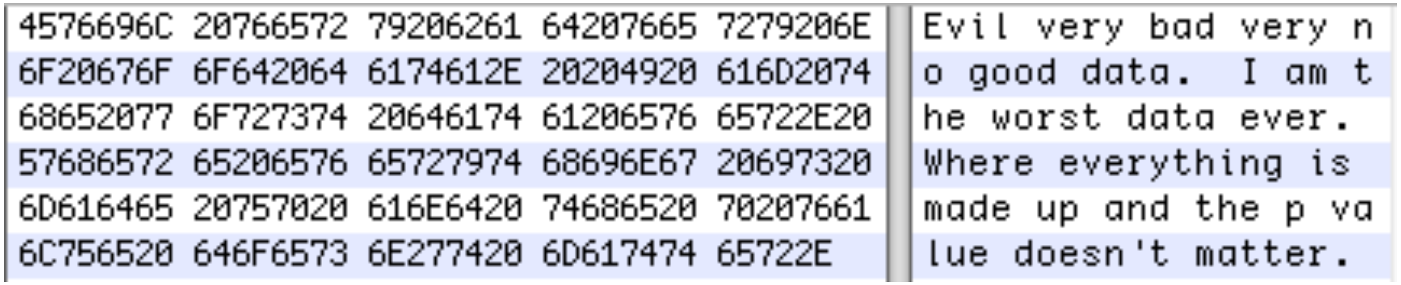
(just kidding…)

Why should you care?

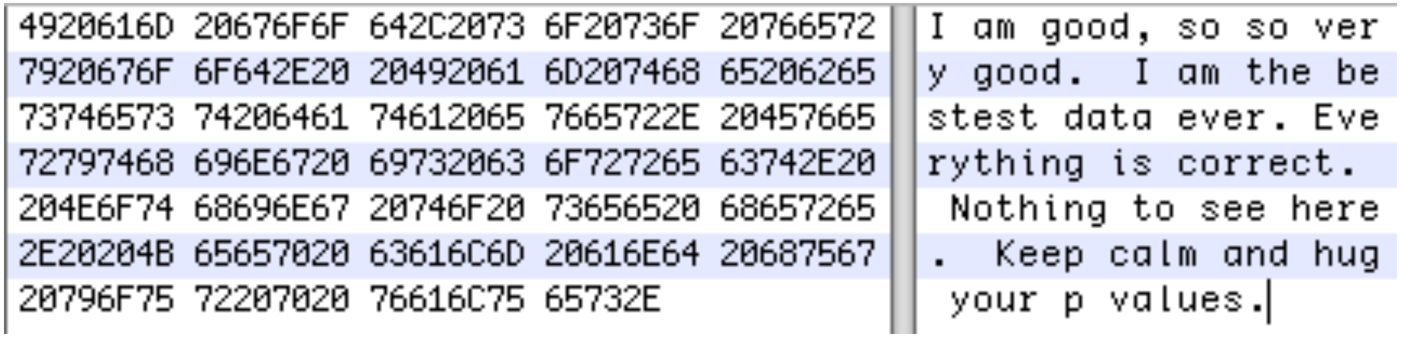
A bad apple can be easy to spot…

(C) Adrian Van Leen for OPENPHOTO.NET CC:PublicDomain

But spotting bad data can be harder…



```
4576696C 20766572 79206261 64207665 7279206E   Evil very bad very n
6F20676F 6F642064 6174612E 20204920 616D2074   o good data.  I am t
68652077 6F727374 20646174 61206576 65722E20   he worst data ever. 
57686572 65206576 65727974 68696E67 20697320   Where everything is 
6D616465 20757020 616E6420 74686520 70207661   made up and the p va
6C756520 646F6573 6E277420 6D617474 65722E     lue doesn't matter.
```



```
4920616D 20676F6F 642C2073 6F20736F 20766572   I am good, so so ver
7920676F 6F642E20 20492061 6D207468 65206265   y good.  I am the be
73746573 74206461 74612065 7665722E 20457665   stest data ever. Eve
72797468 696E6720 69732063 6F727265 63742E20   rything is correct. 
204E6F74 68696E67 20746F20 73656520 68657265    Nothing to see here
2E20204B 65657020 63616C6D 20616E64 20687567   .  Keep calm and hug
20796F75 72207020 76616C75 65732E               your p values.
```

Why do I care?

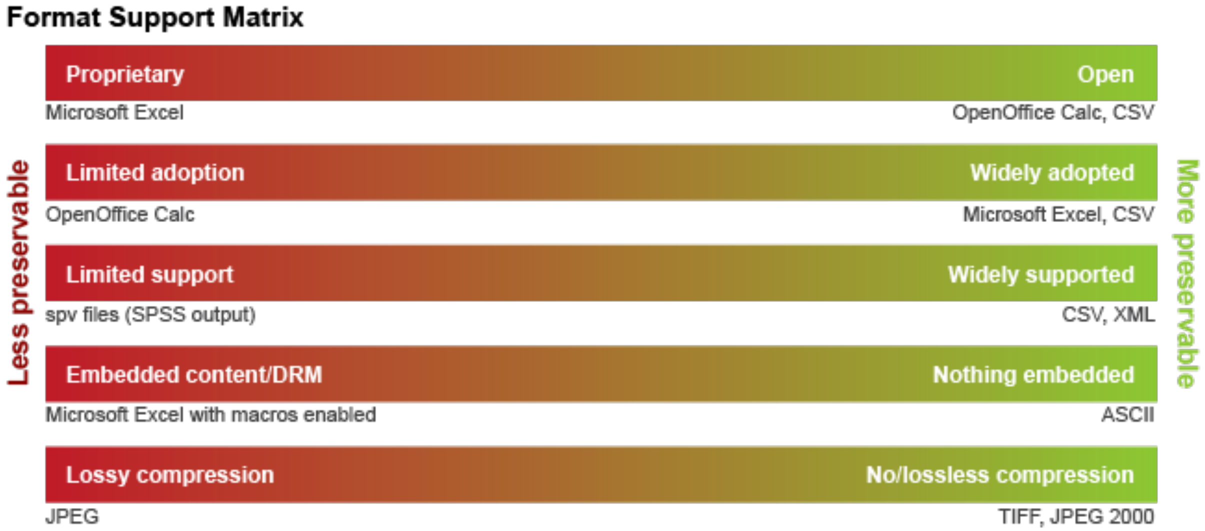
- Your job isn't to read all the scholarly literature about this stuff
- You're here to do research
- Data management is kind of our jam
- And we're here to help

Key behaviors for sanely managing your research data

1. Use open formats
2. Organize your folders, files, and naming conventions
3. Document your design processes and choices
4. BACKUP YOUR DATA!
5. Make a plan and stick to it

Use open formats

- Will you be able to open your SPSS/SAS/ArcGIS files when you leave the university or the site license is dropped?
- Plain text and open source are your friends
  - TXT, CSV, JSON, XML, Open Office, GIMP, R, Python, etc.



Format Support Matrix

Less preservable

More preservable

Proprietary — Open

Microsoft Excel — OpenOffice Calc, CSV

Limited adoption — Widely adopted

OpenOffice Calc — Microsoft Excel, CSV

Limited support — Widely supported

spv files (SPSS output) — CSV, XML

Embedded content/DRM — Nothing embedded

Microsoft Excel with macros enabled — ASCII

Lossy compression — No/lossless compression

JPEG — TIFF, JPEG 2000

http://www.library.illinois.edu/sc/services/data_management/file_formats.html

Organize your stuff: why?

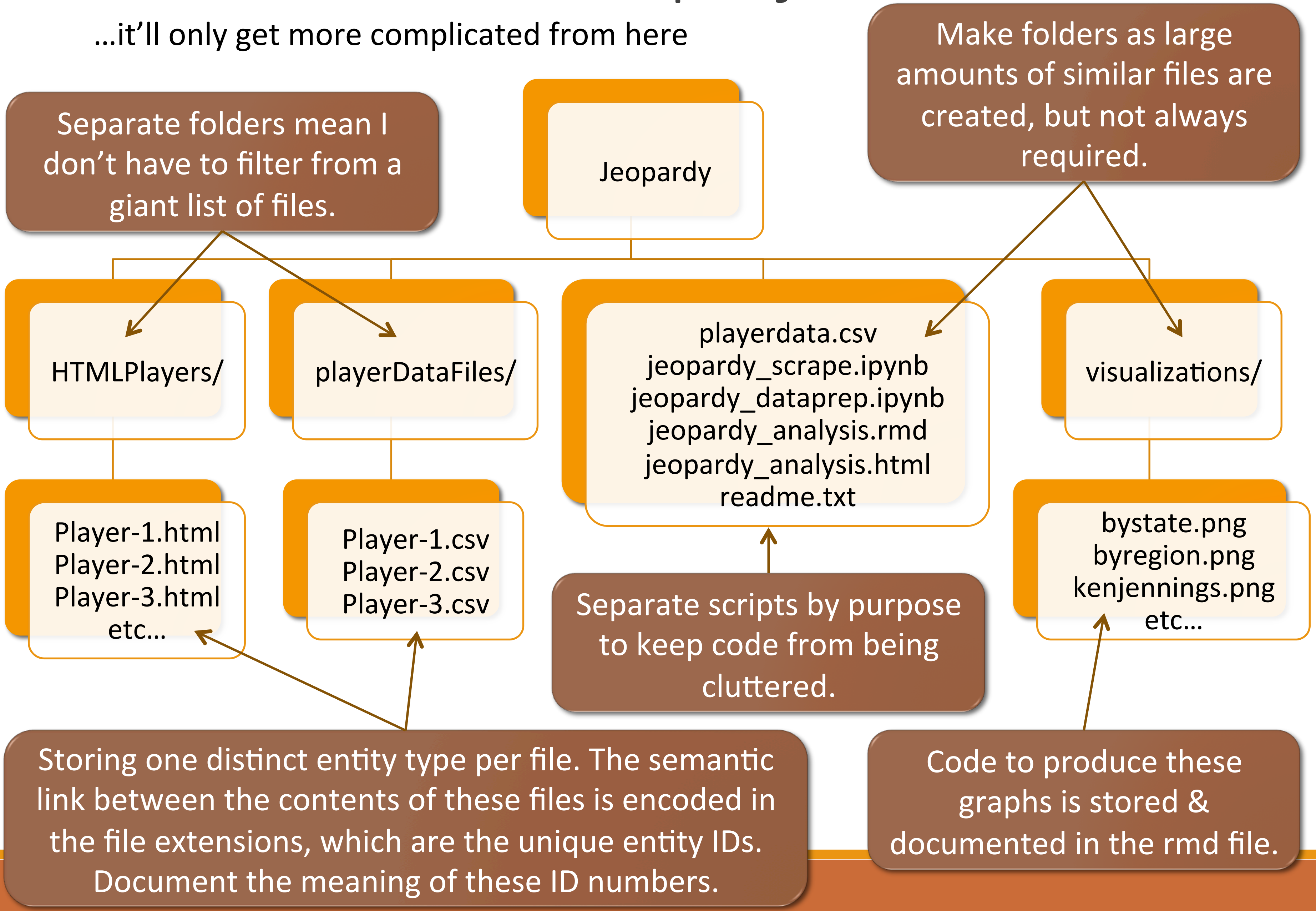
- Computational projects spiral out of control pretty fast
  - How many projects do you have every semester?
  - How many files does each project have?
- Computational projects can be explosive
  - I have 2.7 million XML files on this computer
  - That I've had for 6 months
- A basic project might have:
  - Raw data, clean data, & processed data
  - Scripts (and the previous versions)
  - Various output files
  - Analysis report in Word, LaTeX, or MD
  - All your visualizations

Organize your stuff: names matter

- Keep names short, but use descriptive information
- Avoid spaces or special characters when possible
- Encode meaningful & unique values into your programmatically generated files
  - Datetime: rescrape_report_2016-01-02.csv
  - Iteration: group_run_003.dat
  - Unique ID key: player_id_12094.html
  - Add a version number: processing_v2.0.py
- Key: think about what the project might need and find a balance
  - Too detailed-> you won't follow your own system
  - Not enough detail -> you won't find anything ever
  - Focus on what you can be consistent with

A minimal lone wolf project

…it'll only get more complicated from here



Separate folders mean I don't have to filter from a giant list of files.

Jeopardy

Make folders as large amounts of similar files are created, but not always required.

HTMLPlayers/

playerDataFiles/

playerdata.csv
jeopardy_scrape.ipynb
jeopardy_dataprep.ipynb
jeopardy_analysis.rmd
jeopardy_analysis.html
readme.txt

visualizations/

Player-1.html
Player-2.html
Player-3.html
etc…

Player-1.csv
Player-2.csv
Player-3.csv

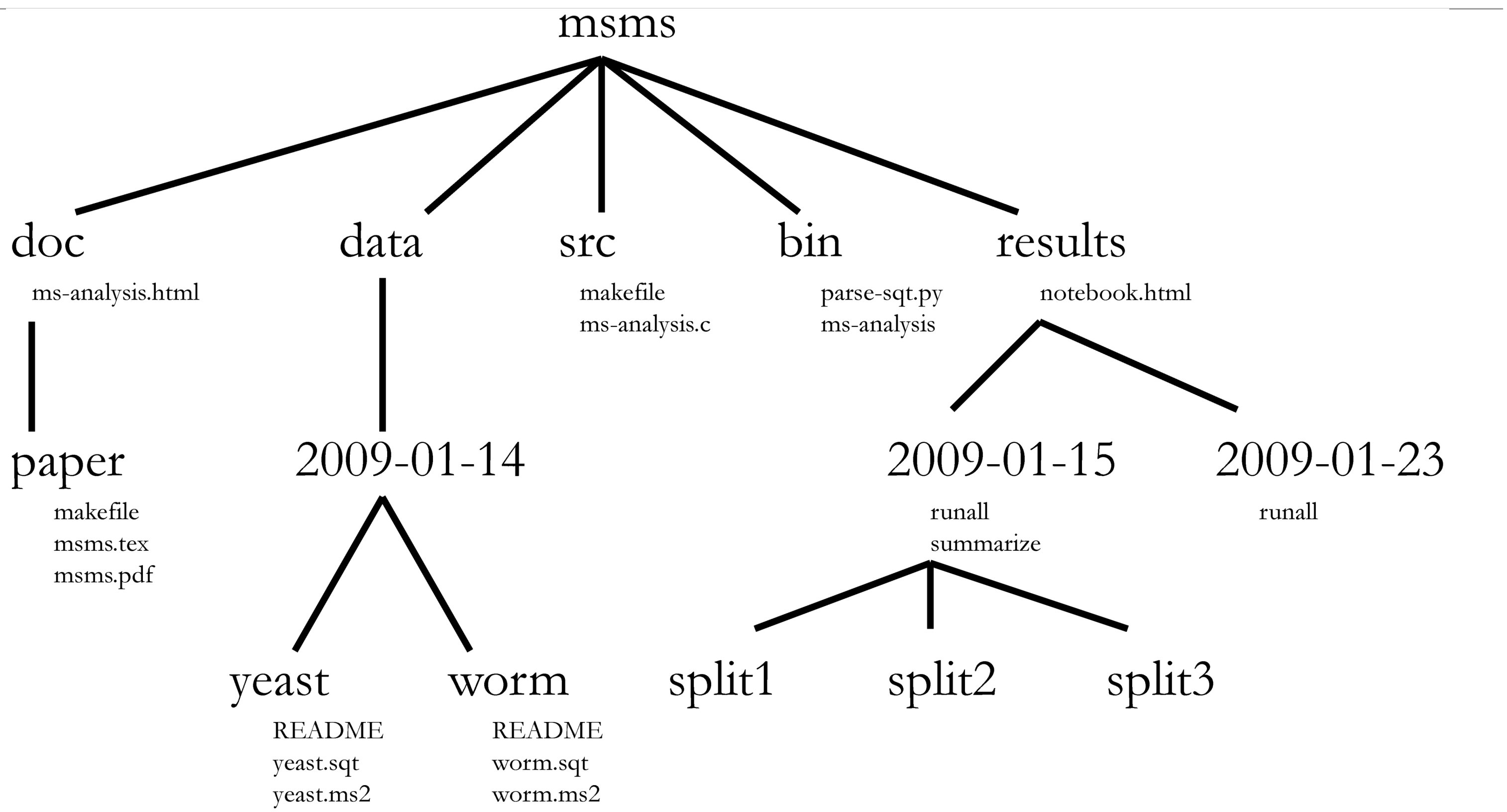
Separate scripts by purpose to keep code from being cluttered.

bystate.png
byregion.png
kenjennings.png
etc…

Storing one distinct entity type per file. The semantic link between the contents of these files is encoded in the file extensions, which are the unique entity IDs. Document the meaning of these ID numbers.

Code to produce these graphs is stored & documented in the rmd file.

Noble (2009)'s Bioinformatics project structure



Noble WS (2009) A Quick Guide to Organizing Computational Biology Projects. PLoS Comput Biol 5(7): e1000424. doi:**10.1371/journal.pcbi.1000424**

Document your computational decisions

- How many processing steps go into creating a single visualization?
- Results don't magically appear out of the blue
  - You did stuff to make those results
- Keep notes for your future self when you need to explain what you've done
- Better yet, retain all those decisions within your code
- Will you remember these choices when you are getting ready to publish on the project?

Practice defensive and executable documentation

- People will question your work
- You will have typos in your graphs.
- Solve both these problems by using scripts to process your data, run your analysis, and produce your figures.
  - E.g. Jupyter, R Markdown, or etc
- Leave your original data intact and use your script to perform all the programmatic transformations before analysis.
  - Documents that you've done (don't forget to add comments to your code) and preserves the original value.

You still have to write stuff down

- Elements of a great readme file:
  1. Names, dates, contact info, and roles of all folk on the project
  2. List of files and their relationships, e.g. a processing workflow
  3. Copyright/licensing dependencies and declarations
  4. Limitations, next steps, FYI, etc. about the data
  5. Funding, grant numbers, and other institutional support information
  6. Technical requirements for running the scripts or opening the data
     - E.g. version numbers for processing software and packages
  7. Data sources, dates, access information, etc

- Also consider, as separate files:
  - Data dictionary, folder definitions, file naming, and computational workflow

BACKUP ALL THE THINGS

- Some day…
  - Your computer will die
  - Your USB drive will be stolen
  - You'll leave the university
- Backup your entire computer
  - As security and permissions allow
- Even your software.
  - And know where you can open your data in a pinch

3-2-1 rule of file storage

- What does 3-2-1
  - Three copies of your data
  - Over two kinds of media
  - One of your copies is remote
- What this might mean for you:
  - one copy on your local hard drive
  - another an external hard drive
  - the whole project is a github repo and/or synced to Box
  - Export your proprietary files out to something open now and then
    - Great to do when wrapping up a project or you finalize results
    - SPSS -> CSV; Word -> TXT, Jupyter -> .py, etc.
- The more you can automate this stuff the less work it is.
  - Box/Google Drive (beware! https://answers.uillinois.edu/page.php?id=54880)
  - End of day commits to github/bitbucket/etc
  - Backup software with external drives (e.g. Time Machine)

Make a plan

- Start early in the process
  - Before you’ve gotten your file structure into a snarled mess
- This work can be emotional
  - Don’t decide how to organize stuff in a fit of rage
  - resultsIHATEYOUSOMUCHv5.rmd
- Aim for sufficient rather than perfect
  - Ignore the database normalization haters
  - Database nerds: simmer down
- You don’t need to have a perfect memory if you can predict what you would have done
  - But do your collaborators work the same way?
- Be flexible and adapt
  - Also use structures that allow for change

BEEN WORKING IN R FOR 4 HOURS
HAVEN'T HIT ANYTHING OR CRIED YET
memegenerator.net

Where to go from here?

- Workshops!
  - Intro to Data Management
    - February 16, 10-11am, Library 314
  - Documentation and Organization for Data
    - February 23, 10-11am, Library 314
  - Making Research Data Public
    - March 1, 10-11am, Library 314
- Open hours!
  - Tuesdays from 3-5 in the Scholarly Commons
- Contact us
  - researchdata@library.illinois.edu
  - @ILresearchdata

- Some slide content adapted from:
  - Brianna Marshall
    - @notsosternlib
  - Cameron Cook
    - @cameron_ccook
  - Heidi Imker
    - @imkerinfo

- Elizabeth Wickes
  - wickes1@illinois.edu
  - @elliewix