So What is Git??

December 11-12, 2013.

What is Git

Version Control Basics

Setting up git

- username
- computer name
- your first repostory

Your First Commit

- staging
- committing

What is a hash??

- git uses ‘hashes’ to track commits
- a hash is generated by a an algorithm run on the content of the commit
- hashes are unique to the commit (1.2 x 10^{24})
- small changes in content result in wildly different hashes - probability of collisions extremely small.
- distributed nature means that sequential commit numbers are meaningless

What is a hash?? (cont'd)

- R can generate hashes using the digest library.

For example try:

```
> library(digest)

> digest('QFC\_workshop', algo='sha1')

> digest('QFC workshop', algo='sha1')
```

Your Second Commit

- what just happened

Reverting to Initial Commit

Git Log

- provides a history of changes lead to current state
- multiple options to control output and format

from a command prompt in your working directory try:

```
> git log
> git log --oneline
```

or equivalently in emacs with magit

- C-c C-g L
- C-c C-g l

Git Diff

- disply line-by-line difference between commits
- by default shows difference between latest commit and current directory contents
- commit numbers and/or file names can be used as arguments
- parts of each changed file are shown for context
- new lines are green and prefixed with '+'
- removed line red and prefixed with '-'

Git Diff - screen capture

What to commit - .gitignore

- only source files need to checked into version control
  - .r, .rnw
  - .dat, .pin, .tpl
- create .gitignore file in projects root
- lists files and directories that will NOT be committed
- an example for admb projects found in ~/workshop/utils/.gitignore

When to commit

- commit early and often
- especially if tests pass or model converges
- immediately before reporting

Creating Branches

- easy to create branches
- branches s

Merging

Remote Repositories

- creating and configuring
- what they are

Pushing and Pulling to Remote Repositories

- workflow

Clone Existing Repository

Hooks

- files that run on when specific actions occur
- git has several hooks available
- post-commit and post-checkout hook used to integreate git and reproducible research
- need to be manually activated in each repository
- each commit or check out will result in file being written to working directory
- contents of the file (commit hash) can then be integrated into reporting products

Gotchas

- reports must be generated **after** commiting working directory