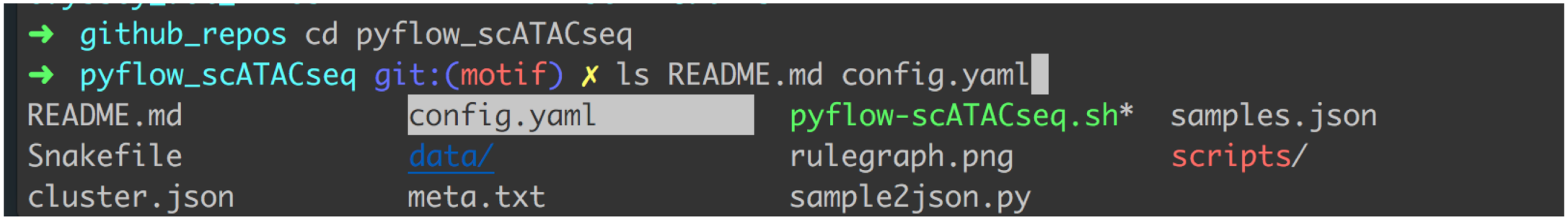
Tools and tricks for a data scientist

03/09/2020 Ming (Tommy) Tang

Oh-my-zsh!

- https://ohmyz.sh/



https://divingintogeneticsandgenomics.rbind.io/post/set-up-my-new-mac-laptop/

Mosh: mobile shell

- https://mosh.org/

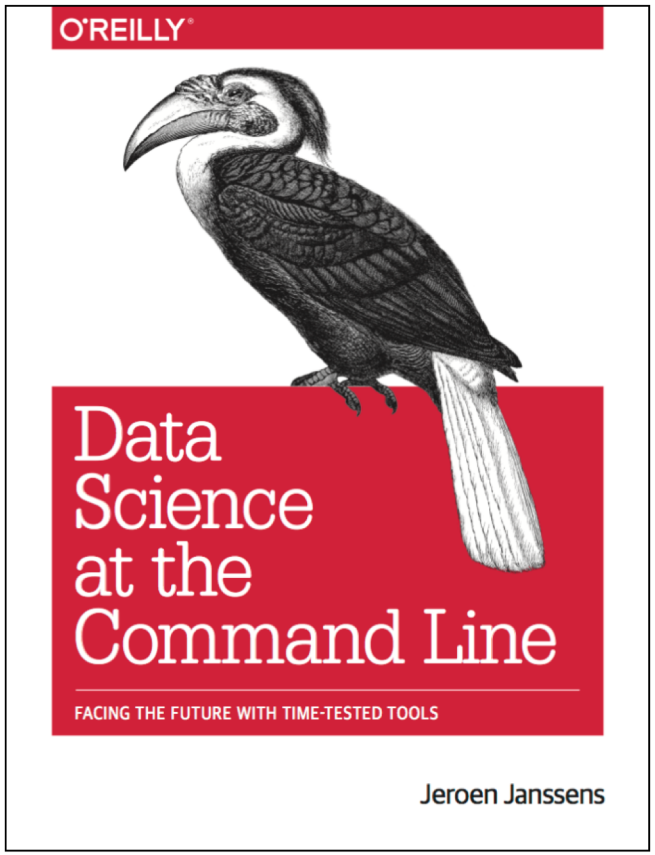
- Mosh + screen/tmux to keep your session persistent.

csvkit

- https://www.datascienceatthecommandline.com/

cd /n/holyscratch01/informatics/mtang

cat mtcars.csv | csvless –S
cat mtcars.csv | head | csvless –S
csvcut –n mtcars.csv



body

- https://github.com/jeroenjanssens/data-science-at-the-command-line/blob/master/tools/body

Executable File | 15 lines (15 sloc) | 472 Bytes | Raw

```
#!/usr/bin/env bash
#
# body: apply expression to all but the first line.
# Use multiple times in case the header spans more than one line.
#
# Example usage:
# $ seq 10 | header -a 'values' | body sort -nr
# $ seq 10 | header -a 'multi\nline\nheader' | body body body sort -nr
#
# From: http://unix.stackexchange.com/a/11859
#
# See also: header (https://github.com/jeroenjanssens/command-line-tools-for-data-science)
IFS= read -r header
printf '%s\n' "$header"
eval $@
```

Cat myfile.txt | body grep “pattern”

Will retain the header

cat mtcars.csv | body grep Ford | csvless -S

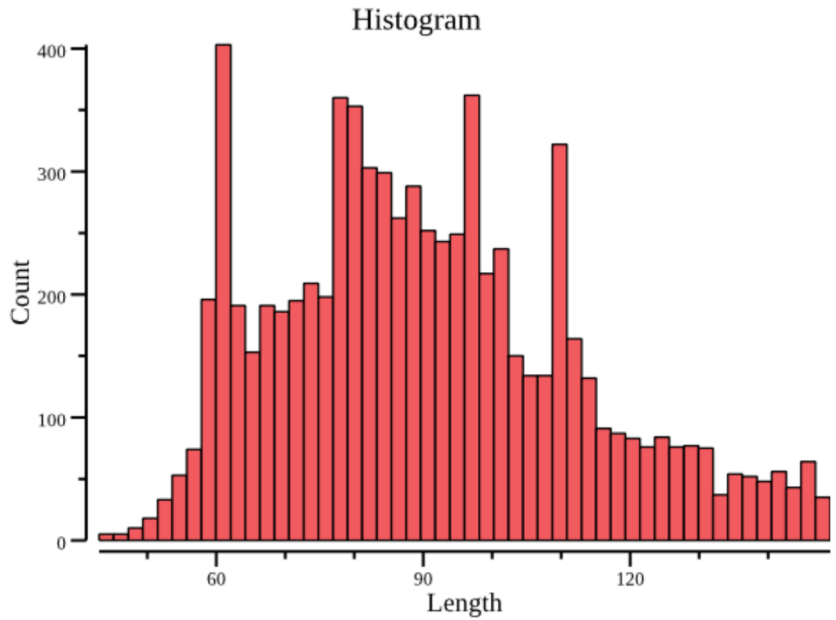
csvtk

- https://github.com/shenwei356/csvtk
- **E.g. cut out columns based on column names in another file.**

- csvtk cut -f $(paste -s -d, columns.txt) mtcars.csv
- **Unix cut can not arrange column orders,**
- **I usually use awk. Csvtk can**
- **Other tools:**
- https://github.com/crazyhottommy/getting-start tools-and-resources#do-not-give-me-excel-files

◦ plot histogram with data of the second column:

```
csvtk -t plot hist testdata/grouped_data.tsv.gz -f 2 | display
```



GNU parallel

Download for all

```bash
cat  2020-02-24_sophia_basespace_run_info.txt | awk '{print$2}' | sed '1d' | grep -v 117072956 | grep -v 11601489 > bs_cli/run_ids.txt
```

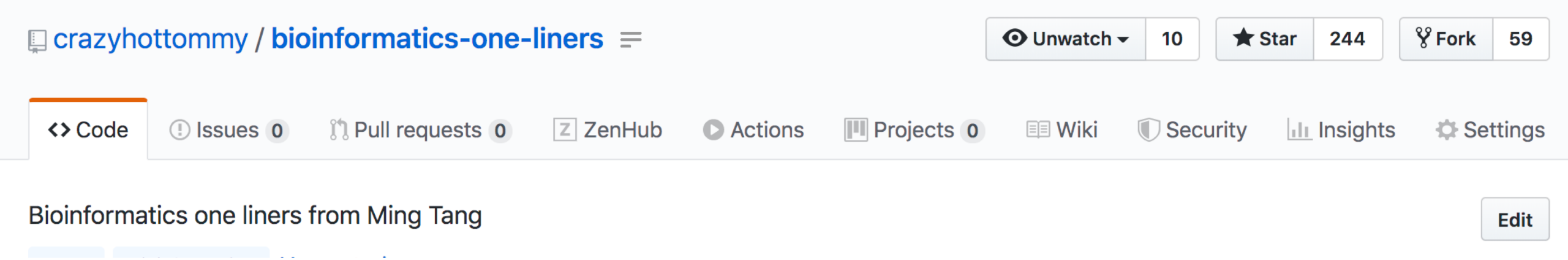
```
cd bs_cli
cat run_ids.txt | parallel -j 24 'echo bs download run -c default -i {} -o {} --log={}.log'

cat run_ids.txt | parallel -j 24 'echo bs download run -c default -i {} -o {} --log={}.log'
```

Most frequently used…

- 1. readlink –e
- 2. realpath
- 3. less –S
- 4. cat –A show hidden characters e.g. ^M, ^I,
- 5. dos2unix

One-liners

- https://github.com/crazyhottommy/bioinformatics-one-liners



Brename: rename your files without a mess

- https://github.com/shenwei356/brename
- Written in go, download the binary, ready to use.
- Regular expression
- undo last -u
- Dry run -d
- only renaming specific paths via include filters :
- brename -p ":" -r "-" -f ".htm$" -f ".html$”
- using capture variables, e.g., $1, $2…
- brename -p "(m)" -r "\$1\$1"

rmate editing remote files (I only know how to quit vim)

- https://divingintogeneticsandgenomics.rbind.io/post/open-files-on-remote-with-sublime-by-ssh/

on remote machine, install `rmate`

```
ssh bio1
curl -o ~/bin/rmate https://raw.githubusercontent.com/aurora/rmate/master/rmate

chmod u+x bin/rmate
```

on your local computer, install `RemoteSubl`

on your **local** computer, open `sublime`, click `tools` –> `Command Palette` –> type Package control:Install Package –> type `RemoteSubl` to install.

change your ssh config file

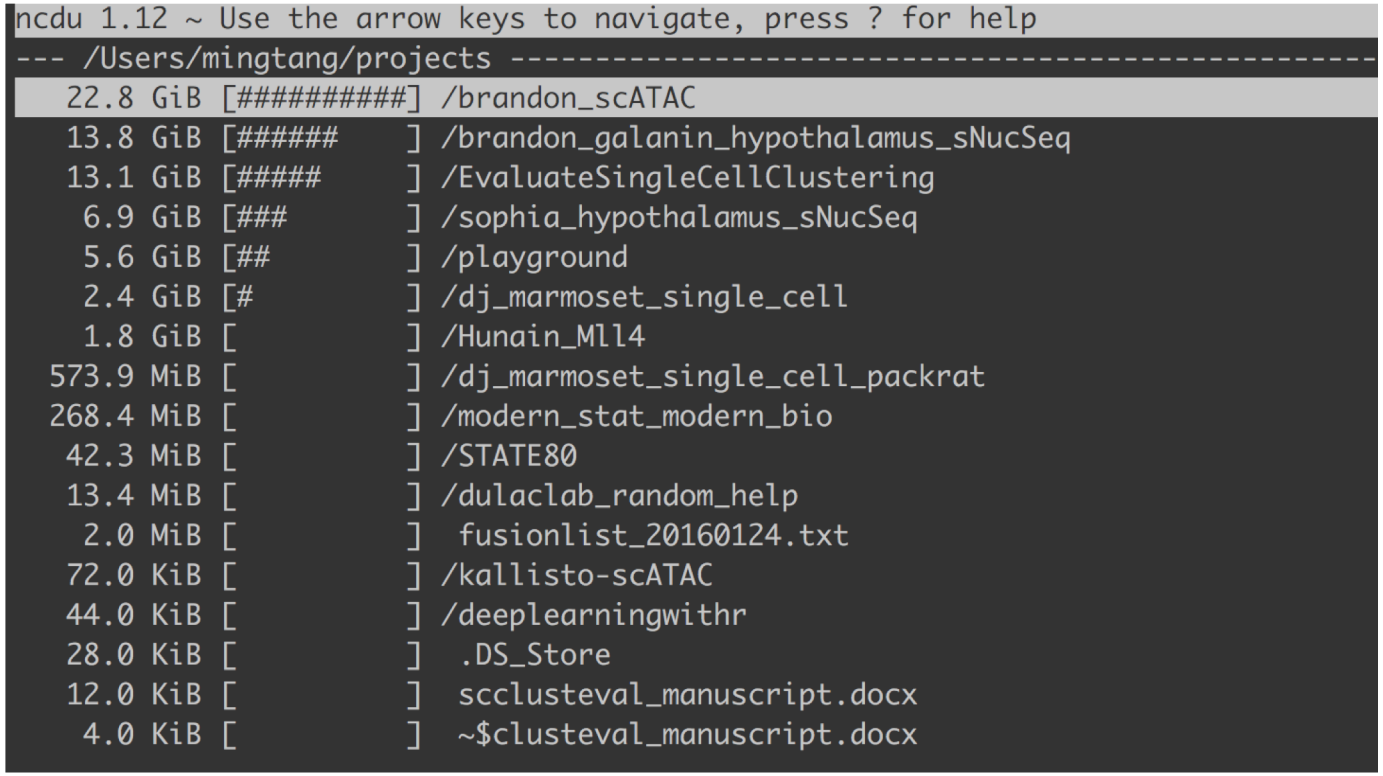
add `RemoteForward 52698 localhost:52698` to your `~/.ssh/config` file.

Now, ssh to remote, and you can do `rmate my.txt` in your remote and open sublime in your local machine.

ncdu

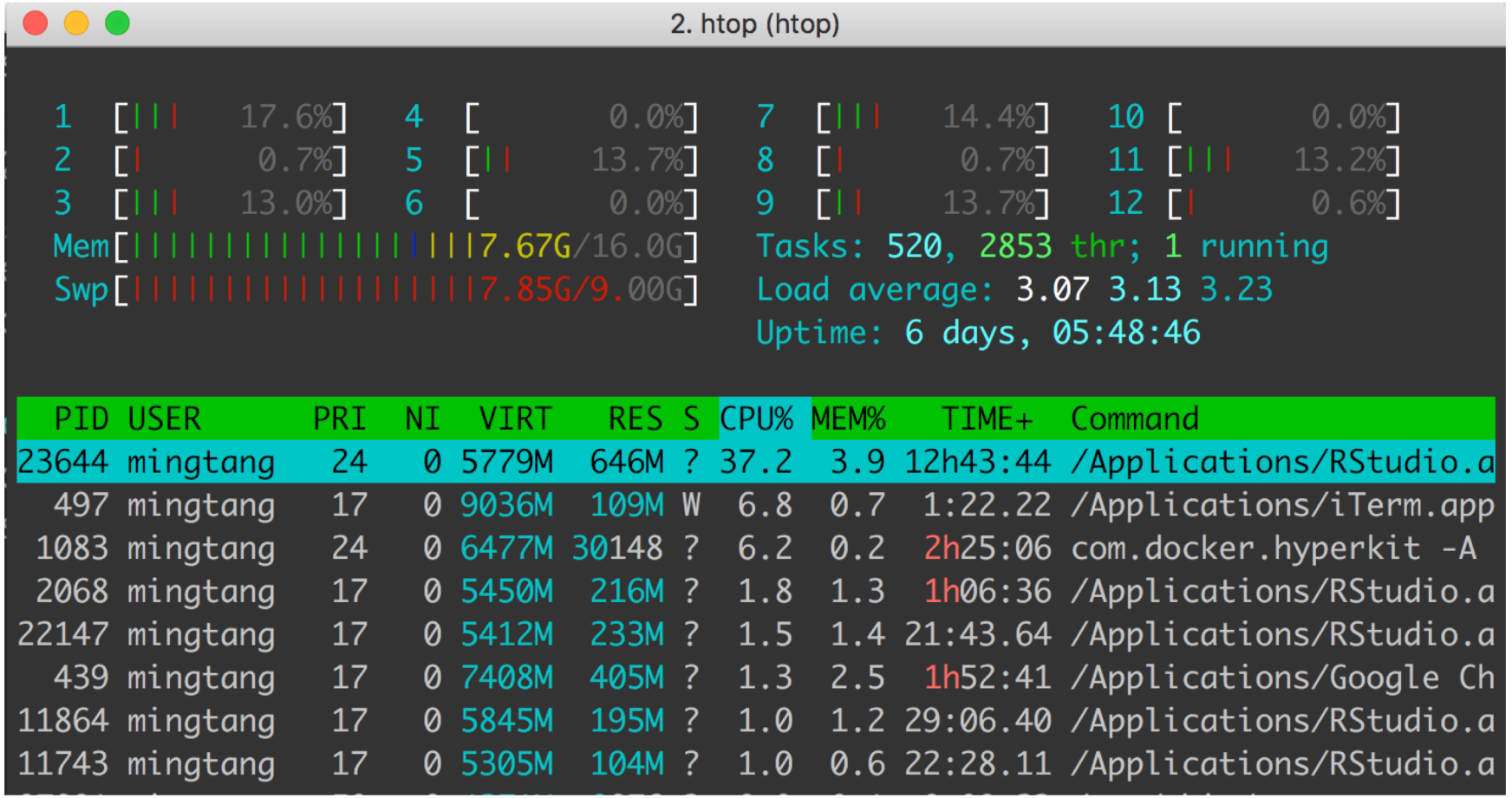
https://anaconda.org/coecms/ncdu

- Ncdu, acronym of **NC**urses **D**isk **U**sage, is a curses-based version of the well-known ‘du’ command. It provides a fast way to see what directories are using the disk space.



higer top: htop

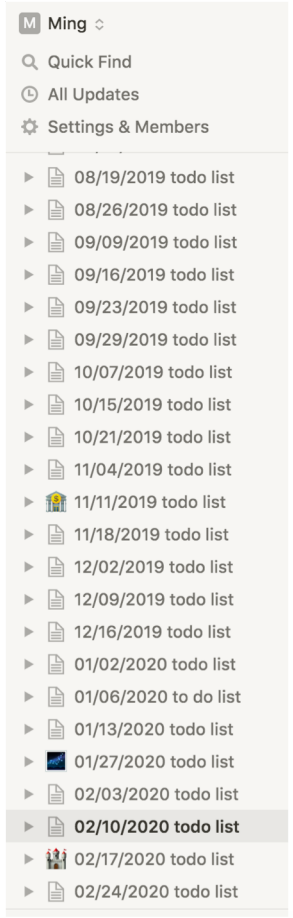
https://hisham.hm/htop/

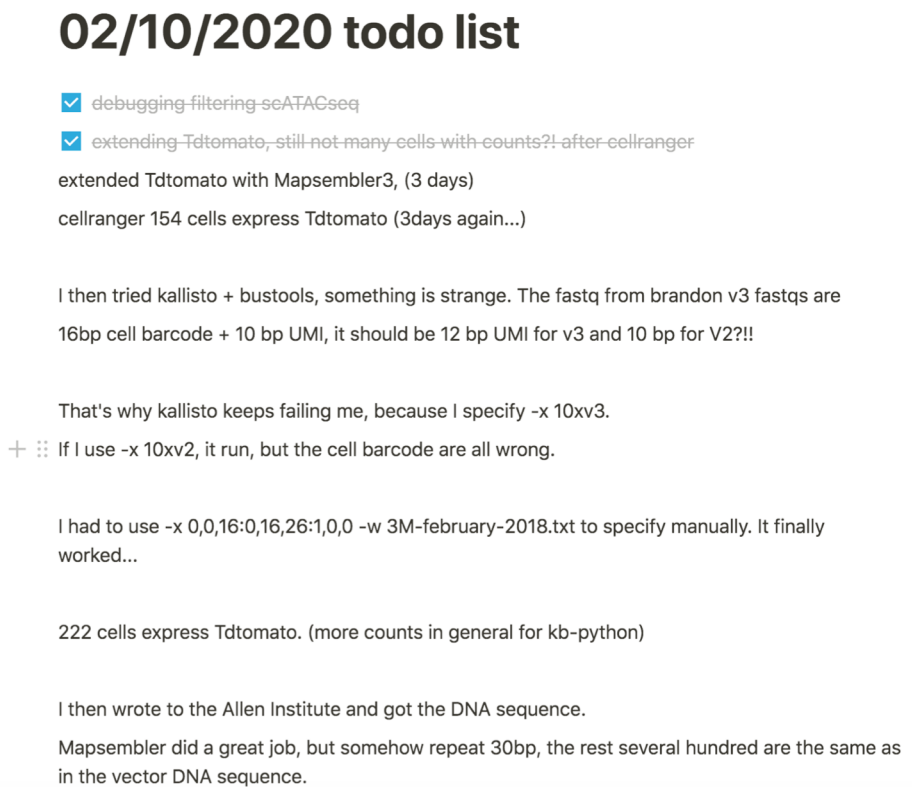


Dat:

- peer-to-peer sharing & live syncronization of files via command line https://dat.foundation.
- npm install -g dat

Notion App for to do list and many more

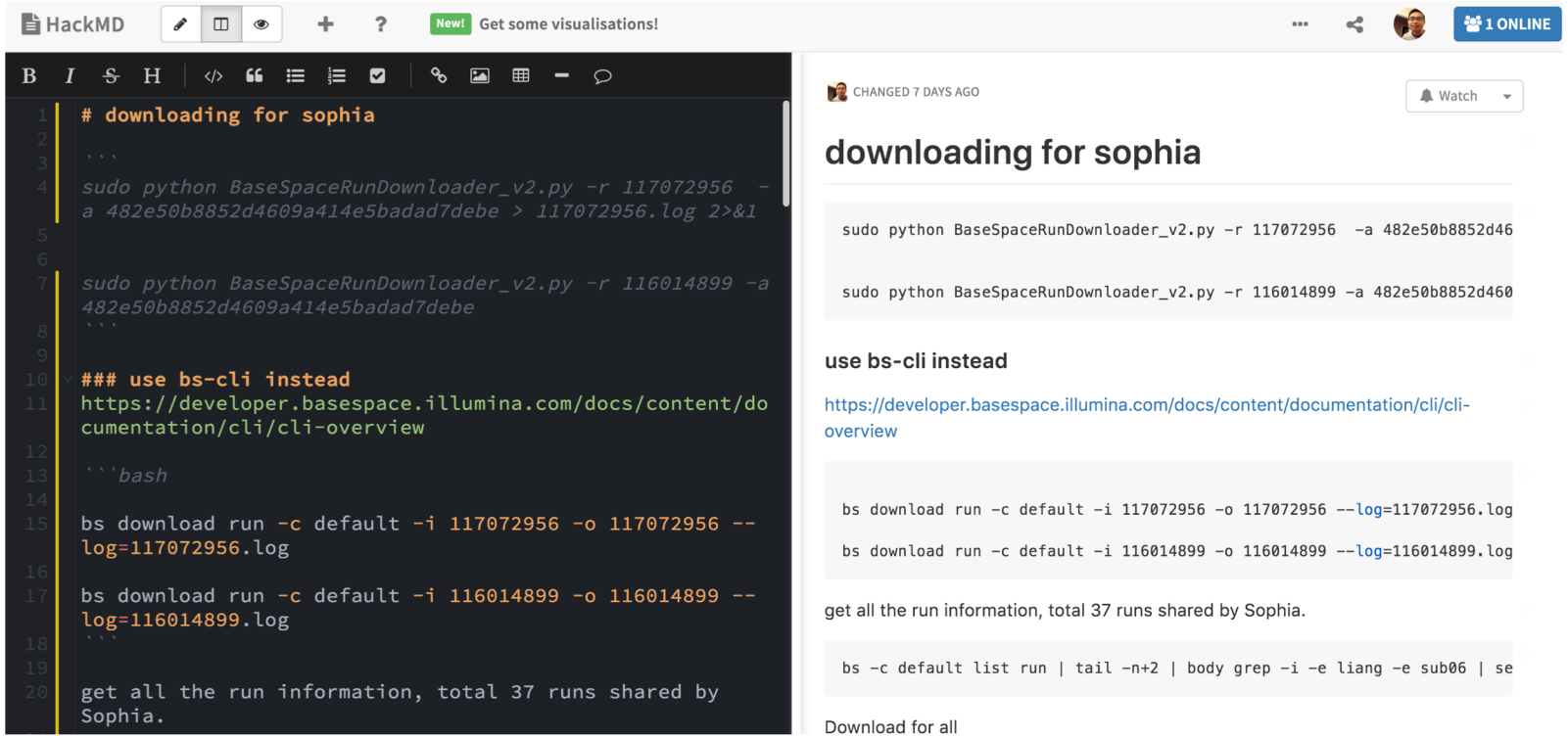




Other tools: https://github.com/crazyhottommy/The-world-of-faculty#digital-tools-for-organizing-a-computational-biology-lab

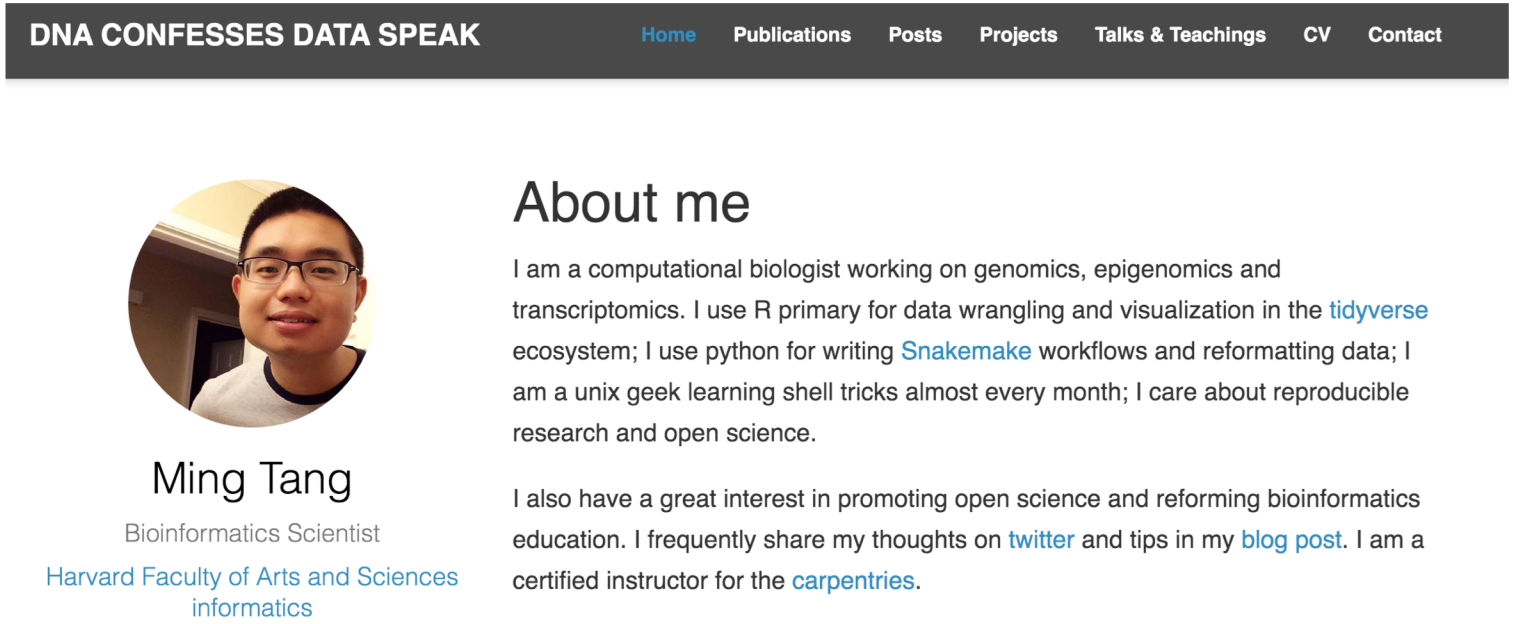
Hackmd for taking notes

- https://hackmd.io/



Take notes and maybe write it to a blog post.

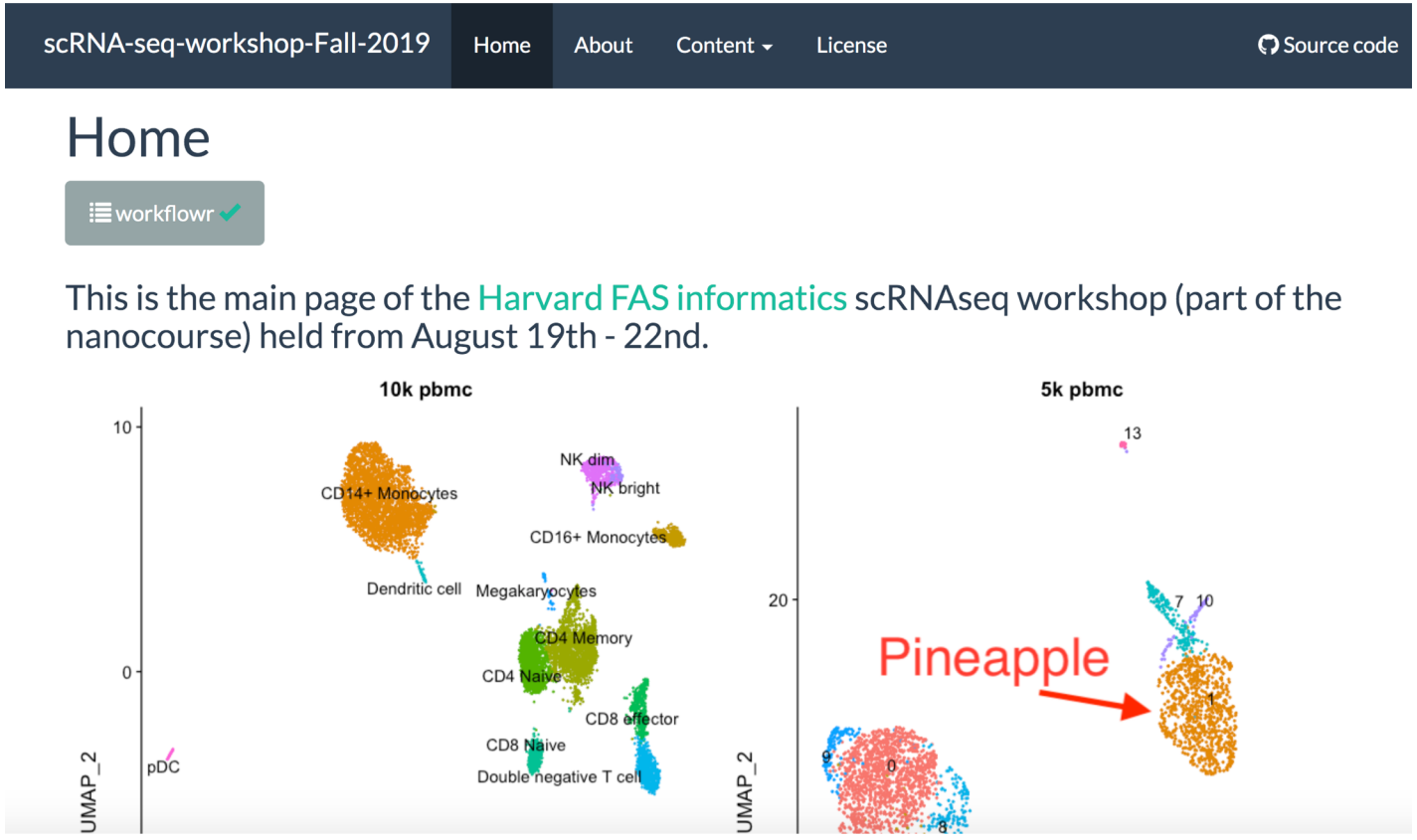
Blogdown for blog posts



https://divingintogeneticsandgenomics.rbind.io/post/hugo-academic-theme-blog-down-deployment-some-details/

Workflowr to make website for teaching, sharing projects

- https://github.com/jdblischak/workflowr

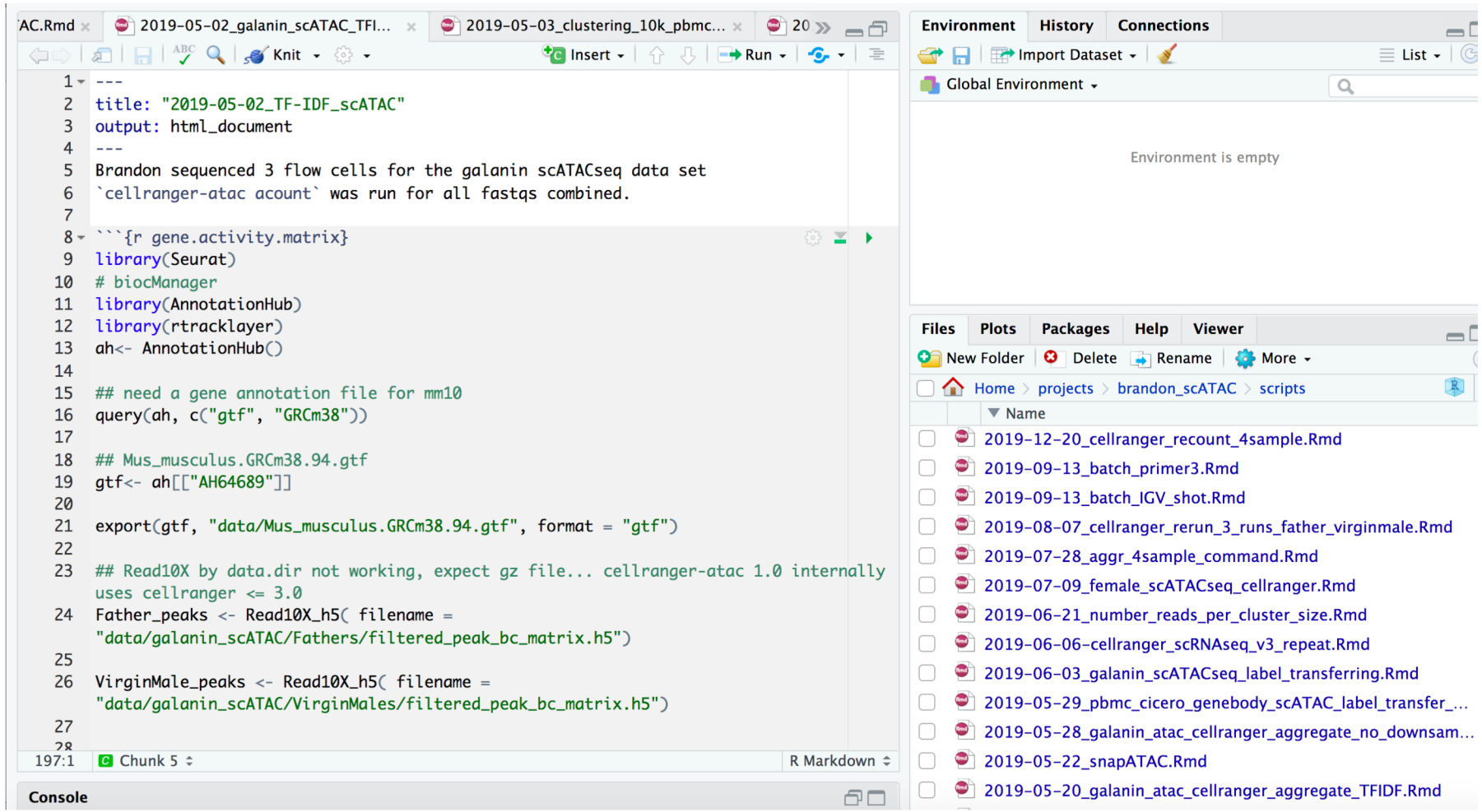


https://crazyhottommy.github.io/scRNA-seq-workshop-Fall-2019/

Command line R utilites

- DocoptR
- https://divingintogeneticsandgenomics.rbind.io/post/use-docopt-to-write-command-line-r-utilities/
- Littler
- http://dirk.eddelbuettel.com/code/littler.html
- Funr
- https://github.com/sahilseth/funr

Rstudio R project



PLOS | COMPUTATIONAL BIOLOGY

Browse | Publish | About

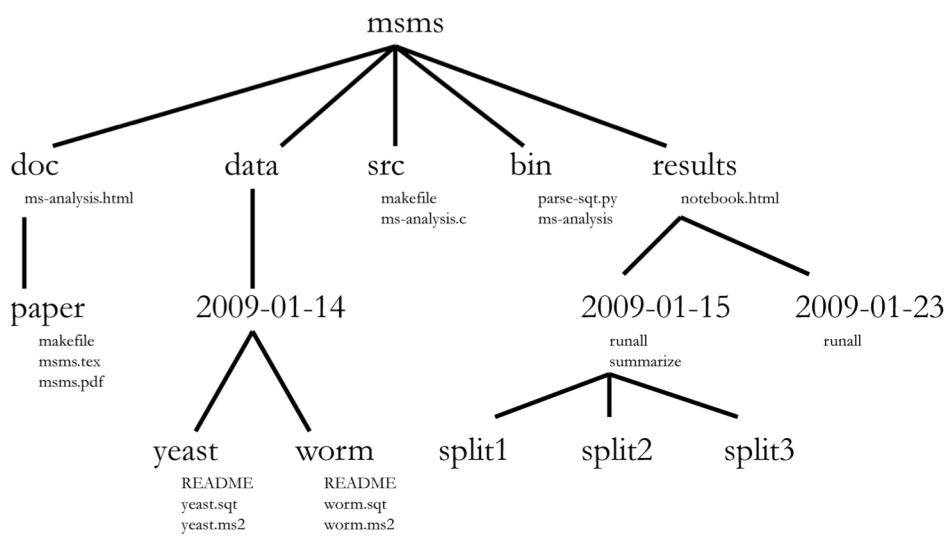
OPEN ACCESS

EDUCATION

A Quick Guide to Organizing Computational Biology Projects

William Stafford Noble

Published: July 31, 2009 • https://doi.org/10.1371/journal.pcbi.1000424



PLOS COMPUTATIONAL BIOLOGY

Browse Publish About

OPEN ACCESS

PERSPECTIVE

Good enough practices in scientific computing

Greg Wilson, Jennifer Bryan, Karen Cranston, Justin Kitzes, Lex Nederbragt, Tracy K. Teal

Published: June 22, 2017 • https://doi.org/10.1371/journal.pcbi.1005510

PLOS BIOLOGY

FIFTEENTH ANNIVERSARY

Browse Publish About

OPEN ACCESS

COMMUNITY PAGE

Best Practices for Scientific Computing

Greg Wilson, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, Kathryn D. Huff, Ian M. Mitchell, Mark D. Plumbley, Ben Waugh, Ethan P. White, Paul Wilson

here::here()

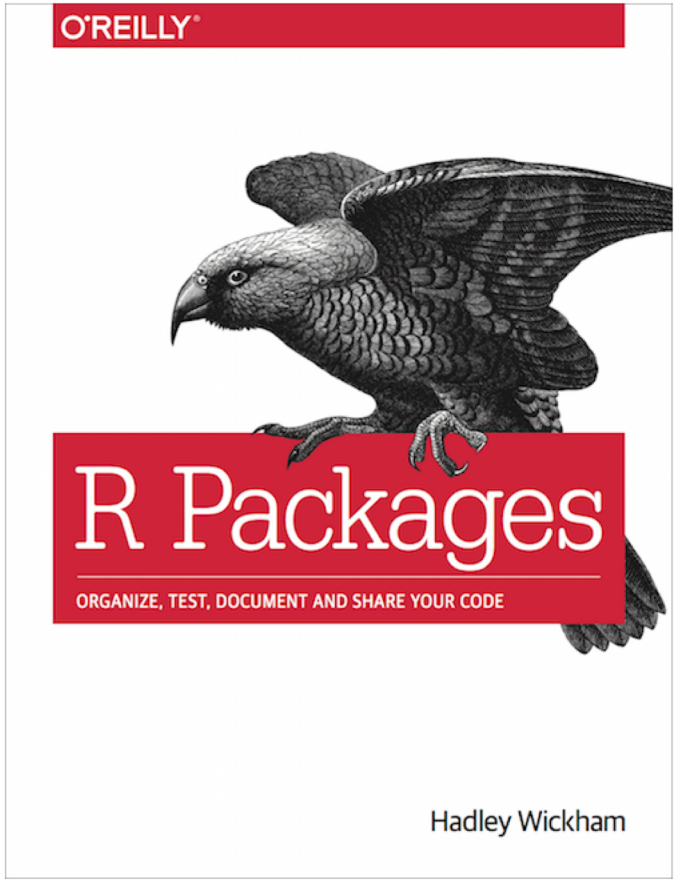
https://www.tidyverse.org/blog/2017/12/workflow-vs-script/
Works with Rproject

> *If the first line of your R script is*
>
> `setwd("C:\Users\jenny\path\that\only\I\have")`
>
> *I will come into your office and SET YOUR COMPUTER ON FIRE 🔥.*

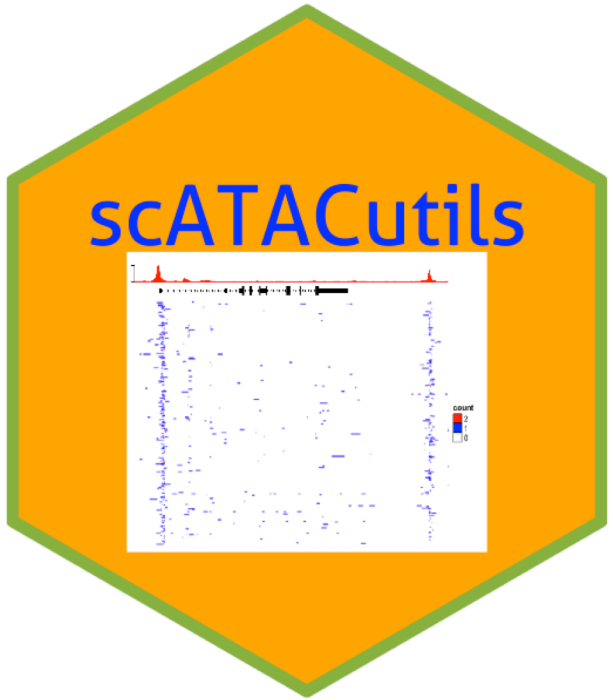
> *If the first line of your R script is*
>
> `rm(list = ls())`
>
> *I will come into your office and SET YOUR COMPUTER ON FIRE 🔥.*

Making R packages

- http://r-pkgs.had.co.nz/



R packages



https://github.com/crazyhottommy/scclusteval
https://github.com/crazyhottommy/scATACutils

Docker + rstudio (Thanks Nathan!)

- Docker/singularity rocker image
- Ssh tunneling to connect to bioinfo1 (enjoy the 1 TB RAM!)

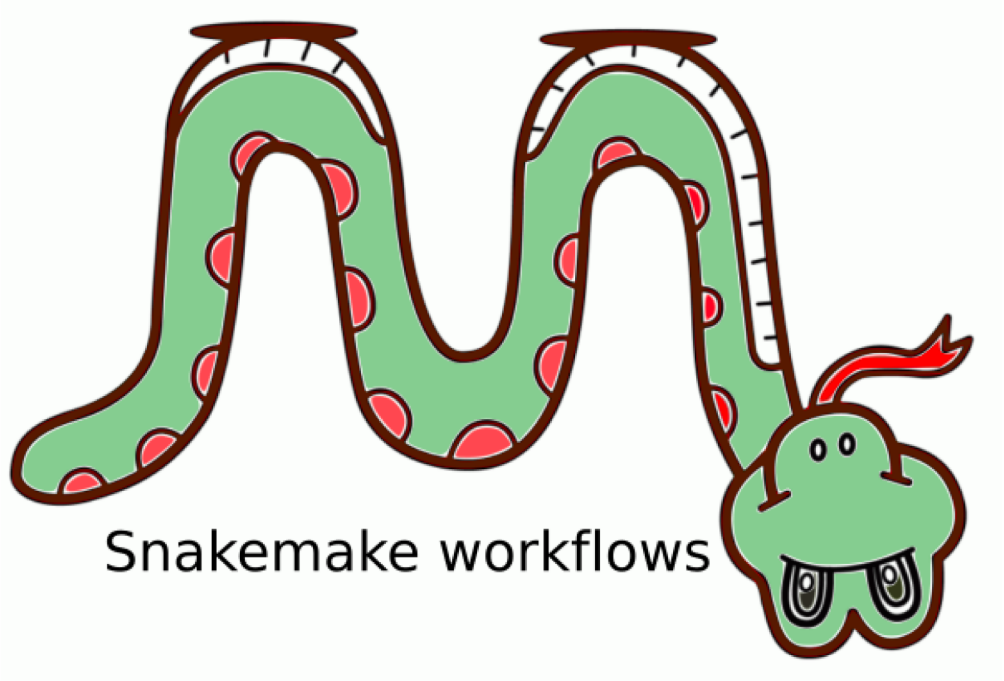
First, go to https://www.rocker-project.org/images/ choose the image you want. I use `tidyverse` heavily, so I downloaded the `tidyverse` image buit upon `Rstudio` image

```
## ssh to remote HPC and pull the docker image by singularity
ssh bio1
mkdir singularity-images; cd !$
singularity pull --name rstudio.simg docker://rocker/tidyverse:latest


# This example bind mounts the /project directory on the host into the Singularity contai
# By default the only host file systems mounted within the container are $HOME, /tmp, /pr
# type in the password you want to set, make it more complicated than this dummy one
PASSWORD='xyz' singularity exec --bind=/project  rstudio.simg rserver --auth-none=0  --au
```

- https://divingintogeneticsandgenomics.rbind.io/post/run-rstudio-server-with-singularity-on-hpc/

Snakemake for pipelines

- https://snakemake.readthedocs.io/en/stable/
- tutorials
- https://github.com/ctb/2019-snakemake-ucdavis
- https://hackmd.io/jXwbvOyQTqWqpuWwrpByHQ?view



Many workflow languages/engines

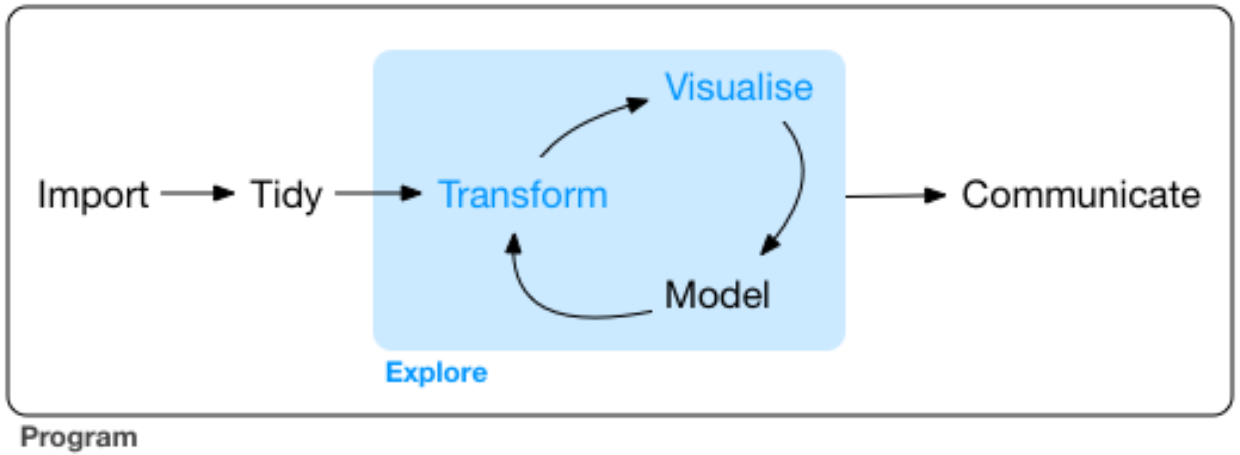
Awesome Pipeline

A curated list of awesome pipeline toolkits inspired by Awesome Sysadmin

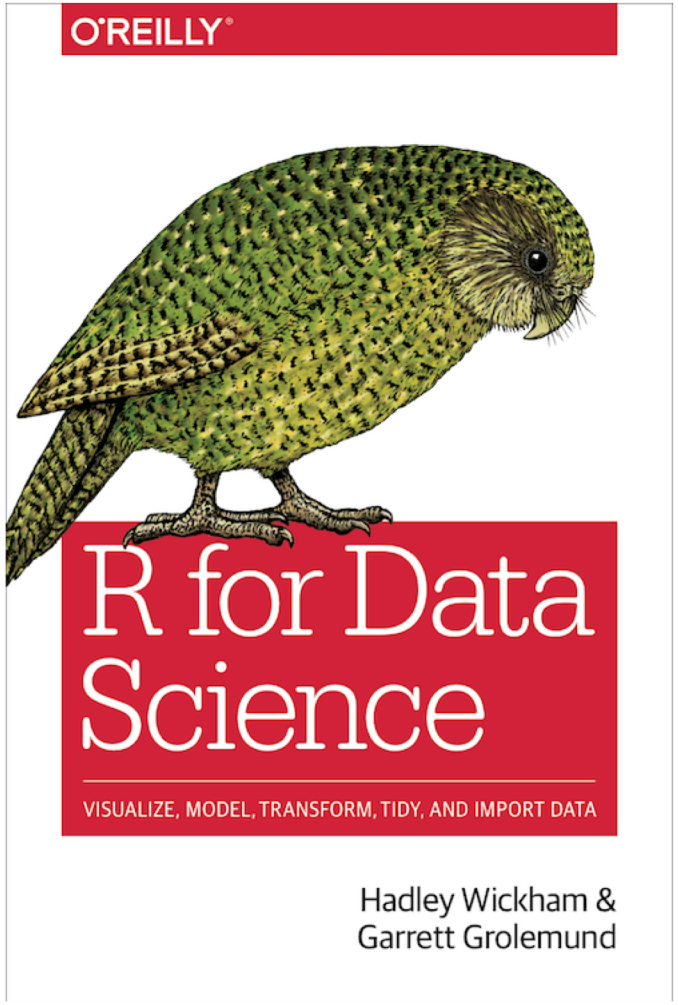
Pipeline frameworks & libraries

- ActionChain - A workflow system for simple linear success/failure workflows.
- Adage - Small package to describe workflows that are not completely known at definition time.
- Airflow - Python-based workflow system created by AirBnb.
- Anduril - Component-based workflow framework for scientific data analysis.
- Antha - High-level language for biology.
- AWE - Workflow and resource management system with CWL support
- Bds - Scripting language for data pipelines.
- BioMake - GNU-Make-like utility for managing builds and complex workflows.
- BioQueue - Explicit framework with web monitoring and resource estimation.
- Bioshake - Haskell DSL built on shake with strong typing and EDAM support
- Bistro - Library to build and execute typed scientific workflows.

Downstream analysis

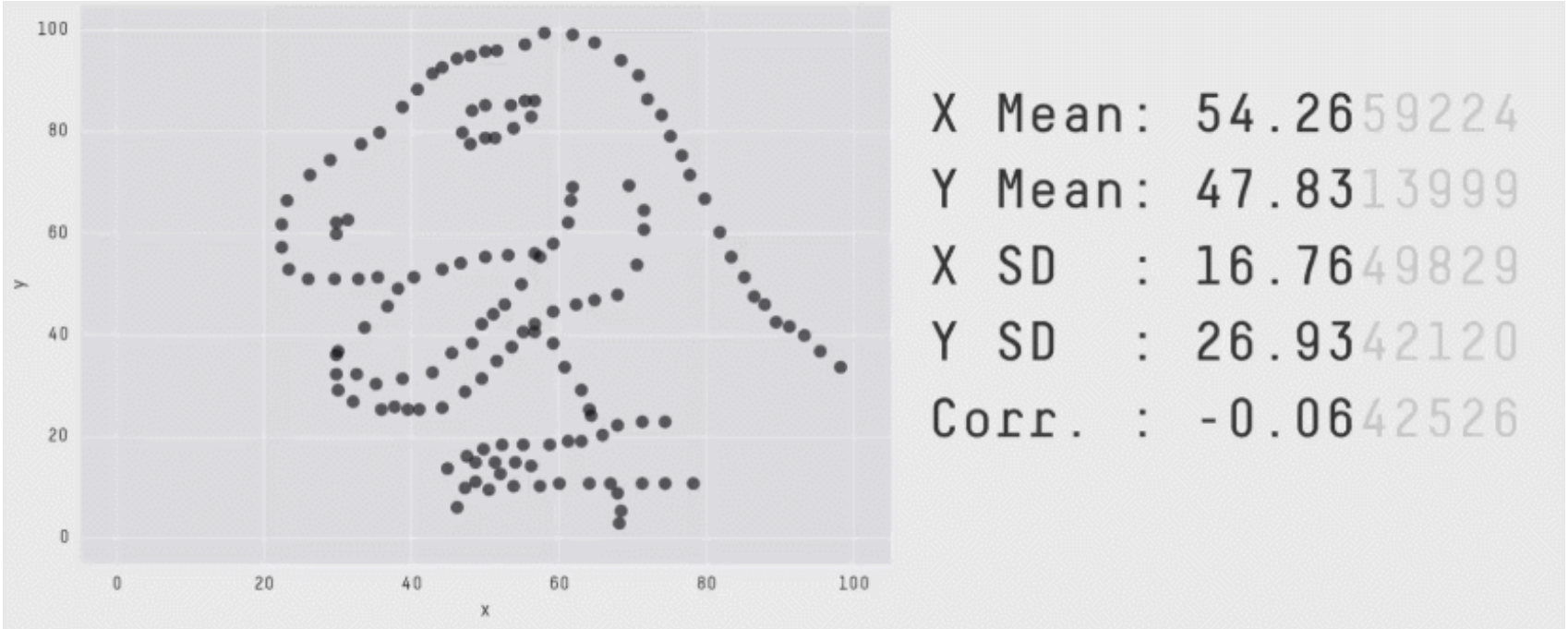


Tidying the data can take 80% of your time
Tidyverse



R for data science by Hadley Wickham & Garrett Grolemund
http://r4ds.had.co.nz/

Data visualization



https://www.r-bloggers.com/the-datasaurus-dozen/

One single suggestion

- Documentation! Documentation! And documentation!

One last suggestion: backup!
Backup by crontab

- https://divingintogeneticsandgenomics.rbind.io/post/crontab-for-backup/

commands for `crontab`:

```
# It took me forever to quit vim :) so avoiding it now.
export EDITOR=nano ;to specify a editor to open crontab file.

crontab -e    Edit crontab file, or create one if it doesn't already exist.
crontab -l    crontab list of cronjobs , display crontab file contents.
crontab -r    Remove your crontab file.
crontab -v    Display the last time you edited your crontab file. (This option is only av
```

crontab file

crontab syntax

```
*     *     *   *    *        command to be executed
-     -     -   -    -
|     |     |   |    |
|     |     |   |    +----- day of week (0 - 6) (Sunday=0)
|     |     |   +------- month (1 - 12)
|     |     +--------- day of        month (1 - 31)
|     +------------ hour (0 - 23)
+-------------- min (0 - 59)
```

#rsync every Sunday 5am.
0 5 * * 0 rsync -avhP --exclude=".aspera" --exclude=".autojump" --exclude=".bash_history' --exclude=".mozilla" --exclude=".myconfigs" --exclude=".oracle_jre_usage" --exclude=".parallel" --exclude=".pki" --exclude=".rbenv" railab:.[^.]* ~/shark_dotfiles >> /var/log/rsync_shark_dotfiles.log 2>&1