R Programming
Lab 01

DATA SCIENTIST'S TOOLBOX OVERVIEW, BASIC R

Lecturer：Professor Pao-Ann Hsiung

Teaching Assistant: Hendrik

Embedded Systems Laboratory
National Chung Cheng University
Chiayi, Taiwan-62102

Learning Objective

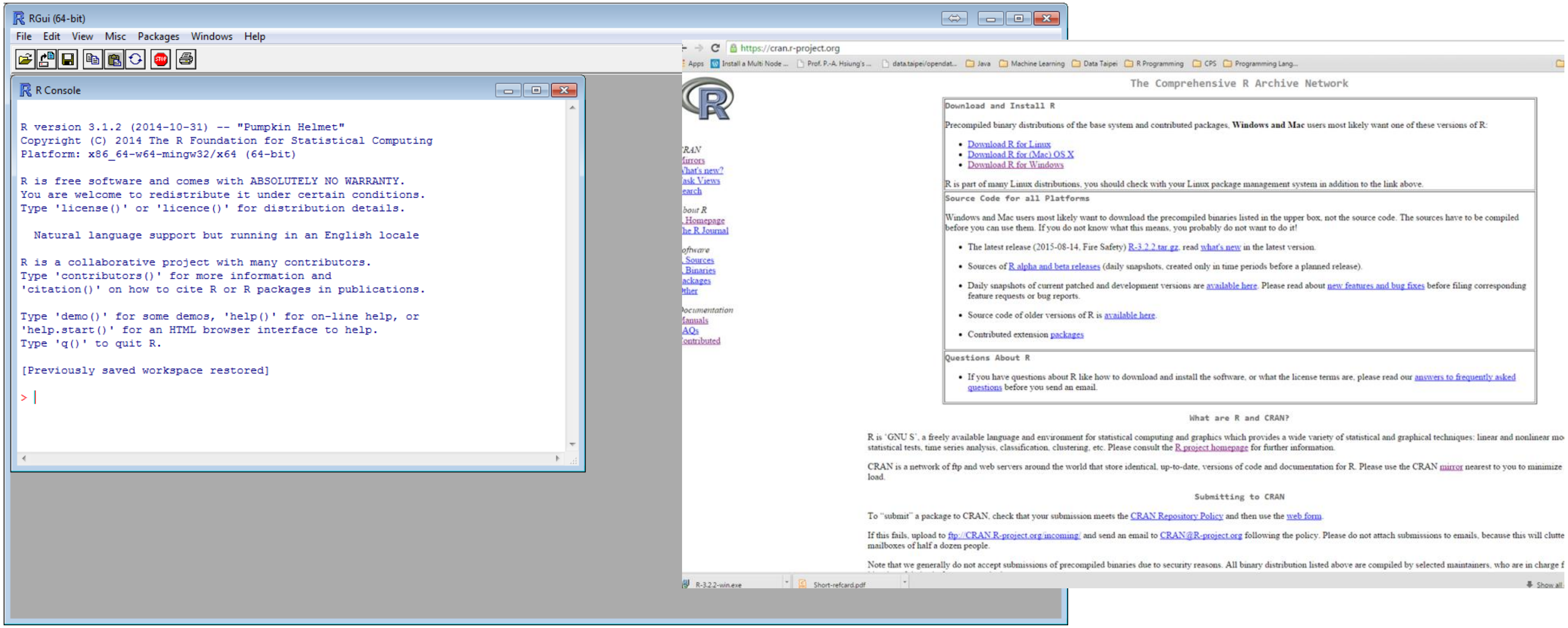
- Install the R and RStudio software packages
- Git and GitHub
- Download and install the swirl package for R
- Execute basic arithmetic operations
- Get to know the sense of R

Outline

- Data Scientist’s Toolbox
  - Introduction to basic tools
  - R&Rstudio
  - R tools
  - Git&Github
  - Getting Help
- Basic R
  - Data input
  - First program
  - R packages
- Demo & Assignment

R & RStudio

- R distribution available on https://cran.r-project.org/
-



```
R version 3.1.2 (2014-10-31) -- "Pumpkin Helmet"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> |
```

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux
- Download R for (Mac) OS X
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2015-08-14, Fire Safety) R-3.2.2.tar.gz, read what's new in the latest version.
- Sources of R alpha and beta releases (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are available here. Please read about new features and bug fixes before filing corresponding feature requests or bug reports.
- Source code of older versions of R is available here.
- Contributed extension packages

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

What are R and CRAN?

R is 'GNU S', a freely available language and environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques: linear and nonlinear mo statistical tests, time series analysis, classification, clustering, etc. Please consult the R project homepage for further information.

CRAN is a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R. Please use the CRAN mirror nearest to you to minimize load.

Submitting to CRAN

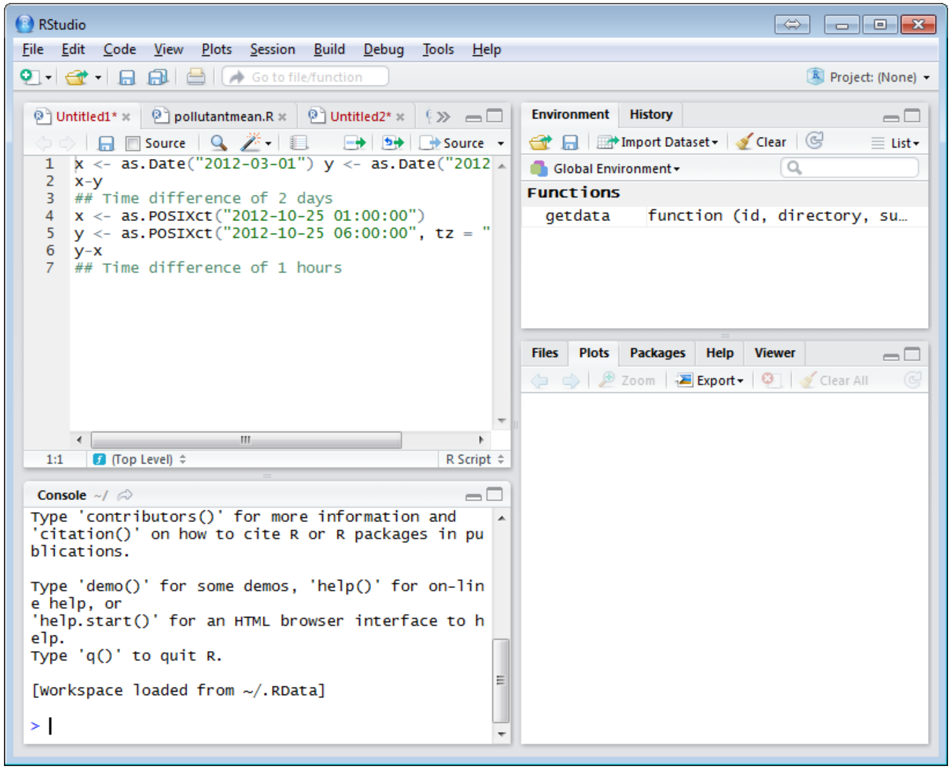
To "submit" a package to CRAN, check that your submission meets the CRAN Repository Policy and then use the web form.

If this fails, upload to ftp://CRAN.R-project.org/incoming/ and send an email to CRAN@R-project.org following the policy. Please do not attach submissions to emails, because this will clutte mailboxes of half a dozen people.

Note that we generally do not accept submissions of precompiled binaries due to security reasons. All binary distribution listed above are compiled by selected maintainers, who are in charge f

R & Rstudio

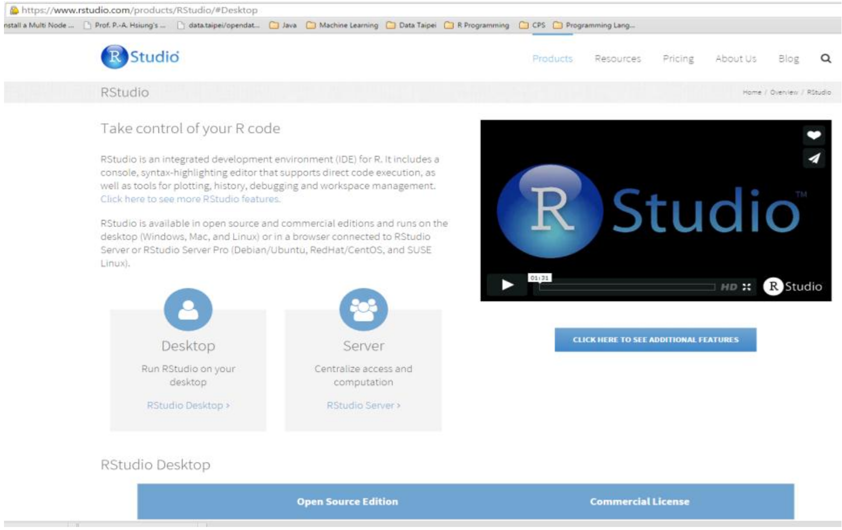
- **Rstudio**: IDE for R includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management
- Available on https://www.rstudio.com/products/rstudio/



```
x <- as.Date("2012-03-01") y <- as.Date("2012
x-y
## Time difference of 2 days
x <- as.POSIXct("2012-10-25 01:00:00")
y <- as.POSIXct("2012-10-25 06:00:00", tz = "
y-x
## Time difference of 1 hours
```

```
Functions
  getdata      function (id, directory, su…
```

```
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in pu
blications.

Type 'demo()' for some demos, 'help()' for on-lin
e help, or
'help.start()' for an HTML browser interface to h
elp.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

> |
```



RStudio

Take control of your R code

RStudio is an integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management. Click here to see more RStudio features.

RStudio is available in open source and commercial editions and runs on the desktop (Windows, Mac, and Linux) or in a browser connected to RStudio Server or RStudio Server Pro (Debian/Ubuntu, RedHat/CentOS, and SUSE Linux).

Desktop

Run RStudio on your desktop

RStudio Desktop ›

Server

Centralize access and computation

RStudio Server ›

CLICK HERE TO SEE ADDITIONAL FEATURES

RStudio Desktop

Open Source Edition | Commercial License

R Tools

- A collection of tools for building R packages in windows
- Available at https://cran.r-project.org/bin/windows/Rtools/
- For detail refer to that website

* Not necessary for linux or mac

Packages

- See available packages on CRAN

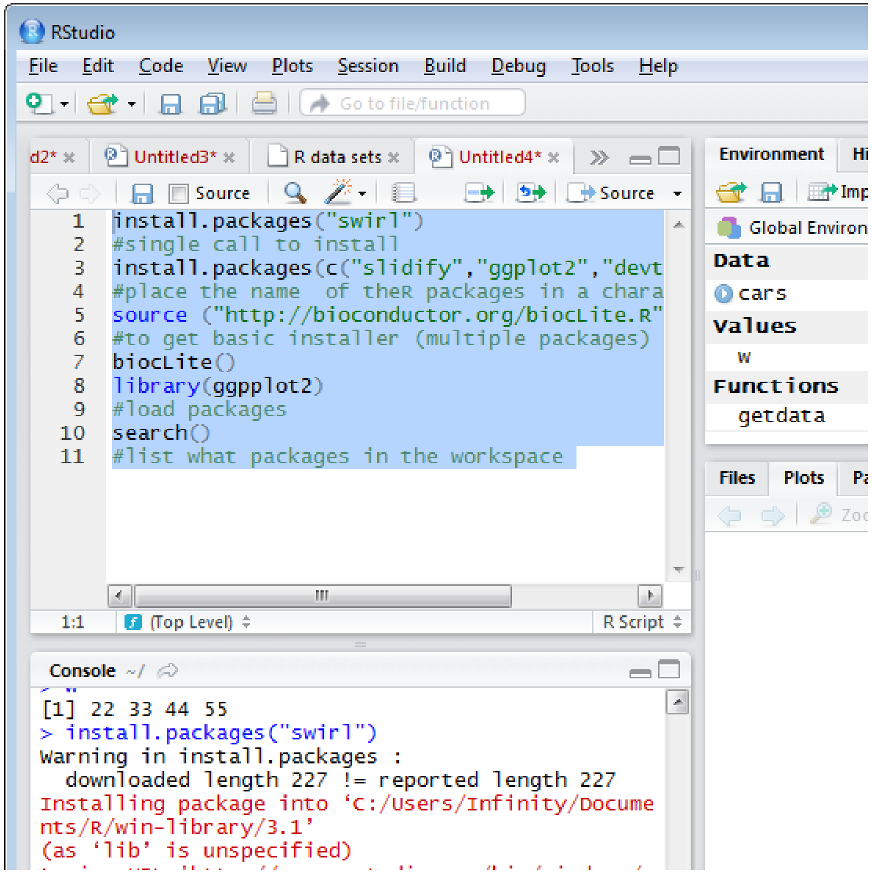
  available.packages()

- Install

  install.packages(“namesPackages")

- Load

  library(“namesPackages")



```
install.packages("swirl")
#single call to install
install.packages(c("slidify","ggplot2","devt
#place the name  of theR packages in a chara
source ("http://bioconductor.org/biocLite.R"
#to get basic installer (multiple packages)
biocLite()
library(ggpplot2)
#load packages
search()
#list what packages in the workspace
```

```
[1] 22 33 44 55
> install.packages("swirl")
Warning in install.packages :
  downloaded length 227 != reported length 227
Installing package into ‘C:/Users/Infinity/Docume
nts/R/win-library/3.1’
(as ‘lib’ is unspecified)
```

Git & GitHub

Vs

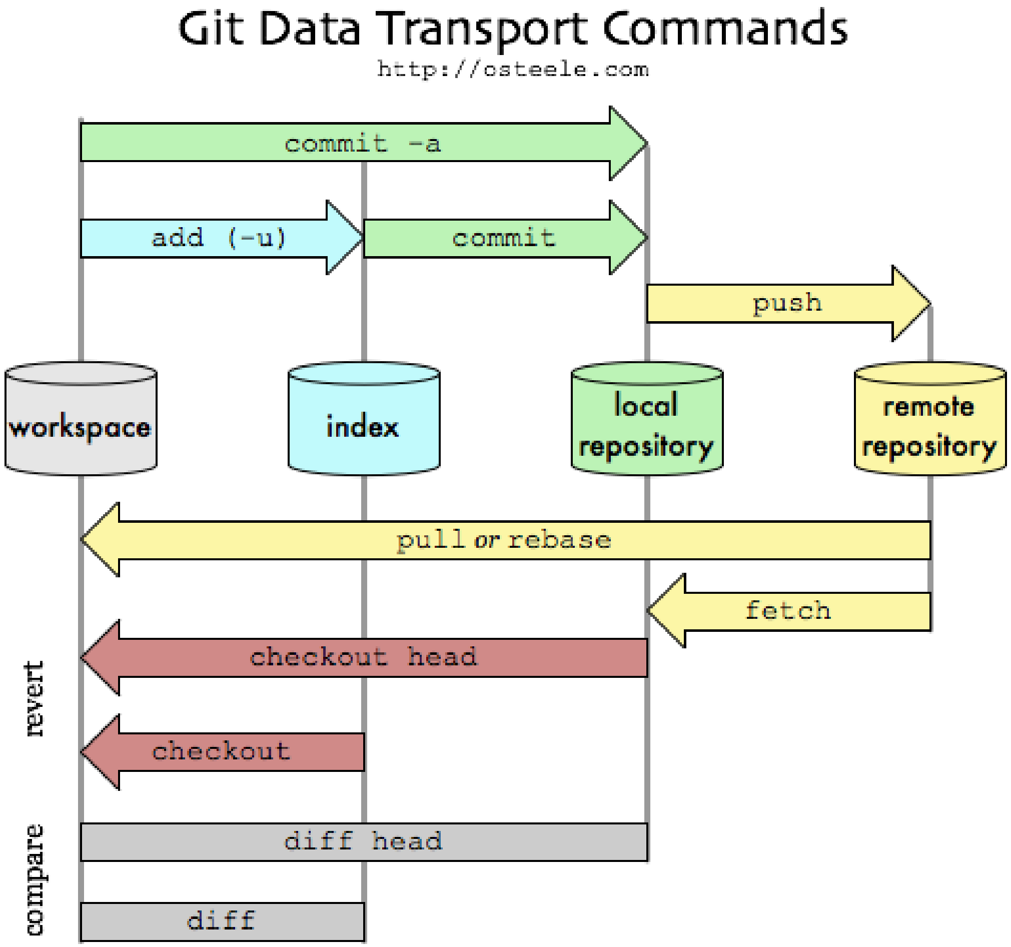
- You don’t need GitHub to use Git
- Git = Local (On your computer); Github=Remote (on the Web)
- GitHub Allow you to:

1. Share your Repositories with others
2. Access other user’s repositories
3. Store remote copies of your repositories (on GitHub’s Server) in case something happens to your local copies (on your computer)

Go to the following website to download https://git-scm.com/downloads

Git Basic

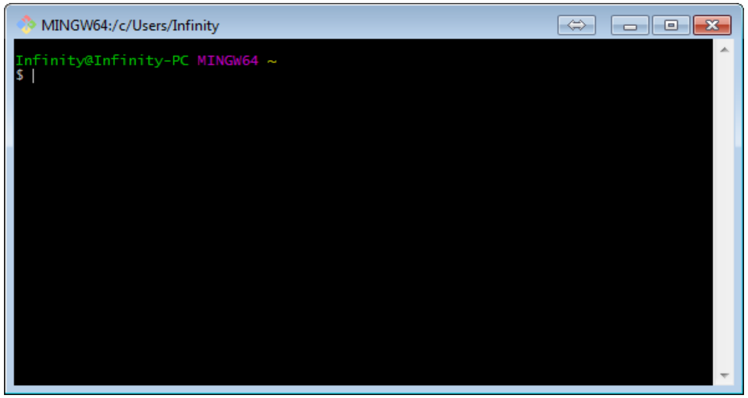
- init
- clone
- add
- status
- commit
- log
- diff
- checkout



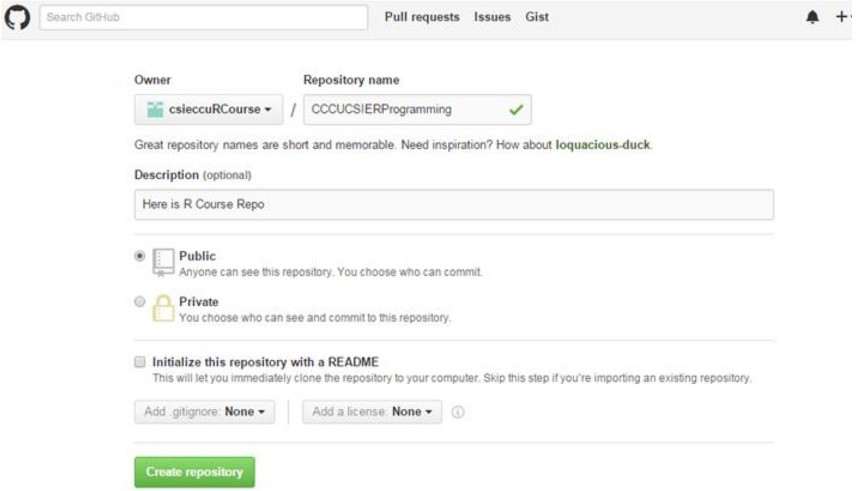
Getting started: Git and Github

- Create a your own github and make a new repository named it *Lab1_studentID*

- Open your git bash, download git here



```
Infinity@Infinity-PC MINGW64 ~
$ |
```



Owner / Repository name

csieccuRCourse ▾ / CCCUCSIERProgramming ✓

Great repository names are short and memorable. Need inspiration? How about **loquacious-duck**.

Description (optional)

Here is R Course Repo

● **Public**
Anyone can see this repository. You choose who can commit.

○ **Private**
You choose who can see and commit to this repository.

□ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾ | Add a license: **None** ▾

Creating GitHub Repo

- Two Methods
  1. Start Repository from scratch
  2. Fork another user’s repository
- Repo From Scratch
  - Go to your profile page(https://github.com/userNameHere) and click on “Create a new repo” in the righthand corner of the page
  - Or go directly to https://github.com/new
  - Select public(Private repos require a paid or educational account)
  - Check the box next to inialize this repository with a README

Creating a Local Copy

- Now you need to create a copy of this repo on your computer
- Open git bash
- Create a directory on your computer where you will store your copy of the repo

```
$ mkdir ~/test-repo
$ cd ~/test-repo
```

Initialize a local git repo

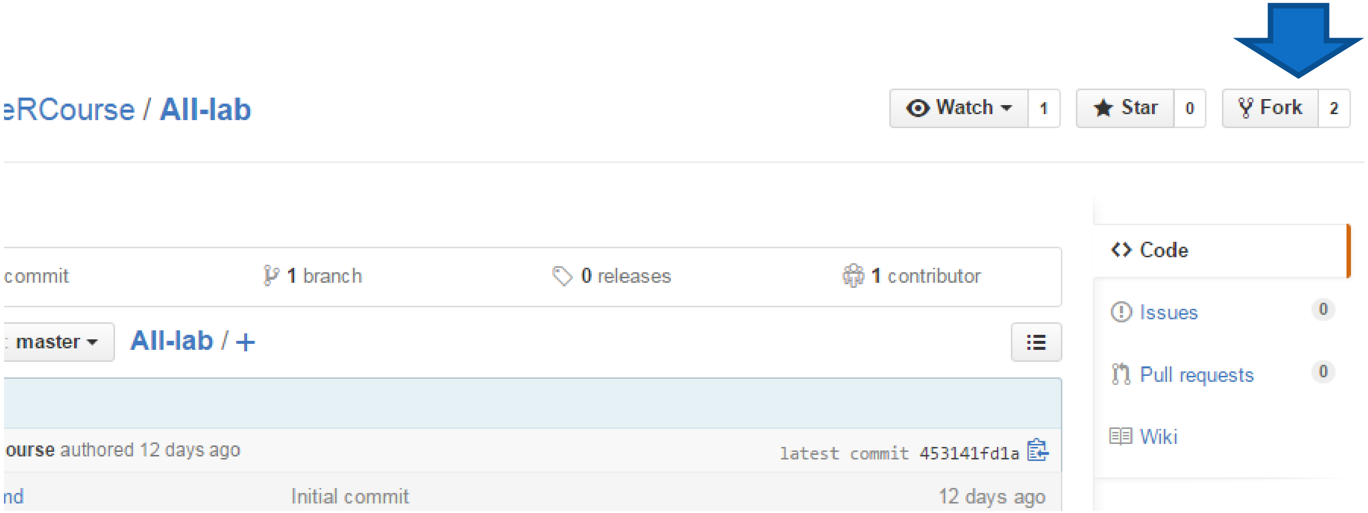
```
$ git init
```

Initialize Point your local repository at the remote repository you just created on the GitHub server

```
$ git remote add origin git@github.com:yourUserNamel/test-repo.git
```

Fork a Another User’s Repository

- The second method of creating a repository is to make a copy of someone else’s
- Begin by navigating to the desired repository on GitHub website and click the “Fork” button

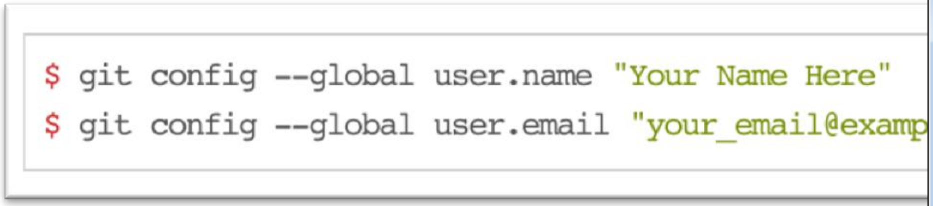


https://help.github.com/articles/fork-a-repo/

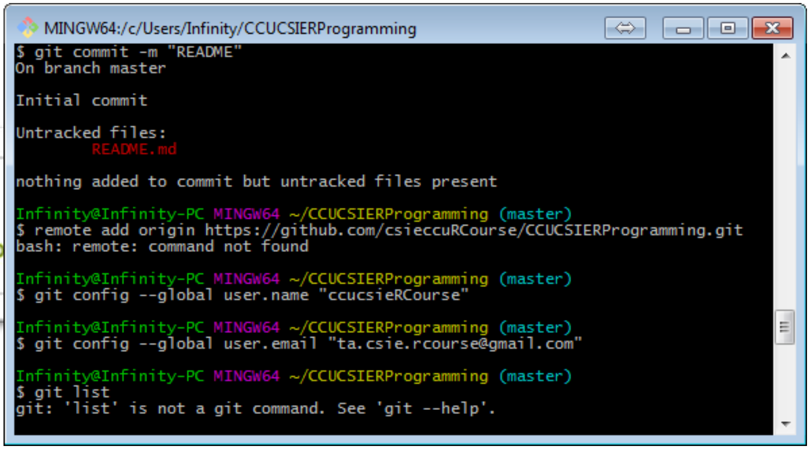
Configure username and email

- Each commit to a Git repository will be tagged with the username of the person who made the commit

- To set your username and email, Enter the following commands in Git Bash



```
$ git config --global user.name "Your Name Here"
$ git config --global user.email "your_email@examp
```

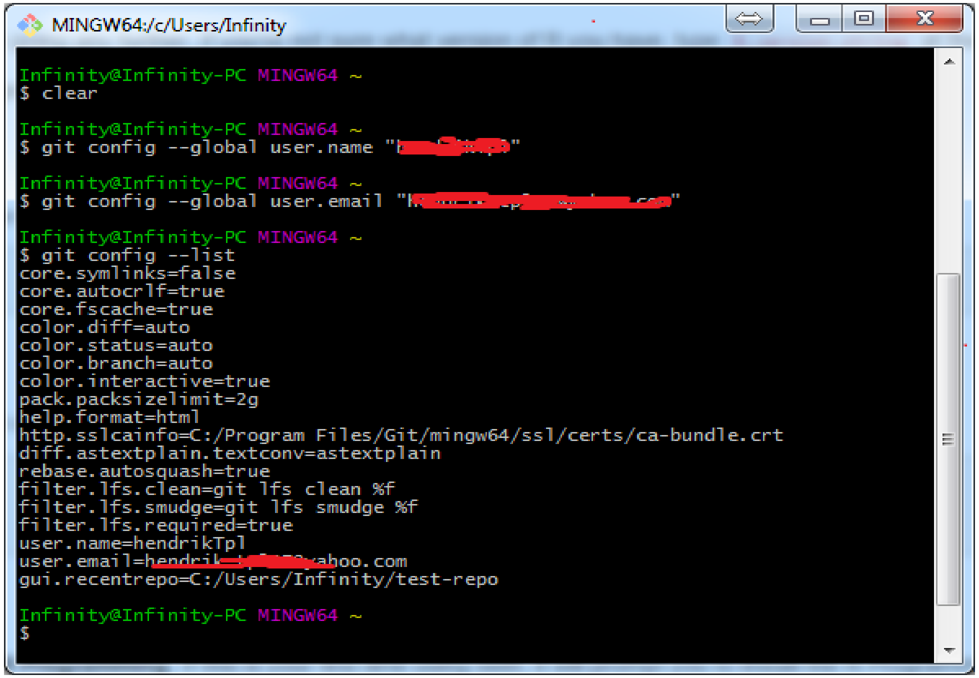


MINGW64:/c/Users/Infinity/CCUCSIERProgramming

```
$ git commit -m "README"
On branch master

Initial commit

Untracked files:
        README.md

nothing added to commit but untracked files present

Infinity@Infinity-PC MINGW64 ~/CCUCSIERProgramming (master)
$ remote add origin https://github.com/csieccuRCourse/CCUCSIERProgramming.git
bash: remote: command not found

Infinity@Infinity-PC MINGW64 ~/CCUCSIERProgramming (master)
$ git config --global user.name "ccucsieRCourse"

Infinity@Infinity-PC MINGW64 ~/CCUCSIERProgramming (master)
$ git config --global user.email "ta.csie.rcourse@gmail.com"

Infinity@Infinity-PC MINGW64 ~/CCUCSIERProgramming (master)
$ git list
git: 'list' is not a git command. See 'git --help'.
```

Cont.

- To confirm your changes, type the following command
  - $ git config –list

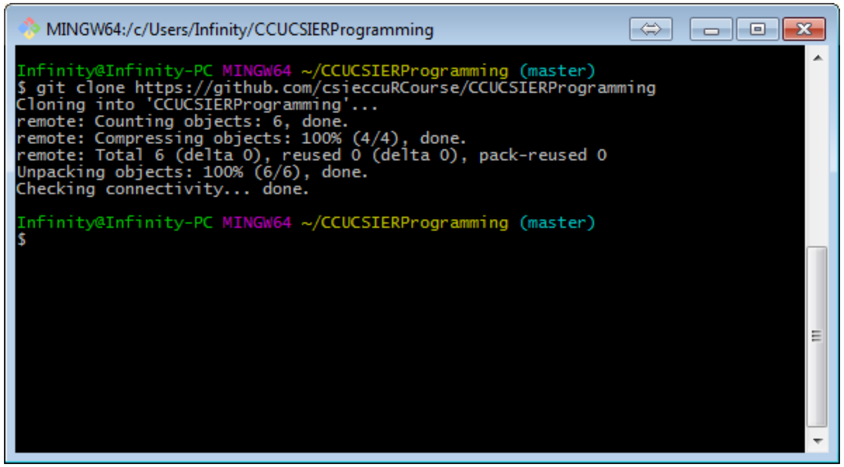


```
Infinity@Infinity-PC MINGW64 ~
$ clear

Infinity@Infinity-PC MINGW64 ~
$ git config --global user.name "[redacted]"

Infinity@Infinity-PC MINGW64 ~
$ git config --global user.email "[redacted]"

Infinity@Infinity-PC MINGW64 ~
$ git config --list
core.symlinks=false
core.autocrlf=true
core.fscache=true
color.diff=auto
color.status=auto
color.branch=auto
color.interactive=true
pack.packsizelimit=2g
help.format=html
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
diff.astextplain.textconv=astextplain
rebase.autosquash=true
filter.lfs.clean=git lfs clean %f
filter.lfs.smudge=git lfs smudge %f
filter.lfs.required=true
user.name=hendrikTpl
user.email=[redacted]ahoo.com
gui.recentrepo=C:/Users/Infinity/test-repo

Infinity@Infinity-PC MINGW64 ~
$
```

Clone

- To make a local repository on your computer
- to get a copy of an existing Git repository on your GitHub Account

```
git clone https://github.com/yourUserName/repoName
```



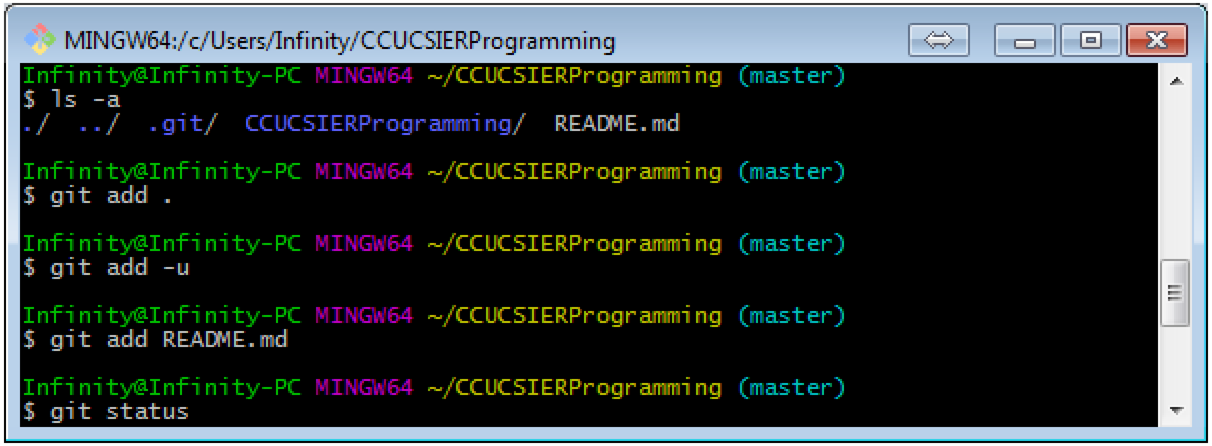
```
Infinity@Infinity-PC MINGW64 ~/CCUCSIERProgramming (master)
$ git clone https://github.com/csieccuRCourse/CCUCSIERProgramming
Cloning into 'CCUCSIERProgramming'...
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.
Checking connectivity... done.

Infinity@Infinity-PC MINGW64 ~/CCUCSIERProgramming (master)
$
```

This will clone repo into your current directory

Add

- Suppose you add new files to a local repository under version control
- You need to let git know that they need to be tracked
- Should do this before commiting

git add . adds all news file
git add –u updates tracking for files that changed names or were deleted
git add –A both

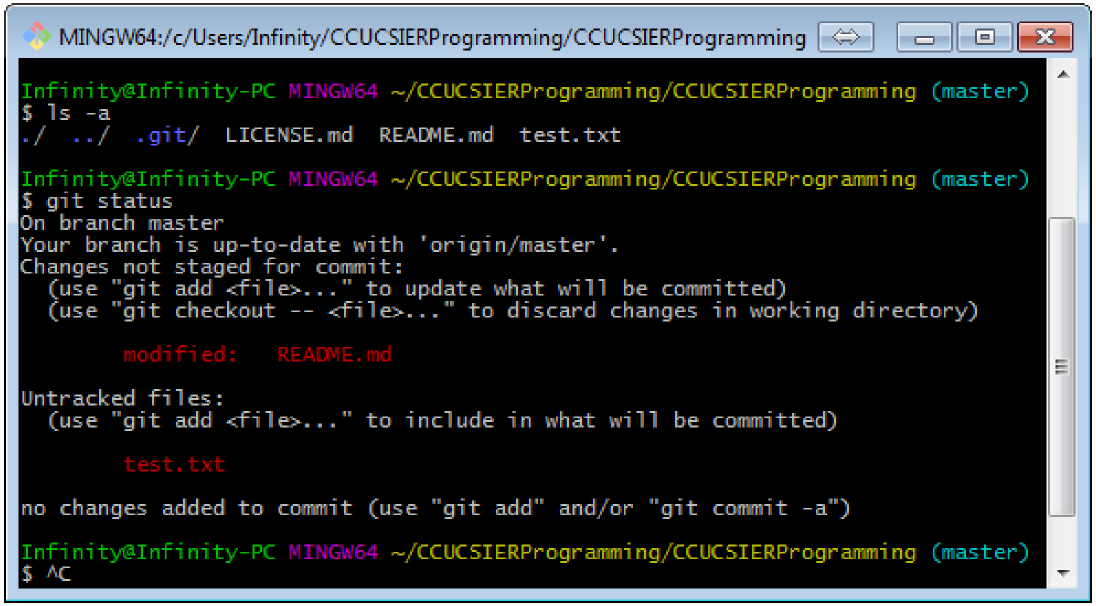


```
Infinity@Infinity-PC MINGW64 ~/CCUCSIERProgramming (master)
$ ls -a
./  ../  .git/  CCUCSIERProgramming/  README.md

Infinity@Infinity-PC MINGW64 ~/CCUCSIERProgramming (master)
$ git add .

Infinity@Infinity-PC MINGW64 ~/CCUCSIERProgramming (master)
$ git add -u

Infinity@Infinity-PC MINGW64 ~/CCUCSIERProgramming (master)
$ git add README.md

Infinity@Infinity-PC MINGW64 ~/CCUCSIERProgramming (master)
$ git status
```

Status

- As you saw in the git add section, in order to see what the status of your staging area is compared to the code in your working directory

```
git status
git status -s
```



MINGW64:/c/Users/Infinity/CCUCSIERProgramming/CCUCSIERProgramming

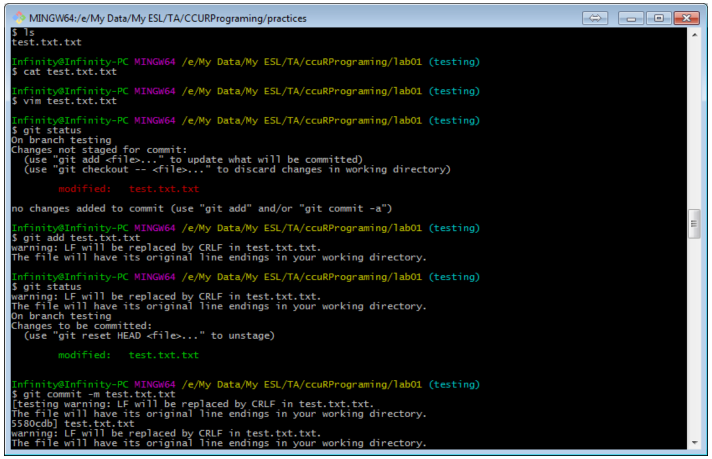
```
Infinity@Infinity-PC MINGW64 ~/CCUCSIERProgramming/CCUCSIERProgramming (master)
$ ls -a
./  ../  .git/  LICENSE.md  README.md  test.txt

Infinity@Infinity-PC MINGW64 ~/CCUCSIERProgramming/CCUCSIERProgramming (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        test.txt

no changes added to commit (use "git add" and/or "git commit -a")

Infinity@Infinity-PC MINGW64 ~/CCUCSIERProgramming/CCUCSIERProgramming (master)
$ ^C
```

Commit

- Commit the files that you've staged in your local repository

```
$ git commit -m 'First commit'
# Commits the tracked changes and prepares them to be pushed to a remote repository. To remove this commit and modify the file, use 'git reset --soft HEAD~1' and commit and add the file again.
```

- This only update your local repo , not the remote on github

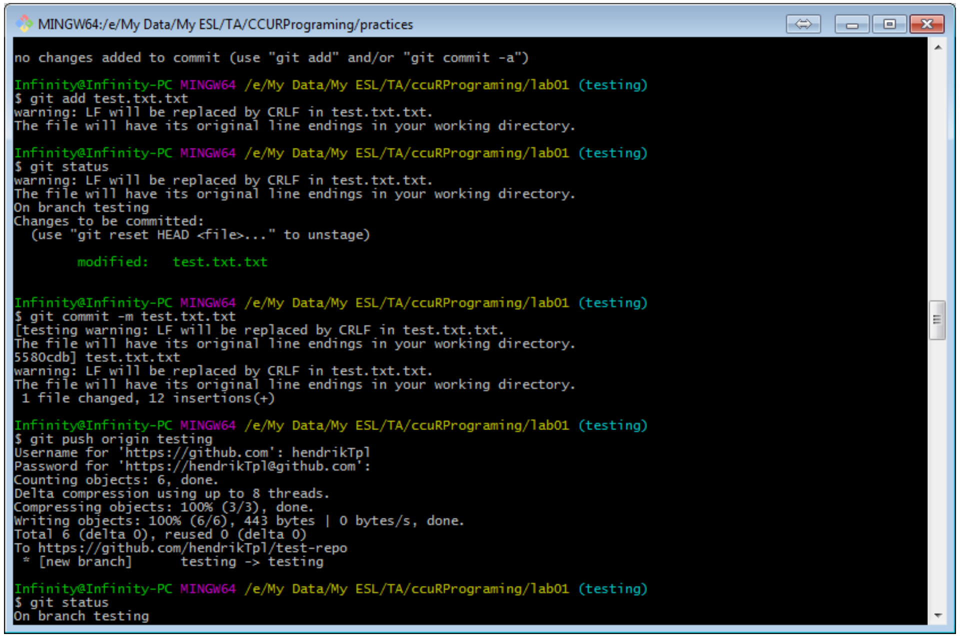


```
$ ls
test.txt.txt

Infinity@Infinity-PC MINGW64 /e/My Data/My ESL/TA/ccuRPrograming/lab01 (testing)
$ cat test.txt.txt

Infinity@Infinity-PC MINGW64 /e/My Data/My ESL/TA/ccuRPrograming/lab01 (testing)
$ vim test.txt.txt

Infinity@Infinity-PC MINGW64 /e/My Data/My ESL/TA/ccuRPrograming/lab01 (testing)
$ git status
On branch testing
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   test.txt.txt

no changes added to commit (use "git add" and/or "git commit -a")

Infinity@Infinity-PC MINGW64 /e/My Data/My ESL/TA/ccuRPrograming/lab01 (testing)
$ git add test.txt.txt
warning: LF will be replaced by CRLF in test.txt.txt.
The file will have its original line endings in your working directory.

Infinity@Infinity-PC MINGW64 /e/My Data/My ESL/TA/ccuRPrograming/lab01 (testing)
$ git status
warning: LF will be replaced by CRLF in test.txt.txt.
The file will have its original line endings in your working directory.
On branch testing
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   test.txt.txt


Infinity@Infinity-PC MINGW64 /e/My Data/My ESL/TA/ccuRPrograming/lab01 (testing)
$ git commit -m test.txt.txt
[testing warning: LF will be replaced by CRLF in test.txt.txt.
The file will have its original line endings in your working directory.
5580cdb] test.txt.txt
warning: LF will be replaced by CRLF in test.txt.txt.
The file will have its original line endings in your working directory.
```

Push

- Update local commits on the remote (Github)

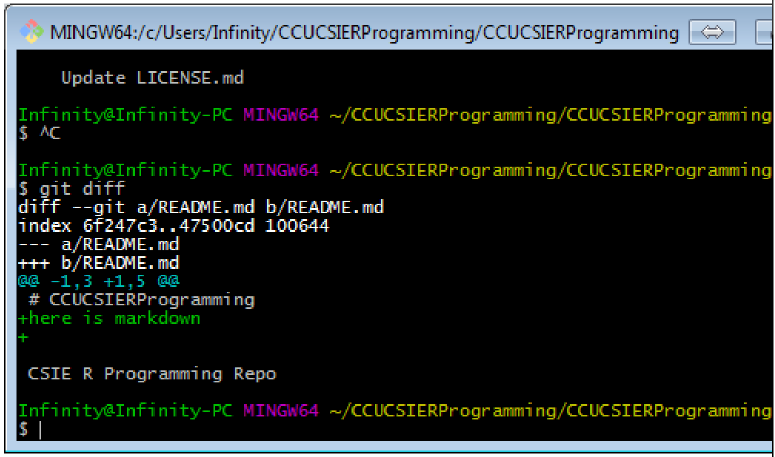
```
$ git push origin master
# Pushes the changes in your local repository up to the remote repository you specified as the origin
```



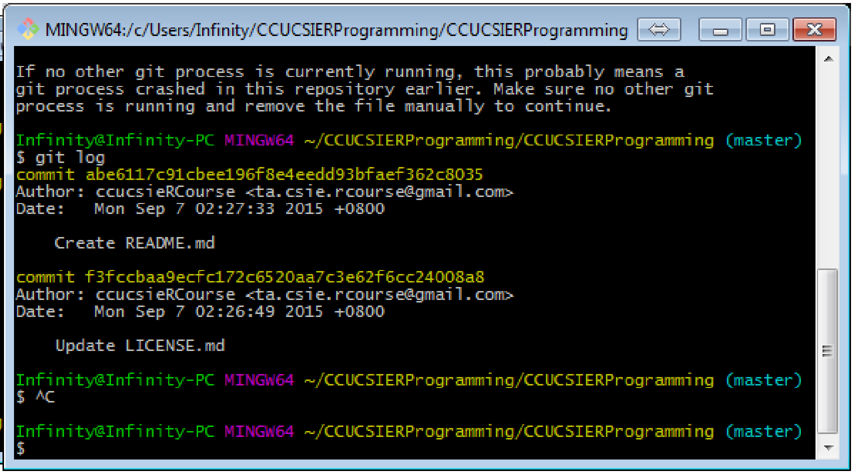
```
no changes added to commit (use "git add" and/or "git commit -a")

Infinity@Infinity-PC MINGW64 /e/My Data/My ESL/TA/ccuRPrograming/lab01 (testing)
$ git add test.txt.txt
warning: LF will be replaced by CRLF in test.txt.txt.
The file will have its original line endings in your working directory.

Infinity@Infinity-PC MINGW64 /e/My Data/My ESL/TA/ccuRPrograming/lab01 (testing)
$ git status
warning: LF will be replaced by CRLF in test.txt.txt.
The file will have its original line endings in your working directory.
On branch testing
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   test.txt.txt


Infinity@Infinity-PC MINGW64 /e/My Data/My ESL/TA/ccuRPrograming/lab01 (testing)
$ git commit -m test.txt.txt
[testing warning: LF will be replaced by CRLF in test.txt.txt.
The file will have its original line endings in your working directory.
5580cdb] test.txt.txt
warning: LF will be replaced by CRLF in test.txt.txt.
The file will have its original line endings in your working directory.
 1 file changed, 12 insertions(+)

Infinity@Infinity-PC MINGW64 /e/My Data/My ESL/TA/ccuRPrograming/lab01 (testing)
$ git push origin testing
Username for 'https://github.com': hendrikTpl
Password for 'https://hendrikTpl@github.com':
Counting objects: 6, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 443 bytes | 0 bytes/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To https://github.com/hendrikTpl/test-repo
 * [new branch]      testing -> testing

Infinity@Infinity-PC MINGW64 /e/My Data/My ESL/TA/ccuRPrograming/lab01 (testing)
$ git status
On branch testing
```

log & diff

- git log to find specific commits in your project history - by author, date,
- git diff to compare two different points in your history - generally to see how two branches differ or what has changed from one version of your software to another content or history.


MINGW64:/c/Users/Infinity/CCUCSIERProgramming/CCUCSIERProgramming

```
    Update LICENSE.md

Infinity@Infinity-PC MINGW64 ~/CCUCSIERProgramming/CCUCSIERProgramming
$ ^C

Infinity@Infinity-PC MINGW64 ~/CCUCSIERProgramming/CCUCSIERProgramming
$ git diff
diff --git a/README.md b/README.md
index 6f247c3..47500cd 100644
--- a/README.md
+++ b/README.md
@@ -1,3 +1,5 @@
 # CCUCSIERProgramming
+here is markdown
+

 CSIE R Programming Repo

Infinity@Infinity-PC MINGW64 ~/CCUCSIERProgramming/CCUCSIERProgramming
$ |
```


MINGW64:/c/Users/Infinity/CCUCSIERProgramming/CCUCSIERProgramming

```
If no other git process is currently running, this probably means a
git process crashed in this repository earlier. Make sure no other git
process is running and remove the file manually to continue.

Infinity@Infinity-PC MINGW64 ~/CCUCSIERProgramming/CCUCSIERProgramming (master)
$ git log
commit abe6117c91cbee196f8e4eedd93bfaef362c8035
Author: ccucsieRCourse <ta.csie.rcourse@gmail.com>
Date:   Mon Sep 7 02:27:33 2015 +0800

    Create README.md

commit f3fccbaa9ecfc172c6520aa7c3e62f6cc24008a8
Author: ccucsieRCourse <ta.csie.rcourse@gmail.com>
Date:   Mon Sep 7 02:26:49 2015 +0800

    Update LICENSE.md

Infinity@Infinity-PC MINGW64 ~/CCUCSIERProgramming/CCUCSIERProgramming (master)
$ ^C

Infinity@Infinity-PC MINGW64 ~/CCUCSIERProgramming/CCUCSIERProgramming (master)
$
```

Branches

- Sometimes you are working on a project with a version being used by many people. You may not want to edit that version
- So you can create a branch with the command

```
git checkout –b branchname
```

- To see what branch you are on type

```
git branch
```

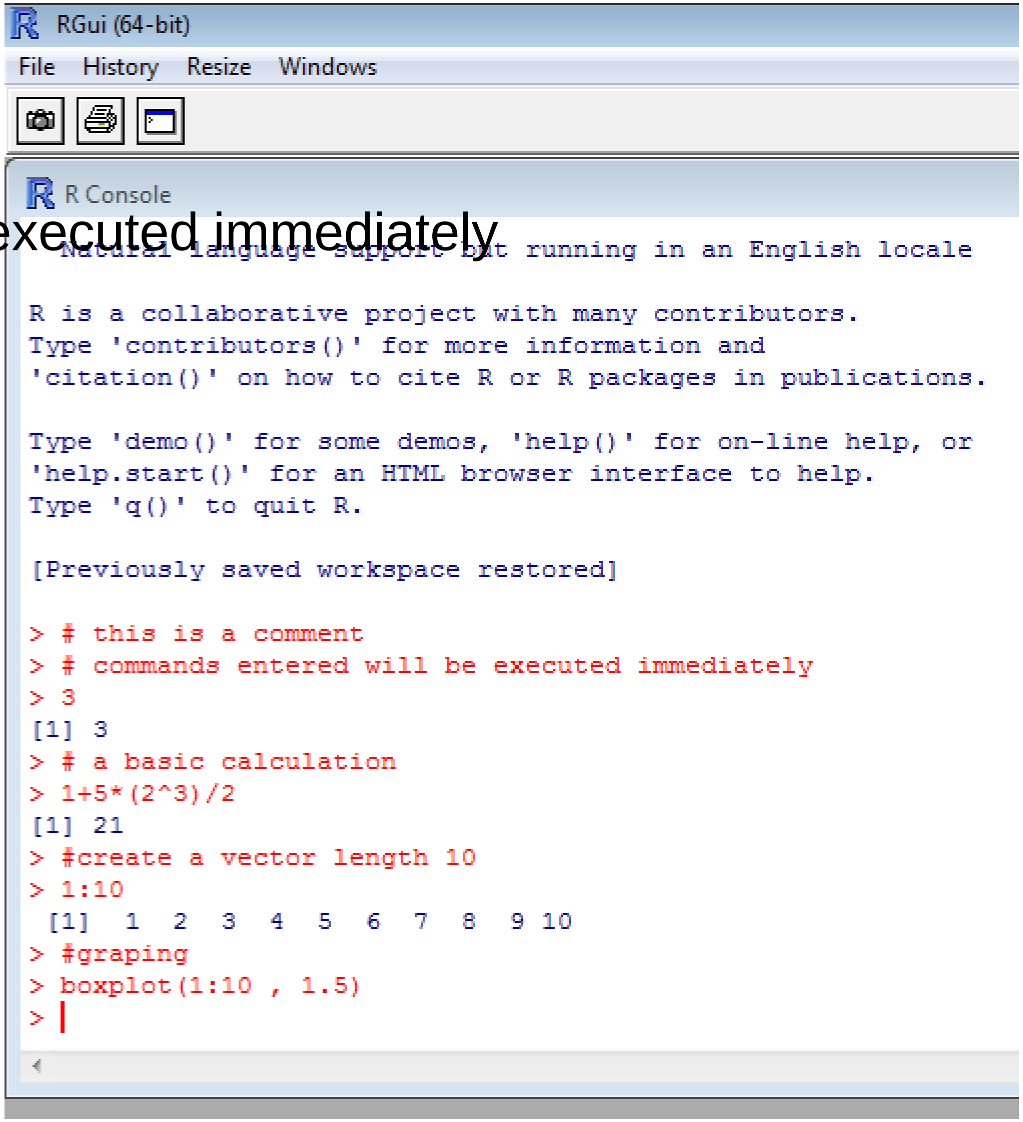
- To switch back to master branch type

```
git checkout master
```

More about git, please kindly go to https://git-scm.com/doc or http://gitref.org/

Getting Started

- Open R console
  - Commands entered will be executed immediately
- Demo some simple tasks
- Calculation
  - E.g. 1+5*(2^3)/2
- Statistical analysis
  - E.g. linear regression
- Graphing
  - E.g. boxplot



```
Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> # this is a comment
> # commands entered will be executed immediately
> 3
[1] 3
> # a basic calculation
> 1+5*(2^3)/2
[1] 21
> #create a vector length 10
> 1:10
 [1]  1  2  3  4  5  6  7  8  9 10
> #graping
> boxplot(1:10 , 1.5)
> 
```

Entering Input & Evaluation

- <- symbol is the assignment operator
- # character indicates a comment

VarNames <- c(“variables”)

```
> x <- 1
> print(x)
   [1] 1
> x
  [1] 1
> msg <- "hello“
> x <- ## Incomplete expression
```

When a complete expression is entered at the prompt, it is evaluated and the result of the evaluated expression is returned

```
> x <- 5 ## nothing printed
> x ## auto-printing occurs
    [1] 5
> print(x) ## explicit printing
    [1] 5
```

The [1] indicates that x is a vector and 5 is the first element

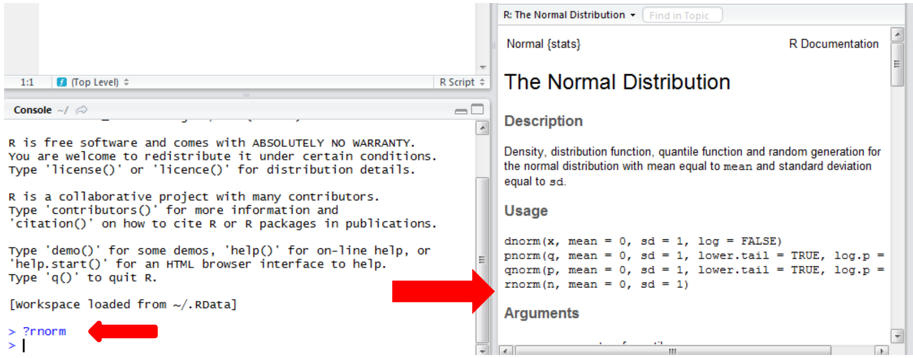
First R

```
• X <-1:10          #create x with 10 numbers
• Print(x)          #print numbers in x
• Y <-2*x-1         #create y based on x (also 10 num)
• Print(y)
• Mean(y)           #calculate average of numbers in y
• Model <- lm(y~x)     #fit a linear regression model
                       # y=B0+(B1*X)
Print(Model)  # print the linear model
                       #B0 = -1 , B1=2
Boxplot(x,y)  #plot distribution of x and y
```

Getting help

- Access help file
  - ?topic
    - E.g. ?rnorm

      for Normal Distribution
- Search help files
  - help.search(“topic”)
- Get arguments
  - args(“topic”)



```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

> ?rnorm
> |
```

Normal {stats} R Documentation

The Normal Distribution

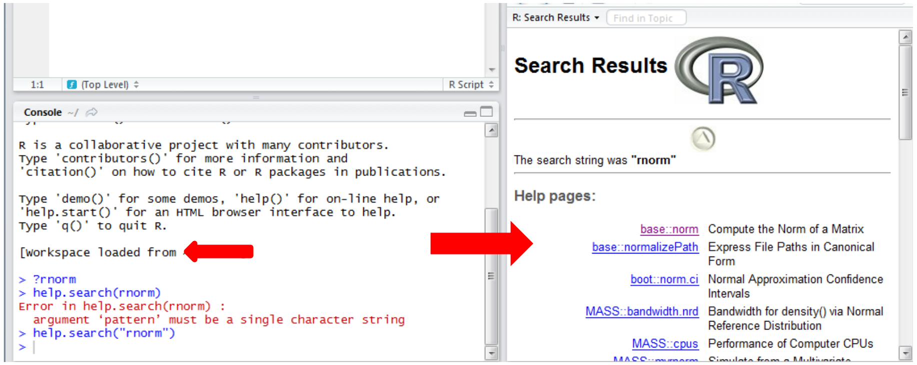
Description

Density, distribution function, quantile function and random generation for the normal distribution with mean equal to `mean` and standard deviation equal to `sd`.

Usage

```
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p =
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p =
rnorm(n, mean = 0, sd = 1)
```

Arguments



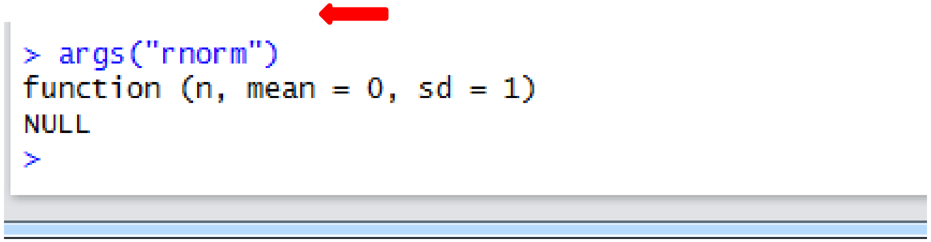
```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from

> ?rnorm
> help.search(rnorm)
Error in help.search(rnorm) :
  argument ‘pattern’ must be a single character string
> help.search("rnorm")
> |
```

Search Results

The search string was **"rnorm"**

Help pages:

base::norm	Compute the Norm of a Matrix
base::normalizePath	Express File Paths in Canonical Form
boot::norm.ci	Normal Approximation Confidence Intervals
MASS::bandwidth.nrd	Bandwidth for density() via Normal Reference Distribution
MASS::cpus	Performance of Computer CPUs
MASS::mvrnorm	Simulate from a Multivariate



```
> args("rnorm")
function (n, mean = 0, sd = 1)
NULL
>
```

Reference: https://cran.r-project.org/doc/contrib/Short-refcard.pdf

DEMO & ASSIGNMENT

Part1(setup your own environment)

You are asked to setup your R & Git

- R
  - Install R
  - Install Rstudio
  - Open R &Rstudio and take a screenshot

- Github
  - Set up a Github account (you may use your own name or a pseudonym).
  - Create a repo called ***Lab1_studentID (***scratch***)***
  - Clone your github into local repo
  - Create a text file called HelloWorld.md
  - Add the line

    ##This is a secondary heading
    ### This is a Tertiary heading
    * first item list
    * second item list
  - Commit every changes
  - Push the document to the ***Lab1_studentID*** repo you created on Github
  - Fork the data sharing repository here https://github.com/ccucsieRCourse/sharing

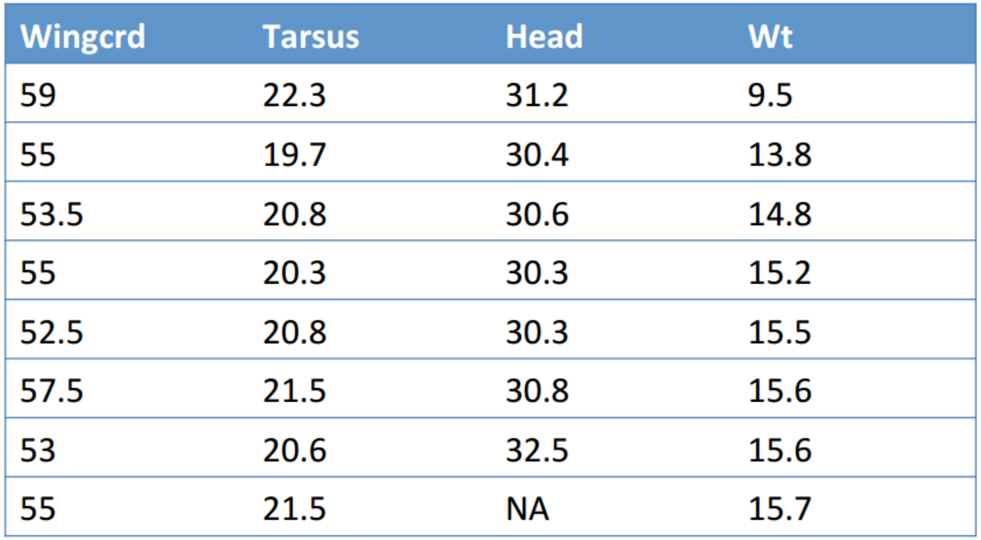
Part2 (My first R)

write your first R

- Do some calculations (e.g. 1*1+2*2+3*3+…+100*100)
- Learn some datasets (e.g. ?inspect)
- Learn some function (e.g. ?seq, ?mean)
- Learn some package (e.g. ?stats, ?swirl)
- Try example in the help file

Part 3 (Simple Data Analysis)

- Table morphometric measurements of eight birds.



Wingcrd	Tarsus	Head	Wt
59	22.3	31.2	9.5
55	19.7	30.4	13.8
53.5	20.8	30.6	14.8
55	20.3	30.3	15.2
52.5	20.8	30.3	15.5
57.5	21.5	30.8	15.6
53	20.6	32.5	15.6
55	21.5	NA	15.7

- Use the c function, create a variable wingcrd to hold the wing lengths for the 8 birds
  - Similary, create Tarsus , head and wt
  - Compute basic statistics of the data (mean, median , min , max)

Swirl exercise(home)

- 1) Make sure you have a recent version version of swirl:

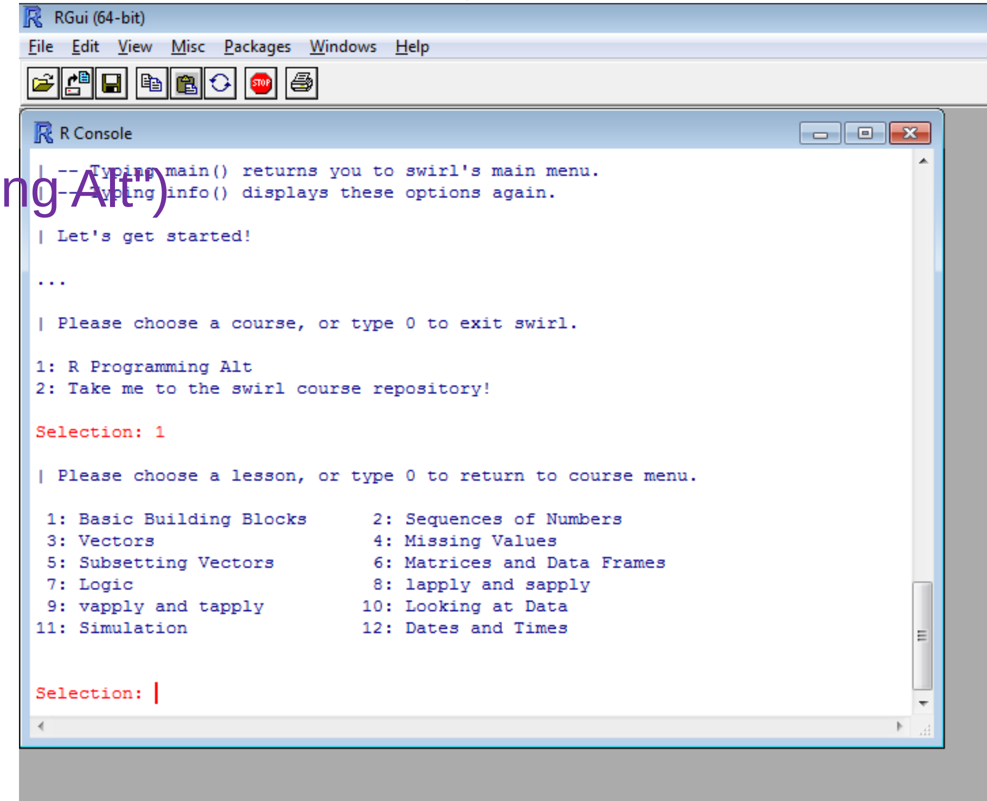
  install.packages("swirl")

- 2) Enter the following from the R console, substituting the name of the course that you wish to install:

  library(swirl)

  install_from_swirl("R Programming Alt")

  swirl()



```
| -- Typing main() returns you to swirl's main menu.
| -- Typing info() displays these options again.

| Let's get started!

...

| Please choose a course, or type 0 to exit swirl.

1: R Programming Alt
2: Take me to the swirl course repository!

Selection: 1

| Please choose a lesson, or type 0 to return to course menu.

 1: Basic Building Blocks      2: Sequences of Numbers
 3: Vectors                    4: Missing Values
 5: Subsetting Vectors         6: Matrices and Data Frames
 7: Logic                      8: lapply and sapply
 9: vapply and tapply         10: Looking at Data
11: Simulation                12: Dates and Times


Selection: 
```

Hand in

- To submit this lab
  - Use your own repo that you created on your Github
  - R scripts are called regards to part number (e.g. part2.R)
  - Zip Lab1_StudentID that contain R&Rstudio screenshot, part2.R , part3.R
  - Commit each file you have modified
  - Submit the link to the forked repository on your Github account

Hand in

- In order to record to assignment you still need to upload to
- The E-course System
  - https://ecourse.ccu.edu.tw/

- Upload StudentID.zip into “Lab1”

GRADING POLICIES

Grading Policies

- There will be 8 Labs =24%
  - Lab1 =2%, Lab2=2%, Lab3=3%, Lab4=3%, Lab5=3%, Lab6=3%, LAb7=4%, Lab8=4%
- Finish All Lab1 in order to get full credit (part1-part3)
  - In case you haven’t finished by the end of class, you can still send it by reducing 20% from full score for each day..
  - e.g. Total score lab1 100, it will 80, 60, 40, ….

References

- Roger, D. Peng.(2015).*R Programming for Data Science*. Victoria, British Columbia : Leanpub.
- Joseph, A. (2012). *R IN A NUTSHELL: A Desktop Quick Reference, 2nd Edition*. Sebastopol, CA : O'Reilly.
- Alain, F. Z. ,Elena N. L., Erik H.W.G. (2009). *A Beginner's Guide to R.* Newburgh,United Kingdom : Springer
- https://www.r-project.org/
- http://gitref.org/
- https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository
- https://www.coursera.org/jhu