Happy Git and GitHub for the useR

@JennyBryan
@jennybc

@STAT545
stat545-ubc.github.io
Slides used in live workshops based on:

happygitwithr.com
Given the self-guided CSAMA lab format … here’s a little guidance for you.

Optional: read Why Git? Why GitHub?

Optional: look at the rest of these slides.

Prove you exist and have a GitHub account 😊
- Put an emoji on this GitHub issue:
  CSAMA lab participants: What is your operating system?

Complete everything in these sections of happygitwithr.com
- Installation
- Connect Git, GitHub, RStudio
- Get help from us if you need it! Setup is half the battle here.
Given the self-guided CSAMA lab format ... here's a little guidance for you.

Depending on your background and ambitions ...

Do something from Early Usage section:
- New project, GitHub first
- Existing project, GitHub first
- Existing project, GitHub last

Or choose something more advanced from Tutorial prompts section:
- Clone a repo
- Fork a repo
- Create a bingo card ... maybe for CSAMA?
- Practice failure: burn it all down
- Search GitHub ... maybe Bioconductor specific searches?
Version control systems (VCS) were created to help groups of people develop software

Git, in particular, is being “repurposed” for activities other than pure software development … like the messy hybrid of writing, coding and data wrangling … and exposing the gory details… and collaborating with others
What would Git adoption feel like?

Install Git. Configure it. Affirm RStudio can find it.

R project? Pre-existing or new. Dedicate a directory to it. Make that an RStudio Project. Make that a Git repository. Do your usual thing but … instead of just saving, you also make commits. Push to GitHub periodically.
RStudio will offer a Git pane to help you make commits, view history and diffs, and push to / pull from GitHub.
You — and possibly other people! — could visit the project on GitHub. For browsing and much more.
Do this now:

Put an emoji on this GitHub issue:

CSAMA lab participants: What is your operating system?
Our battle-tested installation instructions:

1. Installation
2. Happy Git with R

4. Installation pain
5. Register a GitHub account
6. Install or upgrade R and RStudio
7. Install Git
8. Introduce yourself to Git
9. Install a Git client

Connect Git, GitHub, RStudio

10. Connect to GitHub
11. Cache credentials for HTTPS
12. Set up keys for SSH
13. Connect RStudio to Git and GitHub
14. Detect Git from RStudio
15. RStudio, Git, GitHub Hell
By now you should have done this:

Complete everything in these sections of happygitwithr.com
- Installation
- Connect Git, GitHub, RStudio
- Get help from us if you need it! Setup is half the battle here.
NAME

git-update-ref - Update the object name stored in a ref safely

SYNOPSIS

    git update-ref [-m <reason>] (-d <ref> [<oldvalue>] | [--no-deref] <ref> <newvalue> [<oldvalue>] | --stdin [-z])

DESCRIPTION

Given two arguments, stores the <newvalue> in the <ref>, possibly dereferencing the symbolic refs. E.g. git update-ref HEAD <newvalue> updates the current branch head to the new object.

Given three arguments, stores the <newvalue> in the <ref>, possibly dereferencing the symbolic refs, after verifying that the current value of the <ref> matches <oldvalue>. E.g. git update-ref refs/heads/master <newvalue> <oldvalue> updates the master branch head to <newvalue> only if its current value is <oldvalue>. You can specify 40 "0" or an empty string as <oldvalue> to make sure that the ref you are creating does not exist.

It also allows a "ref" file to be a symbolic pointer to another ref file by starting with the four-byte header sequence of "ref:"

More importantly, it allows the update of a ref file to follow these symbolic pointers, whether they are symlinks or these "regular file symbolic refs". It follows real symlinks only if they start with "refs/": otherwise it will just try to read them and update them as a regular file (i.e. it will allow the filesystem to follow them, but will overwrite such a symlink to somewhere else with a regular filename).
NAME

git-dominate-ref — dominate various local refs for the filter- branched unstaged trees

SYNOPSIS

    git-dominate-ref [ --diagnose-pack | --jam-kick-tip | --slide-commit ]

DESCRIPTION

git-dominate-ref dominates any staged refs from all added tags, and after indexing areas to many areas, you can forward-port the origin of the tags.

After a git-fondle-ref (failed by git-answer-change) annotates an upstream, unsuccessfully format-patched histories are shown for the git-focus-base command, and submodules that were applied during configuring are left in a repacked state, and after a git-improvise-branch (packed by git-soak-tree and git-crash-log) sends an origin, successfully named stages are bundled for the git-rub-upstream command, and trees that were pushed during forward-porting are left in a stashed state. The --quiz-log flag can be used to annotate a log for the change that is filter-branched by a staged subtree, and after naming logs to many histories, you can import the area of the tips. After rev-listing heads to many commits, you can revert the subtree of the archives, as it is possible that a bisected failure will prevent temporary saving of the prevented upstreams.
highly recommended stress relief

https://git-man-page-generator.lokaltog.net
https://github.com/Lokaltog/git-man-page-generator
http://www.commitlogsfromlastnight.com
http://starlogs.net
This is a good time to

Explore some of the comic relief resources, so you can find them again when Git despair strikes.
agony : flow
agony : flow
agony reduction
Use a Git client

RStudio might not be enough — some noticeable gaps

I 💗 SourceTree (free, Mac OS + Windows)

More recommendations here:

http://happygitwithr.com/git-client.html
SourceTree, a free Git client for Windows and Mac.
Or do it like this ... it's your call.

heffalump:~ cewing$
heffalump:~ cewing$ source virtualenvs/sublenv/bin/activate
[sublenv]
heffalump:~ cewing$ cd projects/training/training.gotg/
[sublenv]
[master=]
heffalump:training.gotg cewing$ git checkout -b add_foobar
Switched to a new branch 'add_foobar'
[sublenv]
[add_foobar]
heffalump:training.gotg cewing$ touch foobar.txt
[sublenv]
[add_foobar]
heffalump:training.gotg cewing$ git add foobar.txt
[sublenv]
[add_foobar +]
heffalump:training.gotg cewing$ git reset HEAD foobar.txt
[sublenv]
[add_foobar]
heffalump:training.gotg cewing$ rm foobar.txt
[sublenv]
[add_foobar]
heffalump:training.gotg cewing$ deactivate
[add_foobar]
heffalump:training.gotg cewing$ cd ..
heffalump:training cewing$ echo $LESSON_LEARNED
what a nice, informative prompt
heffalump:training cewing$ ✅
RStudio can also act as your Git(Hub) client

http://www.rstudio.com/ide/docs/version_control/overview
This is a good time to do:

From Installation:
- Install a Git client.

From Early Usage section:
- New project, GitHub first
Use GitHub

Or Bitbucket or Gitlab or …

Even if you keep things private and don’t collaborate.

Commit and push early and often!

Why, you ask?
The amount of energy necessary to refute bullshit is an order of magnitude bigger than to produce it.

- Alberto Brandolini
The amount of Git skilz necessary to fix a borked up repo is an order of magnitude bigger than to bork it.

- Me
BURN IT ALL DOWN
This is Git. It tracks collaborative work on projects through a beautiful distributed graph theory tree model.

Cool. How do we use it?

No idea. Just memorize these shell commands and type them to sync up. If you get errors, save your work elsewhere, delete the project, and download a fresh copy.
This is Git. It tracks collaborative work on projects through a beautiful distributed graph theory tree model.

Cool. How do we use it?

No idea. Just memorize these shell commands and type them to sync up. If you get errors, save your work elsewhere, delete the project, and download a fresh copy.
If that doesn't fix it, git.txt contains the phone number of a friend of mine who understands git. Just wait through a few minutes of 'It's really pretty simple, just think of branches as...' and eventually you'll learn the commands that will fix everything.
This is a good time to do:

From Tutorial prompts section:
- Practice failure: burn it all down
increase flow
machine readable

&

human readable
code can be machine & human readable
data

can be

machine & human readable
your project can be machine & human readable
code comments

explanation-of-mystifying-variable-names-and-codes.txt

README
What is here? When did it last change? Who changed it? Why did they change it? Can I have it? Oh, I want that other version.
Commits are how the files evolve
Commit message = short description of what/why changed
“diffs” show what actually changed
Issues for bug reports, feature requests, to do list, ...
This is a good time to

Find a package you use that is developed on GitHub

Explore its code, releases, and issues

or

Find a data analytical project you find interesting on GitHub

Explore its code! Maybe run it!

FiveThirtyEight shares data and code behind its stories and interactives at on GitHub

https://github.com/fivethirtyeight/data
collaboration
in theory

more typical

GitHub

adapted from https://www.atlassian.com/git/tutorial/git-basics#!/clone
Note the contributions to STAT 545 materials from one prof, 3 TAs, and one kind soul from the internet.
“Pull requests” are a mechanism to propose, discuss, and merge changes into a repository.
when you’re the boss:

link to evolving files, don’t attach static copies to email

plain text everything you can

use Git

put it on the internet somewhere

when you’re not the boss:

try to talk everyone into Google Docs 😊
get a pseudo-website for free
The unreasonable effectiveness of GitHub browsability

- Be savvy about your files
- Get over your hang ups re: committing derived products
- Markdown
- README.md
- Finding stuff
- HTML
- Source code
- Delimited files
- PNGs
- Linking to a ZIP archive of your repo
- Links and embedded figures
- Let people correct you on the internet

One of my favorite aspects of GitHub is the ability to inspect a repository's files in a browser. Certain practices make browsing more rewarding and can postpone the day when you must create a proper website for a project. Perhaps indefinitely.
Impressive showing by Git here
Comma (.csv) and tab (.tsv) delimited files are automatically rendered nicely in GitHub repositories. Example: some Lord of the Rings data.
When rendered Markdown is no longer enough …

<my_thing>.github.io
When we push *.html to STAT 545 repo, website updates!
the poor woman’s regression test of a data analysis
Your code’s the same
Your data’s the same
But you updated R + pkgs
Surprise!
replace gdata with readxl
<table>
<thead>
<tr>
<th></th>
<th>-69.86</th>
<th>+69.86</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>Argentina</td>
<td>Americas</td>
<td>2007</td>
<td>75.32</td>
<td>60</td>
<td>Argentina</td>
<td>Americas</td>
</tr>
<tr>
<td>61</td>
<td>-Armenia</td>
<td>1952</td>
<td>62.809</td>
<td>61</td>
<td>+Armenia</td>
<td>NA</td>
<td>1952</td>
</tr>
<tr>
<td>62</td>
<td>-Armenia</td>
<td>1957</td>
<td>64.928</td>
<td>62</td>
<td>+Armenia</td>
<td>NA</td>
<td>1957</td>
</tr>
<tr>
<td>63</td>
<td>-Armenia</td>
<td>1962</td>
<td>67.055</td>
<td>63</td>
<td>+Armenia</td>
<td>NA</td>
<td>1962</td>
</tr>
<tr>
<td>64</td>
<td>-Armenia</td>
<td>1967</td>
<td>69.211</td>
<td>64</td>
<td>+Armenia</td>
<td>NA</td>
<td>1967</td>
</tr>
<tr>
<td>65</td>
<td>-Armenia</td>
<td>1972</td>
<td>70.786</td>
<td>65</td>
<td>+Armenia</td>
<td>NA</td>
<td>1972</td>
</tr>
<tr>
<td>66</td>
<td>-Armenia</td>
<td>1977</td>
<td>70.595</td>
<td>66</td>
<td>+Armenia</td>
<td>NA</td>
<td>1977</td>
</tr>
<tr>
<td>67</td>
<td>-Armenia</td>
<td>1982</td>
<td>70.916</td>
<td>67</td>
<td>+Armenia</td>
<td>NA</td>
<td>1982</td>
</tr>
<tr>
<td>68</td>
<td>-Armenia</td>
<td>1987</td>
<td>68.408</td>
<td>68</td>
<td>+Armenia</td>
<td>NA</td>
<td>1987</td>
</tr>
<tr>
<td>69</td>
<td>-Armenia</td>
<td>1992</td>
<td>68.663</td>
<td>69</td>
<td>+Armenia</td>
<td>NA</td>
<td>1992</td>
</tr>
<tr>
<td>70</td>
<td>-Armenia</td>
<td>1997</td>
<td>70.377</td>
<td>70</td>
<td>+Armenia</td>
<td>NA</td>
<td>1997</td>
</tr>
<tr>
<td>71</td>
<td>-Armenia</td>
<td>2002</td>
<td>71.403</td>
<td>71</td>
<td>+Armenia</td>
<td>NA</td>
<td>2002</td>
</tr>
<tr>
<td>72</td>
<td>-Armenia</td>
<td>2007</td>
<td>71.965</td>
<td>72</td>
<td>+Armenia</td>
<td>NA</td>
<td>2007</td>
</tr>
<tr>
<td>73</td>
<td>-Aruba</td>
<td>1952</td>
<td>60.437</td>
<td>73</td>
<td>+Aruba</td>
<td>NA</td>
<td>1952</td>
</tr>
<tr>
<td>74</td>
<td>-Aruba</td>
<td>1957</td>
<td>64.381</td>
<td>74</td>
<td>+Aruba</td>
<td>NA</td>
<td>1957</td>
</tr>
<tr>
<td>75</td>
<td>-Aruba</td>
<td>1962</td>
<td>66.606</td>
<td>75</td>
<td>+Aruba</td>
<td>NA</td>
<td>1962</td>
</tr>
<tr>
<td>76</td>
<td>-Aruba</td>
<td>1967</td>
<td>68.336</td>
<td>76</td>
<td>+Aruba</td>
<td>NA</td>
<td>1967</td>
</tr>
<tr>
<td>77</td>
<td>-Aruba</td>
<td>1972</td>
<td>70.941</td>
<td>77</td>
<td>+Aruba</td>
<td>NA</td>
<td>1972</td>
</tr>
<tr>
<td>78</td>
<td>-Aruba</td>
<td>1977</td>
<td>71.83</td>
<td>78</td>
<td>+Aruba</td>
<td>NA</td>
<td>1977</td>
</tr>
<tr>
<td>79</td>
<td>-Aruba</td>
<td>1982</td>
<td>74.116</td>
<td>79</td>
<td>+Aruba</td>
<td>NA</td>
<td>1982</td>
</tr>
<tr>
<td>80</td>
<td>-Aruba</td>
<td>1987</td>
<td>74.494</td>
<td>80</td>
<td>+Aruba</td>
<td>NA</td>
<td>1987</td>
</tr>
<tr>
<td>81</td>
<td>-Aruba</td>
<td>1992</td>
<td>74.108</td>
<td>81</td>
<td>+Aruba</td>
<td>NA</td>
<td>1992</td>
</tr>
<tr>
<td>82</td>
<td>-Aruba</td>
<td>1997</td>
<td>73.811</td>
<td>82</td>
<td>+Aruba</td>
<td>NA</td>
<td>1997</td>
</tr>
<tr>
<td>83</td>
<td>-Aruba</td>
<td>2002</td>
<td>73.451</td>
<td>83</td>
<td>+Aruba</td>
<td>NA</td>
<td>2002</td>
</tr>
<tr>
<td>84</td>
<td>-Aruba</td>
<td>2007</td>
<td>74.239</td>
<td>84</td>
<td>+Aruba</td>
<td>NA</td>
<td>2007</td>
</tr>
<tr>
<td>85</td>
<td>-Australia</td>
<td>1950</td>
<td>69.02</td>
<td>85</td>
<td>+Australia</td>
<td>NA</td>
<td>1950</td>
</tr>
<tr>
<td>86</td>
<td>-Australia</td>
<td>1951</td>
<td>68.72</td>
<td>86</td>
<td>+Australia</td>
<td>NA</td>
<td>1951</td>
</tr>
</tbody>
</table>
subtle fig changes due to ggplot2 release