

326.212 Statistical Computing and Labs. 실습 제 1 강 : R과 Git 소개

최영원

서울대학교 통계학과

Autumn 2015

이번 강좌에서 다룰 내용

- R 및 R studio
- DVCS - git
- Lecture Page

Section 1

R 및 R studio

R 소개

- R은 통계 계산과 그래픽을 위한 프로그래밍 언어이자 소프트웨어 환경
- 뉴질랜드 오클랜드 대학의 로스 이하카와 로버트 젠틀맨에 의해 시작되어 현재는 R 코어 팀이 개발
- GPL 하에 배포되는 S 프로그래밍 언어의 구현
- 통계 소프트웨어 개발과 자료 분석에 널리 쓰이고 있음
- 패키지 개발이 용이하여 통계학자들 사이에서 통계 소프트웨어 개발에 사용됨
- UNIX, Mac, Windows 모두에서 동작함

R 첫삽 뜨기

- R은 커맨드라인(command line) 기반 환경임
 - ▶ 거의 대부분의 일처리가 잘 정의된 텍스트 명령어로 이루어짐.
 - ▶ 명령어들 다수가 정해진 문법을 따라야 함 (구두점 및 명령어들간에 따라야 할 규칙).
 - ▶ 결과가 잘못되었을 때 나오는 피드백이나 에러메시지가 그다지 친절하지는 않음.

R 설치하기: 윈도우 PC

- ① Comprehensive R Network 방문: <http://cran.r-project.org/>. “Download and Install R” 섹션에서 “Download R for Windows” 링크 클릭.

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 (2013-09-25, Frisbee Sailing) [R-3.0.2.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](#)

최신 버전 확인: 2015년 9월 현재 3.2.2

R 설치하기: 윈도우 PC (cont'd)

② “Subdirectories” 섹션에서 “base” 링크 클릭.

R for Windows

Subdirectories:

- [**base**](#) (Red circle drawn around this link)
- [contrib](#)
- [Rtools](#)

Binaries for base distribution (managed by Duncan Murdoch). This is what you want to [install R for the first time](#).

Binaries of contributed packages (managed by Uwe Ligges). There is also information on [third party software](#) available for CRAN Windows services and corresponding environment and make variables.

Tools to build R and R packages (managed by Duncan Murdoch). This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Duncan Murdoch or Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the [R FAQ](#) and [R for Windows FAQ](#).

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

R 설치하기: 윈도우 PC (cont'd)

- ③ 다음 페이지에서 “Download R X.X.X for Windows” 같은 링크가 보이면 그것을 클릭 (X.X.X는 R 버전을 의미함. 예를 들어 R 3.0.2). 파일을 실행 또는 저장 후 실행.



If you want to double-check that the package you have downloaded exactly matches the package distributed by R, you can compare the [md5sum](#) of the .exe to the [true fingerprint](#). You will need a version of md5sum for windows: both [graphical](#) and [command line versions](#) are available.

Frequently asked questions

- [How do I install R when using Windows Vista?](#)
- [How do I update packages in my previous version of R?](#)
- [Should I run 32-bit or 64-bit R?](#)

Please see the [R FAQ](#) for general information about R and the [R Windows FAQ](#) for Windows-specific information.

Other builds

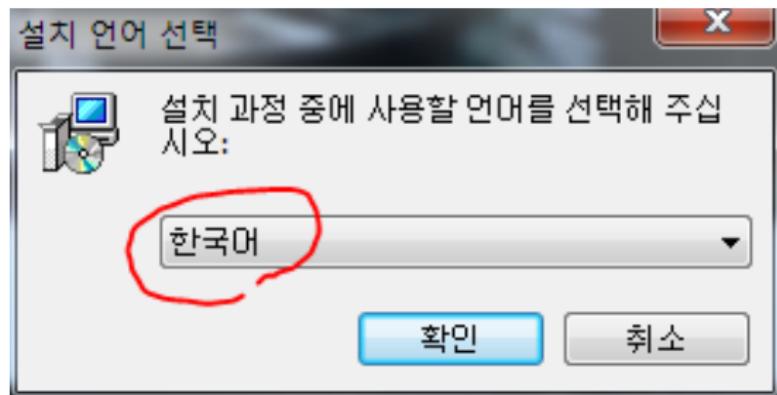
- Patches to this release are incorporated in the [r-patched snapshot build](#).
- A build of the development version (which will eventually become the next major release of R) is available in the [r-devel snapshot build](#).
- [Previous releases](#)

Note to webmasters: A stable link which will redirect to the current Windows binary release is
<http://<CRAN MIRROR>/bin/windows/base/release.htm>.

Last change: 2013-09-25, by Duncan Murdoch

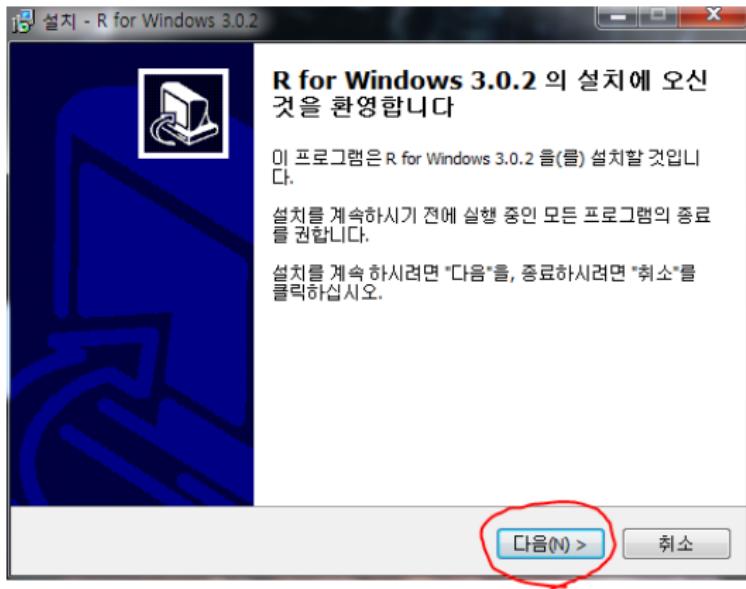
R 설치하기: 윈도우 PC (cont'd)

- ④ 사용할 언어 선택에서 “한국어” 선택.



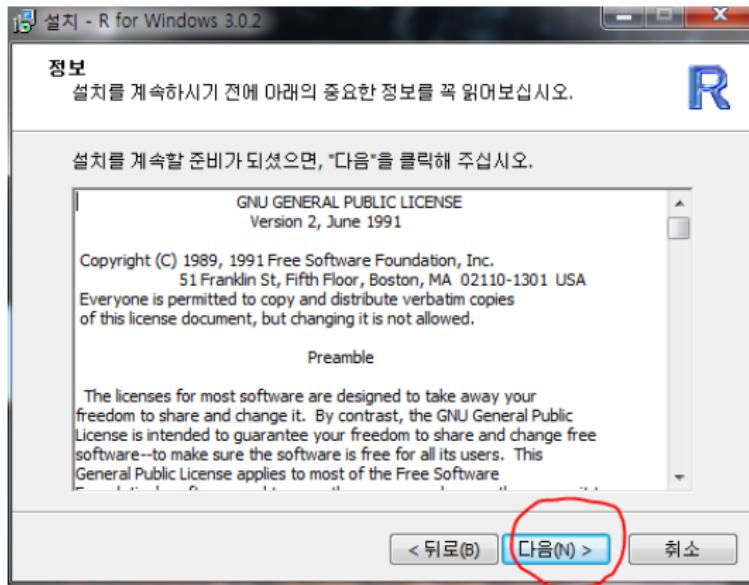
R 설치하기: 윈도우 PC (cont'd)

- ⑤ R 설치 마법사가 나타나면 마법사 창 아래의 “다음” 버튼을 클릭.



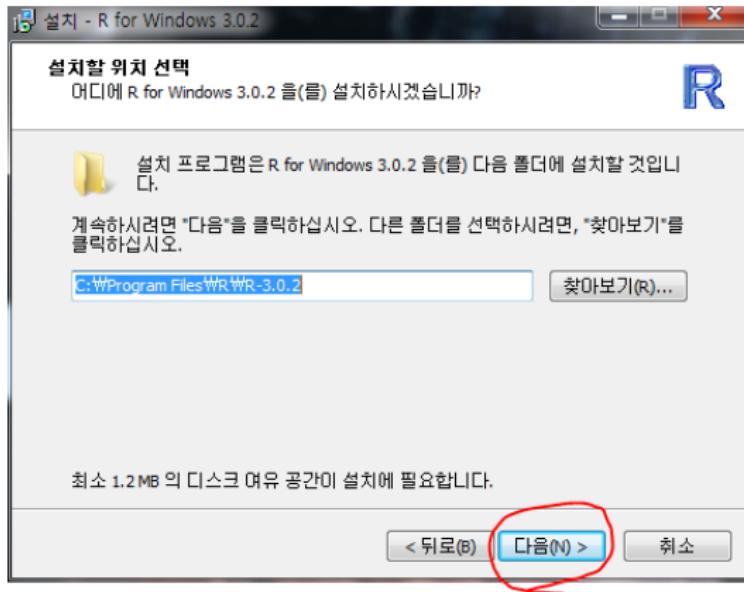
R 설치하기: 윈도우 PC (cont'd)

- ⑥ 다음 페이지 상단에 “정보”가 나타나면 “다음” 버튼을 클릭.



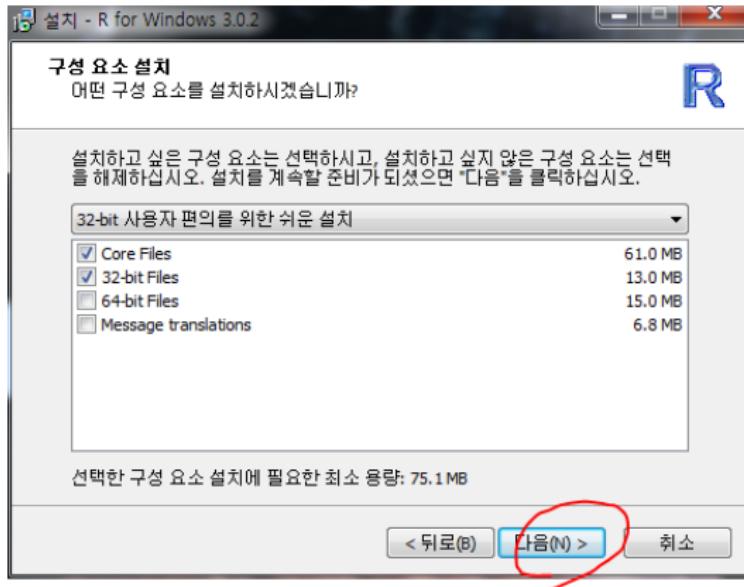
R 설치하기: 윈도우 PC (cont'd)

- ⑦ 다음 페이지 상단에 “설치할 위치 선택”이 나타남. 전산실습실에서는 매일 C드라이브가 재설정되므로 D:\comp_stat\R-3.1.2로 변경한다. 하단의 “다음” 버튼을 클릭.



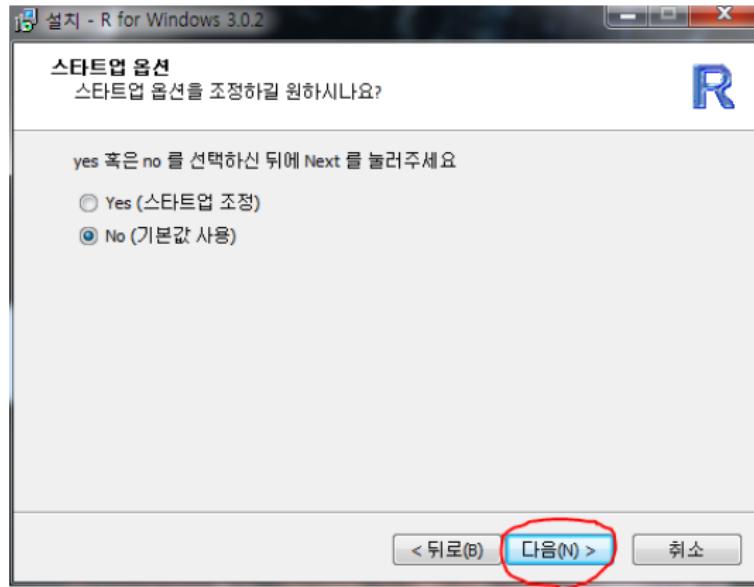
R 설치하기: 윈도우 PC (cont'd)

- ⑧ 다음 페이지 상단에 “구성 요소 설치”가 나타나면 다시 하단의 “다음” 버튼을 클릭.



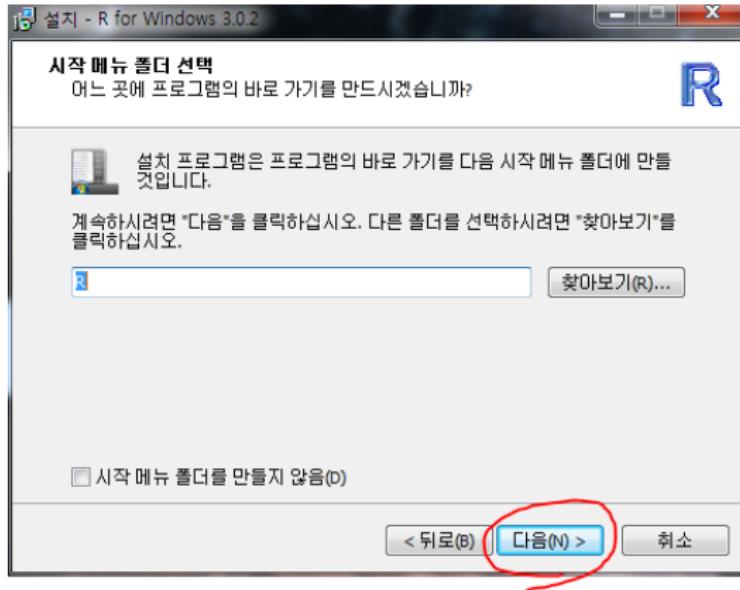
R 설치하기: 윈도우 PC (cont'd)

- ⑨ 다음 페이지 상단에 “스타트업 옵션”이 나타나면 다시 하단의 “다음” 버튼을 클릭.



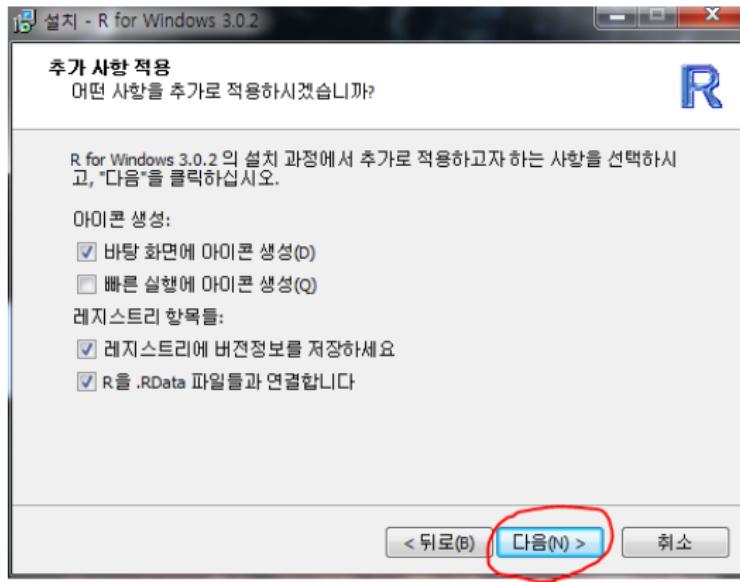
R 설치하기: 윈도우 PC (cont'd)

- ⑩ 다음 페이지 상단에 “시작 메뉴 폴더 선택”이 나타나면 다시 하단의 “다음” 버튼을 클릭.



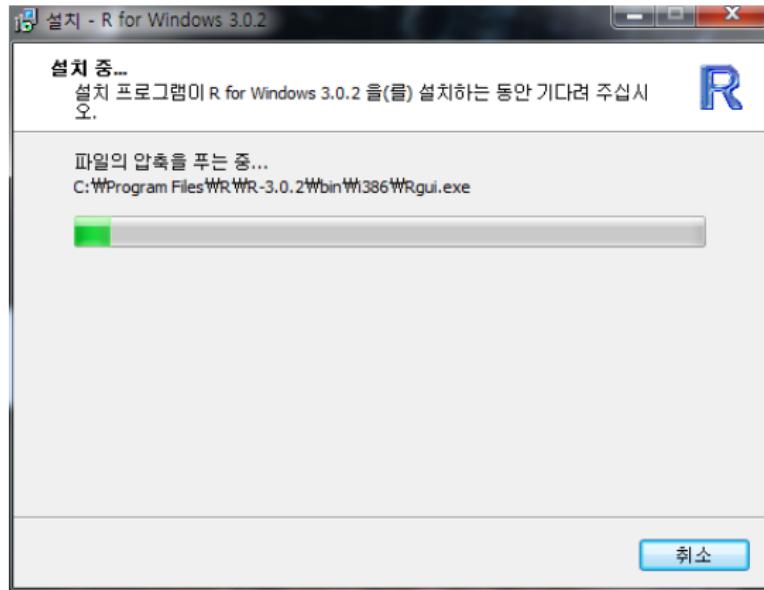
R 설치하기: 윈도우 PC (cont'd)

- ⑪ 다음 페이지 상단에 “추가 사항 적용”이 나타나면 다시 하단의 “다음” 버튼을 클릭.



R 설치하기: 윈도우 PC (cont'd)

⑫ R 설치에는 수 분 정도가 소요.



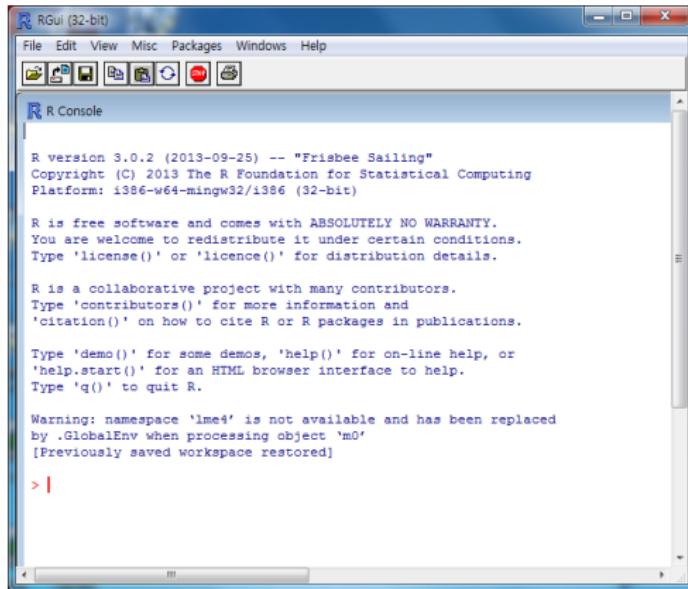
R 설치하기: 윈도우 PC (cont'd)

- ⑬ 설치가 완료되면 “R for Windows X.X.X 설치 완료” 창이 나타남.
“완료” 버튼 클릭.



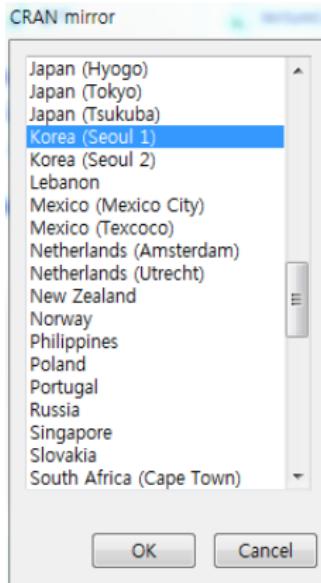
R 설치하기: 윈도우 PC (cont'd)

- 14 데스크탑 왼쪽 하단의 “시작” 버튼을 클릭하고 “모든 프로그램” 선택. 메뉴에서 “R” 또는 “R X.X.X”을 선택하여 “R” 시작. R 콘솔이 나타남



R 설치하기: 윈도우 PC (cont'd)

- 15 R 콘솔에 `update.packages()`를 입력하고 엔터를 누름. “CRAN Mirror” 창이 뜨면 “Korea (Seoul 1)” 또는 “Korea (Seoul 2)”를 선택. 업데이트할 패키지가 있으면 각 패키지마다 `Update (y/N/c)?`로 업데이트 여부를 물어옴. 모두 “y”를 입력하고 엔터.



R 설치하기: 튜토리얼

- 동영상 튜토리얼: [How to install R for Windows](#)
- 동영상 튜토리얼: [How to install R for Mac](#)
- 보다 자세한 튜토리얼:
<http://a-little-book-of-r-for-biomedical-statistics.readthedocs.org/en/latest/src/installr.html>

R 맛보기

간단한 계산

`2*3`

`## [1] 6`

`10-3`

`## [1] 7`

R 맛보기 (cont'd)

변수 및 대입

```
x <- 2*3
```

```
x
```

```
## [1] 6
```

```
x = 2*3
```

```
x
```

```
## [1] 6
```

R 맛보기 (cont'd)

벡터

```
myvector <- c(8, 6, 9, 10, 5)  
myvector
```

```
## [1] 8 6 9 10 5
```

```
myvector[4]
```

```
## [1] 10
```

- c for combine
- 같은 형의 원소만 가능

R 맛보기 (cont'd)

리스트

```
mylist <- list(name="Fred", wife="Mary", myvector)
mylist

## $name
## [1] "Fred"
##
## $wife
## [1] "Mary"
##
## [[3]]
## [1] 8 6 9 10 5

mylist[[2]]

## [1] "Mary"

mylist$wife

## [1] "Mary"

mylist[[3]]

## [1] 8 6 9 10 5
```

R 맛보기 (cont'd)

테이블

```
mynames <- c("Mary", "John", "Ann", "Sinead", "Joe", "Mary", "Jim", "John",
table(mynames)
```

```
## mynames
##      Ann      Jim      Joe      John      Mary      Simon      Sinead
##      1        1        1        2        2        1        1
```

```
mytable <- table(mynames)
mytable[[4]]
```

```
## [1] 2
```

```
mytable[["John"]]
```

```
## [1] 2
```

R 맛보기 (cont'd)

수학함수

```
log10(100)
```

```
## [1] 2
```

```
exp(1)
```

```
## [1] 2.718282
```

R 맛보기 (cont'd)

함수 (프로그래밍 언어에서의)

```
mean(myvector)      # 내장함수
```

```
## [1] 7.6
```

```
myfunction <- function(x) { return(20 + (x*x)) } # 사용자정의함수  
myfunction(10)
```

```
## [1] 120
```

```
myfunction(25)
```

```
## [1] 645
```

```
fish(myvector)      # 미정의함수
```

```
## Error in eval(expr, envir, enclos): could not find function
```

R 에러 메시지

- `fish()` 함수가 정의되지 않았으므로 에러메시지를 내보냄.
- 사용자가 R이 못 알아듣는 명령을 내려도 컴퓨터가 고장나거나 하는 일 없이 조용히 에러메시지만 내보냄.
- 에러메시지가 항상 큰 도움이 되는 것은 아니지만, 조심스럽게 살펴보면 많은 경우 문제를 파악하고 해결할 수 있음.

시행착오에서 배우기

- ① R은 기본적으로 프로그래밍 언어이므로, 교재의 명령들을 직접 입력하여 수행해 보는 것이 배우는 가장 빠른 길임.
- ② 교재의 명령이 제대로 실행되지 않을 경우 왜 그럴까를 항상 생각해야 함.
 - ▶ 스펠링 또는 구두점
 - ▶ 대소문자 구분: R은 대소문자를 구분함.
 - ▶ 명령어를 정확히 입력했는데도 실행이 안되는 경우 온라인 또는 주변 사람들에게 도움을 요청.
 - ▶ 무엇이 잘못되었는지를 깨닫게 되면 R에 대해 하나 더 배운 것임.
- ③ 배운 것을 조합하여 새로운 시도를 해 볼 필요가 있음.
 - ▶ 예를 들어,

```
avg <- mean(myvector)
```

```
avg
```

```
## [1] 7.6
```

R Studio

- R을 위한 통합환경, 또는 IDE(Integrated Development Environment)
 - ▶ 윈도우 응용 프로그램 개발에서의 Visual Studio와 비슷한 역할
- R 사용에 필요한 부가 정보들을 한눈에 보여줌.

IDE가 왜 필요한가?

- 지금까지는 각종 R 명령들을 “R 콘솔”에서 입력.

```
Last login: Sun Oct 13 21:27:00 on ttys003
Jhwon@Joong-Ho-Mons-Air:~$ R

R version 2.14.2 (2012-02-29)
Copyright (C) 2012 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-apple-darwin9.8.0/x86_64 (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.

> x <- c(1,2,3,4)
> x
[1] 1 2 3 4
> █
```

IDE가 왜 필요한가? (cont'd)

- “콘솔”이란 오래 전 컴퓨터가 집 한 채만하던 시절 집 한켠에 모시고 운영자가 그 거대한 기계를 조정하기 위한 “마스터 제어판”을 이르던 말.



source: <http://asterhost.info/retro-delight-gallery-of-early-computers-1940s%2D%80%93%2D1960s/>

IDE가 왜 필요한가? (cont'd)

- 한번 입력한 명령을 다시 입력하는 수고를 덜어줌.
- 매번 손으로 입력하는 것은 실수를 유발하기 쉬움. 즉, 재사용성을 원함.
- 데이터 중심으로 작업이 진행되므로, 데이터의 내용과 구조를 모두 들여다 볼 필요가 자주 있으나, R 콘솔로 하기에는 매우 짜증나는 작업임.
- R용 IDE로서 R Studio는 명령어와 데이터 모두 재사용이 용이하도록 도와줌.

R Studio 설치

① <http://www.rstudio.com/> 에 접속해 “Download RStudio” 클릭

Welcome to RStudio - Open source and enterprise-ready professional software for R

[Download RStudio](#) [Discover Shiny](#) [shinyapps.io Login](#)

Powerful IDE for R
RStudio IDE is a powerful and productive user interface for R. It's free and open source, and works great on Windows, Mac, and Linux.

[Learn More >](#)

R Packages
Our developers and expert trainers are the authors of several popular R packages, including ggplot2, plyr, lubridate, and others.

[Learn More >](#)

Bring R to the web
Shiny is an elegant and powerful web framework for building interactive reports and visualizations using R — with or without web development skills.

[Learn More >](#)

R Studio 설치 (cont'd)

② “Download RStudio Desktop” 선택

The screenshot shows the RStudio website homepage. At the top, there's a navigation bar with links for Products, Resources, Pricing, About Us, and Blog, along with a search icon. Below the navigation, a breadcrumb trail shows Home / Download / RStudio. The main content area has a heading "Take control of your R code". It describes RStudio as an integrated development environment (IDE) for R, mentioning its features like a console, syntax-highlighting editor, direct code execution, plotting tools, history, debugging, and workspace management. A link is provided to see more RStudio features. Below this, it says RStudio is available in open source and commercial editions and runs on desktop (Windows, Mac, and Linux) or in a browser connected to RStudio Server or RStudio Server Pro (Debian/Ubuntu, RedHat/CentOS, and SUSE Linux). Two large buttons are shown: "RStudio Desktop" (with a yellow bounding box around it) and "RStudio Server". A video player window on the right shows a video titled "R Studio™" with a play button and a progress bar. Below the video is a button labeled "CLICK HERE TO SEE ADDITIONAL FEATURES". At the bottom, there are two buttons: "Open Source Edition" and "Commercial License".

R Studio 설치 (cont'd)

② “Download RStudio Desktop” 선택

The screenshot shows the RStudio website's download page for RStudio Desktop. At the top, there is a navigation bar with links for Products, Resources, Pricing, About Us, and Blog. Below the navigation bar, the page title "RStudio Desktop" is displayed. Underneath the title, there are two main tabs: "Open Source Edition" (which is highlighted with a blue background) and "Commercial License".

Overview

- Access RStudio locally
- Syntax highlighting, code completion, and smart indentation
- Execute R code directly from the source editor
- Quickly jump to function definitions
- Easily manage multiple working directories using projects
- Integrated R help and documentation
- Interactive debugger to diagnose and fix errors quickly
- Extensive package development tools

All of the features of open source; plus:

- A commercial license for organizations not able to use AGPL software
- Access to priority support

Support Community forums only **Pricing** Free

License AGPL v3 **Support** Priority Email Support
8 hour response during business hours (ET)

Pricing \$995/year

[DOWNLOAD RSTUDIO DESKTOP](#) [BUY NOW](#)

R Studio 설치 (cont'd)

③ 사용자 컴퓨터 사양에 맞는 설치 파일을 선택

 R Studio

Products Resources Pricing About Us Blog 

[Home](#) / [Overview](#) / [RStudio](#) / [Download RStudio](#)

[Download RStudio](#)

RStudio is a set of integrated tools designed to help you be more productive with R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management.

If you run R on a Linux server and want to enable users to remotely access RStudio using a web browser please download RStudio Server.

Do you need support or a commercial license? Check out our commercial offerings

RStudio Desktop 0.99.473 — Release Notes

RStudio requires R 2.11.1 (or higher). If you don't already have R, you can download it [here](#).



 Share your R code on the web with Shiny
Click here to learn more

Installers for Supported Platforms

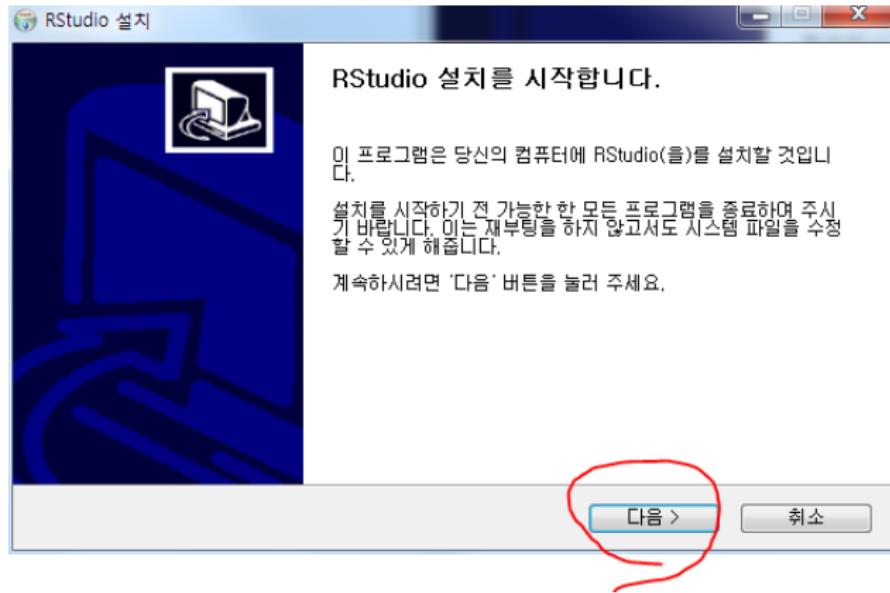
Installer	Size	Date	MD5
RStudio 0.99.473 - Windows (32-bit)	79.8 MB	2015-08-12	c6c79173f7ba6406d4c1720b740a95
RStudio 0.99.473 - Mac OS X 10.6+ (64-bit)	56.2 MB	2015-08-12	43966915d76923ac14524c4d93943
RStudio 0.99.473 - Ubuntu 12.04+ (Debian 6+ (32-bit))	77.4 MB	2015-08-12	469929d3d501ae11aa64c0593c0d0e
RStudio 0.99.473 - Ubuntu 12.04+ (Debian 6+ (64-bit))	83.9 MB	2015-08-12	4952a2e501c9163fec43c9e993b5a73
RStudio 0.99.473 - Fedora 19+ / Red Hat 7+ / openSUSE 13.1+ (32-bit)	76.8 MB	2015-08-12	159167574b19102a202cd486223f15
RStudio 0.99.473 - Fedora 19+ / Red Hat 7+ / openSUSE 13.1+ (64-bit)	77.7 MB	2015-08-12	8e9532774d1f0463267b3972acd0

Zip/Tarballs

(2015년 11월 현재 최신 버전은 RStudio v0.99.473)

R Studio 설치 (cont'd)

④ 실행 또는 저장 후 실행



R Studio 설치 (cont'd)

- 동영상 튜토리얼
- 백견이불여일행!

The screenshot shows the RStudio interface. On the left, the 'Console' tab displays R code and its output. The code includes comments about histograms and density plots, and it generates a histogram for the 'petalwidth' variable. The histogram is shown in the main pane, featuring a red outline and a teal fill. The x-axis is labeled 'petalwidth' and ranges from -1.3 to 0.7. The y-axis is labeled 'COUNT' and ranges from 0 to 50. A legend indicates two categories: 0 and 1. The histogram bars for category 0 are teal, and those for category 1 are red.

```
## Why we need on IDE? (cont'd)
55
56 1. We don't want to retype the commands we typed yesterday
57
58 2. It is too easy to make a mistake if we are doing every little task by hand, i.e. we want
reusability.
59
60 3. Because we are working with data, we also need a way of closely inspecting the data, both its
contents and its structure. It gets pretty tedious to do this on the R console.
61
62 As an IDE for R, R Studio allows us to control and monitor both our code and our text in a way
that supports the creation of reusable elements.
63
64 ## Picture is worth a thousand words
65
66 * http://www.rstudio.com/
67 * http://www.youtube.com/watch?v=7aKnpZ35nEo
68
69
70 1 | (Top Level) | R Markdown 1
71
72 > petalwidthhist <- hist(iris$petalwidth, 3, position="identity") + geom_vline(xintercept=1)
73 > print(p)
74
75 > start_hist <- binwidth(p)
76 > start_hist
77 [1] 0.07955703
78 > start_hist * 30
79 [1] 0.87954947
80 > sd <- sd(iris$petalwidth)
```

R 프로그래밍

- C언어에서는 기본적인 함수, 예를 들어 `printf`를 사용하기 위해 `stdio.h`를 인클루드해야 했다. 앞서 사용한 `mean`, `log10`, `exp`와 같은 함수는 어디에서 왔을까? 또한 이런 함수가 있는지는 어떻게 알까?
 - ▶ 온라인 커뮤니티
 - ▶ 일부는 오픈소스 커뮤니티에서 R에 사용될 함수들을 개발하고 있다.
 - ▶ 대부분은 여러분과 비슷한 문제를 해결하기 위해 온라인 게시판에 질문을 던지고, 그 해결책을 올리고 있다. 해결책은 대부분 R 코드 조각으로 올라온다.
 - ▶ 프로그래밍을 배우는 가장 좋은 방법은 다른 사람을 코드를 보는 것이니 이러한 온라인 커뮤니티 검색을 생활화하자.

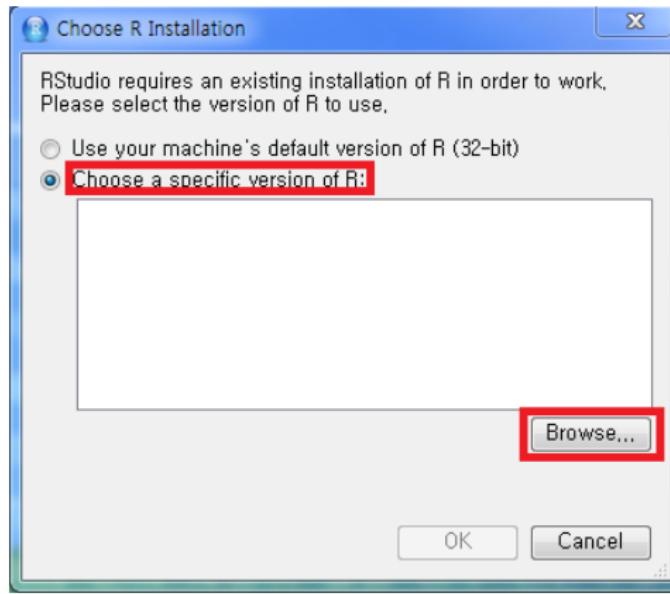
StackOverflow

<http://stackoverflow.com/questions/tagged/r>

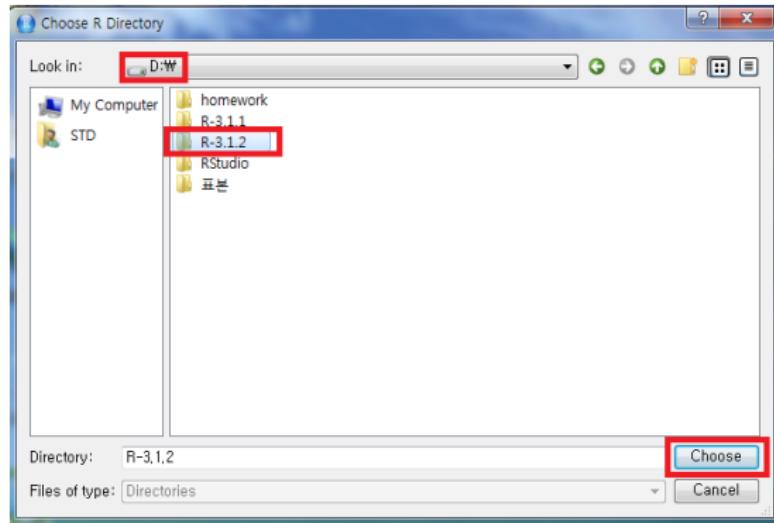
RStudio 전산실습실에서 사용하기

- 통계학과 전산실습실 컴퓨터는 매일 재설정되므로 R에 대한 기본 설정이 남아 있지 않음.
- RStudio를 시작할 때 아래와 다음과 같은 과정을 거치면 정상적으로 R을 로드할 수 있음.
- R은 D:\comp_stat\R-3.1.2, RStudio는 D:\comp_stat\RStudio에 설치되어 있다고 가정.
- RStudio 실행: D:\comp_stat\RStudio\bin\RStudio.exe

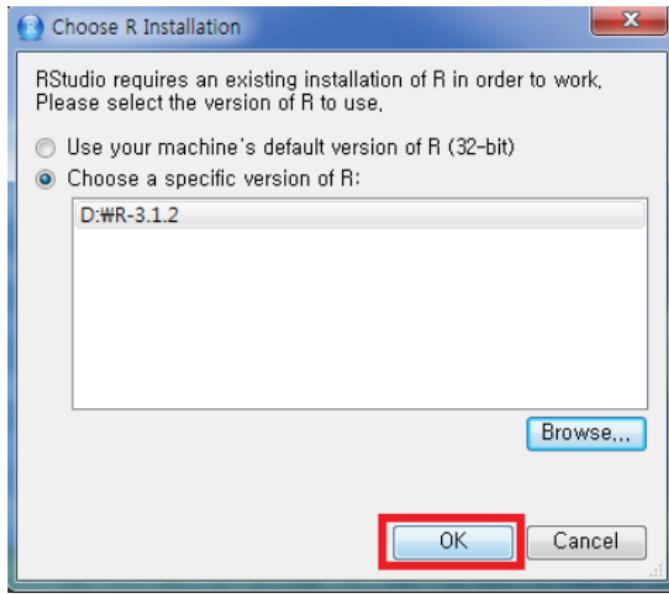
RStudio 전산실습실에서 사용하기 (2)



RStudio 전산실습실에서 사용하기 (3)



RStudio 전산실습실에서 사용하기 (4)



정리

- R은 다양한 통계 기법 지원과 그래픽 능력을 갖춘 오픈 소스 데이터 분석 소프트웨어로서 그 사용이 최근 급격히 늘고 있다.
- 오픈 소스의 특성상, 필요한 기능이 대두되면 그 구현이 대단히 빠르며, 이 점이 사용자 수 증가에 큰 요인이 되었다.
- R은 다양한 컴퓨터 환경을 지원하며, 설치가 매우 쉽다.
- R은 기본적으로 프로그래밍 언어이며, 명령어 기반으로 동작한다.

정리 (cont'd)

- 통합개발환경(IDE)인 R Studio를 사용하여 새로운 함수 작성 및 패키지 관리를 쉽게 할 수 있다.
- 패키지는 추가적인 함수들의 집합이며 R 커뮤니티의 개발자들이 오픈 소스로 작성하고 있다.

참고자료

- Avril Coghlan, A Little Book of R for Biomedical Statistics
<http://a-little-book-of-r-for-biomedical-statistics.readthedocs.org/>
- Jeffrey Stanton, Introduction to Data Science
<http://jsresearch.net/wiki/projects/teachdatascience>
- Quick R
- <http://stackoverflow.com/questions/tagged/r>
- <https://stat.ethz.ch>

Section 2

DVCS - Git

버전 관리를 들어본 적 없는 사람들을 위한 DVCS

- DVCS : Distributed Version Control System
- 숙제를 위한 코드



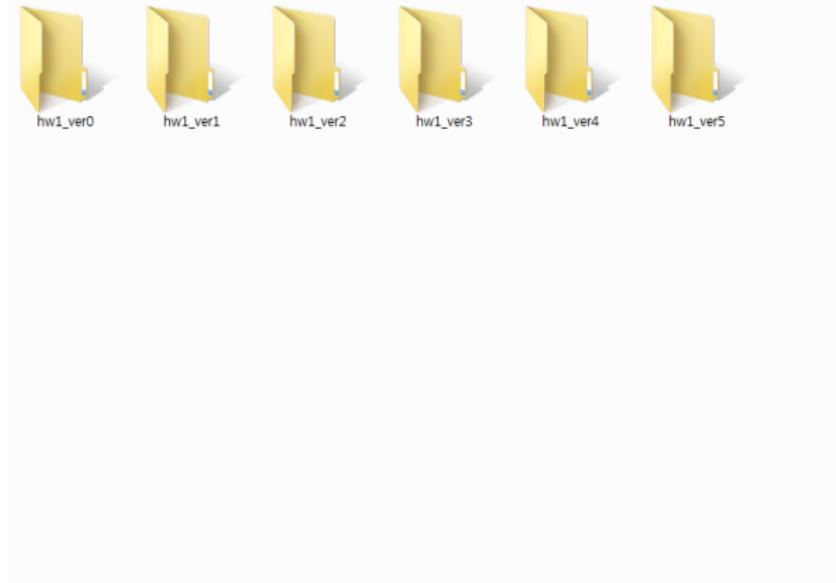
버전 관리를 들어본 적 없는 사람들을 위한 DVCS (cont'd)

- 수정이 있었다.



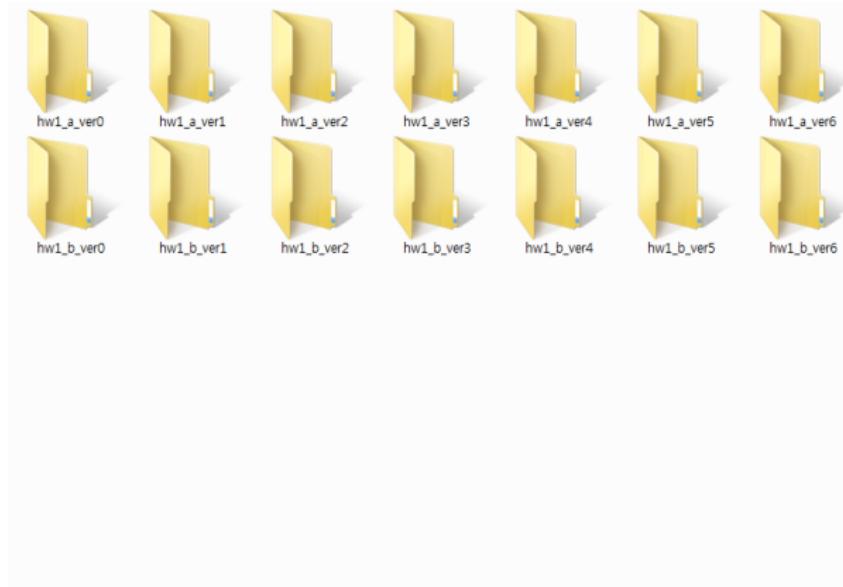
버전 관리를 들어본 적 없는 사람들을 위한 DVCS (cont'd)

- 많은 수정이 있었다....



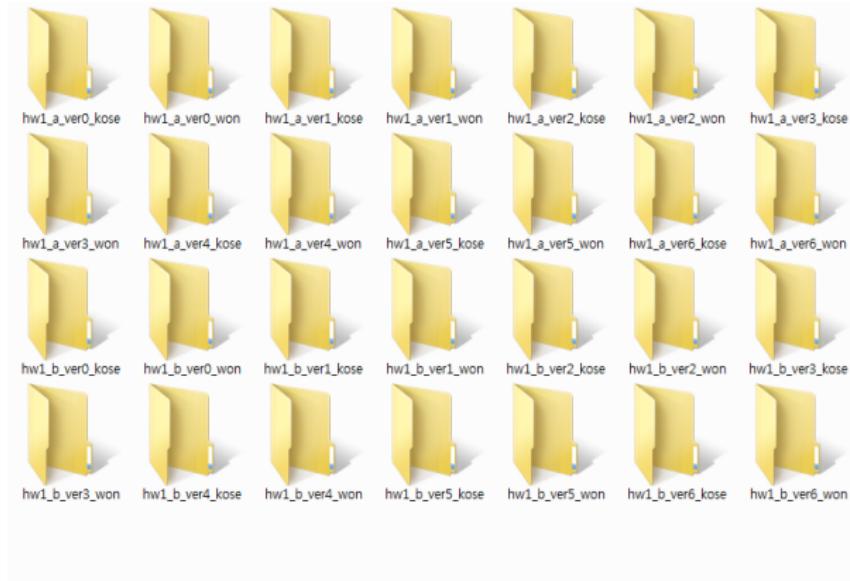
버전 관리를 들어본 적 없는 사람들을 위한 DVCS (cont'd)

- 숙제를 위한 코드가 하나가 아니라면?



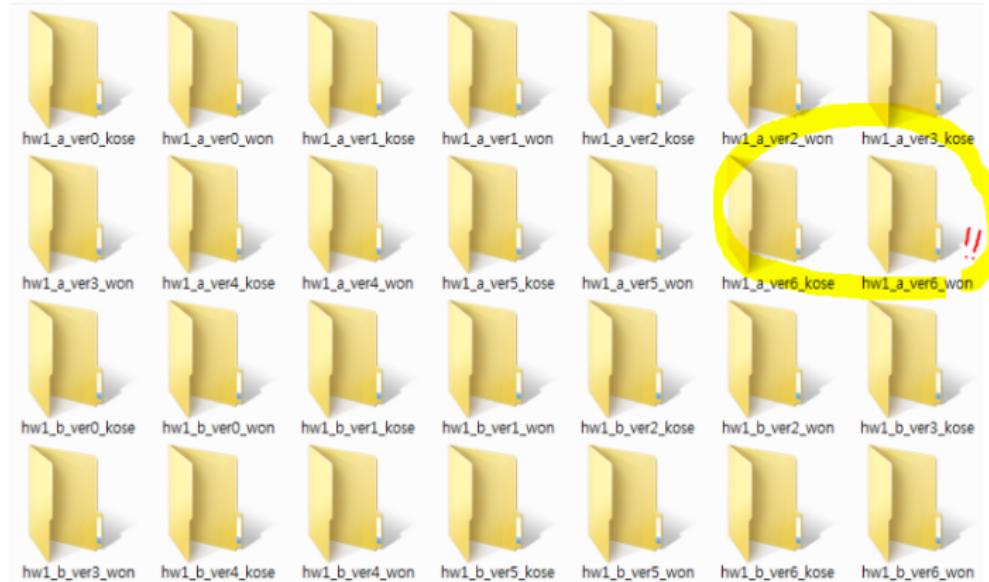
버전 관리를 들어본 적 없는 사람들을 위한 DVCS (cont'd)

- 팀 프로젝트라면?usb?



버전 관리를 들어본 적 없는 사람들을 위한 DVCS (cont'd)

- 합쳤는데 버그가!!



버전 관리를 들어본 적 없는 사람들을 위한 DVCS (cont'd)

- 버전 관리를 전문적으로 해주는 소프트웨어가 있다.

어떤 소프트웨어가 있나?

많다



source: http://www.slideshare.net/ibare/dvcs-git?qid=b93af986-f188-4813-abd4-eb13631bf51f&v=default&b=&from_search=2

Git?

- 분산형 소스 코드 버전 관리 시스템
- 리눅스에서 사용하던 Bitkeeper의 라이센스 이슈로, Linus Torvalds가 직접 개발
- Git: 재수없고 멍청한 놈, 자식
- GitHub의 등장과 함께 폭발적으로 성장

The screenshot shows the official Git website at git-scm.com. The header features the Git logo (a red diamond with a white 'g') and the text "git --distributed-even-if-your-workflow-isn't". A search bar is located in the top right. The main content area has a light blue background with a grid pattern. It contains two sections of text in English, each with a small icon. To the right is a diagram illustrating Git's distributed nature with five computer nodes connected by a network of colored lines (red, green, blue) representing data flow between repositories.

Git is a [free and open source](#) distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is [easy to learn](#) and has a [tiny footprint with lightning fast performance](#). It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like [cheap local branching](#), convenient staging areas, and [multiple workflows](#).

 Learn Git in your browser for free with [Try Git](#).

Git의 개념 및 특징

- 소스 코드의 변경 이력을 관리
 - ▶ 특정 시점 및 변경 추적
 - ▶ 원하는 시점으로 파일의 상태 변경이 가능(유연성을 제공)
- 분산형 소스 코드 버전 관리 시스템
 - ▶ 지역 저장소와 원격 저장소로 구분
 - ▶ 지역 저장소에서 개별 버전 관리
 - ▶ 원격 저장소로 내용 공유
 - ▶ 인터넷 연결이 되지 않아도 버전 관리가 가능
 - ▶ 원격 저장소 장애로 내용물 유실시에도 지역 저장소에서 복구 가능

Git의 개념 및 특징 (cont'd)

- 브랜치 모델

- ▶ 비 선형적 개발 지원
- ▶ 병행개발을 통한 지속적이고 안정적인 소프트웨어 개발
- ▶ 다양한 버전의 시도 및 관리
- ▶ 신규기능, 버그수정, 실험적 작업을 위해 사용

- 데이터 무결성

- ▶ 모든 변경 사항에 대한 스냅샷을 저장

완전 초보를 위한 기본 용어

- 작업 공간(Working Directory)

- ▶ 내 컴퓨터(local)의 소스 코드 작업이 이루어지는 폴더
 - ▶ 작업 공간에서 Git 초기화를 하면 저장소가 만들어진다.

- 저장소(Repository)

- ▶ 프로젝트가 거주(live)하는 저장 공간. 종종 'repo'라고 불린다.
 - ▶ 컴퓨터의 로컬 폴더일 수도 있고, 온라인 호스트(ex. GitHub)의 저장 공간일 수도 있다.
 - ▶ Git 정보가 저장되어 있다(보이지는 않는다).
 - ▶ 작업 공간에서 저장소로 의미 단위의 작업 스냅샷을 전달해(커밋, Commit) 버전을 관리한다.

- 스테이징 영역(Staging Area)

- ▶ 커밋될 예정인 파일의 내용을 준비하는 곳
 - ▶ 쇼핑몰에서 물건을 구매할 때 먼저 장바구니에 넣는 것과 유사

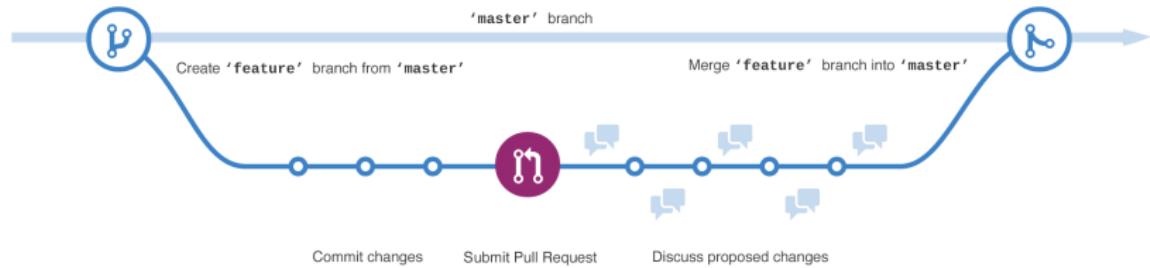
완전 초보를 위한 기본 용어 (cont'd)

- 지역 저장소(Local Repository)
 - ▶ 작업이 실제로 이루어지는 컴퓨터의 저장소
 - ▶ remote를 통해 원격 저장소와 연결
- 원격 저장소(Remote Repository)
 - ▶ Git server를 구축해서 쓰는 경우 해당 server에 만들어진 저장소
 - ▶ 온라인 호스트(ex. GitHub)를 이용하는 경우, 호스트에서 제공하는 온라인 상의 저장 공간
- 커밋
 - ▶ 그 시점의 지역 저장소의 '스냅샷'을 찍어, 프로젝트를 이전의 어떤 상태로든 재평가하거나 복원할 수 있는 체크포인트를 만든다.
 - ▶ 사용자가 의미 있는 작업 뭉치를 판단해 커밋을 명령한다.
 - ▶ 커밋 시 작업 내용을 간략하게 comment로 남긴다.

완전 초보를 위한 기본 용어 (cont'd)

- 브랜치

- ▶ 병행 개발을 위한 분기 생성
- ▶ 협업의 경우, 작업자들은 메인 프로젝트에서 브랜치를 따와서(branch off), 변경을 통해 자신만의 버전을 만든다.
- ▶ 작업이 끝나면 메인 디렉토리인 'master'에 브랜치를 다시 합친다 (Merge).



완전 초보를 위한 기본 명령어

- 모든 명령어는 'git'으로 시작한다.
- git init
 - ▶ 깃 저장소 초기화
 - ▶ 작업 공간을 로컬 저장소로 만든다.
- git config
 - ▶ git을 설치한 컴퓨터에서 작업자의 정보를 설정 시 사용.
- git help
 - ▶ 21개의 가장 많이 사용하는 명령어를 보여준다.
 - ▶ 다음과 같이 각 명령어에 대한 더 자세한 설명을 볼 수 있다.
`git help init`

완전 초보를 위한 기본 명령어 (cont'd)

- git status
 - ▶ 저장소의 상태를 확인하는 명령어
 - ▶ add나 commit 전에 명령어를 통해 변경사항과 저장소 상태를 확인
 - ▶ 저장소의 작업 브랜치, 저장소에서 tracked 중인 파일, 커밋이 필요한 변경 사항을 확인할 수 있다.
- git add
 - ▶ git이 이 파일을 지켜보게(tracked) 만들어, git의 저장소 스냅샷에 포함.
`git add 'filename'`
 - ▶ 스테이지 영역에 파일이 올라가게 된다.
 - ▶ -all 옵션을 통해 저장소의 모든 파일을 스테이지 영역에 올릴 수 있다.
`git add --all`
- git commit
 - ▶ 저장소의 스냅샷을 찍기 위한 명령어로 로컬 저장소로 변경 내용을 전달한다.
 - ▶ 보통 간단한 변경 사항에 대한 comment를 함께 남긴다.
`git commit -m "Comment"`

완전 초보를 위한 기본 명령어 (cont'd)

- git branch

- ▶ 새 브랜치 'labs'를 만들어 병행 작업을 하도록 해 준다.

```
git branch labs
```

- ▶ master 브랜치와 별개로 변경 사항을 labs 브랜치에서 시험해 볼 수 있다.

- git checkout

- ▶ 현재 위치하고 있지 않은 저장소 혹은 브랜치로 옮겨갈 수 있게 해준다.

```
git checkout master
```

- git merge

- ▶ 생성한 브랜치에서 작업이 완료되어 다른 브랜치로 병합하고자 할 때 사용한다.

```
git merge labs
```

- ▶ labs 브랜치에서 만든 모든 변경사항을 master 브랜치에 반영한다.
 - ▶ 이 때 충돌이 생기면 이를 보고한다.

완전 초보를 위한 기본 명령어 (cont'd)

- git remote

- ▶ 지역 저장소와 원격 저장소 간의 연결을 위한 명령어
- ▶ add 옵션을 통해 원격 저장소의 주소를 알려줄 수 있다.

```
git remote add https://...
```

- ▶ -v 옵션을 통해 현재 지역 저장소가 바라보는 원격 저장소의 주소를 알 수 있다.

```
git remote -v
```

- git push

- ▶ 지역 저장소에서 원격 저장소로 작업을 올리기 위해 사용한다.

```
git push origin gh-pages
```

- ▶ 지역 저장소의 origin에서 원격 저장소의 gh-pages로 commit한 사항을 push한다.

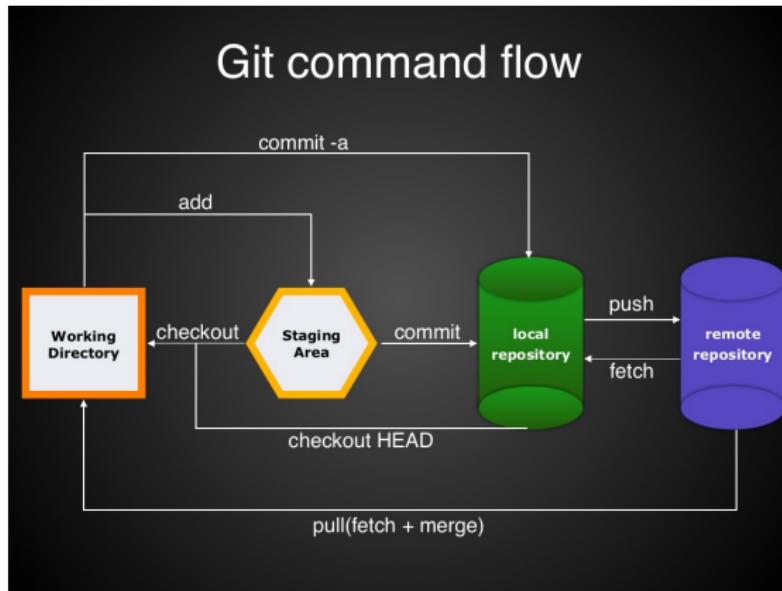
완전 초보를 위한 기본 명령어 (cont'd)

- git pull
 - ▶ 원격 저장소로부터 변경 내용을 지역 저장소로 가져온다.
 - ▶ 협업하는 경우, pull 명령어를 통해 하루를 시작하는 것을 추천한다.
- git clone
 - ▶ 온라인 호스트 등에서 제공되는 원격 저장소를 작업하고자 하는 컴퓨터로 다운로드하고 지역 저장소를 만든다.

```
git clone https://github.com/username/repo
```
 - ▶ 주로 온라인 호스트에서 자신의 원격 저장소로 가져오자 하는 내용을 fork한 후 자신의 컴퓨터에 지역 저장소를 만들어 작업하는 순서로 진행한다.

Git의 흐름

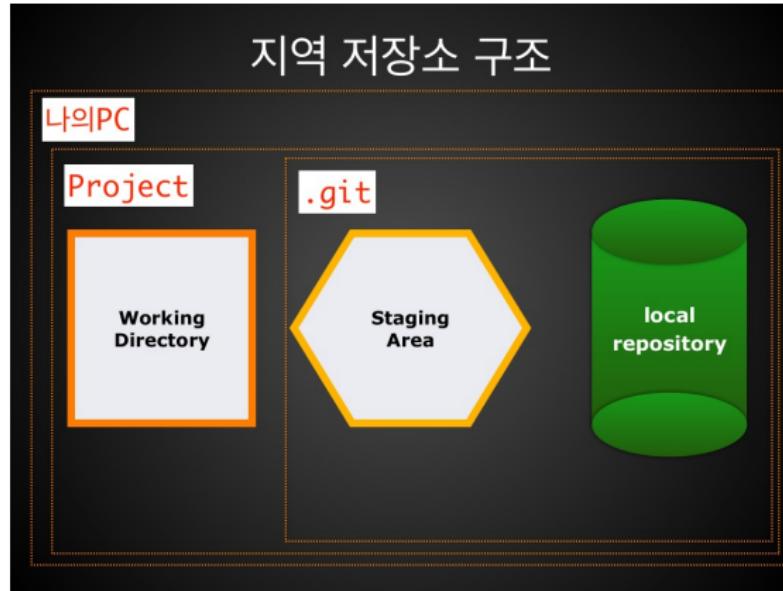
- Git Command Flow



source: http://www.slideshare.net/chandler0201/git-50655983?qid=1378b3a8-0389-4876-b1b0-c3c47fe54977&v=qf1&b=&from_search=4

Git의 흐름 (cont'd)

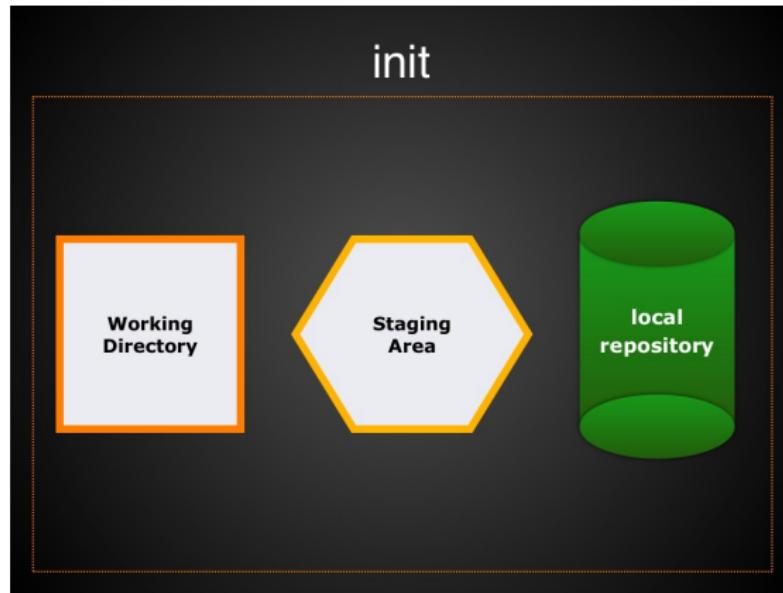
- 지역 저장소 구조



source: http://www.slideshare.net/chandler0201/git-50655983?qid=1378b3a8-0389-4876-b1b0-c3c47fe54977&v=qf1&b=&from_search=4

Git의 흐름 (cont'd)

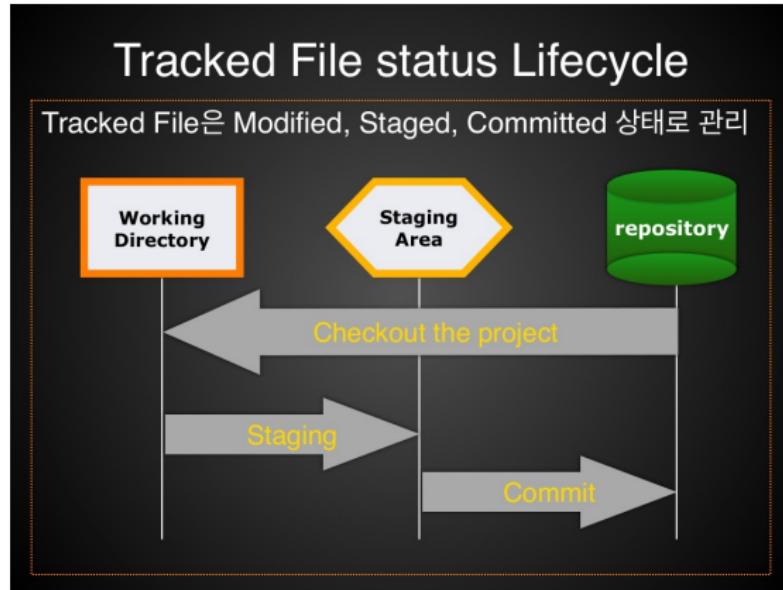
- 지역 저장소 init



source: http://www.slideshare.net/chandler0201/git-50655983?qid=1378b3a8-0389-4876-b1b0-c3c47fe54977&v=qf1&b=&from_search=4

Git의 흐름 (cont'd)

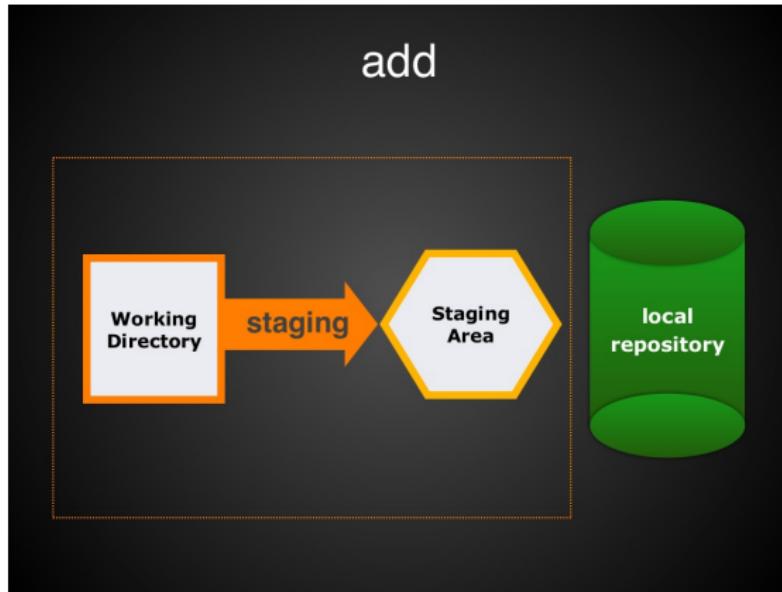
- Tracked file status lifecycle



source: http://www.slideshare.net/chandler0201/git-50655983?qid=1378b3a8-0389-4876-b1b0-c3c47fe54977&v=qf1&b=&from_search=4

Git의 흐름 (cont'd)

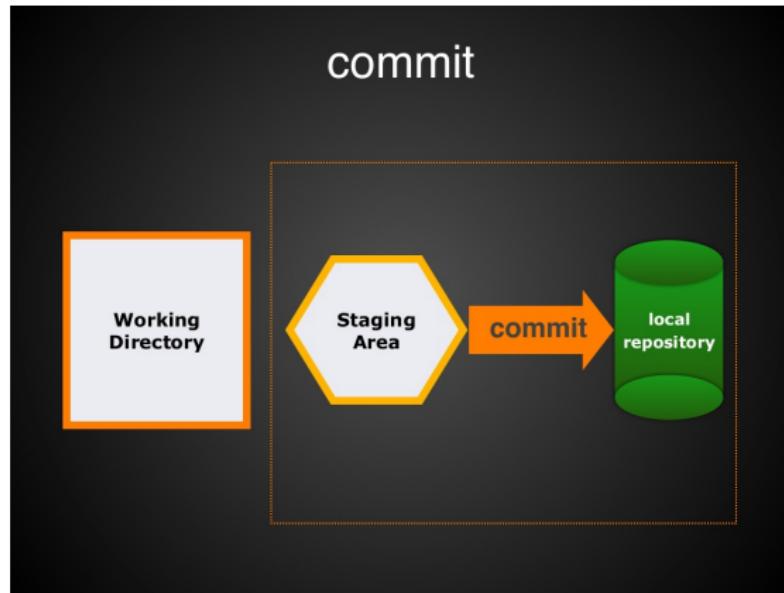
- 작업공간에서 스테이지 영역으로 add



source: http://www.slideshare.net/chandler0201/git-50655983?qid=1378b3a8-0389-4876-b1b0-c3c47fe54977&v=qf1&b=&from_search=4

Git의 흐름 (cont'd)

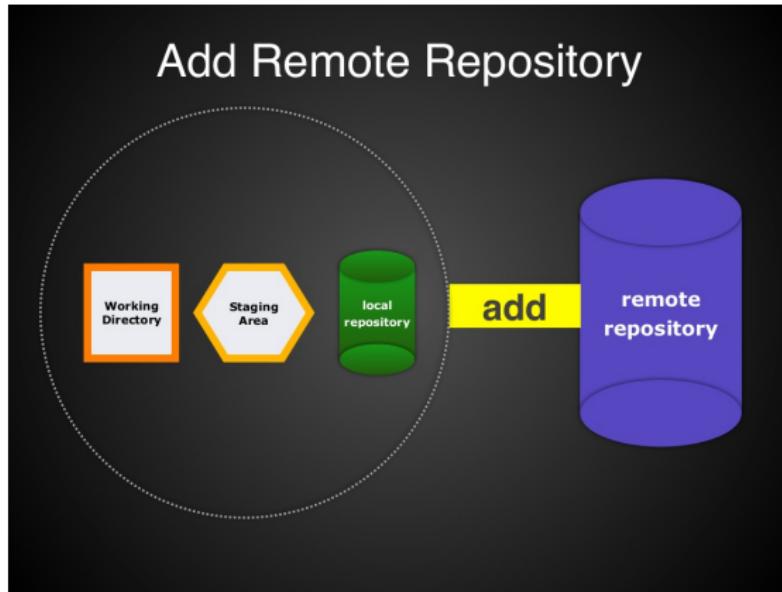
- 스테이지 영역에서 지역 저장소로 commit



source: http://www.slideshare.net/chandler0201/git-50655983?qid=1378b3a8-0389-4876-b1b0-c3c47fe54977&v=qf1&b=&from_search=4

Git의 흐름 (cont'd)

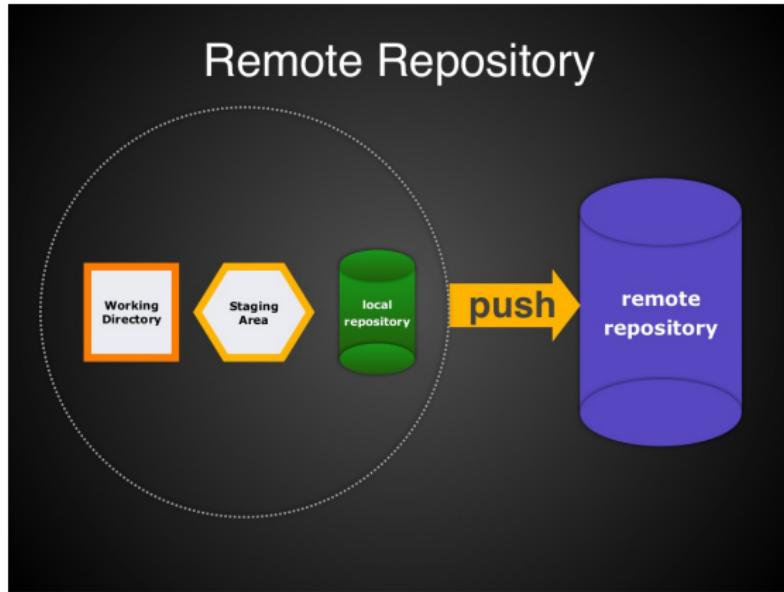
- 지역 저장소에서 원격 저장소로 remote 추가



source: http://www.slideshare.net/chandler0201/git-50655983?qid=1378b3a8-0389-4876-b1b0-c3c47fe54977&v=qf1&b=&from_search=4

Git의 흐름 (cont'd)

- 지역 저장소에서 원격 저장소로 push



source: http://www.slideshare.net/chandler0201/git-50655983?qid=1378b3a8-0389-4876-b1b0-c3c47fe54977&v=qf1&b=&from_search=4

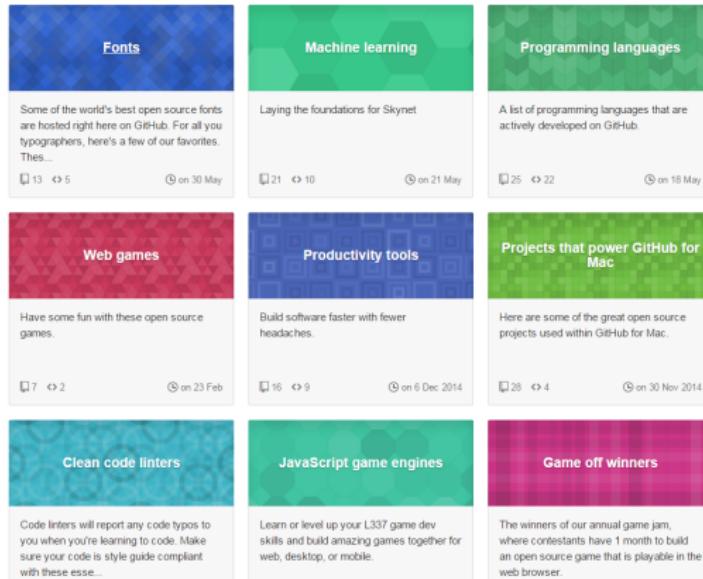
GitHub

- 웹을 통한 프로젝트 버전 관리
- 협업을 위한 코드 공유
- 평가 측정 등의 소셜 네트워크 기능



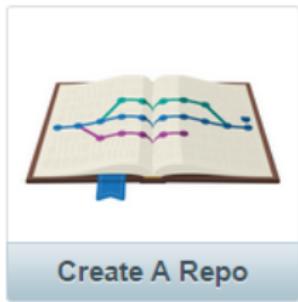
GitHub (cont'd)

- 유명한 오픈 소스 대부분이 여기에 있다



|<https://github.com/explore/>|

GitHub를 사용하기 위한 흐름



|<https://help.github.com/>|

Try Git

- Try Git
- Code School과 GitHub에서 제공하는 튜토리얼
- 튜토리얼을 완수하세요!

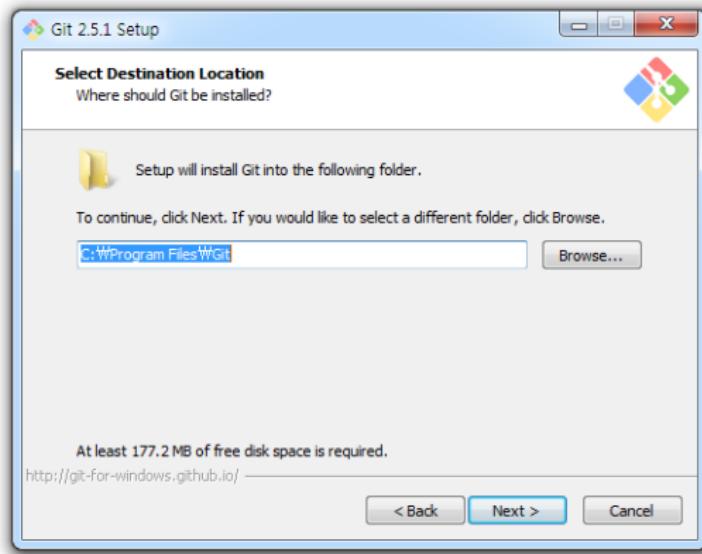


Git 설치하기

- 이렇게 좋은 Git을 직접 설치해봅시다.
- Window를 위한 다운로드 (대부분 자동으로 적절한 버전이 다운로드된다.)
- 다운로드 후 실행

Git 설치하기 (cont'd)

- 설치 폴더를 지정해준다. D:\comp_stat\내에 설치.



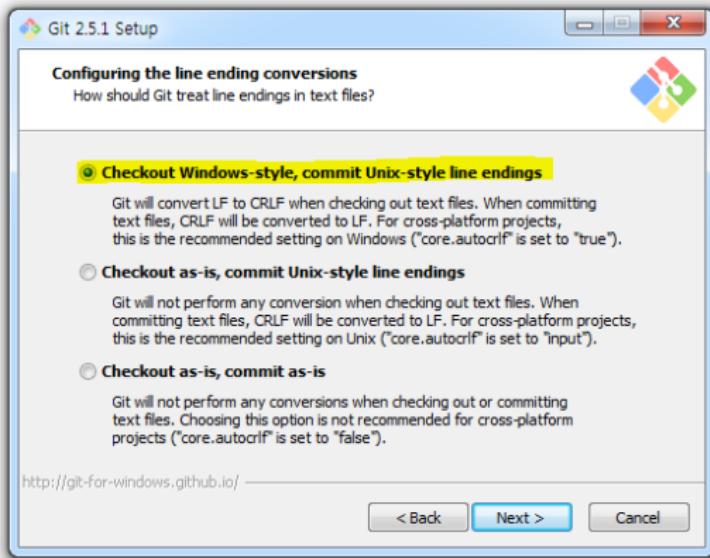
Git 설치하기 (cont'd)

- 표시된 부분 선택 후 Next 클릭.



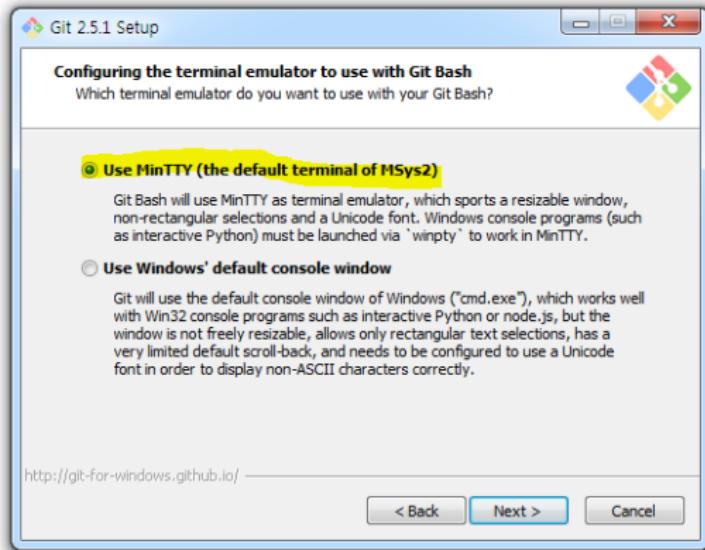
Git 설치하기 (cont'd)

- 표시된 부분 선택 후 Next 클릭.



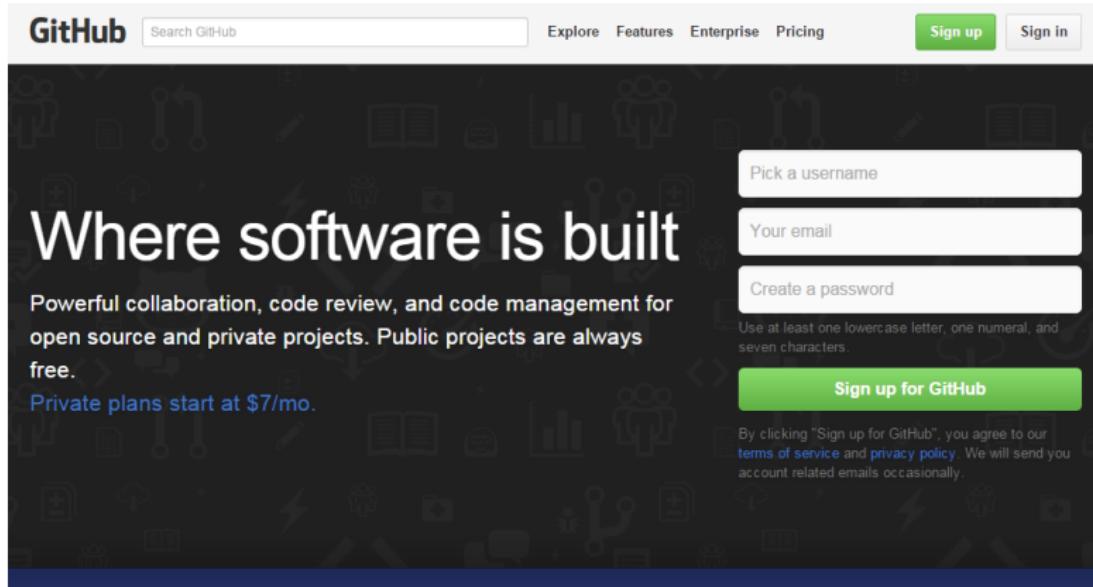
Git 설치하기 (cont'd)

- 표시된 부분 선택 후 Next 클릭.



처음으로 Git/GitHub 설정하기

- Github 가입



처음으로 Git/GitHub 설정하기 (cont'd)

- 로컬 컴퓨터에서 Git 초기 설정하기

- ▶ Git bash를 실행한다.
 - ▶ 자신을 소개한다. 인용부호 안에 GitHub를 가입할 때 정한 Username을 적는다.

```
git config --global user.name "USERNAME"
```

- ▶ GitHub에 가입할 때 정한 이메일을 말해준다.

```
git config --global user.email "Your@email.com"
```

온라인 저장소 만들기

- GitHub에서 온라인 저장소를 만들어봅시다.

The screenshot shows a GitHub user profile for 'Young-won'. On the left, there's a feed of recent pull requests:

- ttyong318 opened pull request snu-stat/students#5**: practice0. pull request, 1 commit with 4 additions and 0 deletions.
- youjin2 opened pull request snu-stat/students#4**: upload, 1 commit with 4 additions and 0 deletions.
- soohwanjo opened pull request snu-stat/students#3**: pull, 2 commits with 4 additions and 0 deletions.
- inguri0403 opened pull request snu-stat/students#2**: Create inguri0403.json

On the right, there are sections for 'GitHub Universe: The Big Bang' (two full days of how to build, collaborate, and deploy great software. Join us on October 1 & 2 in San Francisco), 'Repositories you contribute to' (snu-stat/students, snu-stat/snu-stat.github.io, snu-stat/slides, snu-stat/syllabus, snu-stat/prep_LectureNotes), and 'Your repositories' (with a '+ New repository' button highlighted by a yellow box). A search bar at the bottom says 'Find a repository...' and filter buttons for 'All', 'Public', 'Private', 'Sources', and 'Forks' are shown.

온라인 저장소 만들기

- Private은 유료!

Owner Repository name

 Young-won  / 

Great repository names are short and memorable. Need inspiration? How about [adamant-lamp](#).

Description (optional)

 **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** | ⓘ

Create repository

로컬 저장소 만들기

- ① 먼저 작업 공간을 만든다.
- ② git 사용을 위해 초기화한다.

```
git init
```

- ③ 작업을 진행한다. 여기서는 'Readme.md' 파일을 만들었다고 가정한다.
- ④ 로컬 저장소의 상황을 확인한다.

```
git status
```

- ⑤ 작업한 파일을 스테이지 영역으로 올린다.

```
git add 'Readme.md'
```

- ⑥ 로컬 저장소에 변경 사항을 저장한다.

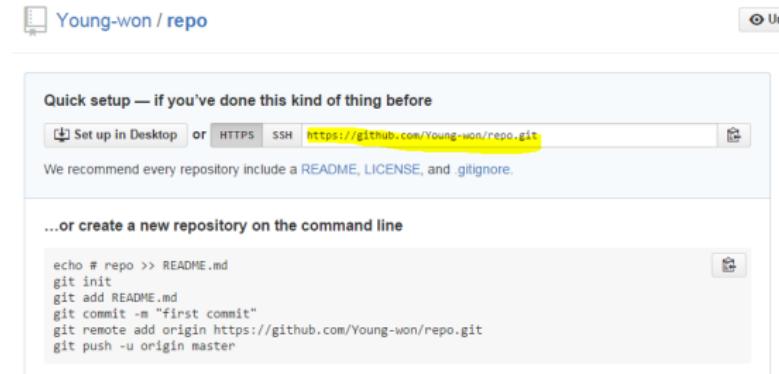
```
git commit -m 'Add Readme.md'
```

로컬 저장소와 원격 저장소 연결하기

- 앞서 만든 GitHub의 원격 저장소와 연결한다.

```
git remote add origin https://github.com/username/repo
```

- 주소는 GitHub의 해당 저장소에서 확인할 수 있다.



로컬 저장소와 원격 저장소 연결하기 (cont'd)

- ③ 다음과 같이 제대로 연결되었는지 확인한다.

```
git remote -v
```

- ④ 로컬 저장소에서 원격 저장소로 변경 사항을 업로드한다.

```
git push origin master
```

참고자료

- Github Guides
- Pro Git 오픈 소스북, 3장 까지만 읽으면 기본적인 내용을 익힐 수 있다.
- Try Git Code School과 GitHub에서 제공하는 튜토리얼
- Git 입문자를 위한 가이드
- 버전 관리를 들어본 적 없는 사람을 위한 DVCS - Git(SlideShare)
- 완전 초보를 위한 깃허브

Section 3

Lecture Page

Lecture Page 소개

- <https://snu-stat.github.io>
- 앞으로 모든 강의 자료는 이곳에 올라옵니다.
- 수업을 위한 private repository가 제공됩니다. 숙제는 앞으로 이를 통해 진행하게 됩니다.



강의목표

본 과목은 컴퓨터 프로그래밍에 익숙하지 않은 통계학 전공 학부생을 대상으로 하며, 프로그래밍의 기본 개념과 이를 이용해 통계학의 문제를 계산적으로 접근하는 방법에 대해 익숙해지는 것을 목표로 한다. 이 과목에서 배우는 내용은 크게 다음과 같다.

Git 실습 (cont'd)

- 실습 : Students repo를 fork한 후, 자신의 GitHub 정보를 업로드합니다.

- 1 Student repo를 자신의 원격 저장소로 fork해가세요.

The screenshot shows a GitHub repository page for 'snu-stat / students'. The 'Fork' button in the top right corner is highlighted with a yellow box. The repository has 15 commits, 1 branch, 0 releases, and 4 contributors. The README.md file contains the following content:

```
Student directory

Pull requests를 연습하기 위한 저장소입니다.

_data/TERM_YEAR/ 디렉토리의 USERNAME.json 파일을 수정하고, pull request를 통해 제출해주세요. 다음의 예시를 참고하세요.

이 저장소의 모든 내용은 https://github.com/advanced-js/students를 참고하여 작성되었습니다.
```

On the right side of the page, there are sections for Code, Issues, Pull requests, Wiki, Pulse, Graphs, and Settings. Under the Code section, there is an HTTPS clone URL field with the value <https://github.com/>, and Clone in Desktop and Download ZIP buttons.

Git 실습 (cont'd)

- ② 시작-Git-Git bash를 실행한 후 작업할 디렉토리로 이동하세요.

```
Welcome to Git (version 1.9.5-preview20150319)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.
sh.exe": nnexport: command not found
sh.exe": nnexport: command not found

won@WON-PC ~
$ cd d:\comp_stat

won@WON-PC /d/comp_stat
$
```

Git 실습 (cont'd)

③ 자신의 원격 저장소의 주소를 보고 git bash에서 clone을 진행하세요.

The screenshot shows a GitHub repository page for 'Young-won / students'. The repository has 15 commits, 1 branch, 0 releases, and 4 contributors. The 'gh-pages' branch is selected. A merge pull request from 'ietyong318/gh-pages' is shown. The repository contains files like README.md and test.js. The 'Code' tab is open, showing the repository's contents. The 'HTTPS Clone URL' field is highlighted in yellow, containing the URL `https://github.com/username/repo.git`. Below it, there are buttons for 'Clone in Desktop' and 'Download ZIP'.

The terminal window below shows the command to clone the repository:

```
welcome to Git (version 1.9.5-preview20150319)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.
sh.exe": nexport: command not found
sh.exe": nexport: command not found

won@WON-PC ~
$ cd d:/comp_stat
won@WON-PC /d/comp_stat
$ git clone https://github.com/username/repo.git
```

Git 실습 (cont'd)

- ④ 다운로드된 repo에서 예제 파일을 보고 본인의 GitHub username으로 만든 username.json 파일을 추가하세요.
- ⑤ 다음으로 해당 디렉토리에서 상태를 확인하세요.

```
git status
```

```
won@WON-PC ~  
$ cd d:/comp_Stat  
  
won@WON-PC /d/comp_Stat  
$ git clone https://github.com/Young-won/students  
Cloning into 'students'...  
remote: Counting objects: 77, done.  
remote: Compressing objects: 100% (26/26), done.  
remote: Total 77 (delta 5), reused 0 (delta 0), pack-reused 44  
Unpacking objects: 100% (77/77), done.  
Checking connectivity... done.  
  
won@WON-PC /d/comp_Stat  
$ git status
```

- ⑥ 원격 저장소와의 연결도 확인하세요.

```
git remote -v
```

Git 실습 (cont'd)

- ⑦ 이제 원격 저장소로 변경사항을 전달합시다.
- ⑧ 먼저 add를 통해 스테이지에 변경사항을 올리고, 스테이지의 내용을 지역 저장소로 commit합니다.

```
git add "_data\autumn_2015\username.json"  
git commit -m "Upload username.json"
```

- ⑨ commit이 완료되었으면 지역 저장소의 내용을 원격 저장소로 push 합니다.
- ⑩ 지역 저장소와 원격 저장소의 브랜치를 주의해서 push 명령을 수행합니다.

```
won@WON-PC /d/comp_Stat/students (gh-pages)  
$ git remote -v  
origin https://github.com/Young-won/students (fetch)  
origin https://github.com/Young-won/students (push)  
  
won@WON-PC /d/comp_Stat/students (gh-pages)  
$ git add '_data\autumn_2015\username.json'  
  
won@WON-PC /d/comp_Stat/students (gh-pages)  
$ git commit -m 'Upload username.json'  
[gh-pages 4efffd4] Upload username.json  
 1 file changed, 4 insertions(+)  
  create mode 100644 _data\autumn_2015\username.json  
  
won@WON-PC /d/comp_Stat/students (gh-pages)  
$ git push origin gh-pages  
Username for 'https://github.com': Young-won  
Password for 'https://Young-won@github.com':
```

Git 실습 (cont'd)

- ⑪ 원격 저장소에서 master 브랜치로 pull request를 보내면 담당자가 merge를 진행합니다.

The screenshot shows a GitHub repository page for 'Young-won / students'. The repository was forked from 'snu-stat/students'. The main interface includes fields for 'Description' and 'Website', and summary statistics like 16 commits, 1 branch, 0 releases, and 4 contributors. A sidebar on the right lists repository features: Pull requests (highlighted in yellow), Wiki, Pulse, Graphs, and Settings. The 'Pull requests' section shows 0 pull requests.

Young-won / students

forked from snu-stat/students

Description

Short description of this repository

Website

Website for this repository (optional)

Save or Cancel

16 commits 1 branch 0 releases 4 contributors

Branch: gh-pages ➔ students / +

This branch is 1 commit ahead of snu-stat:gh-pages.

Upload username.json

Young-won authored 10 minutes ago

_data Upload username.json latest commit 4eeffd426a

Pull requests 0

Wiki

Pulse

Graphs

Settings

Git 실습 (cont'd)

- ⑪ 원격 저장소에서 master 브랜치로 pull request를 보내면 담당자가 merge를 진행합니다.

The screenshot shows a GitHub repository page for 'Young-won / students'. At the top, there are buttons for 'Unwatch' (with 1 follower), 'Star' (0 stars), 'Fork' (6 forks), and a 'New pull request' button, which is highlighted with a yellow box. Below these are tabs for 'Pull requests' (selected), 'Labels', and 'Milestones'. A search bar contains the filters 'is:pr is:open'. The main area is currently empty, showing a large white space.

Git 실습 (cont'd)

- ① 원격 저장소에서 master 브랜치로 pull request를 보내면 담당자가 merge를 진행합니다.

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.

The screenshot shows a GitHub interface for comparing branches. At the top, there are dropdown menus for 'base fork' (snu-stat/students), 'base' (gh-pages), '...', 'head fork' (Young-won/students), and 'compare' (gh-pages). A green checkmark indicates that the branches are 'Able to merge'. Below this, a button labeled 'Create pull request' is highlighted with a yellow box. To its right, instructions say 'Discuss and review the changes in this comparison with others.' and a question mark icon. At the bottom, summary statistics are shown: 1 commit, 1 file changed, 0 commit comments, and 1 contributor.

base fork: snu-stat/students ▾ base: gh-pages ▾ ... head fork: Young-won/students ▾ compare: gh-pages ▾

✓ Able to merge. These branches can be automatically merged.

Create pull request

Discuss and review the changes in this comparison with others. ?

1 commit 1 file changed 0 commit comments 1 contributor

Git 실습 (cont'd)

- ⑪ 원격 저장소에서 master 브랜치로 pull request를 보내면 담당자가 merge를 진행합니다.

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

The screenshot shows the GitHub interface for creating a pull request. At the top, there are dropdown menus for 'base fork' (snu-stat/students), 'base' (gh-pages), 'head fork' (Young-won/students), and 'compare' (gh-pages). A green checkmark indicates that the branches are 'Able to merge'. The main area has tabs for 'Write' and 'Preview', with 'Write' selected. It includes a 'Leave a comment' text area and a note about attaching images. On the right, there are sections for 'Labels' (None yet), 'Milestone' (No milestone), and 'Assignee' (No one—assign yourself). At the bottom, a large green button labeled 'Create pull request' is highlighted with a yellow box.

Q&A

- 강의를 위한 GitHub Page는 etl에 다시 공지됩니다.