Lecture 10: Supercomputing at CRC

LING 1340/2340: Data Science for Linguists

Na-Rae Han

Objectives

- Supercomputing at CRC
  - Server access through SSH
  - Running jobs

- HW3/ML wrap-up

- Let's share your HW3
  - Copy your HW3 jupyter notebook into https://github.com/Data-Science-for-Linguists-2021/Class-Exercise-Repo/tree/main/hw3_shared
  - Rename your JNB file to have your name in it "_narae"
  - Push to your repo, create a pull request

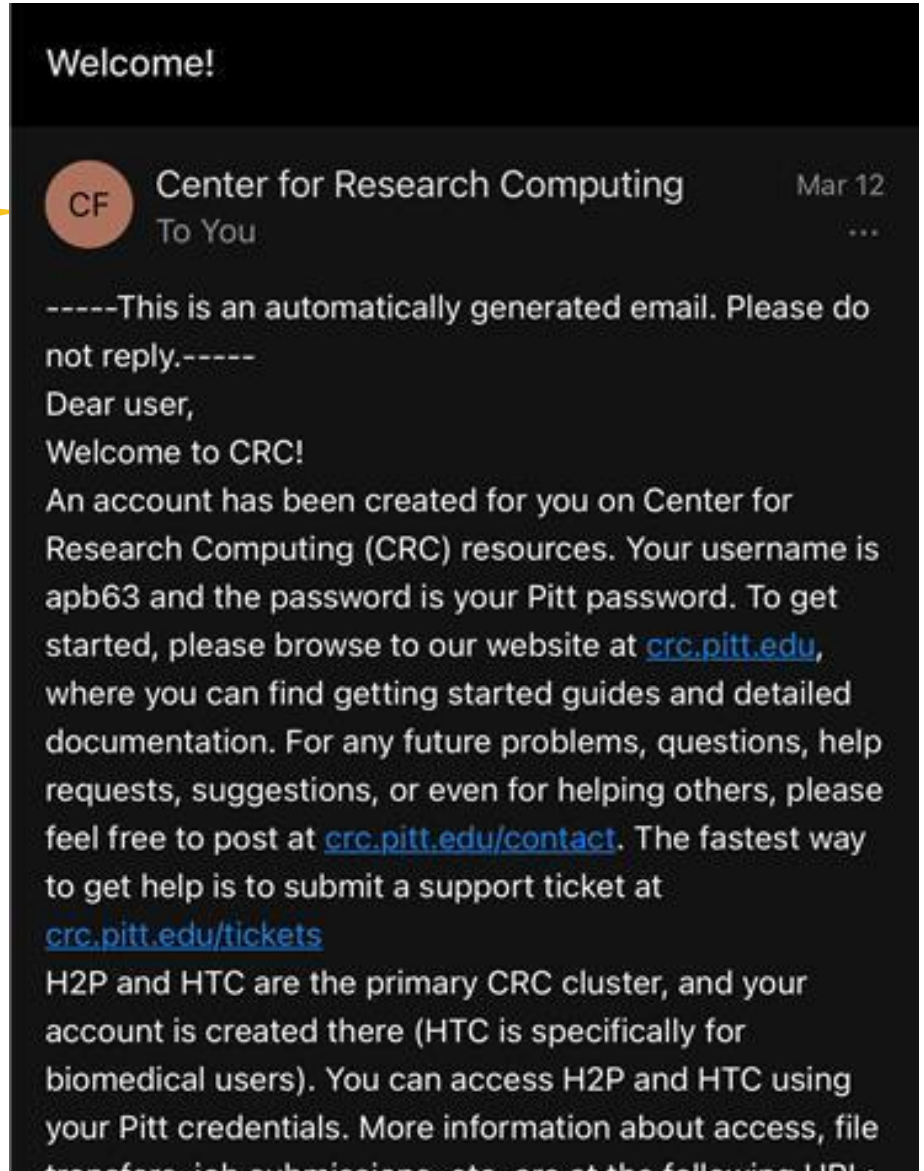
Let us now supercompute.

By Argonne National Laboratory's Flickr page - originally posted to Flickr as Blue Gene / PFrom Argonne National Laboratory Uploaded using F2ComButton, CC BY-SA 2.0, https://commons.wikimedia.org/w/index.php?curid=6412306

You got a supercomputing account.

- You received this mysterious email:

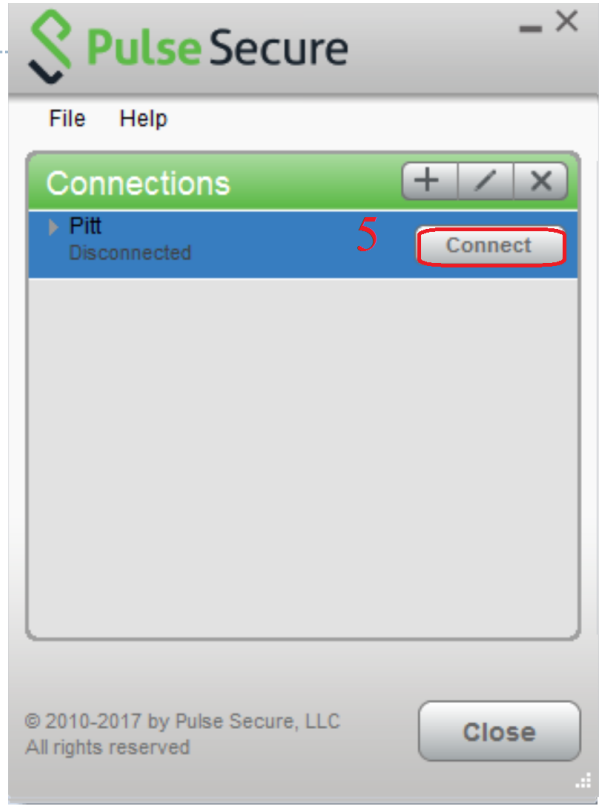
I got you all an account at Pitt's **Center for Research Computing** (CRC)



- CRC: Center for Research Computing
  - https://crc.pitt.edu
  - Handy links in "Resource" page!

Accessing CRC's cluster

- Your laptop should be running a **Secure Remote Access client**.
  - Install and run PulseSecure →
  - Details in the h2p cluster user guide: https://crc.pitt.edu/resources/h2p-user-guide
- Remote-access your account via SSH:
  - `ssh yourpittid@h2p.crc.pitt.edu`
- Getting your bearings:
  - Where are you? `pwd`
  - What is your user 'group'? `groups`
  - Is python installed on this machine? `which python`
  - What are your configuration files? `ls -a`
    - `.bash_profile`
      ← Customize with your own aliases, etc.
    - `.bash_history`
      ← Bash commands you typed in are logged here.



Shared data space for the group

- You are all assigned to our class **group**: `ling2340_2021s`

```
naraehan@login0:~
[naraehan@login0 ~]$  groups
nhan ling1340-2017f pyling ling1340-2019s ling2340_2021s
[naraehan@login0 ~]$
```

- Our shared datasets are stored in: `/bgfs/ling2340_2021s/shared_data/`

```
naraehan@login0:/bgfs/ling2340_2021s/shared_data/yelp_dataset_2021
[naraehan@login0 ~]$  cd /bgfs/ling2340_2021s/
[naraehan@login0 ling2340_2021s]$  ls
shared_data
[naraehan@login0 ling2340_2021s]$  cd shared_data/
[naraehan@login0 shared_data]$  ls
enable1.txt  ETS_Corpus_of_Non-Native_Written_English  nltk_data  word_vectors  yelp_dataset_2021
[naraehan@login0 shared_data]$  cd yelp_dataset_2021/
[naraehan@login0 yelp_dataset_2021]$  ls
Dataset_User_Agreement.pdf          yelp_academic_dataset_checkin.json  yelp_academic_dataset_tip.json
yelp_academic_dataset_business.json  yelp_academic_dataset_review.json   yelp_academic_dataset_user.json
[naraehan@login0 yelp_dataset_2021]$
```

ETS corpus, Yelp dataset, and nltk data!

Na-Rae's .bash_profile

- PATH configuration
- Prompt in pink!! Add this line:

```
export PS1="\[\e[0;35m\][\u@\h \W]\$ \[\e[m\]"
```

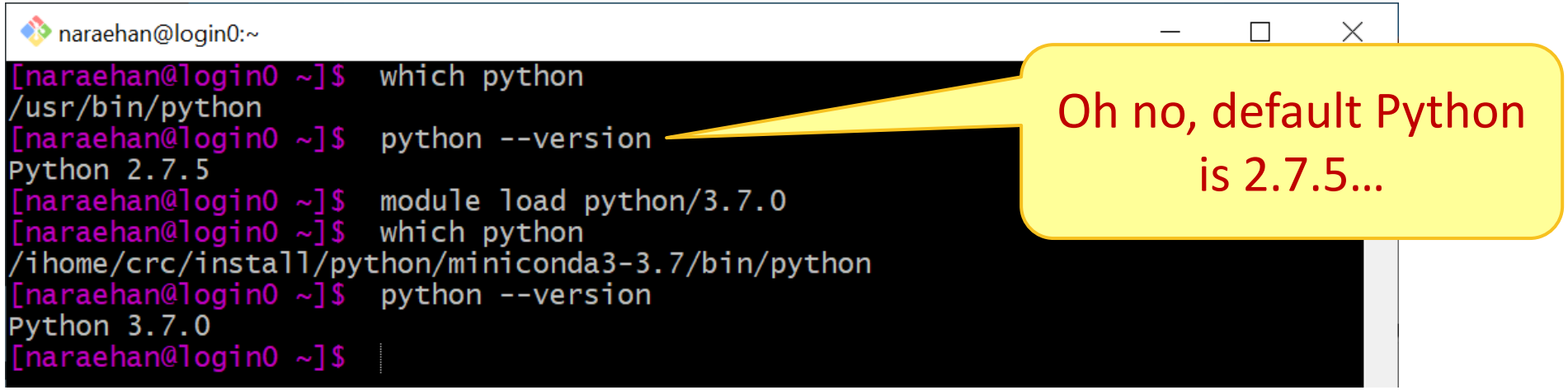
- Some aliases

```
[naraehan@login1 ~]$ cat .bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/.local/bin:$HOME/bin
export PATH

# Prompt in pink color!
export PS1="\[\e[0;35m\][\u@\h \W]\$ \[\e[m\]"

# perl-style regex, color
alias grep='grep -P --color'
```

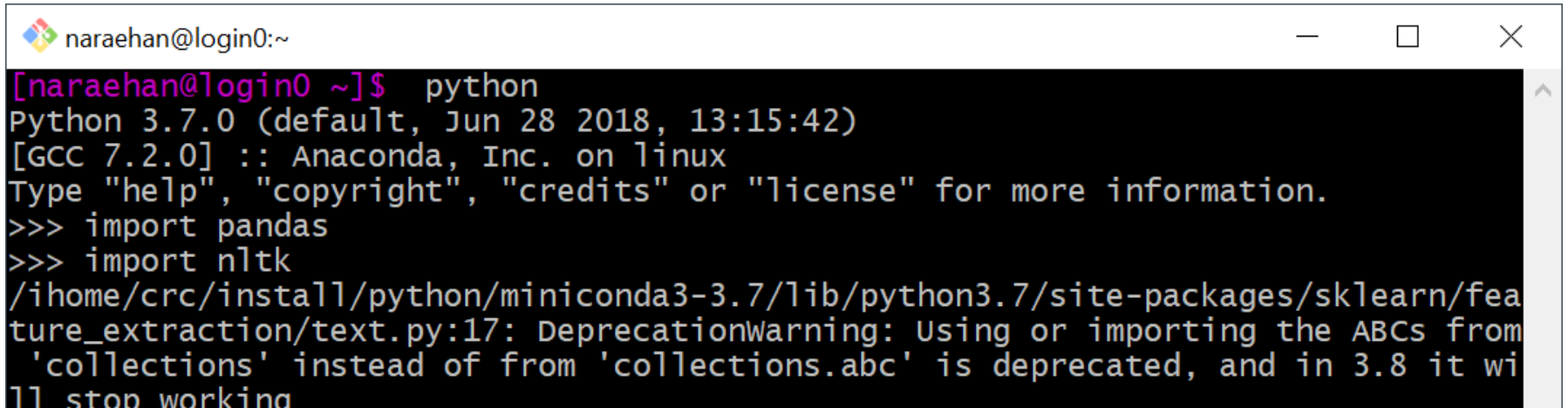
If you edit this file, changes take effect after logging back in.

For immediate effect, run:
`source .bash_profile`

Using the right python module



- We have to "load" the correct python module via `module load python/3.7.0`
- Popular data science libraries are already installed (pandas, sklearn, nltk...):



Using CRC clusters

- Job submissions
  - On a computing cluster, many people are using the same resources so we have a "job queue" that accepts job submissions
  - CRC and many other clusters use Slurm for managing and scheduling these jobs.

- What this means:
  - You don't directly execute your Python script. (A big NO-NO)
  - You create a BASH SCRIPT to run a PYTHON SCRIPT (job).

Before you get carried away

- Do NOT yet run any commands/jobs that may be resource-intensive.
- This is a powerful super-computer, shared by many research groups at Pitt.
  - Our class as a group has a limited, shared allocation. We have a reserve of **10000 Service Units** (SUs), which is 10k hours of computing time.
  - You do not want to accidentally initiate a run-away process and hog resources.
- There are PROPER ways to run jobs.
  - We will show you now.

Slurm Jobs

- To make a slurm job script, you basically need to write a bash script of what you would do to run your program on the command line. This is just a text file, usually with a **.sh** ending.
- Also need some slurm configs
- Example (let's call this `hello.sh`)

```
#!/usr/bin/env bash

#SBATCH --job-name=hello
#SBATCH --output=hello.out
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --partition=smp
#SBATCH --cluster=smp

echo "hello world"
```

<-- Copy this into a file and name it something like **hello.sh**

Below are some other Slurm config options (prefix with #SBATCH) as in hello.sh. EVEN MORE at https://slurm.schedmd.com/sbatch.html

Option	Environment Variables
--output	–
--time	(Format: DAYS-HOURS:MINUTES:SECONDS)
--job-name	SLURM_JOB_NAME
--nodes	SLURM_NNODES
--ntasks	SLURM_NTASKS
--cpus-per-task	SLURM_CPUS_PER_TASK
--ntasks-per-node	SLURM_NTASKS_PER_NODE
--partition	SLURM_JOB_PARTITION
--mem	SLURM_MEM_PER_NODE
--account	SLURM_JOB_ACCOUNT

Job management commands

So from the directory with our `hello.sh` script, we can submit it with `sbatch hello.sh`

This should run pretty much instantly and we can check our `hello.out` output file.

Command	Description
`sinfo`	Quick view of partitions
`sbatch <job>`	Submit your job[a]
`squeue`	View all jobs
`squeue -u <user>`	Look at **your** jobs
`scancel <jobid>`	Cancel your job
`crc-sinfo.py`	`sinfo` wrapper
`crc-squeue.py`	`squeue` wrapper
`crc-scancel.py <jobid>`	`scancel` wrapper
`crc-usage.pl`	View your group's usage

To-do #11 redux on CRC: setting up

(1) New location of yelp review data file:

```
/bgfs/ling2340_2021s/shared_data/yelp_dataset_2021/yelp_academic_dataset_review.json
```

(2) We'll sample 1 million lines (shuffled):

```
shuf /bgfs/ling…/…/yelp_academic_dataset_review.json -n 1000000 > ~/review_1mil.json
```

(3) Copy our python script. Running it on this data will look like: (but don't run this!!)

```
python process_reviews.py review_1mil.json
```

(4) But before that, we should load the appropriate python environment:

```
module load python/3.7.0
```

(5) Now we can toss all this into a bash script. Let's call it `todo11.sh`

(6) Start with `hello.sh` (make a copy, then edit)

(7) Change the bash commands at the bottom to run our script for To-do 11, and change the job name and output file to something like `todo11` and `todo11.out`

todo11.sh

```
#!/usr/bin/env bash

#SBATCH --job-name=todo11
#SBATCH --output=todo11.out
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --partition=smp
#SBATCH --cluster=smp

module load python/3.7.0
python process_reviews.py review_1mil.json
```

SLURM Job script

Python script

process_reviews.py
(from To-do #11)

```
import pandas as pd
import sys
from collections import Counter

filename = sys.argv[1]

df = pd.read_json(filename, lines=True, encoding='utf-8')
print(df.head(5))

wtoks = ' '.join(df['text']).split()
wfreq = Counter(wtoks)
print(wfreq.most_common(20))
```

To-do #11 redux on CRC

- Submit your job:
  - `sbatch todo11.sh`
- and check status with:
  - `squeue -u <user-id>`
  - Done when squeue no longer shows job (keep re-running with up arrow)
  - Or: add `-i 10` to auto-run every 10 seconds (Ctrl+c to get out)
- Check the output with:
  - `cat todo11.out`
- Success! 1 million reviews weren't much of a challenge

```
naraehan@login0:~
[naraehan@login0 ~]$  ls
hello.sh  old2              pyling           shared_data  try    w2v
old       process_reviews.py  review_1mil.json  todo11.sh    vault
[naraehan@login0 ~]$  sbatch todo11.sh
Submitted batch job 2824655 on cluster smp
[naraehan@login0 ~]$  squeue -u naraehan
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           2824655       smp   todo11 naraehan  R       0:05      1 smp-n111
[naraehan@login0 ~]$  squeue -u naraehan
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           2824655       smp   todo11 naraehan  R       0:15      1 smp-n111
[naraehan@login0 ~]$  squeue -u naraehan -i 10
Tue Mar 23 23:05:16 2021
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           2824655       smp   todo11 naraehan  R       0:23      1 smp-n111

Tue Mar 23 23:05:26 2021
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           2824655       smp   todo11 naraehan  R       0:33      1 smp-n111

Tue Mar 23 23:05:36 2021
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           2824655       smp   todo11 naraehan  R       0:43      1 smp-n111

Tue Mar 23 23:05:46 2021
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           2824655       smp   todo11 naraehan  R       0:53      1 smp-n111

Tue Mar 23 23:05:56 2021
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)

^C
[naraehan@login0 ~]$  ls
hello.sh  old2              pyling           shared_data  todo11.sh  vault
old       process_reviews.py  review_1mil.json  todo11.out   try        w2v
[naraehan@login0 ~]$  cat todo11.out
                review_id  ...                 date
0  BiRkTZrn8x_zspT8sRt-Qw  ... 2018-10-10 23:43:25
1  24pRZcBLDM_kbTvF1bCl-w  ... 2014-04-12 16:42:59
2  xOxa2icr6U4E7G93trPwfw  ... 2017-12-18 22:13:04
3  JoFOhMZF9DWiB9_SRWgKCA  ... 2018-12-31 23:45:56
4  Fk2LP1ft8TrQcrUS19w8OA  ... 2019-06-20 21:04:28

[5 rows x 9 columns]
[('the', 4516590), ('and', 3799781), ('I', 2887729), ('a', 2788950), ('to', 2697926),
 1878049), ('of', 1603693), ('is', 1313227), ('for', 1258119), ('in', 1170009), ('The
8), ('it', 940556), ('with', 896436), ('my', 896209), ('that', 883825), ('but', 76059
', 725910), ('have', 677351), ('you', 666733), ('this', 635577)]
```

How did the job go?

- Job ID was shown earlier →
- Check finished job's stats by:
  - `seff <job-id>`
- Our Python script on 1million reviews used:
  - 9.4GB of memory (RAM)
  - 55 seconds of CPU time

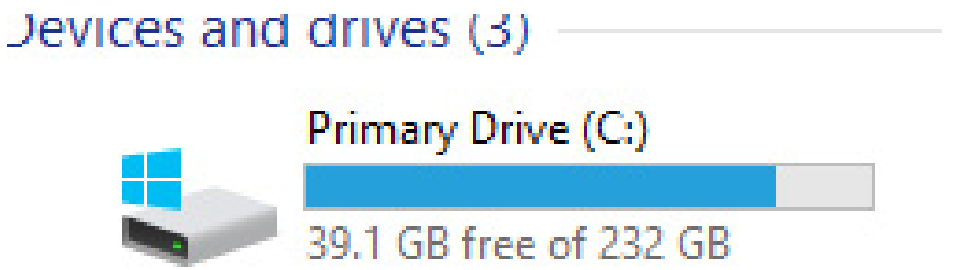
```
[naraehan@login0 ~]$  sbatch todo11.sh
Submitted batch job 2824655 on cluster smp
[naraehan@login0 ~]$  squeue -u naraehan
             JOBID PARTITION     NAME     USER ST
           2824655       smp   todo11 naraehan  R
```

```
[naraehan@login0 ~]$  seff 2824655
Job ID: 2824655
Cluster: smp
User/Group: naraehan/nhan
State: COMPLETED (exit code 0)
Cores: 1
CPU Utilized: 00:00:55
CPU Efficiency: 90.16% of 00:01:01 core-walltime
Job Wall-clock time: 00:01:01
Memory Utilized: 9.43 GB
Memory Efficiency: 240.38% of 3.92 GB
[naraehan@login0 ~]$
```

Quick aside

- Memory here refers to Random Access Memory (RAM)
  - You probably have 4, 8 or 16 GB on your laptop
  - Running programs uses RAM to store temporary data (in our case opened file content, variables, lists, DataFrame, etc) that they use or produce
  - Stuff stored in RAM is removed when a program terminates, or if your computer shuts off.
  - Running out of RAM on your laptop will probably cause your computer to freeze/crash
  - Expensive per GB

- NOT disk drive -->
  - Disk space stores files long-term
  - Cheap per GB, 256+GB is pretty standard.

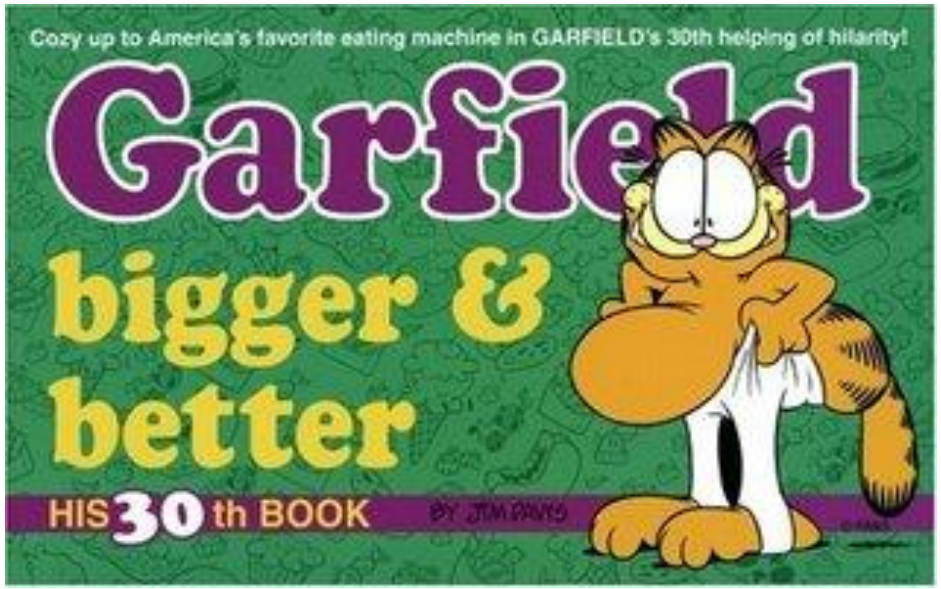


To-do #12: bigger data + better code

- Take 1: use 4 million reviews
- Take 2: use 4 million reviews, with a new (better!) python script

  ← Compare Take 1 vs. Take 2

- Take 3 (optional): all 8.6 million reviews, with the new (better!) python script



Wrapping up

- To-do #12
- PyLing! Next Wed 6pm.

- Next class
  - Joey presents: computational efficiency
  - More CRC exploration: Jupyter Hub, clustering