

Introductions

Jon. Library. Yangyou Chuan
Rob. Institute of Collaborative Biotechnologies. So much deliciousness, can't decide. True true.
Danielle. Technology Management Department. Mapa? Mapo? Tofu
Dawn. Letters and Science IT. The kung pao cauliflower from trader joe's
Kimberlee. PhD student in Chemical Engineering. I like all food equally
Paul. Materials. All types of Chow Mein!
Andrew. Senior undergrad studying Pharmacology. Sushi - fatty tuna
Adreina. History Department (PhD). Dumplings. kung pao shrimp and beef with broccoli sauce
Nazanin : noodle soup with fresh vegetables
Mary-Michelle: Library. Dim sum
Jason, Environmental Studies, Hokkien noodles
Philipp. Materials Research Laboratory, can never remember how it's called (describe it!)
Jenny. PhD student in Mechanical Engineering/BioEngineering. Beef & Broccoli
Xina. English. Dumplings.
Kristina, Post-Doc in Physics/Astrophysics. Mu shu pork, and Schezwan food in general.(I like veggie pizza)
Dana. PhD student in Ecology, Evolution, and Marine Biology. Jian dui.
Maria Teresa, Mech Eng, Szechuan Eggplants
Kai. PhD student in Ecology, Evolution, and Marine Biology. Shu Mai.
Sharon Solis, Research Computing Consultant, chao nian gao (stir fry sticky rice cake)
Ian: library staff, dumplings! SteamEm
Yuan:library staff, dumplings, too.
Mary Donovan, Postdoc at UCSB Marine Science Institute

How do we create a shell 'alias' ???

try this command:

```
alias rm="rm -i"
```

This will alias the rm command to rm -i. this makes rm interactive by default.

Also see this short tutorial: <https://mijingo.com/blog/creating-bash-aliases>

Welcome to The Carpentries Etherpad!

This pad is synchronized as you type, so that everyone viewing this page sees the same text. This allows you to collaborate seamlessly on documents.

Use of this service is restricted to members of The Carpentries community; this is not for general purpose use (for that, try etherpad.wikimedia.org).

Users are expected to follow our code of conduct: https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html

All content is publicly available under the Creative Commons Attribution License: <https://creativecommons.org/licenses/by/4.0/>

Link to data-shell download page: <https://ucsbcarpentry.github.io/2019-02-21-UCSBLibrary/shell-novice/setup/index.html>

Bash commands (notes on the board): <https://ucsbcarpentry.github.io/2019-02-21-UCSBLibrary/files/2019-02-21-shell-commands-white-board.jpg>

wc - count lines (or words)

pwd - print working directory

ls - list (-f, -l)
cd - change dir
.. - shortcut to parent
<tab> - auto complete
man - manual (windows: --help)
<up arrow> - repeat, history
sort - sort file
~ - home directory
mkdir - make directory
cat - show contents
^c (ctrl+c) - exit/escape
head - prints first few lines
touch - create a file
mv - move, rename
* - matches 0 or many
rm - remove, delete
rmdir - directory removal
grep - look in files
find - find files by patterns
echo - print arguments
mv - move/rename
clear - clear window
git init, add, commit, status, diff, pull, push

LUNCH: the cost of entry

- give us a used sticky.
- one thing you learned or liked
one thing you're confused about

Ian's real world command line piping and sorting example:

given a file listing like in:

<https://ucsbcarpentry.github.io/2019-02-21-UCSBLibrary/files/file-listing-sampler.txt>

use these unix command:

```
cut -d _ -f 2 repo-file-listing-sampler.txt |sort |uniq -c|sort -n -r
```

When composing a git "Commit" message try to answer the question of "Why" you're making the changes. Giving importance/context of your changes made, useful to you as a future reader and your collaborators. Helpful to add this into your workflow

A good online tutorial for Git: <https://www.atlassian.com/git/tutorials/undoing-changes>

An example of open science in R

<https://rdcu.be/bn2iI>

https://github.com/fishymary/Hawaii_regimes

<https://zenodo.org/record/1293045#.XHAjT5NKhYI>

Some R resources for more fun and learning:

Rstudio cheatsheets: <https://www.rstudio.com/resources/cheatsheets/>

R for Data Science: <https://r4ds.had.co.nz/>

Tidyverse: <https://www.tidyverse.org/>

Day 2!!!!!! *****

Logistics

Schedule

colorspace is that awesome color picker package.

https://raw.githubusercontent.com/swcarpentry/r-novice-gapminder/gh-pages/_episodes_rmd/data/gapminder_data.csv

For Windows users, there is an extra step to set Git Bash for your Terminal . Here they are (<https://support.rstudio.com/hc/en-us/articles/115010737148-Using-the-RStudio-Terminal#appendix>)

Terminal#appendix)

Tools->Global Options-Terminal

- **New Terminals open with Git Bash:** Use the git-bash shell found with Git for Windows, if installed
- New Terminals open with Command Prompt: Use the standard Windows command prompt (both 32 and 64-bit options will be presented if system supports 64-bit)
- **New Terminals open with PowerShell:** Use the Windows PowerShell, if available
- **New Terminals open with Bash (Windows Subsystem for Linux):** requires 64-bit Windows-10, and must be separately installed

why does R use <- instead of = ?

is there some disadvantage of using '=' ? using '<-' is harder to type

There is an article, which may answer your question:

<https://colinfay.me/r-assignment/>

GGPLOT inspiration:

<https://www.economist.com/graphic-detail/>

<https://www.r-graph-gallery.com/>

Ian asks: I don't understand the difference between a 'vector' and a 'factor'.

- - - a vector is a way of storing data, other types include data.frames, lists, and matrices
 - - a factor is one datatype, other types include characters, numeric, etc

What are the limitations of these data types?

When should I use one or the other?

- - - factors are used primarily in statistical analyses to represent categorical data. They can also be useful in optimizing your computer's performance since integers are easier/faster to process than large character strings
- great questions and answers here!

Wes Anderson movies color palettes
<https://github.com/karthik/wesanderson>