Wharton PhD Tech Camp

Session 2

Alex Miller

Ph.D. Student, Information Systems

Wharton, OID

Goal for today

- Finish R (quickly!)
  - Cover Notebooks, briefly
- Understand basic Python usage
- Quick primer on parallel computing
- Exercises:
  - Use either Python or R
  - If you want to learn both, you will need to spend some time outside of class
  - Goal is to be somewhat comfortable with Python by Monday, so we can get to the interesting stuff

Clarification about Git

- Mac users have built-in a Unix-shell (accessed via “Terminal”)
  - “git” is a thrid-party program is usually pre-installed on most Macs

- Windows users only have Command Prompt (not Unix-like)
  - “GitBash” is a program that emulates a Unix-like environment for Windows users, and comes pre-installed with git (and a few other Unix-like programs)

R, Continued

Important things you won’t learn elsewhere

- R Markdown and R Notebooks
    - Use them!
    - (Please!) Never copy your code into a word document again!
    - Even better: Use the `stargazer` or `texreg` packages for reproducible tables!

Python

Why use Python?

- You can do almost everything in R that you can do in Python
  - But they have different strengths
- R
  - More statistical
  - Typically (but not always) more state-of-the-art statistics packages
- Python
  - More for general purpose scripting
  - More widely-used in industry; lots of general purpose packages exist

Choosing between Python and R

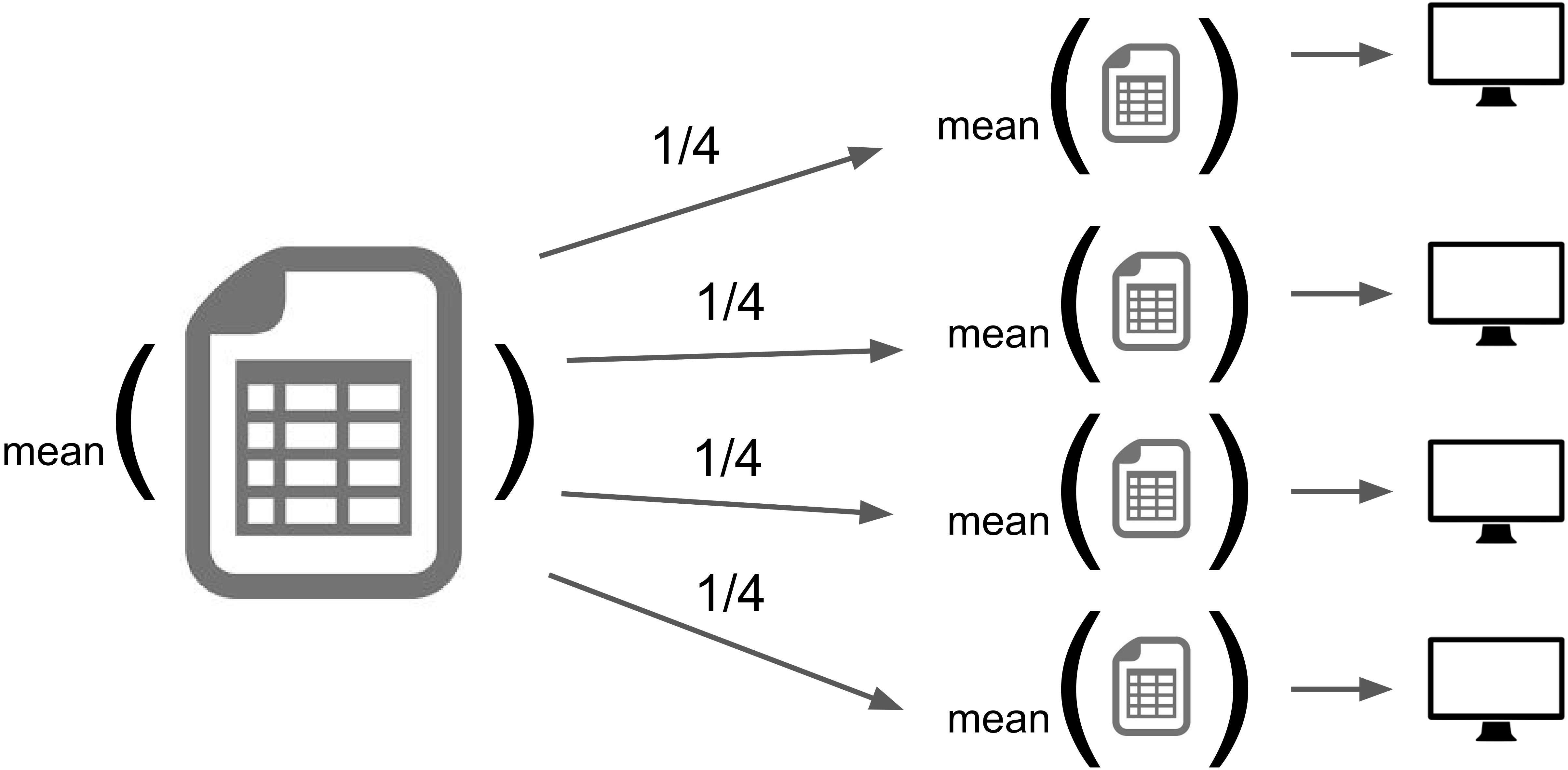
- R for:
  - munging/aggregating/slicing/dicing data
  - regression models and basic statistical analysis
  - ggplot2
- Python for:
  - general scripting, data cleaning
  - state-of-the-art machine learning
- But!
  - rpy2 and Jupyter means you don’t really have to choose
  - I can show you how to setup if you are really interested

Parallelization

What is parallelization?

- When you want to do multiple computations at the same time.
- Suppose you had a dataset with 1,000,000 rows, and you wanted to calculate some function on all the rows (say, average)
  - If each operation takes .01 seconds and you have to serially compute all of them, it will take 2.78 hours
  - What if we could do many calculations at the same time?

Example



Parallelization

- Can be done on different cores of a single machine
- Can also be distributed to many computers

- The Grid!
  - provides up to 64 cores (most modern computers have 4-8)
  - can also just free up your local machine if you have long calculation that cannot be parallelized