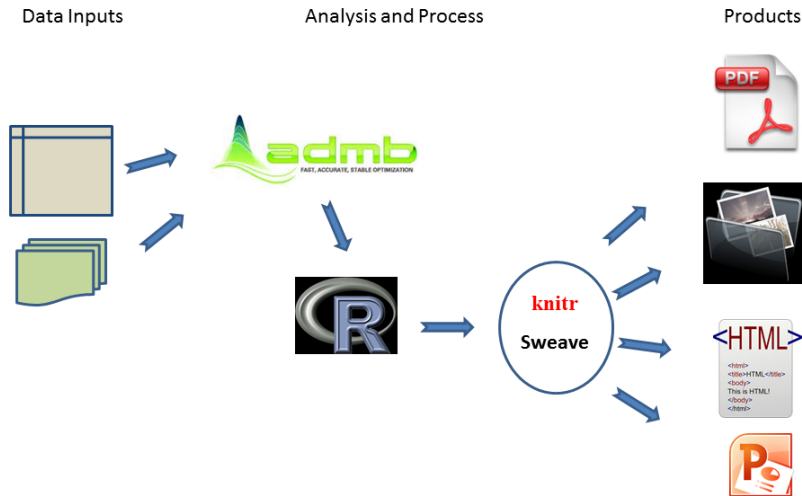


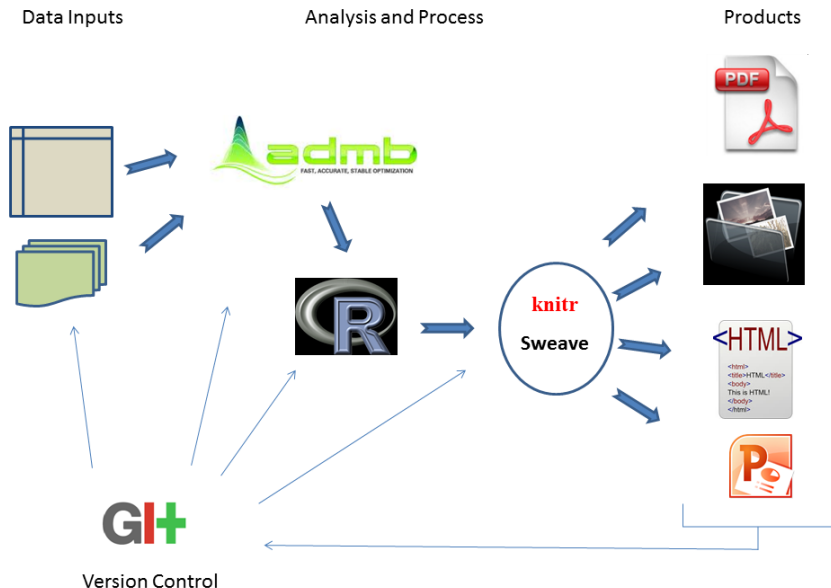
So What is Git??

December 11-12, 2013.

Our Alternative Work-flow (cont'd)



Our Alternative Work-flow (cont'd)



What is Git

Version Control Basics

Setting up git

from a command prompt:

```
> git config --global user.name "<your name>"  
> git config --global user.email "you@there.com"
```

Your First Repository

from a command prompt:

```
> cd <project root directory>
> git init
> dir /a
```

emacs and git: magit

- git command line tool
- several GUI's available
- git plug-in 'magit'

Your First Commit

- staging
- committing

What is a hash??

- git uses 'hashes' to track commits
- a hash is generated by an algorithm run on the content of the commit
- hashes are unique to the commit (1.2×10^{24})
- small changes in content result in wildly different hashes - probability of collisions extremely small.
- distributed nature means that git can't use sequential commit numbers

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 |

Git Diff

- display 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 be 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/utis/.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 integrate 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

Hooks

post-checkout hook

```
prefixes=". Sweave SweaveSEP"
echo $GIT_DIR
for pref in $prefixes
do
  git log -1 --date=short \
  --pretty=format:"\usepackage[%
    shash={%h},
    lhash={%H},
    authname={%an},
    authemail={%ae},
    authsdate={%ad},
    authidate={%ai},
    authudate={%at},
    commname={%an},
    commemail={%ae},
    commsdate={%ad},
    commidate={%ai},
```

Hooks

results in gitHeadInfo.gin:

```
\usepackage[%  
    shash={dabb2eb},  
    lhash={dabb2eb433a5d14bc45a8dae8aad7f43208d990},  
    authname={Adam Cottrill},  
    authemail={adam.cottrill@ontario.ca},  
    authsdate={2013-10-07},  
    authidate={2013-10-07 10:52:12 -0400},  
    authudate={1381157532},  
    commname={Adam Cottrill},  
    commemail={adam.cottrill@ontario.ca},  
    commsdate={2013-10-07},  
    commidate={2013-10-07 10:52:12 -0400},  
    commudate={1381157532},  
    refnames={ (HEAD, master)}  
{gitsetinfo}
```

Gotchas

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

Recap

- git distributed version control system
- designed for collaborative use
- magit emacs plugin for git
- hooks extend functionality