ABHIJIT DASGUPTA, PHD

BEST PRACTICES FOR DATA ANALYSIS

PREAMBLE

THERE IS NO “BEST” PRACTICE

- There are some principles
- There are some safety nets
- There are some tried and trusted procedures
- Everything else is what works for you, your team and your organization
- I’m sharing what works for me

OUR COMMON OBJECTIVES

- Maximize
  - Time to think about a project
  - Reliability/Reproducibility
- Minimize
  - Data errors
  - Programmer/Analyst errors
  - Programming time
  - Re-orientation time when re-visiting

OUR COMMON INCLINATION

- Once we get a data set
  - Dig in!!
  - Start “playing” with tables and figures
  - Try models on-the-fly
  - Cut-and-paste into reports/presentations

DON’T DO THIS!!

A STORY FROM SEVEN YEARS AGO

- 25 year study of rheumatoid arthritis
- 5600 individuals
- Propensity-adjusted survival analyses, plus several cool analyses
- Needed data cleaning, validation and munging and some custom computations, along with parallel computations
- Lots of visualizations and graphs

A STORY FROM SEVEN YEARS AGO

- Resulted in a muddle of 710 files (from 4 data files)
- Unwanted cyclic loops for intermediate data creation, and difficult to determine where I created them
- Lots of ad-hoc decisions and function creation within scripts
- Almost impossible to re-factor and clean up
- Had to return to project for 3 papers and revision cycles!!

WHO IS YOUR MOST LIKELY CLIENT?

- Yourself
    - 3 months, 1 year, 5 years from now
- The biggest reason for maintaining good practices is your own mental sanity

MY 80/20 RULES

MORE TIME THAN YOU WANT, LESS TIME THAN YOU NEED

- 80% of a project is discussion with collaborators about understanding central and peripheral questions
- Of remaining 20%,
    - 80% of time is cleaning and munging the data to make it usable for answering the questions, and exploring the data
    - Remaining time is analysis and reporting

PROJECT ORGANIZATION

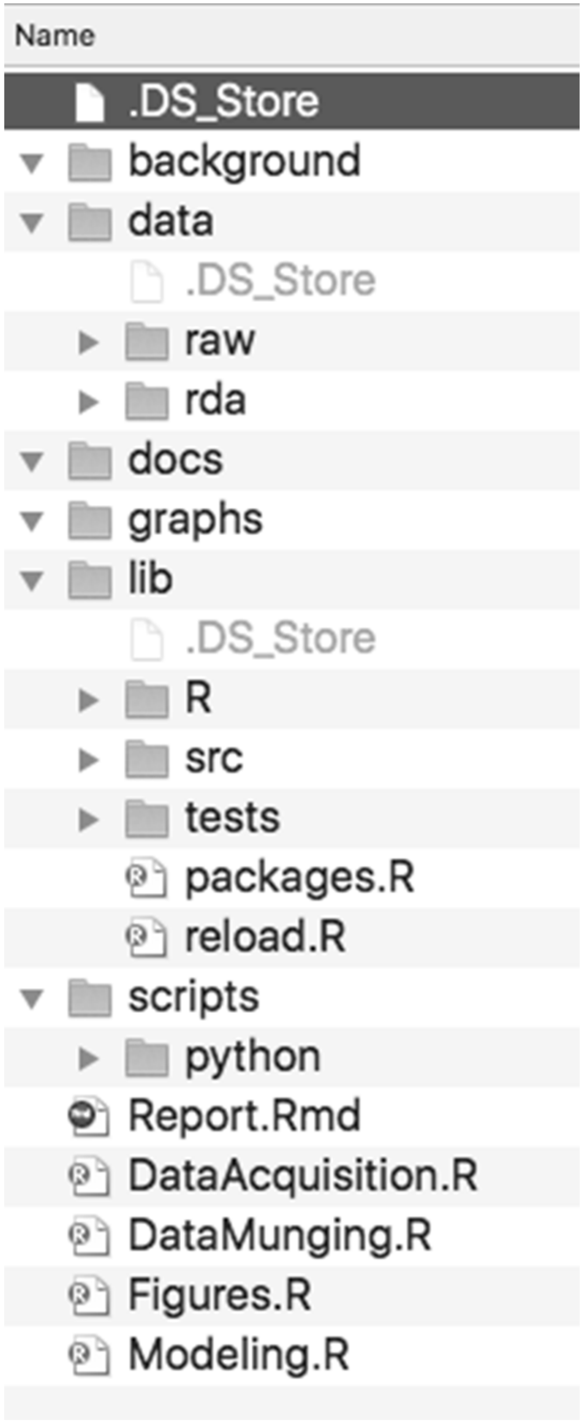
USE A TEMPLATE TO ORGANIZE EACH PROJECT

- Before you even get the data
- Set up a particular folder structure where
  - you know what goes where
  - you can have canned scripts/packages set things up
- Make sure it's the SAME STRUCTURE EVERY TIME
- Next time you visit, you don't have to go into desperate search mode

USE A TEMPLATE TO ORGANIZE EACH PROJECT

- In R, the package ProjectTemplate can help
  - Too involved for my taste
- In Python create virtualenv or conda environments and populate with a particular structure
- Maybe you use a Docker container to hold the template
- Figure out your general use case, and set this up to what you’re comfortable with

MY STRUCTURE



Name	
.DS_Store	
▼ background	Background materials
▼ data	
.DS_Store	
▶ raw	Raw data (storage, not to be touched)
▶ rda	Intermediate and final R data sets
▼ docs	Generated documents (docx, html, pdf)
▼ graphs	Graphs (pdf, png, tiff)
▼ lib	
.DS_Store	
▶ R	Custom R functions (all of them, without exception)
▶ src	Custom C/C++ functions
▶ tests	Unit tests
packages.R	List of R packages for the project
reload.R	Automated loading of functions and packages in a separate environment
▼ scripts	
▶ python	Scripts for file management and conversion
Report.Rmd	
DataAcquisition.R	
DataMunging.R	
Figures.R	
Modeling.R	

FILE NAMING CONVENTIONS

- Be explicit about what a file does in the name
  - File1.R, Script1.py, Program3.sas don’t help you
  - DataMunging.R, FitRandomForest.py is much better
  - Saves a lot of time and heartache

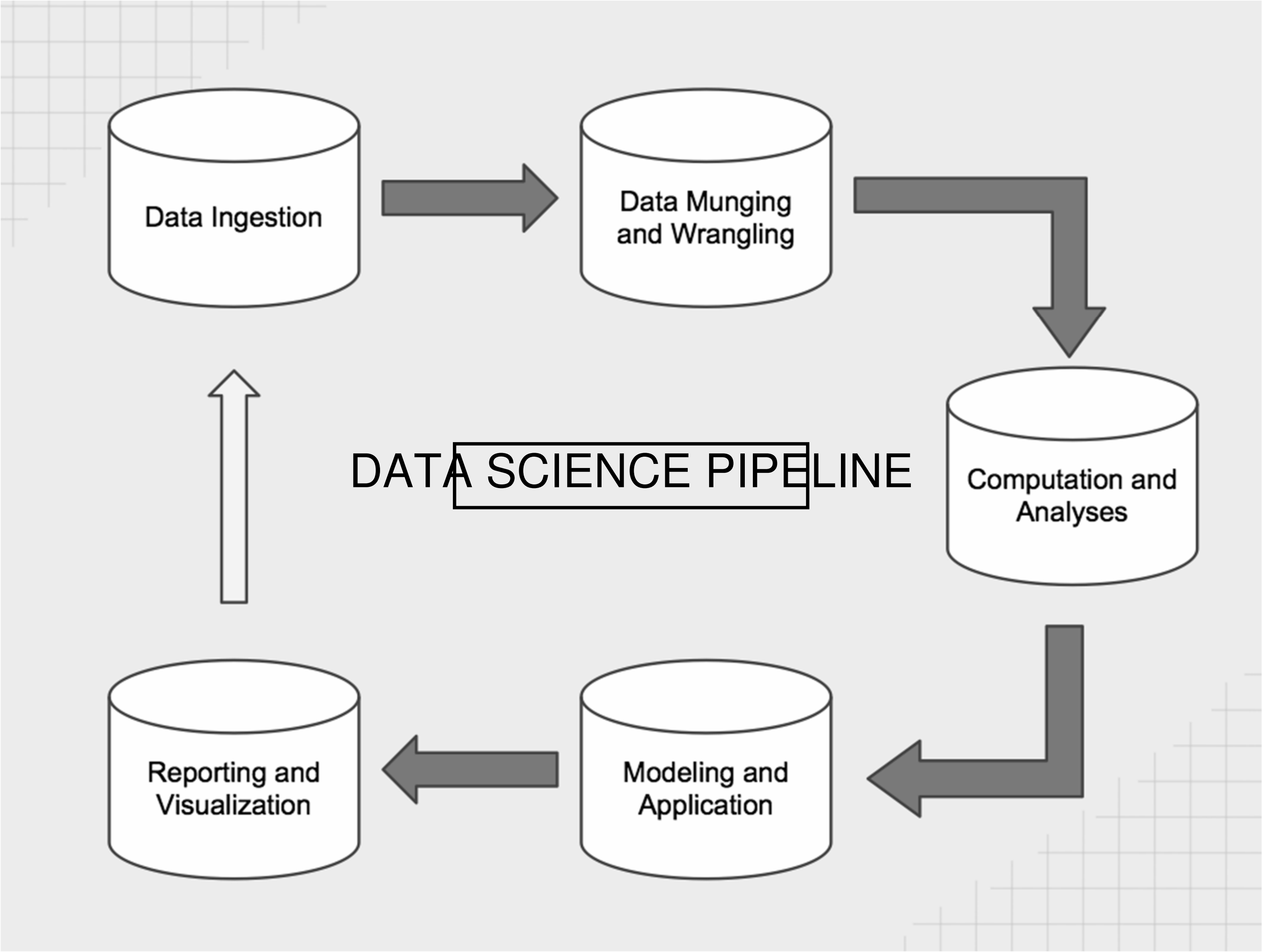
DOCUMENTATION

- Create at least one README
    - Write down the driving questions and purposes of the project
        - You know you won’t remember in 3 months

DOCUMENTATION

- Document code and functions as much as you can
    - Don’t be stingy!! You’ll thank yourself
    - Easy tools today
        - roxygen
        - Python’s __doc__ system
        - Comments

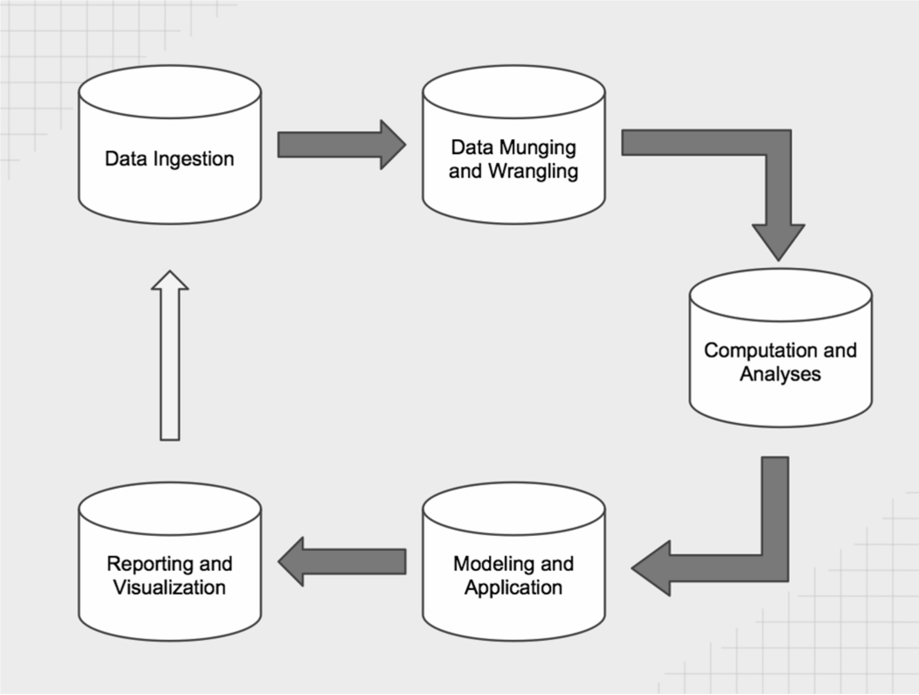
THE ANALYST’S PIPELINE



Practical Data Science Cookbook, Ojeda, Murphy, Bengfort, Dasgupta, 2014

CHOICES AND DECISIONS

- At each point in the cycle you have to make
  - conscious decisions
  - choices you’ll live with
- The idea of best practices is to make many of these decisions and choices safe, automatic and “no-brainers”, so we can concentrate on doing the fun stuff and not regret things later



DATA INGESTION

MICROSOFT EXCEL: THE 800 TONNE GORILLA

- Omnipresent, so “must be great”
- Easy to use (but easy enough to be dangerous)
- Not safe!!
  - Actions can’t be captured and reliably repeated
  - Errors can unknowingly and irretrievably cascade
  - Often unintended behavior based on formats and data types

MICROSOFT EXCEL: THE 800 TONNE GORILLA

- Train collaborators that colors aren’t machine readable.
- If they want to categorize rows, make a new variable, instead of coloring them
- A Jackson Pollock spreadsheet will only create unnecessary headaches (or migraines, or suicidal thoughts) for you. Don’t accept them.
- Spreadsheets are often poorly formatted, so go slowly and understand.
  - New tools like Jenny Bryan’s jailbreakr package

MICROSOFT EXCEL: THE 800 TONNE GORILLA

- Okay for data transport and storage
- May be good for “quick and dirty” stuff
  - This has a way of growing a life of its own
- I use it solely for data transport and storage.
- I keep it in read-only mode

MICROSOFT EXCEL: THE 800 TONNE GORILLA

- If you love Excel and can’t do without it:
  - Make sure you make a copy that you keep pristine
    - First thing you should do
    - ….with any data you receive
  - Be very very very careful
    - See Anil Potti / Duke U debacle
    - See Reinhart & Roganoff debacle
    - *The Economist*, “Excel errors and science papers” 9/7/16

RAW DATA STORAGE

- Text files
  - CSV, TSV
- Open data formats
  - RData, pickle
- Open-source databases
  - SQLite, Postgres

Makes data sharing very easy
Does not limit access to the data
Can always be opened and verified
Operating system agnostic

RAW DATA STORAGE

- Insist upon or create a data dictionary
  - You won’t remember what “rt_tst_29” is
  - Verifies both purpose and type of data in each column
  - Capture study design and other metadata about study

Jeff Leek’s datasharing repo

IMPORTING DATA

- Use scripts
  - Be explicit about options if needed
- R, Python, SAS, Matlab, Stata, shell, Perl, C/C++, Java
- Verify
  - size of data
  - types of variables

DATA MUNGING &

MANIPULATE DATA WITH CARE

- Keep a pristine copy of the data!!!
- Use scripts to manipulate data so you can reproduce
  - More importantly, you can catch analyst errors and fix
- Systematically verify and clean
  - Create your own SOP
- Document what you find and do
  - Lab notebook (Carl Boettiger's example)

MANIPULATE DATA WITH CARE

- The laws of unintended consequences are vicious, unforgiving and appear all too often during this stage
- For example, data types changed (factor to integer, say), and that cascades to later analyses
- Test your data at every stage to see if the structure you expect is still there

TIDY DATASETS ARE ALL ALIKE, BUT EVERY MESSY DATASET IS MESSY IN ITS OWN WAY

Hadley Wickham

PREFER TIDY DATASETS

- Each variable must have its own column
- Each observation must have its own row
- Each value must have its own cell

Computers like this

Search “hadley tidy data”, “tidyr”, “tidyverse”

CODE SUGGESTIONS

- Comment your code
  - Why you made a choice
- Develop naming conventions
  - Under_scored or CamelCase
  - Be explicit (what the heck does d37 mean again?)
  - You can follow established coding styles
    - Google has suggestions for R and Python

FORGIVING MISTAKES

- Use some kind of version control system
  - I use *git* for code and small data sets
  - I use Dropbox for larger data sets
- Learn a version control system
  - Git, Subversion, Mercurial
  - Use it!!

AVOIDING MISTAKES

- When you want to do something new that might corrupt existing data and analyses
  - Create a git branch
  - Develop and validate your work
  - When you’re sure, merge it back in

TRACK DATA PROVENANCE THROUGH THE PIPELINE

- Typically:

  Raw data >> Intermediate data >> Final data >>
  data for sub-analyses >> data for final tables and figures

- Catalog where you create each data set, and where it’s ingested
  - Script files have a habit of multiplying
- Make sure there are no loops!!

REALLY EXPLORE THE DATA, TAKE YOUR TIME

- Be fluent and creative in
    - summarizing the data
    - visualizing the data
    - discerning relationships
- Data immersion

SHARE PRELIMINARY ANALYSIS FOR A SNIFF

- Typically the analyst doesn’t have deep domain knowledge
- Share initial explorations with your colleagues so they pass the “sniff” test
  - Are data types what you expect
  - Are data ranges what you expect
  - Are distributions what you expect
  - Are relationships what you expect

SHARE PRELIMINARY ANALYSIS FOR A SNIFF

- Are anomalies you found reasonable?
  - Instrument error
  - Data recording error
  - True outlier
  - Wrong theoretical framework
  - Wrong study design
- This stuff is really important and requires deliberate brain power
- May require feedback loop and more thinking about the problem

COMPUTATION & ANALYSES

THE COMPUTER FOLLOWS DIRECTION REALLY WELL

- Use scripts/functions/macros to derive quantities you need for other functions
- Don’t hard-code numbers!
-
  ```
  runif(n=nrow(dat), min=min(dat$age), max=max(dat$age))
  ```

  vs

  ```
  runif(n=135, min=6, max=85)
  ```
- Minimizes potential errors in data transcription
  - These are really hard to catch

CREATE FUNCTIONS RATHER THAN COPY-PASTE CODE

- If you’re doing something multiple times, create a function
- Put the function in a separate file, stored in a particular place
  - Don’t hide it in general script files where other analyses are going on
  - Name the file so you know what’s in it
  - One function or one set of related functions (modules) per file
- Write the basic documentation NOW!

MACROS IN EXCEL

- Do macros rather than manipulation by hand
- Document your macros in one sheet of your file

TEST, TEST, TEST

- Write tests for your functions and check
- Many unit testing frameworks
- Test before prime time, not after you’ve moved further along
- Errors hard to catch later

MODELING
AND
APPLICATI

THINK!!

- Does my understanding of the data support the models and tests I planned to run?
- Do I need to modify the standard stuff?
- Do I need help?

STATISTICAL TESTING

- There are many guides about what the “right” test to do in different situations are (textbook cases)
- Unfortunately no situation you will face is ideal
- Think if the tests you’re using are right for the data you’re using it on
- Leverage computational testing (permutations, bootstraps) if needed
- Reflect if the test results gel with your exploratory “feel” for the data

STATISTICAL MODELING & MACHINE LEARNING

- Don’t just do one model
  - Computational time is really cheap!!
- Try many modeling “looks” at the data, see what they all say
  - Data is very susceptible to the Rashomon effect
  - If different models are saying qualitatively the same thing, maybe gain confidence in what you’re seeing

STATISTICAL MODELING AND MACHINE LEARNING

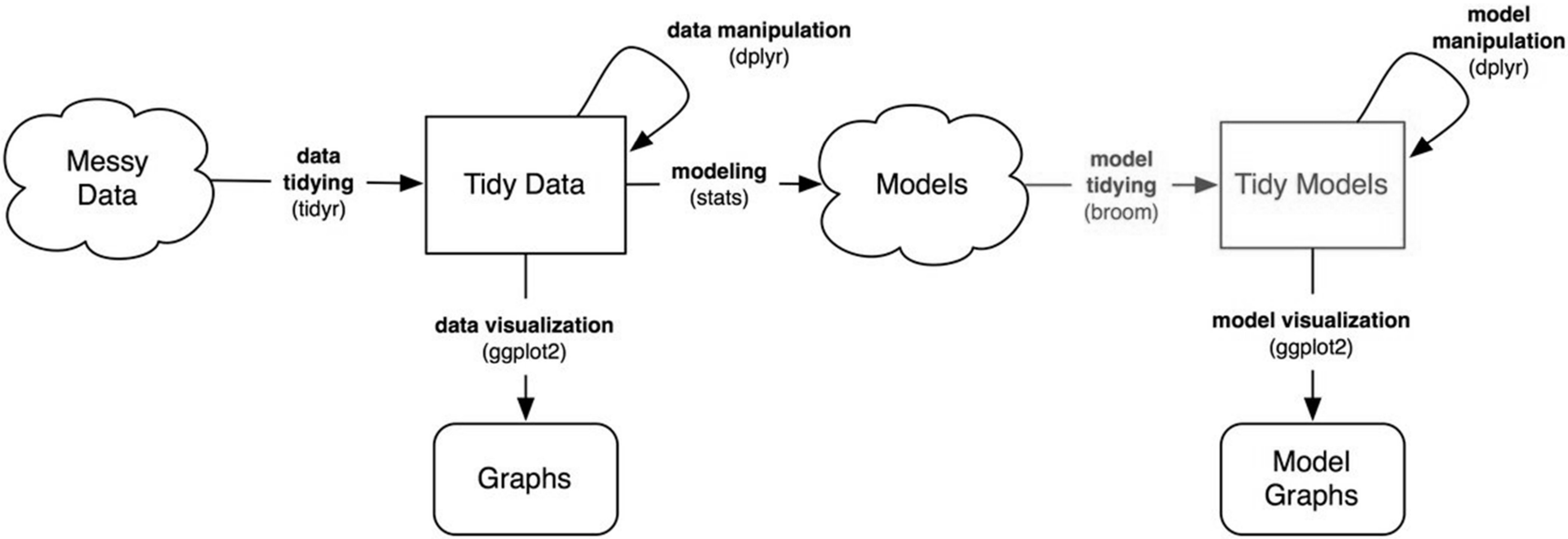
- Resist the temptation to do “cookie-cutter” models and be satisfied
- Watch out for confounders
- Think about the implications of your results
- Diagnostics are key!!
    - Adequate fit
    - Reasonable functional form
    - External validity

FEEDBACK LOOPS

- Modeling is inherently a feedback loop
- It’s okay to loop all the way back to the basic questions posed
- Make sure all the links ring true and are defensible
- Appreciate that no model is perfect
- Understand limitations

FEEDBACK LOOPS

- Compare model results
    - Tidy’ing up the results really helps
        - broom, estout
    - Use tables and graphs to evaluate your models



David Robinson, JSM 2016

REPORTIN
G &
VISUALIZA

KNOW WHERE FINAL TABLES AND FIGURES COME FROM

- I create separate files for creating figures and tables for the final paper
- Make sure you're using the right data to create them
    - It's been a long road, easy to get confused
    - Metadata helps!

LITERATE PROGRAMMING ROCKS!!

- Code and text intertwined
- Makes reports automated, reliable
- Prevents hard coding or copy-paste, which are error-prone
- RMarkdown, Sweave, Jupyter notebooks, Matlab notebooks
- Lots of capabilities today!!

KEEP VISUALS SUCCINCT

- Doesn’t need to be fancy, colorful
- Needs to be clear, to-the-point
- Make the points you're making obvious in the figure
- Sometimes you may need dynamic graphs to make the point
    - NY Times, Guardian, fivethirtyeight, other data journalism sites
- Read Tufte at least once

BAD VISUALS

- Learn when a visual is bad
- Plenty of examples of good and bad
  - FlowingData
  - The D3.js website
- Think if you need to make extra effort or mental gymnastics to understand a visualization
  - If you do, change the visualization
- Same goes for tables, by the way

TABLES

- Don’t report numbers to 8 decimals in tables
  - 2-3 is plenty
- Really work on formatting a table so it is clear

REPRODUCIBLE PRESENTATIONS

- Use literate programming to create presentations
- Automate as much as you can
- Easy to create periodic reports and presentations

CLOSING THOUGHTS

THERE ARE NO “BEST” PRACTICES

- The computer can be your friend, but also your enemy
- Once something works for you, capture it, and modify it as you find other things that might be better
- Adopt a “Mack Truck” or sharing mentality
  - If someone else had to do my analysis, could they do it from what I have?
  - Could they do it if they had the data and understood what I’m doing?
  - Could they actually use what I have developed?

VALIDATE, VALIDATE, VALIDATE

- Data analysis is a chain
  - Every link has to ring true
- Recognize potential for errors
- Recognize potential for starting over
- Ensure provenance, veracity of all data, original and derived
- Use code instead of menus to help validation of the analytic process

CLOSING THOUGHTS

THIS TOPIC IS OPEN, FLUID AND EVER-CHANGING

- Do the best you can today
- See if new technology can help
- Contribute to this discussion online, off line, and in your environment
- All of us together will develop the next practices