DSCI353-353m-453: Class 0b-p Bash Git BitBucket

2001-353-353m-453-01b-p-BashGitBitBucket

Roger H. French, Peitian Wang

16 January, 2020

Contents

1.2.3 R Learning Resources

- Peng: R Programming for Data Science (Book, in readings)
- Roger Peng's Youtube Playlist for 4 weeks of Coursera R Programming

1.2.4 SDLE Teatime Learning

- 2016 year was intro to datascience, R, Python, Git, LaTeX
- 2017 was more advanced topics including Hadoop and Spark and SparklyR
- 2018 continued with more advanced topics and review

1.2.4.1 SDLE TeaTime Learnings Materials are available Online

- 2018 SDLE Teatime Repo
  - 2018 contains the prior years code

  - 2016 SDLE Teatime Repo
  - 2017SDLE Teatime Repo
- SDLE Teatime Youtube Videos and Playlists

1.2.4.2 What we need setup by now for class

1. Setup Data Science Slack for class - Use case.edu email address
2. Setup Bitbucket Account - Use case.edu email address
3. Setup VDI

- Rstudio
- Drag icons of R, Rstudio, Git Bash, Spyder, Jupyter Notebook, Slack to desktop
- Slack client on ODS VDI
- Can put slack app on phone, or on your notebook

4. Setup Git - make H:\Git folder - git config name and email
5. Setup StackExchange
6. Setup Kaggle.com account - Use your caseID email as a google account for login
7. Setup CWRU HPC Data Science Cluster Account - I'll provide more information shortly
8. Git Clone - For Class-Prof Repo
    - Fork the prof repo up on bitbucket
        - remove `-prof` and rename your student repo with your `-caseID`
    - Clone your fork of the Prof Class Repo
        - Down onto your ODS VDI's H: drive in the `H:/Git/` folder
        - For those "new to R" 18-sdle-tea-time
    - for quick introduction to data science techniques and tools
    - `git clone git@bitbucket.org:cwrudsci/18-sdle-teatime.git`

1.2.5 Some students may not have forked the class repo?

- DSCI-353-353m-453 group in CWRU-DSCI team

1.2.6 Bash: The language of the Linux Console

- Bash is the command line processor of the Linux Console
- R has its own command line processor for the R Console
- Bash is the default Console for both Linux and for Mac
    - Mac's are based on BSD-Unix OS
    - A close variant of Linux, only different by the licensing
- Windows uses the DOS command line processor in its 'Command Prompt'

1.2.6.1 On our ODS VDI's we use Git Bash to work with Git

- MinGW64 is a little Linux OS running inside Windows
    - It has the standard Bash commands
    - And tools like vim (the visual text editor)

1.2.6.2 Lets see some Bash Commands we'll be using

- `ls` is the "list" command, to get a directory of files and folders
- `pwd` is the "present working directory" command, to know where you are
- `cd` is "change directory"
- `..` refers to the directory one up from where you are

- so `cd ..` moves you up one directory
- and `cd Downloads` would move you down into Downloads directory (if it exists)
- To copy a file use `cp`
- To move a file use `mv`

- To make a new directory use `mkdir`

1.2.6.3 A good resource for Bash Commands and Man pages

- Is An A-Z Index of the Bash command line for Linux
- There are many other resources too

1.2.7 Now lets start working with our local Git Server

- Using Git Bash to talk to it
  - Git is also a linux program
- All Git commands are entered at the Bash Prompt
- All Git commands start with `git` ‘ * So that the Bash prompt know who to send the subsequent command to

1.2.7.0.1 Check your Git Server Configuration

```
`git config --list`
```

1.2.7.0.2 Essential `git config --global`'s, Set your user info

- `git config --global user.name "[name]"`
- `git config --global user.email "[email address]"`
- `git config --global color.ui auto`

1.2.7.1 First we need to go up to Bitbucket and “Fork” the Prof. Repo

- This will give you a copy of Prof. Repo
  - In your personal account area
  - You want to change the ending from “Prof” to your caseID

1.2.7.2 Now you want to open Git Bash on Windows

- You need to save your Repos on your H: drive, NOT C drive
  - C Drive is restricted
  - H Drive is your personal area that follows your caseID login
- So in Git Bash
  - `pwd` will tell you your present working directory
  - `cd ..` moves up a directory
  - `pwd` to see where you have moved
  - Now change to H: `cd /h`
  - `pwd` see where you are
  - `ls` see what files are there
  - `mkdir Git` this will make a new directory at H:Git
    - So you’ll keep all your repositories under H:Git

1.2.7.3 Important Note: Windows ignores case, Linux and BSD-Unix (Mac) respect case

- So Git and git are the same on windows for a folder
- They are totally different on Linux or Mac
- Best practice Use capitals sparingly
- About only useful place is in CamelBack filenames
  - Since I said, no spaces in filenames
  - To make things readable, you can do CamelBack
  - Example: 1906ITFun-ADSwR-ThisIsMyReport-Name.rmd

1.2.8 Now lets Clone your personal class repo

- Now you want to Clone your personal class repo
  - This is a one time operation
  - To copy all the files and folders down to your local computer
- In Git Bash, you want to be at H:Git or h:Git Check with `pwd`
- Now go to your personal class repo on Bitbucket
  - And find the clone command
  - **For windows choose https protocol (Not ssh)**
- Copy the command
  - Its something like this
  - `git clone https://vuvlab@bitbucket.org/cwrudsci/20s-dsci353-353m-453-prof.git`
- Now that that is on your clipboard
- Go to you Git Bash, and use "Shift-Insert" (Not "Cntrl-v")
  - To copy it onto the Bash Command line
  - Hit enter, and watch a full copy of your repo being copied locally

1.2.9 For class repos

- Before each class, or whenever you want
- Up on Bitbucket
  - You should sync your fork
  - With my Prof repo
  - To get the latest file version and new files
  - After syncing
  - Now `git pull` to bring the updated files to your local git server

1.2.10 Now lets pull and push changes from to your repo

- cd into your repo's top folder
  - This can be done with tab completion
  - `cd 20s-d` and hit tab, it auto completes
- Now type `git pull` To see if there are any changes up on bitbucket And to pull these down and merge them in

1.2.11 Making local changes, Adding, Commmitting and Pushing

- Now change a local file by adding something into it
- Now you add this changed file to be tracked by Git
  - `git add --all :/`
- Now commit your changes
  - `git commit -m 'I have changed the readme.md'`

- Now push your changes up to Bitbucket, to your personal repo
  - `git push`

1.2.12 Links

- https://www.r-project.org
- https://help.ubuntu.com/community/UsingTheTerminal