

# 빅데이터의 이해와 활용

[2022-2 2주차 강의자료]

1장. 빠르게 시작하기

박 혜 승 교 수



# 1장 빠르게 시작하기

- 1.1 R 소개하기
- 1.2 R의 필요성
- 1.3 R 설치하기
- 1.4 RStudio
- 1.5 간단한 예
- 1.6 마치며

# 1.1 R 소개하기

## » R 소개하기

- R은 통계 처리, 데이터 탐색 및 분석, 시각화를 위한 강력한 프로그래밍 언어이자 개발 환경
- 사용자와 개발자가 서로의 경험을 공유하면서 7500개 이상의 패키지 개발에 적극적으로 참여하는 무료 오픈 소스 소프트웨어
- R은 이러한 강력하고 빠르게 성장하는 커뮤니티를 기반으로 다양한 분야의 문제를 다룸

<https://cran.r-project.org/web/views/>

# 1.1 R 소개하기

## » R 소개하기

- R 프로그래밍 언어는 1993년까지 거슬러 올라갈 만큼 그 기원이 오래됨
- 지난 10년 사이 데이터 관련 연구 산업이 급속히 성장함
- R 언어는 데이터 과학 분야에서 기본으로 채택하는 만국 공통어가 됨
- 일반적으로 R은 단순한 프로그래밍 언어 그 이상임
- 종합적인 컴퓨팅 환경인 동시에 강력하고 적극적인 커뮤니티임
- 빠르게 성장하고 확장하는 생태계라고 볼 수 있음

# 1.1 R 소개하기

## » 프로그래밍 언어로서 R

- R은 프로그래밍 언어로서 지난 20여 년 동안 진화와 발전을 거듭해옴
- 종합적인 통계 연산, 데이터 탐색과 시각화를 좀 더 쉽고 유연하게 수행할 수 있게 한다는 목표가 분명함
- 사용의 편리함과 유연함을 동시에 추구하다 보면 당연히 갈등을 겪게 됨
- 수십 가지가 넘는 다양한 함수를 활용하여 마음대로 데이터를 변환하고 복잡한 도표를 만들 수 있다면 여러 상황에 매우 유연하게 대처할 수 있을 것
- 물론 이러한 기능을 올바르게 익히고 적절히 활용하는 것은 쉽지 않음
- R은 바로 이러한 균형을 잘 유지하고 있음

# 1.1 R 소개하기

## » 컴퓨팅 환경으로서 R

- R은 컴퓨팅 환경으로서 가볍고 사용하기 쉬움
- MATLAB이나 SAS 등 다른 유명한 통계 소프트웨어와 비교할 때 R은 훨씬 작고 배포하기도 너무 쉬움
- 이 책에서는 R의 거의 모든 작업을 RStudio를 사용하여 처리함
- 이 통합 개발 환경에서는 구문 강조 기능, 자동 완성 기능, 패키지 관리, 그래픽 뷰어, 도움말 뷰어, 환경 뷰어, 디버깅 등 다양한 기능을 제공함
- 이러한 기능은 생산성 향상에 큰 도움을 줌

# 1.1 R 소개하기

## » 커뮤니티로서 R

- R은 커뮤니티로서 강력하고 활기가 넘침
- Try R(<http://tryr.codeschool.com/>)을 방문하면 대화형 튜토리얼을 이용하여 R 언어의 기초를 가볍게 경험할 수 있음
- 실제로 코딩할 때 혼자서 모든 문제를 해결할 수는 없음
- 구글에서 R과 관련한 질문을 검색하거나 스택 오버플로(<http://stackoverflow.com/questions/tagged/r>)에서 웬만한 답변을 찾을 수 있을 것
- 원하는 질문에서 답변을 얻을 수 없다면 스택 오버플로에 질문을 올림
- 아마 2~3분 이내로 원하는 답변을 받을 수 있을 것

# 1.1 R 소개하기

## » 커뮤니티로서 R

- 어떤 패키지를 사용하기에 앞서 이것이 어떻게 동작하는지 알아보려면 온라인 저장소(또는 repo)에 방문함
- 소스 코드를 직접 확인할 수 있음
- 많은 저장소가 깃허브(GitHub)(<https://www.github.com>)에 호스팅됨
- 물론 깃허브로 훨씬 더 다양한 일도 할 수 있음
- 패키지가 올바르게 동작하지 않는다면 이 문제를 이슈로 정리하여 버그 리포팅을 할 수도 있음



# 1.1 R 소개하기

## » 커뮤니티로서 R

- 패키지 목적에 맞는 기능이 필요하다면, 요구 사항에 대한 이슈를 만들어 새로운 기능을 추가해 달라고 요청할 수도 있음
- 버그를 수정하고 새로운 기능을 구현하여 패키지 개발에 직접 기여하고 싶다면, 먼저 프로젝트를 포크(fork)하고 코드를 알맞게 수정함
- 저장소 관리자에게 풀 리퀘스트(pull request)를 보내고 패키지 관리자에게 변경 사항 승인을 요청할 수 있음
- 변경 사항을 받아들이면 패키지에 대한 정식 기여자가 되는 것
- 놀랍게도 R을 비롯한 패키지 수천 개는 전 세계에 있는 기여자가 만든 것

# 1.1 R 소개하기

## » 생태계로서 R

- R은 하나의 생태계로서 IT 산업을 넘어 모든 데이터 관련 분야에서 급속하게 성장하여 자신만의 영역을 확장하고 있음
- 사용자 대다수는 전문 개발자가 아니라 오히려 데이터 분석가 혹은 통계학자임
- 이들이 최상위 레벨의 코드를 작성하는 것은 아니지만, R 언어라는 생태계에 최첨단 도구들을 공급하는 것은 틀림없음
- 같은 도구를 다시 개발할 필요 없이 이미 있는 도구를 자유롭게 내려받아 사용할 수 있음
- IT 산업(일반적으로 데이터 과학 관련 학계와 산업계) 외부의 최첨단 기술을 이러한 생태계에서 보편적으로 사용 가능한 도구들에 바로 적용할 수 있다는 강점이 여기에 있음
- 현장 지식과 데이터 과학에서 생산성을 향상시켜 실제 가치로 전환하는 것을 좀 더 원활하게 함

# 1.2 R의 필요성

## » R의 필요성

- R은 다양한 통계 소프트웨어 가운데 다음 관점에서 ‘좀 더 앞선다’는 평가를 받고 있음
- 무료: R은 무료  
라이선스를 따로 구입할 필요가 없음  
R을 비롯한 대부분의 확장 패키지를 사용할 수 있는 재정적인 진입 장벽 또한 전혀 없음
- 오픈 소스: R과 대부분의 패키지는 완전히 오픈 소스  
개발자 수천 명이 패키지의 소스 코드를 지속적으로 검토함  
해결해야 하는 버그가 있는지 아니면 개선할 점은 없는지 확인함  
문제가 발생하면 직접 소스 코드를 파헤침  
문제가 있는 곳을 찾아 해결하는 데 기여할 수도 있음

# 1.1 R 소개하기

## » R의 필요성

- **인기:** R은 데이터 마이닝, 데이터 분석과 시각화를 수행하는 데 가장 널리 사용하는 통계 프로그래밍 언어이자 플랫폼  
인기가 많다는 것은 그만큼 같은 언어를 ‘사용하는’ 사람이 많다는 것을 의미  
또 이는 다른 사용자와 의사소통도 더 쉽다는 의미
- **유연성:** R은 동적 스크립트 언어  
함수형 프로그래밍이나 객체 지향 프로그래밍 같은 여러 패러다임의 프로그래밍 스타일을 허용할 만큼 유연성이 매우 뛰어남  
유연한 메타 프로그래밍을 지원함  
유연성으로 고도로 커스터마이징되면서 동시에 종합적인 데이터 변환과 시각화를 수행할 수 있음

# 1.1 R 소개하기

## » R의 필요성

- 재현성: 그래픽 사용자 인터페이스 기반의 소프트웨어를 사용하려면 원하는 메뉴를 선택하고 해당하는 버튼을 클릭함  
스크립트를 작성하지 않고 자동으로 수행한 작업을 정확하게 그대로 재현하기는 쉽지 않음  
대부분의 과학 연구 분야와 산업 응용 분야에서는 여러 가지 이유에서 재현성이 필요함  
R 스크립트를 사용하면 사용자가 컴퓨팅 환경과 데이터로 수행하는 작업을 정확하게 설명하기가 용이함  
모든 작업을 처음부터 완전히 재현할 수도 있음

# 1.1 R 소개하기

## » R의 필요성

- 풍부한 자원: R 관련해서 이미 엄청나게 많은 온라인 자료가 있고 계속해서 폭발적으로 증가함

대표적인 리소스는 바로 확장 패키지

지금 이 시점에서 CRAN(Comprehensive RArchive Network)은 패키지 약 1만 5523개를 제공함(2020년 3월 30일 기준)

CRAN은 전세계 미러 서버의 네트워크를 의미하며, 동일한 최신 R과 패키지들을 제공받을 수 있음

다변량 분석, 시계열 분석, 계량경제학, 베이지안 추론, 최적화, 금융, 유전학, 화학계량학, 계산물리학 등 거의 모든 데이터 관련 분야에서 개발자 8832명이 패키지를 제작하고 관리함

CRAN Task View(<https://cran.r-project.org/web/views/>)에서 분야별로 정리한 내용을 살펴볼 수 있음

엄청난 수의 패키지 외에도 정기적으로 개인 블로그를 남기거나 스택 오버플로 질문에 답변을 달고, 자신의 생각이나 경험, 추천할 만한 예제를 공유하는 수많은 저자가 있음

또 R-bloggers(<http://www.r-bloggers.com>),

RDocumentation(<http://www.rdocumentation.org>), METACRAN(<https://www.r-pkg.org/>) 블로그 등 R 전문 웹 사이트도 많이 있음

# 1.1 R 소개하기

## » R의 필요성

- 강력한 커뮤니티: R 커뮤니티에는 단순히 개발자만 있는 것이 아님

대다수는 통계, 계량 경제학, 금융, 생물정보학, 기계공학, 물리학, 의학 등 다양한 분야에서 R을 사용하는 전문가임

정말 많은 개발자가 R 언어를 사용하는 오픈 소스 프로젝트나 패키지 개발에 적극적으로 참여함

이 커뮤니티는 데이터 분석, 탐색, 시각화를 좀 더 쉽고 재미있게 하는 것이 목표임

R과 관련한 어떤 문제에 봉착했다면 일단 관련 내용을 검색해 보자

아마 벌써 그 문제와 관련한 몇 가지 답변을 찾을 수 있을 것

아무런 답변도 찾지 못했다면 스택 오버플로에 질문을 남겨 보면 아마 곧 누군가 답변을 줄 것

# 1.1 R 소개하기

## » R의 필요성

- 최신 기술: R 사용자 중에는 통계학, 계량경제학 혹은 다른 관련 학문의 전문가가 많음  
꽤 많은 논문 저자가 새로운 논문을 발표할 때, 논문에서 다룬 최신 기술이 포함된 새로운 패키지들도 함께 공개함

이는 아마도 새로운 통계 검정 방법, 패턴 인식 방법 혹은 더 나은 최적화 알고리즘이 될 수 있음

그것이 무엇이든 간에 이렇듯 R 커뮤니티에는 다른 사람보다 앞서 최첨단 데이터 기술을 실제 문제에 적용해 보고, 기능을 개선시키며 그 가능성을 보여 줄 수 있는 특권이 주어진 셈임



## 1.3 R 설치하기

### » R 설치하기

- R을 설치하려면 먼저 R 공식 사이트(<https://www.r-project.org>)에 접속하여 R을 내려받아야 함

<https://cran.r-project.org/mirrors.html>

▼ 그림 1-1 가까운 위치의 미러 서버 선택

Korea

<https://ftp.harukasan.org/CRAN/>

<https://cran.yu.ac.kr/>

<https://cran.seoul.go.kr/>

<http://healthstat.snu.ac.kr/CRAN/>

<https://cran.biodisk.org/>

<http://cran.biodisk.org/>

# 1.3 R 설치하기

▼ 그림 1-2 해당하는 운영 체제 선택

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.

▼ 그림 1-3 install R for the first time 클릭

R for Windows

Subdirectories:

<a href="#">base</a>	Binaries for base distribution. This is what you want to <b>install R for the first time.</b>
<a href="#">contrib</a>	Binaries of contributed CRAN packages (for R >= 2.13.x; managed by Uwe Ligges). The Windows services and corresponding environment and make variables.
<a href="#">old_contrib</a>	Binaries of contributed CRAN packages for outdated versions of R (for R < 2.13.x; managed by Uwe Ligges).
<a href="#">Rtools</a>	Tools to build R and R packages. This is what you want to build your own packages on.

# 1.3 R 설치하기

## » R 설치하기

- 현재 시점에서 가장 최신 버전은 3.6.2
- 이 책에서 소개하는 예제들은 모두 윈도우와 리눅스에서 이 버전을 기준으로 함
- 대부분은 이전 버전이나 다른 운영 체제와 큰 차이가 없을 것

▼ 그림 1-4 윈도우용 R 설치 파일

[Download R 3.6.2 for Windows](#) (83 megabytes, 32/64 bit)

[Installation and other instructions](#)

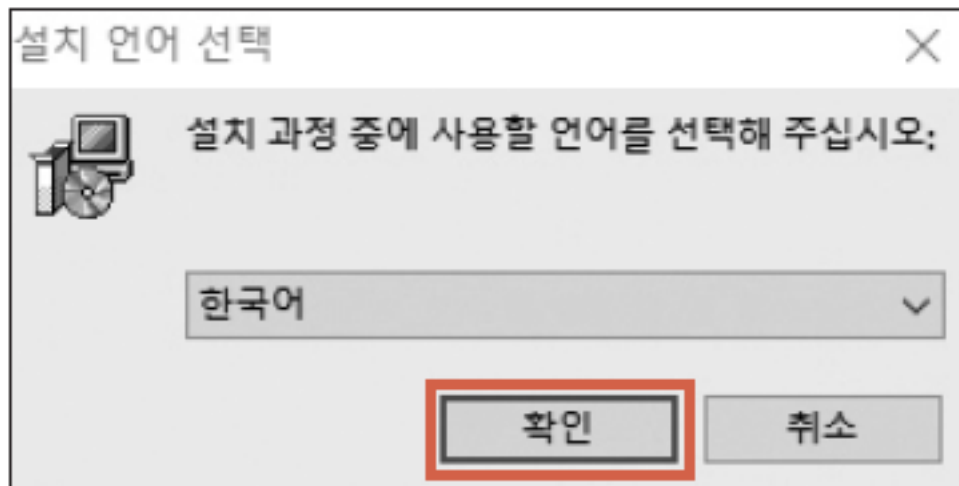
[New features in this version](#)

# 1.3 R 설치하기

## » R 설치하기

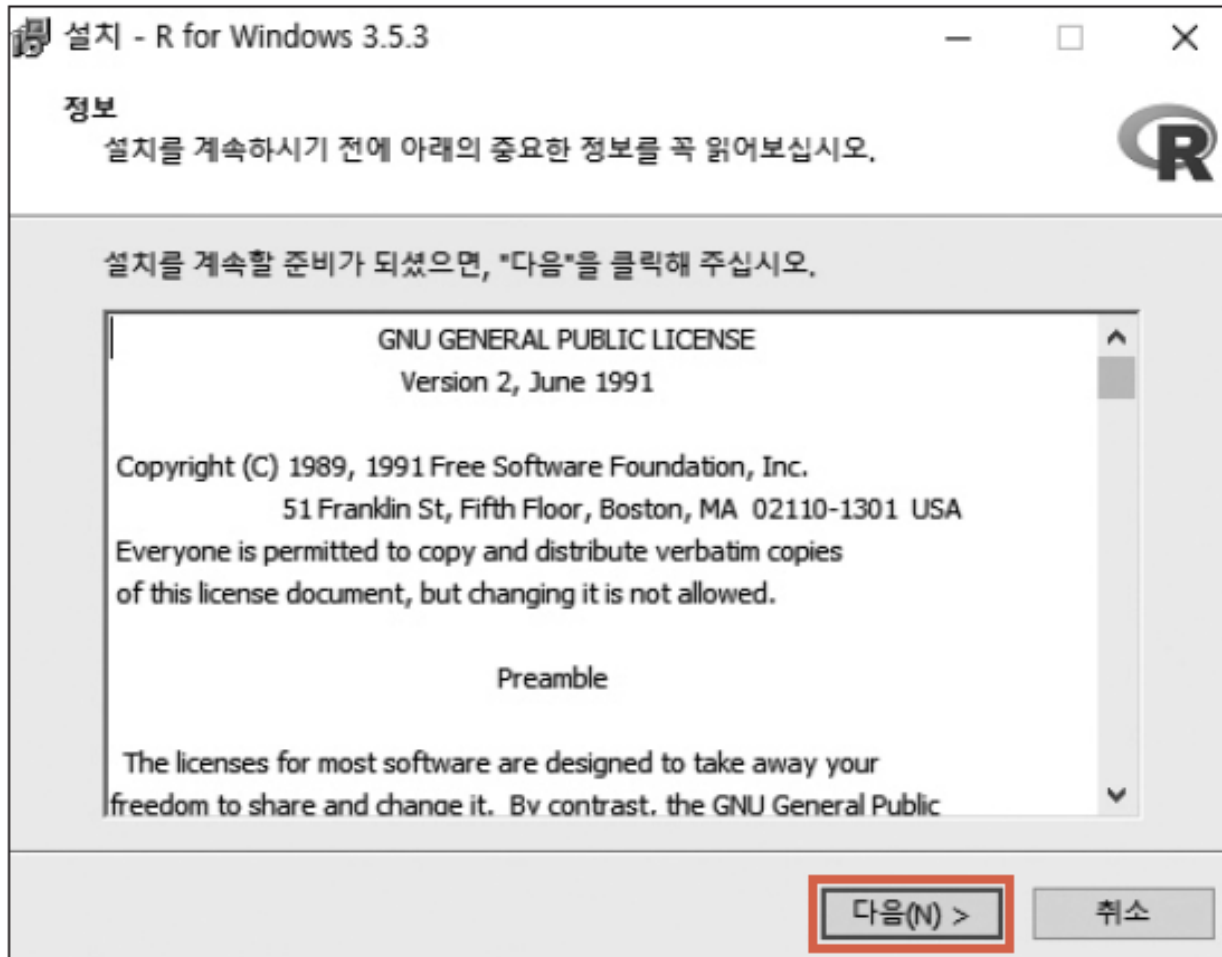
- 원도 사용자라면 가장 최신 버전의 설치 파일을 내려받음
- 내려받은 파일을 실행하여 R을 설치함
- 설치 과정 자체는 무난하지만, 아마도 많은 사용자는 이후 몇 단계를 거치면서 어려움을 겪을 수 있음

▼ 그림 1-5 설치할 언어 선택



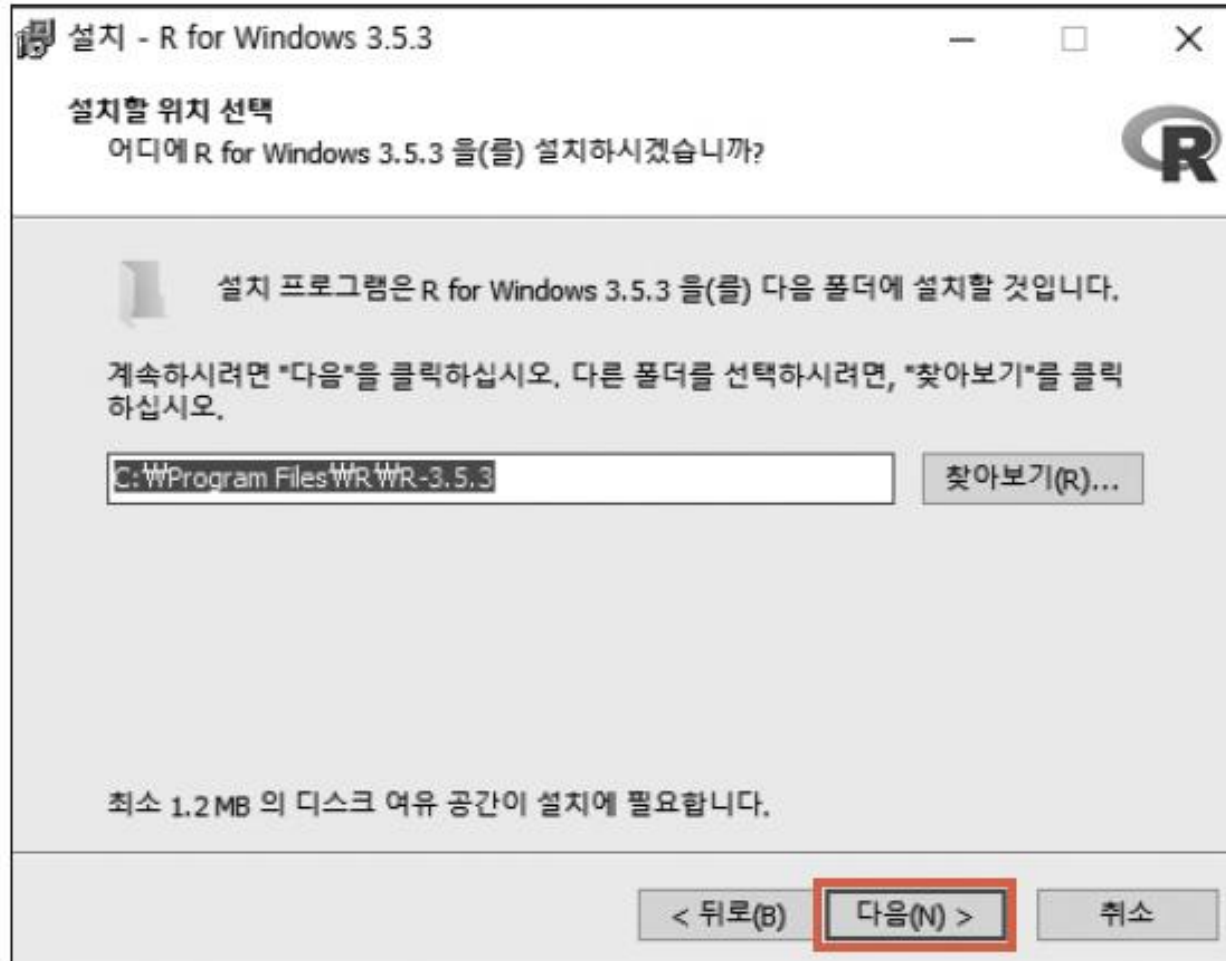
# 1.3 R 설치하기

▼ 그림 1-6 설치 전 중요 정보



# 1.3 R 설치하기

▼ 그림 1-7 설치 폴더 선택



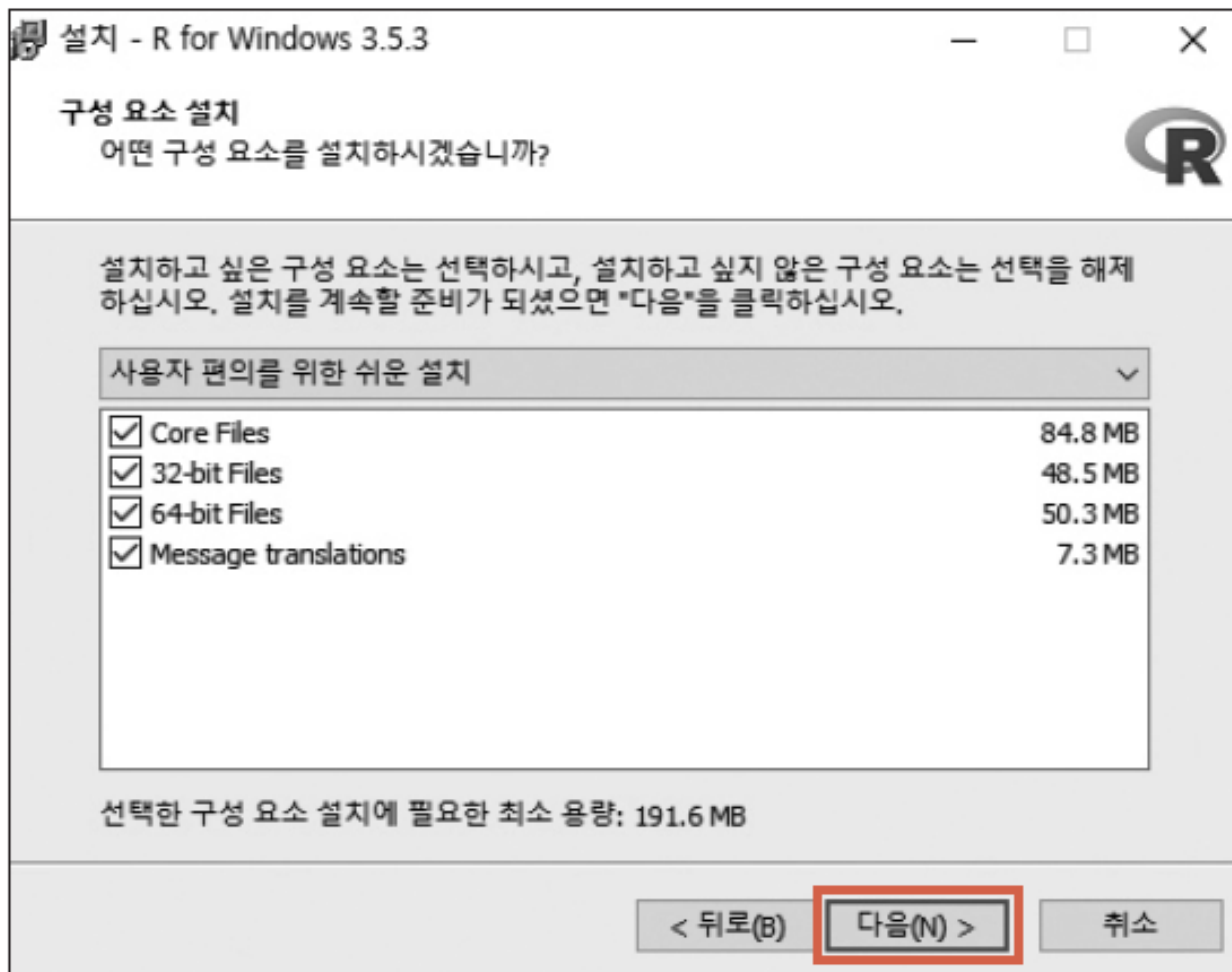
# 1.3 R 설치하기

## » R 설치하기

- 구성 요소 설치 화면에는 설치 가능한 네 가지 구성 요소가 나옴
- 여기서 Core Files는 R의 핵심 라이브러리를 의미
- Message translations는 지원 가능한 언어로, 경고나 오류 메시지를 다양한 언어 버전으로 번역해 주는 기능을 제공함
- 64비트 R은 32비트보다 단일 프로세스에서 더 많은 데이터를 처리할 수 있음
- 최근 몇 년 사이 구매한 컴퓨터를 사용하고 있다면, 기본적으로 64비트 프로그램을 지원하는 운영 체제를 탑재했을 가능성이 높음
- 기본 설정은 64-bit Files로 되어 있음
- 32비트 운영 체제를 사용한다면 64비트 시스템을 설치하지 않는 이상은 안타깝지만 64비트 R은 사용할 수 없음

# 1.3 R 설치하기

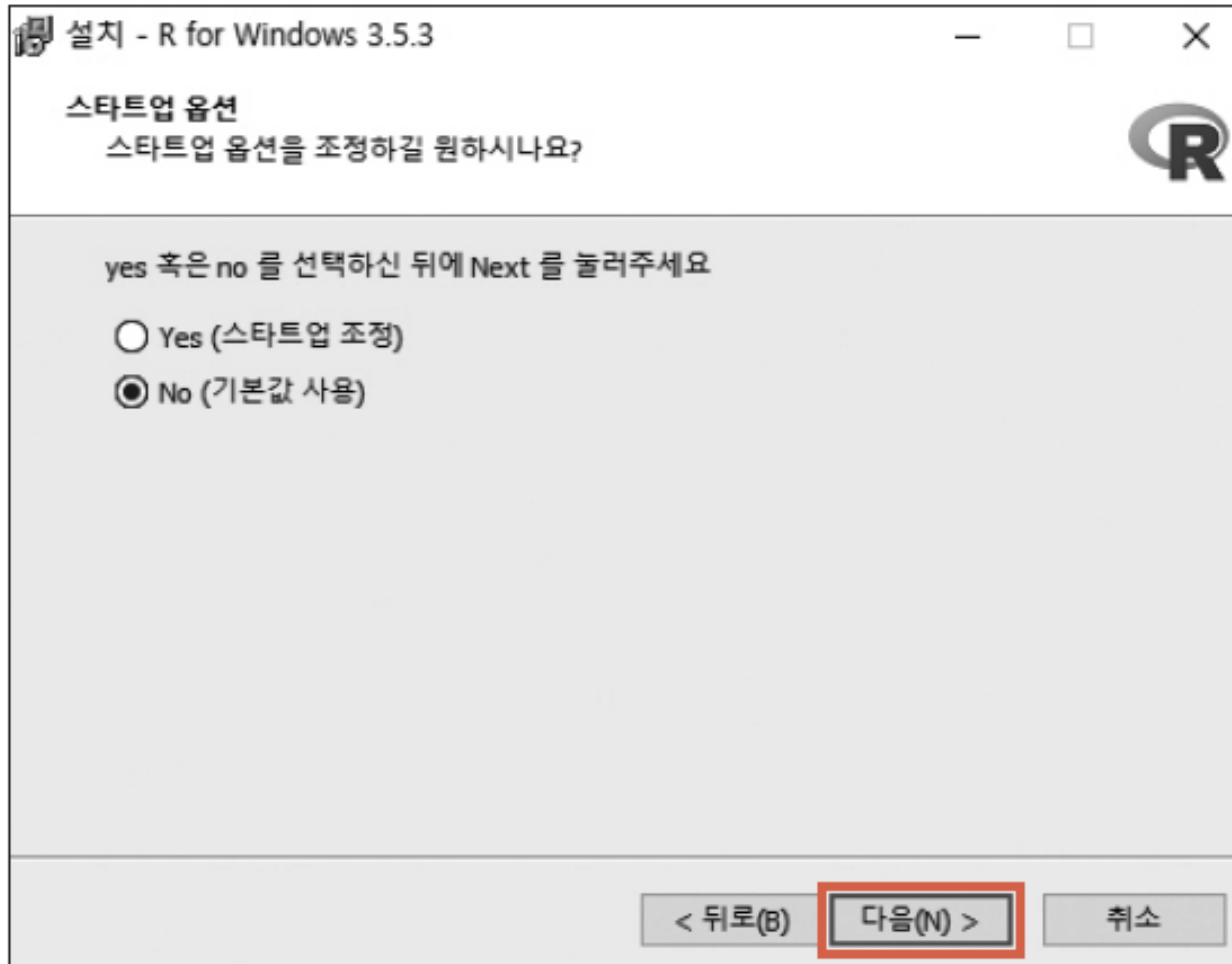
▼ 그림 1-8 구성 요소 설치





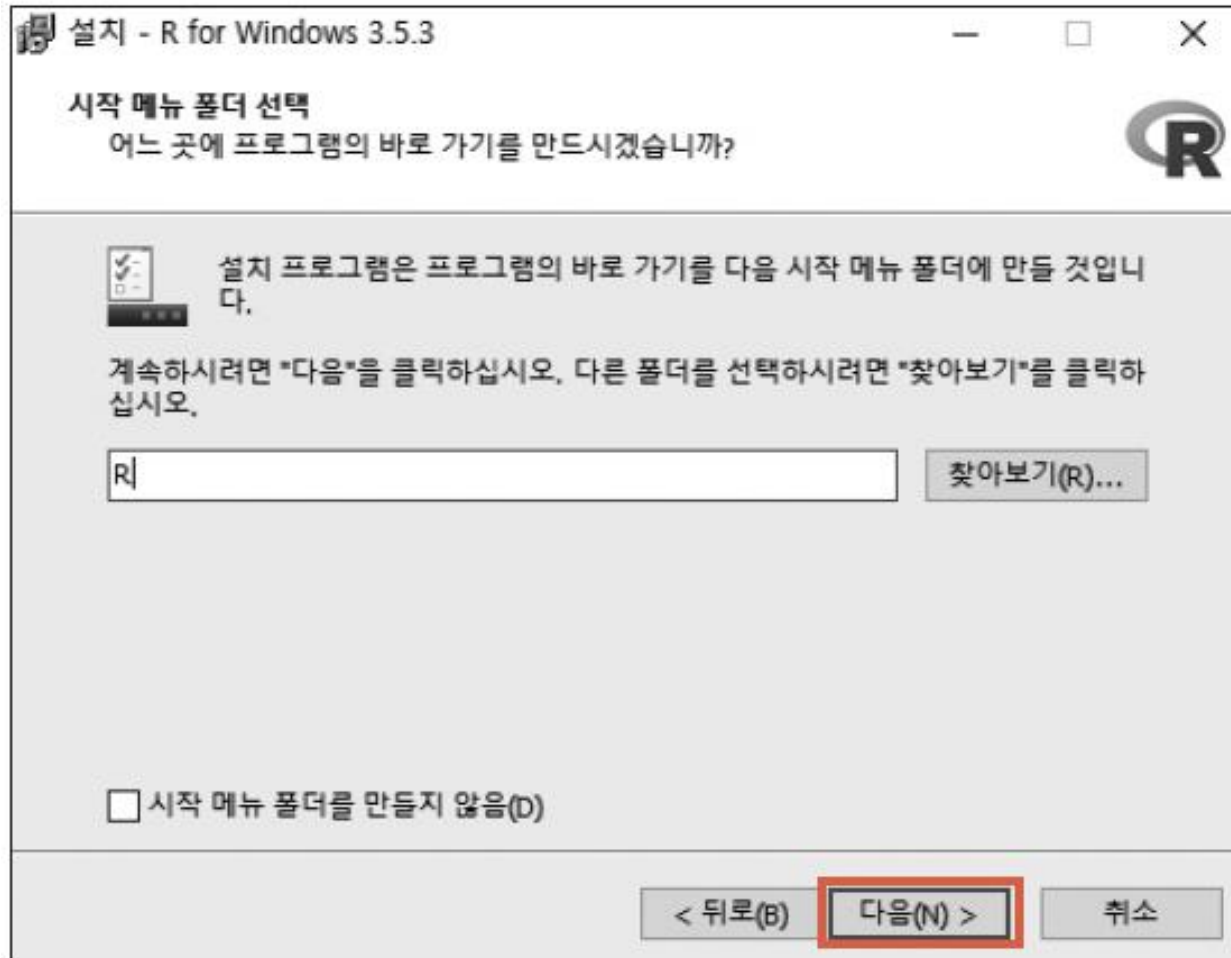
# 1.3 R 설치하기

▼ 그림 1-9 스타트업 옵션 선택



# 1.3 R 설치하기

▼ 그림 1-10 바로 가기 선택



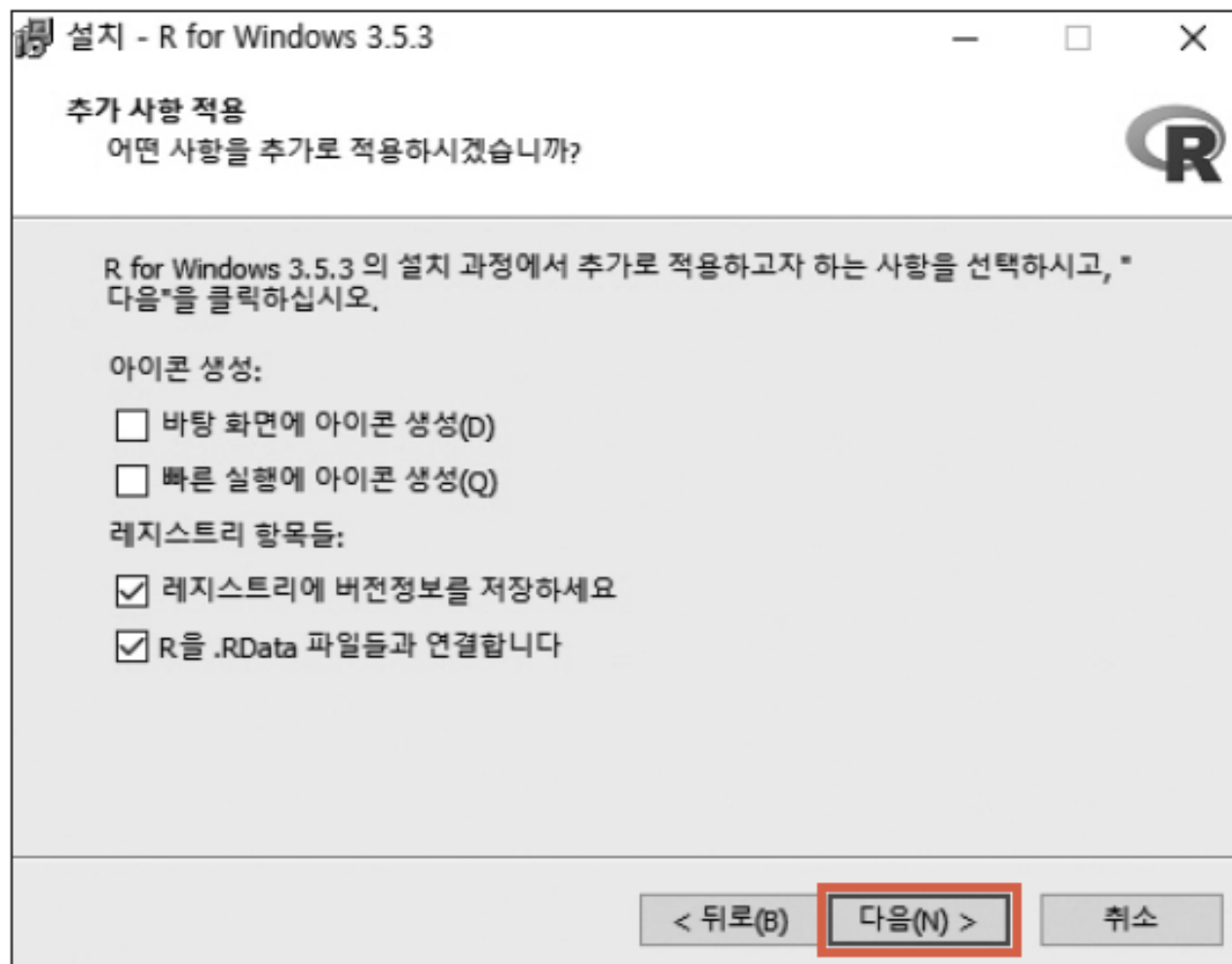
# 1.3 R 설치하기

## » R 설치하기

- 조금 헛갈릴 수 있는 또 다른 옵션은 레지스트리에 R 버전 숫자를 저장할지 말지를 결정하는 부분임
- 이 옵션은 R의 어떤 버전이 이미 설치되었는지 다른 프로그램이 쉽게 파악할 수 있게 도움
- 한 버전의 R만 사용하는 것이 확실하다면 그냥 기본 설정으로 진행해도 좋음

# 1.3 R 설치하기

▼ 그림 1-11 추가 사항 선택

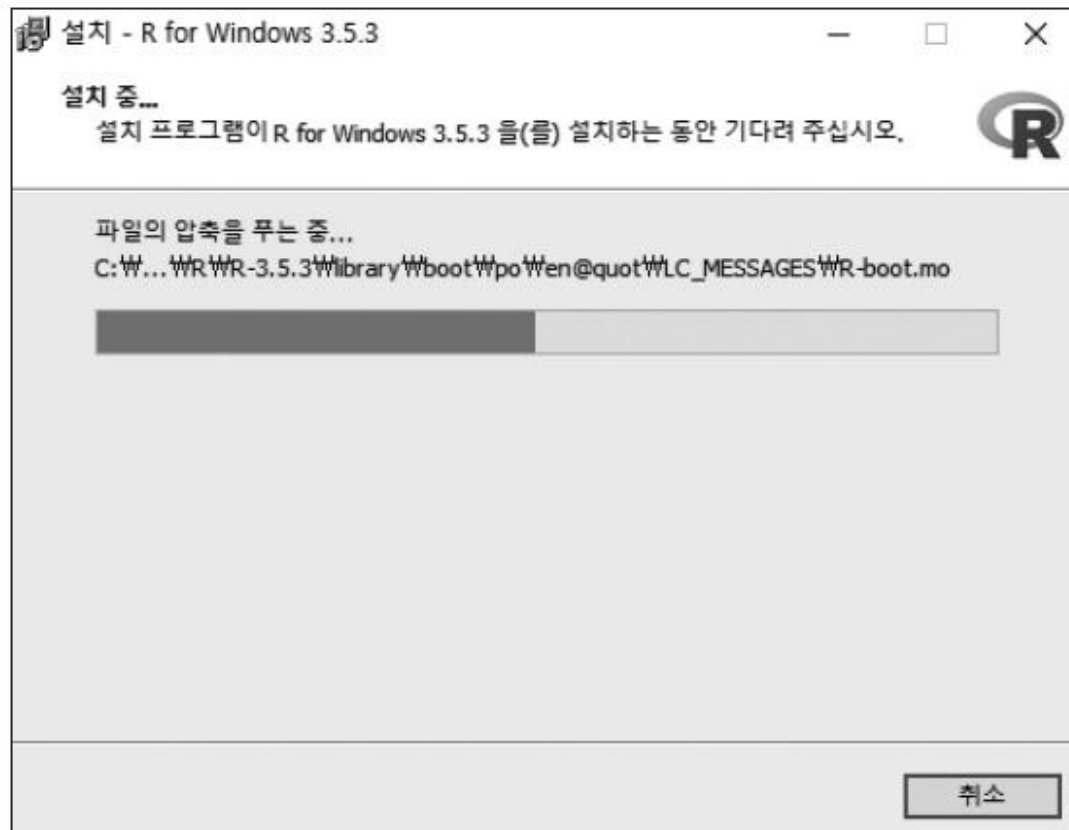


# 1.3 R 설치하기

## » R 설치하기

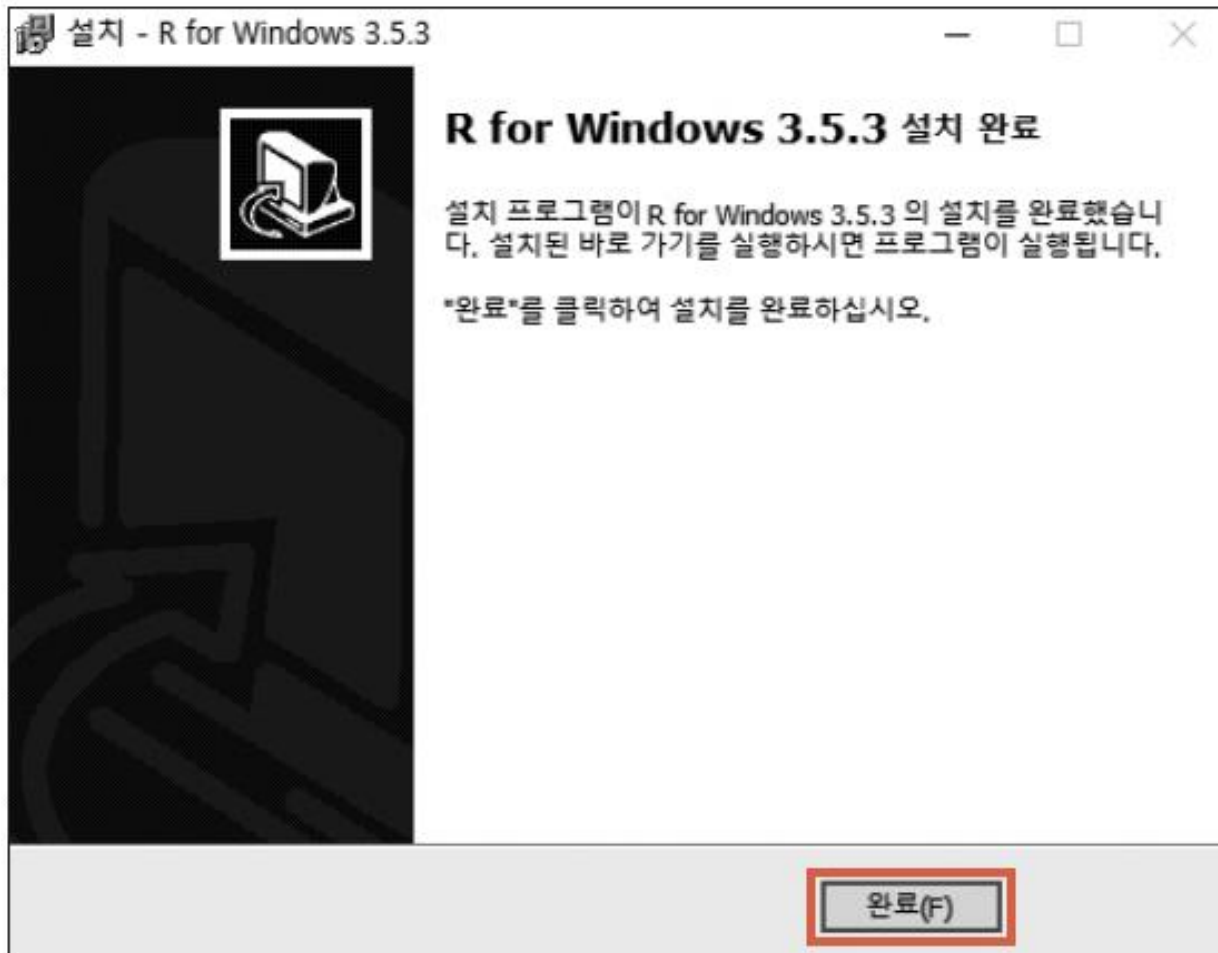
- 이 단계를 지나면 실제로 설치를 시작하면서 필요한 파일을 하드 드라이브에 복사함

▼ 그림 1-12 R 설치



# 1.3 R 설치하기

▼ 그림 1-13 R 설치 완료



# 1.3 R 설치하기

## » R 설치하기

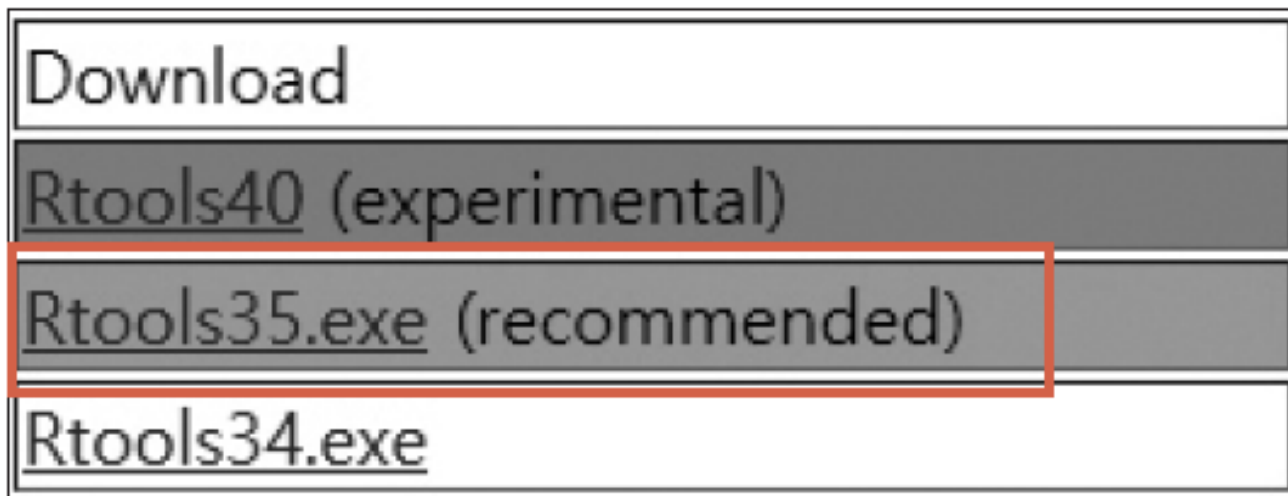
- 설치 과정에서 프로그램 바로 가기를 만들었다면 두 가지 R 관련 바로 가기를 볼 수 있을 것
- R은 명령 프롬프트를 실행하고, R GUI는 굉장히 단순한 GUI를 실행함
- 이 방법으로 R을 당장 사용할 수도 있지만, 꼭 이러한 방법으로 사용해야 하는 것은 아님
- 개인적으로는 R 스크립트를 작성하고, 디버깅할 때는 RStudio를 사용하길 추천함
- RStudio가 아무리 강력한 도구라고 해도 R을 제대로 설치하지 않았다면 동작하지 않을 것
- 다시 말해 R이 백엔드라면 RStudio는 이 백엔드를 더 잘 활용하도록 돕는 프론트엔드라고 할 수 있음

# 1.3 R 설치하기

## » R 설치하기

- 윈도우 사용자라면 Rtools(<http://cran.rstudio.com/bin/windows/Rtools>)를 설치함
- C++ 코드를 작성하고 컴파일해서 R에서 사용하게 돕는 도구
- C/C++ 코드가 포함된 패키지를 설치할 때도 필요함

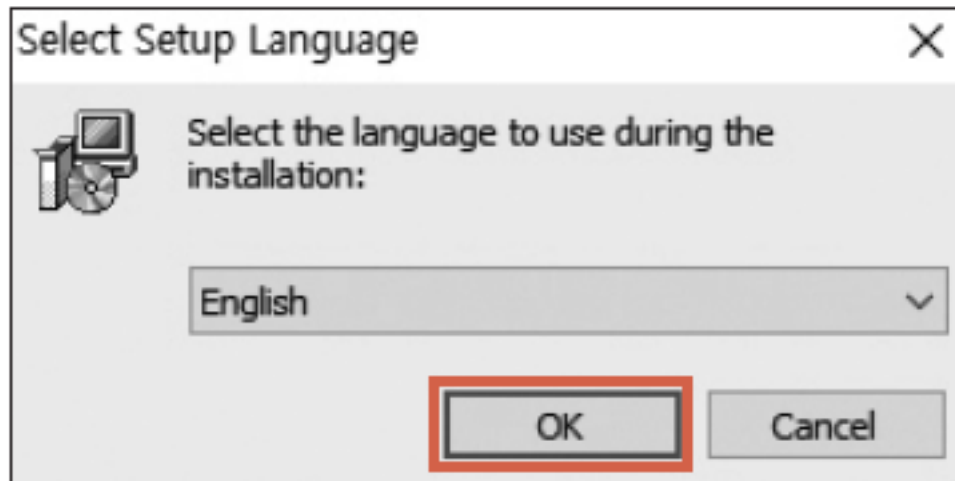
▼ 그림 1-14 Rtools 내려받기





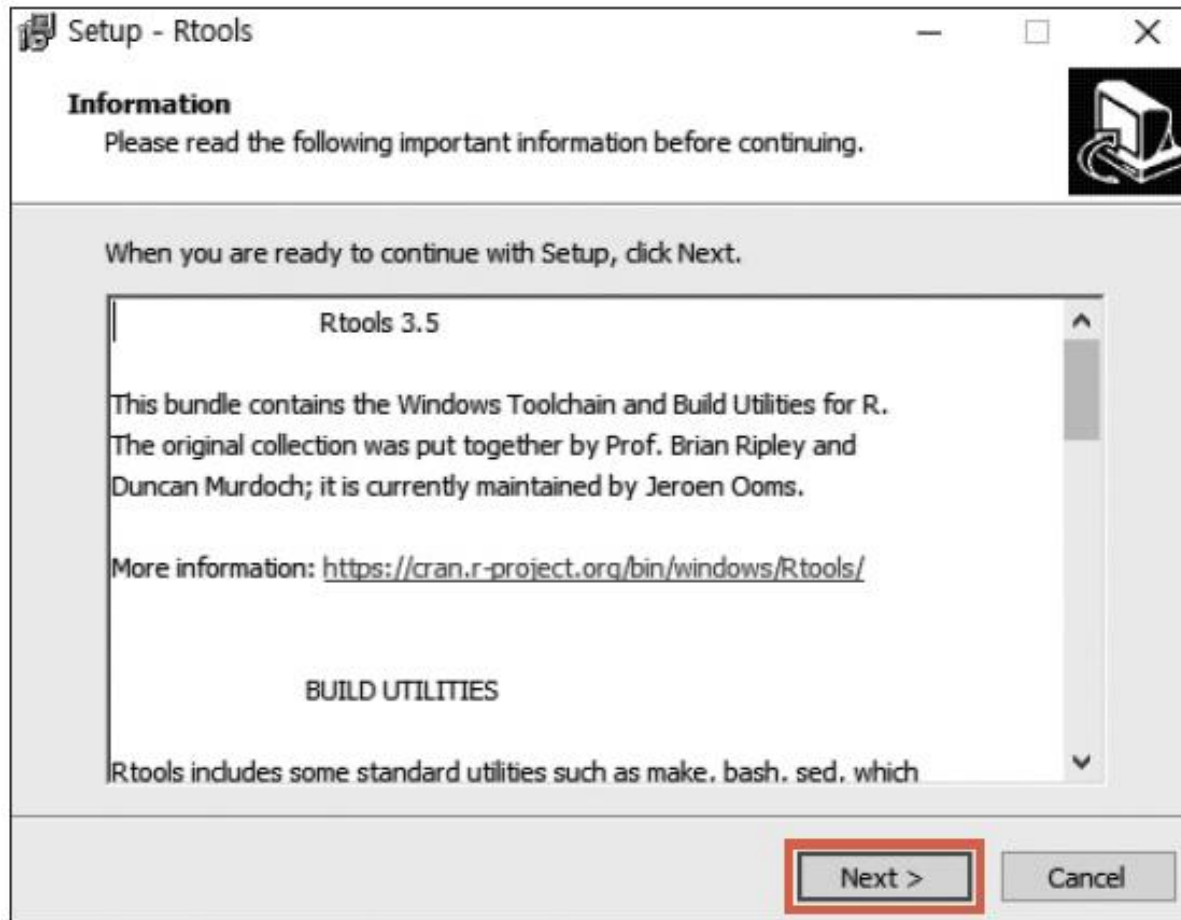
## 1.3 R 설치하기

▼ 그림 1-15 설치 언어 선택



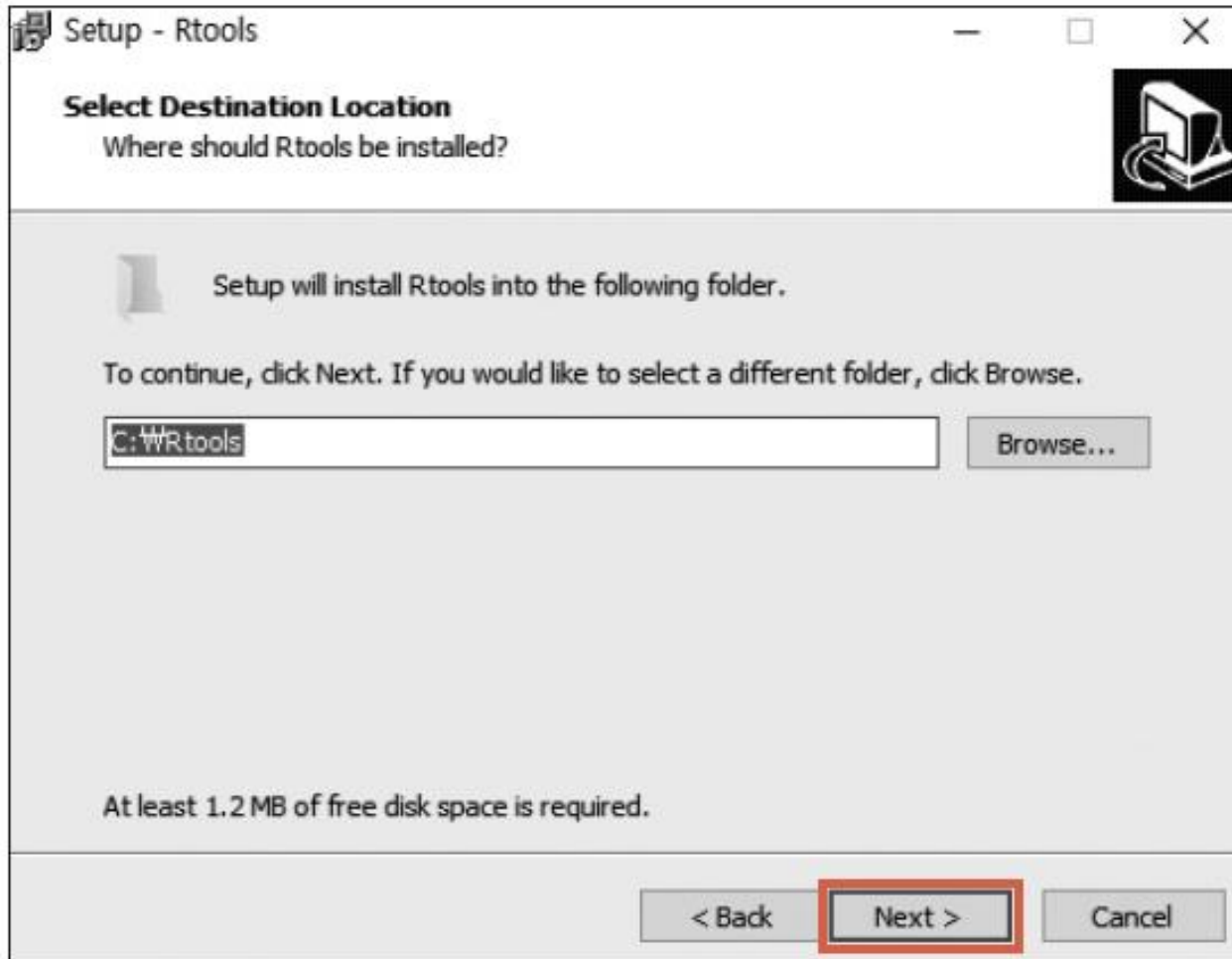
# 1.3 R 설치하기

▼ 그림 1-16 설치 전 중요 정보



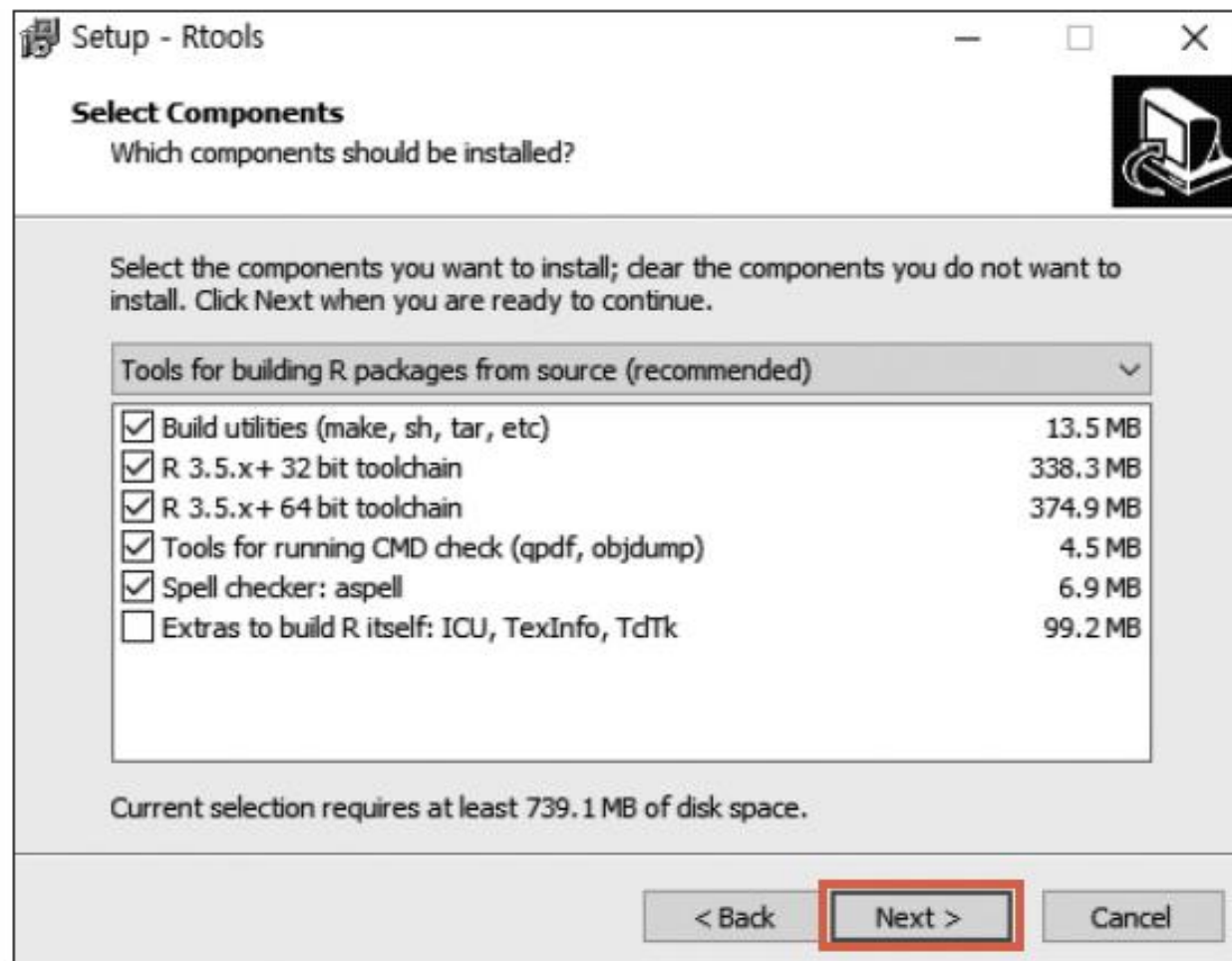
# 1.3 R 설치하기

▼ 그림 1-17 설치 폴더 선택



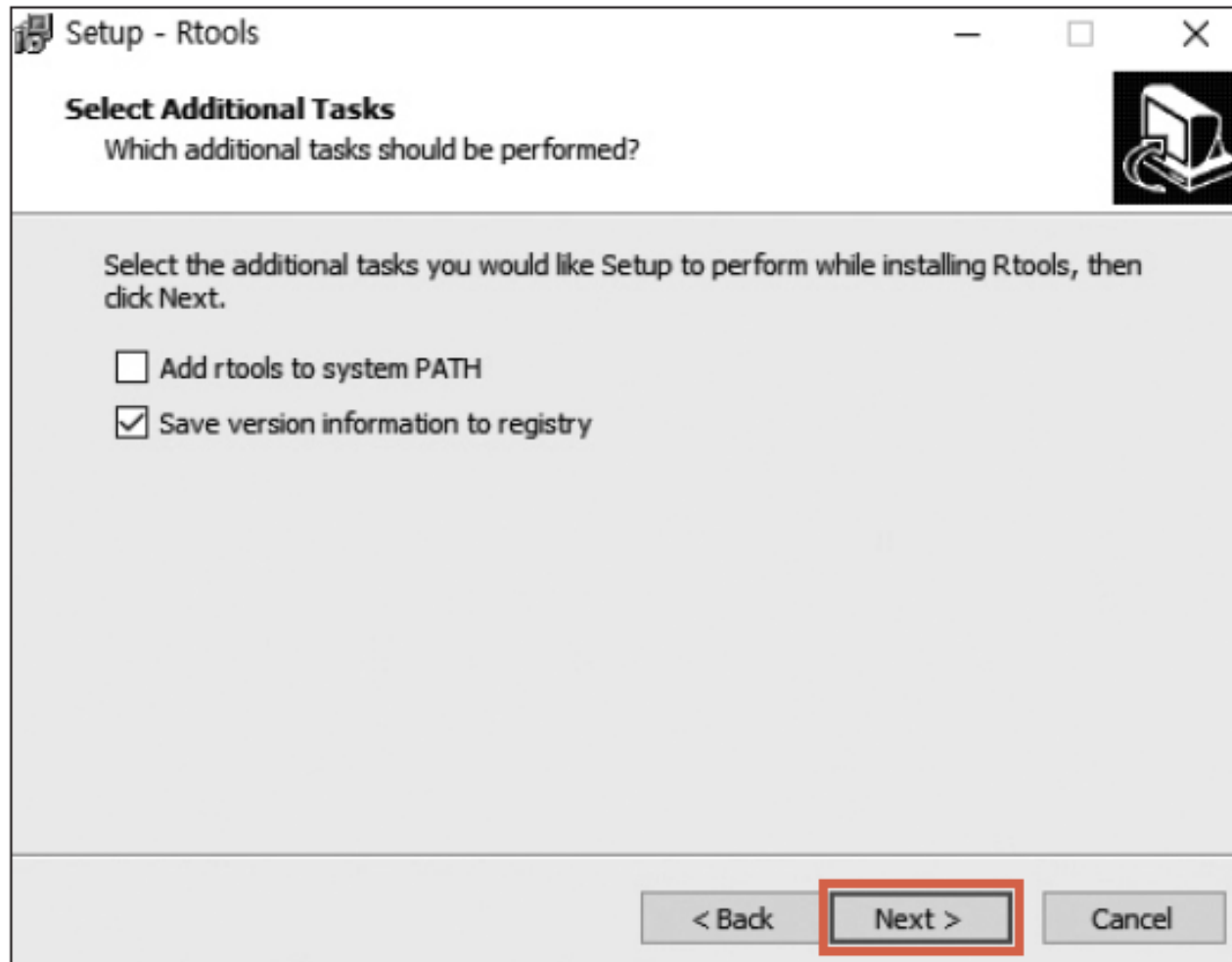
# 1.3 R 설치하기

▼ 그림 1-18 구성 요소 설치



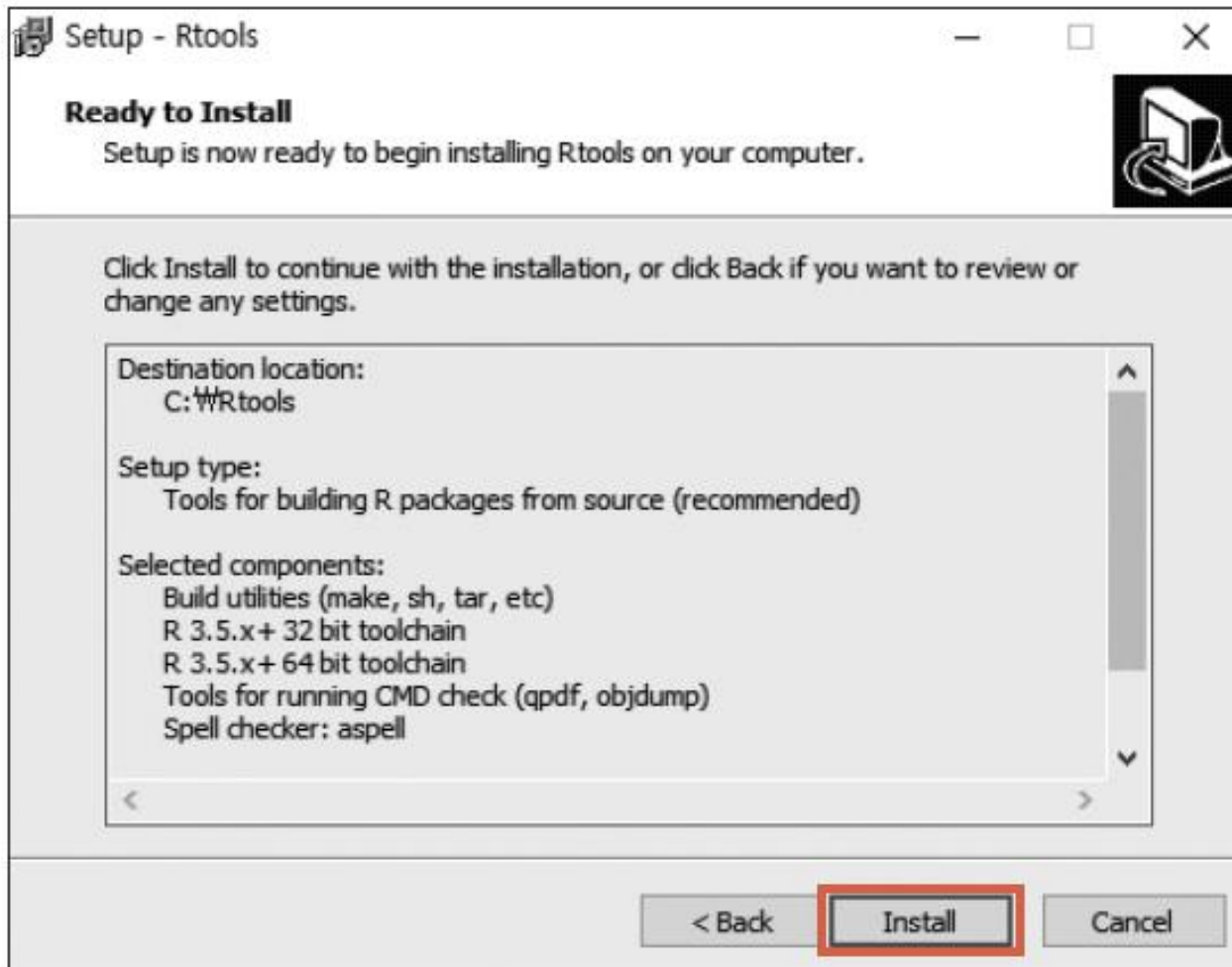
# 1.3 R 설치하기

▼ 그림 1-19 추가 사항 선택



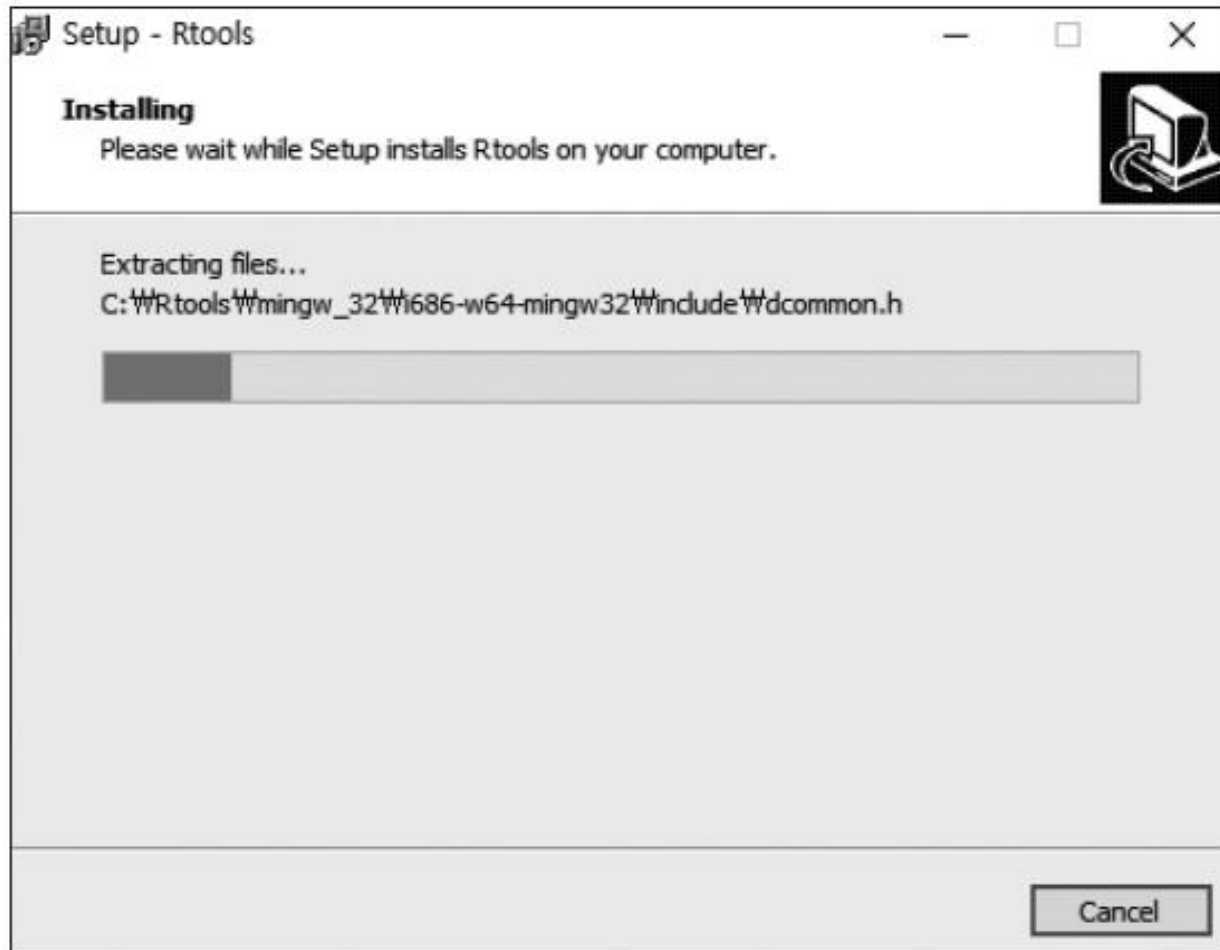
# 1.3 R 설치하기

▼ 그림 1-20 설치 구성 요소 및 설정 확인



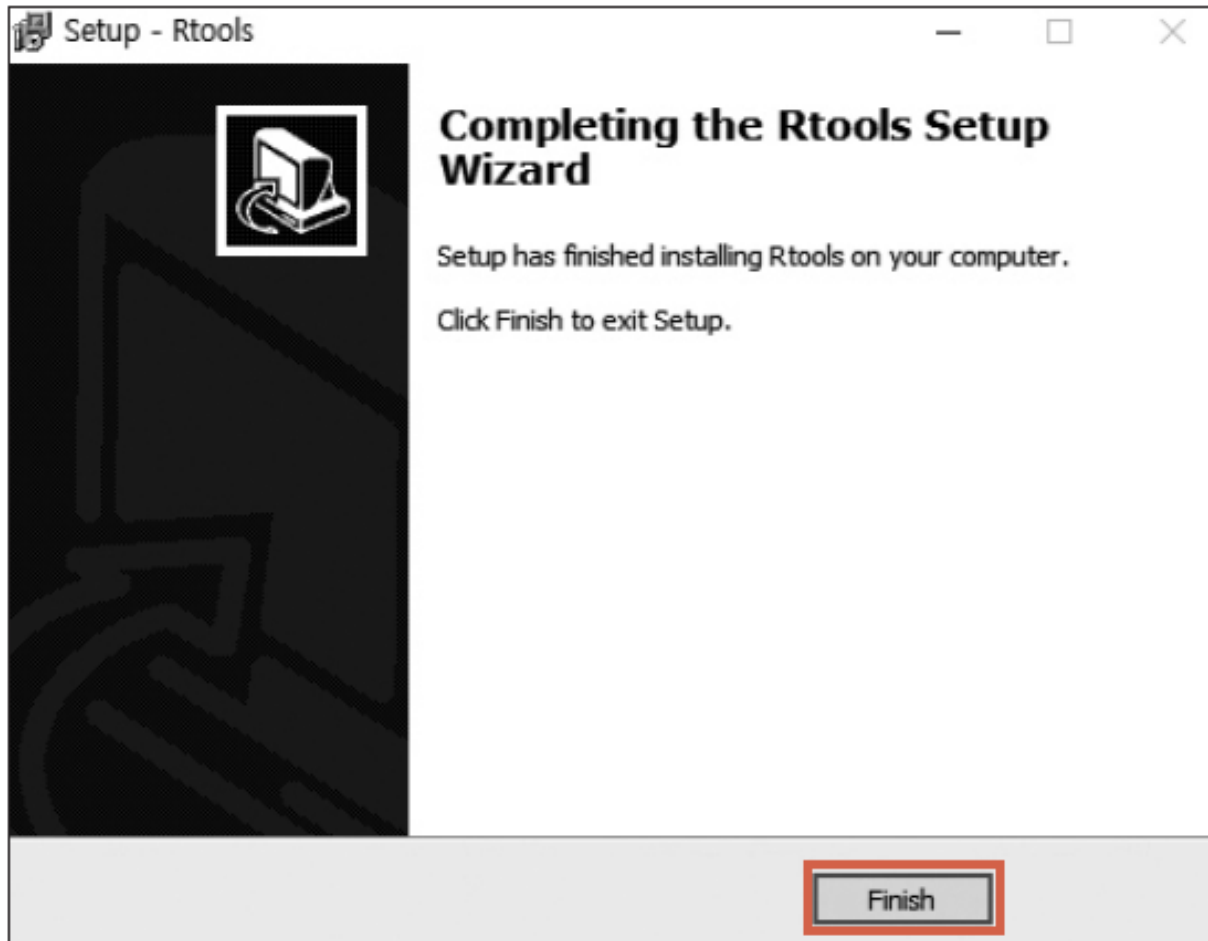
# 1.3 R 설치하기

▼ 그림 1-21 Rtools 설치



# 1.3 R 설치하기

▼ 그림 1-22 Rtools 설치 완료





# 1.4 RStudio

## » RStudio

- RStudio는 R 프로그래밍을 위한 강력한 사용자 인터페이스
- 무료 오픈 소스이면서 윈도우, macOS, 리눅스 등 여러 플랫폼에서 사용할 수 있는 도구
- RStudio에는 데이터 분석과 시각화에 생산성을 엄청나게 향상시켜 주는 강력한 기능들이 있음
- 구문 강조, 자동 완성, 멀티 탭 뷰, 파일 관리, 그래픽 뷰포트, 패키지 관리, 통합 도움말 뷰어, 코드 포매팅, 버전 관리, 인터랙티브한 디버깅 등 많은 기능을 지원함
- 가장 최신 Rstudio를 내려받음(<https://www.rstudio.com/products/rstudio/download/>)
- 새로운 기능을 포함한 프리뷰 버전을 사용하고 싶다면  
<https://www.rstudio.com/products/rstudio/download/preview/>에서 내려받음
- RStudio에는 R이 포함되어 있지 않다는 점을 기억함
- RStudio를 사용하려면 먼저 R을 설치해야 함



# 1.4 RStudio

▼ 그림 1-23 사용 목적에 따른 다양한 버전의 RStudio(이 책의 모든 내용은 무료 버전으로 구현 가능)

RStudio Desktop	RStudio Desktop	RStudio Server	RStudio Server Pro
Open Source License	Commercial License	Open Source License	Commercial License
<b>Free</b>	<b>\$995</b> /year	<b>Free</b>	<b>\$4,975</b> /year (5 Named Users)
<b>DOWNLOAD</b>	<b>BUY</b>	<b>DOWNLOAD</b>	<b>BUY</b>
<a href="#">Learn more</a>	<a href="#">Learn more</a>	<a href="#">Learn more</a>	<a href="#">Evaluation</a>   <a href="#">Learn more</a>

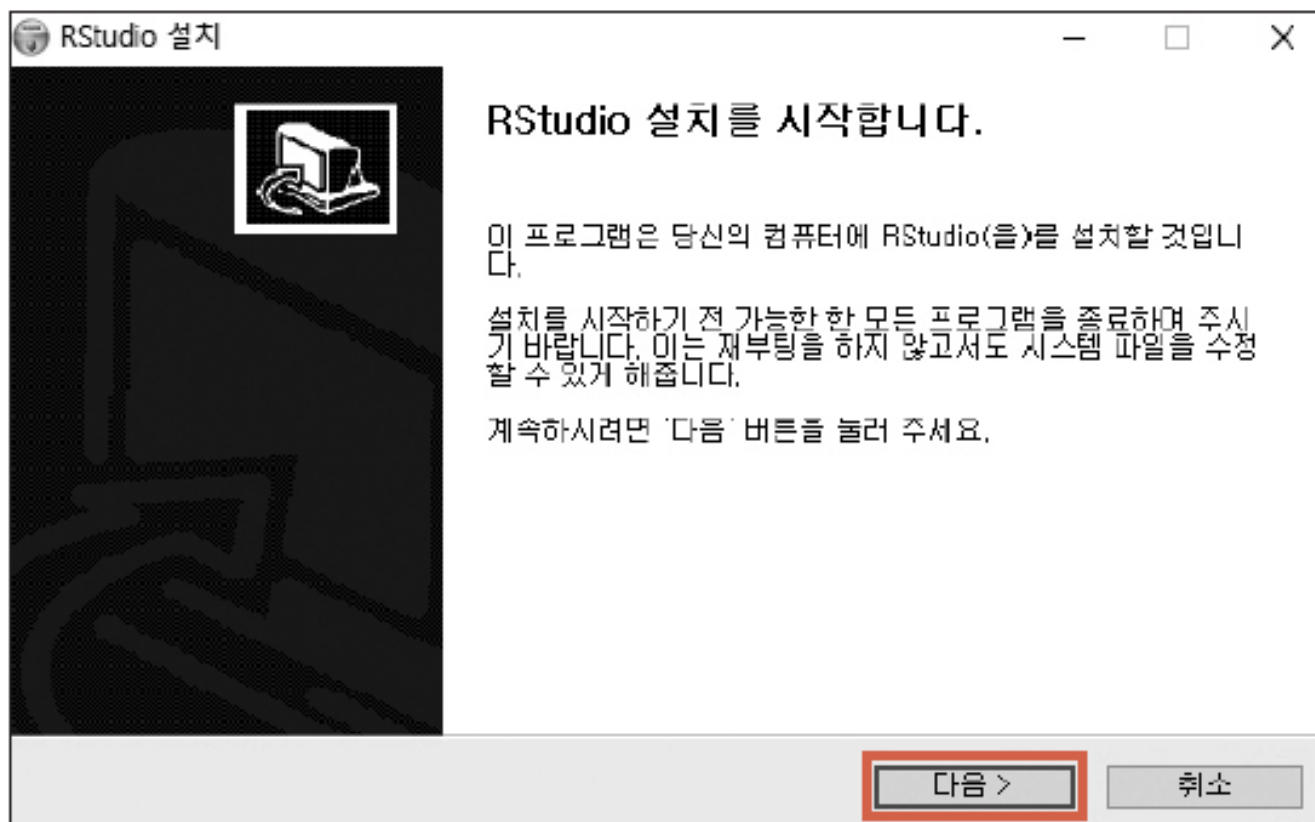
# 1.4 RStudio

▼ 그림 1-24 해당 운영 체제 선택

OS	Download
Windows 10/8/7	 RStudio-1.2.5033.exe
macOS 10.12+	 RStudio-1.2.5033.dmg

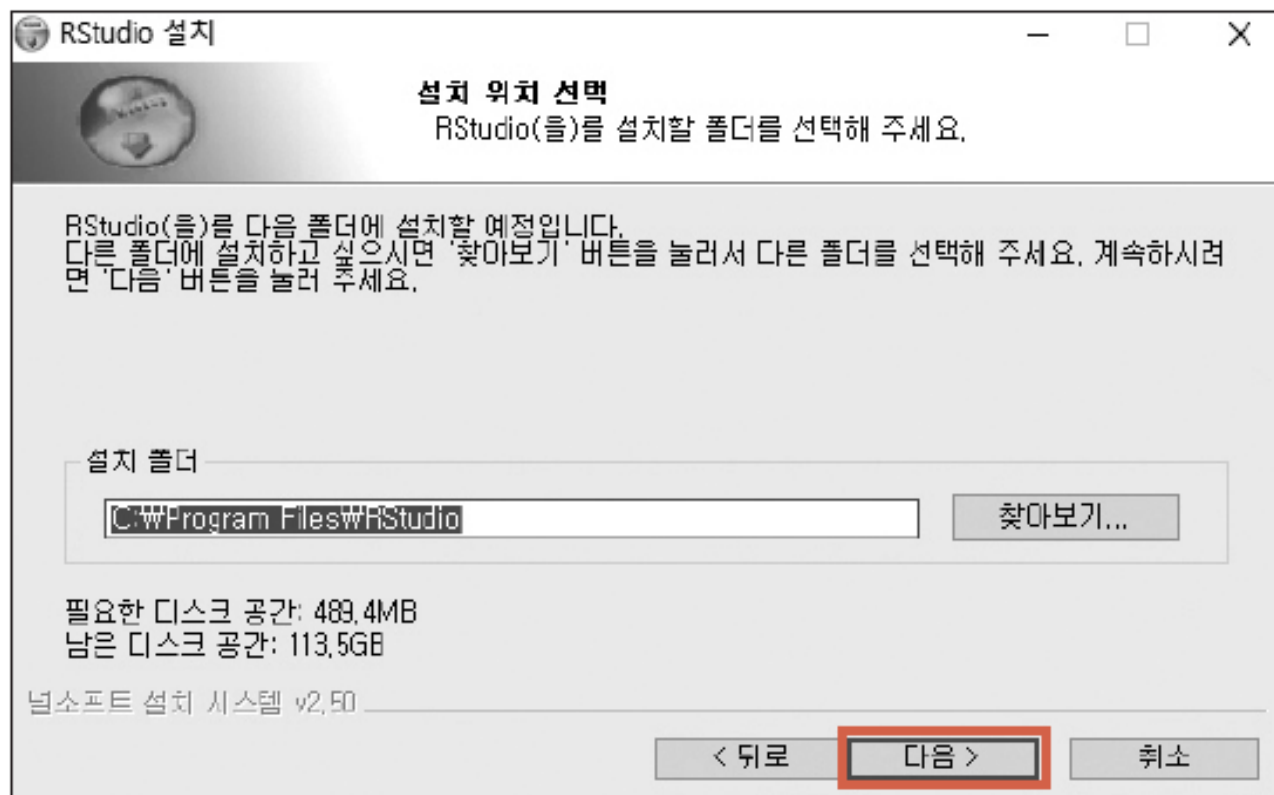
# 1.4 RStudio

▼ 그림 1-25 RStudio 설치 동의



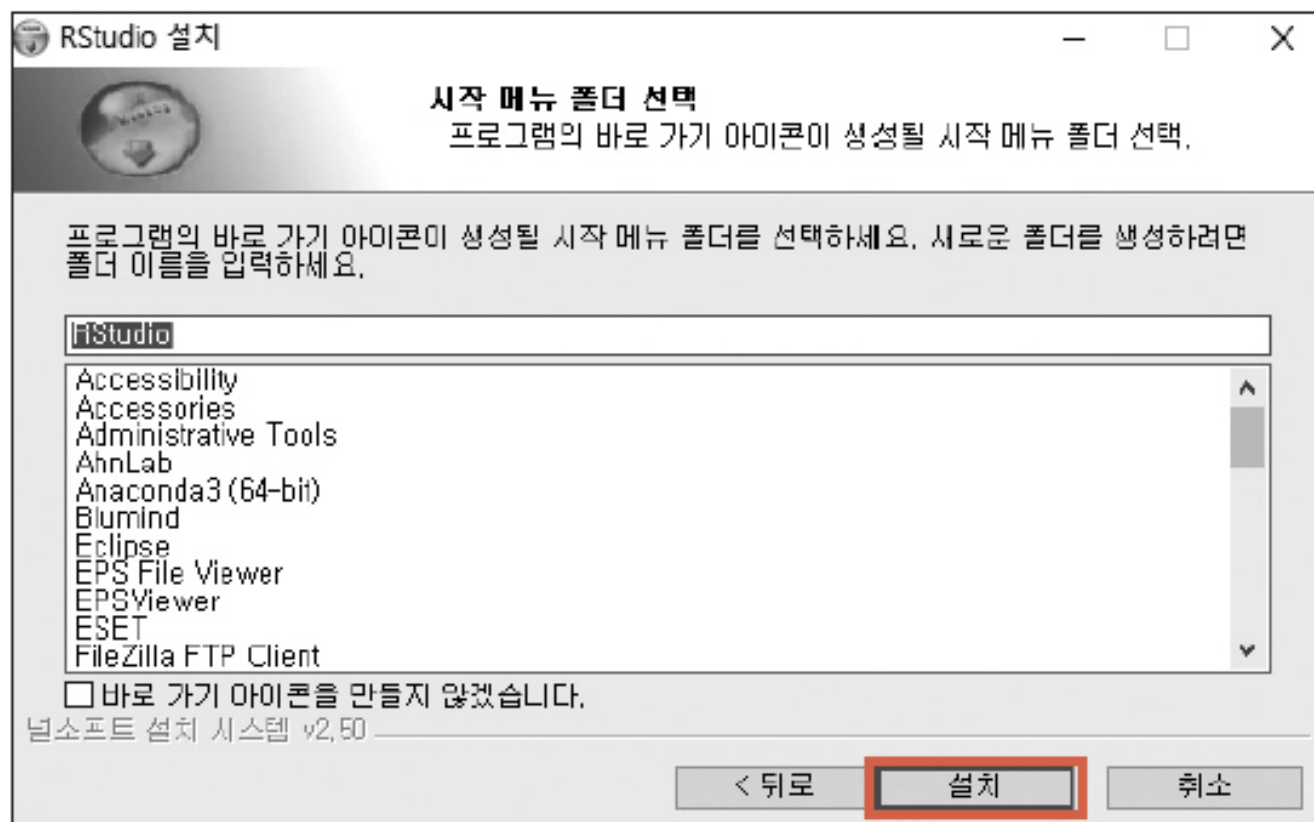
# 1.4 RStudio

▼ 그림 1-26 설치 폴더 선택



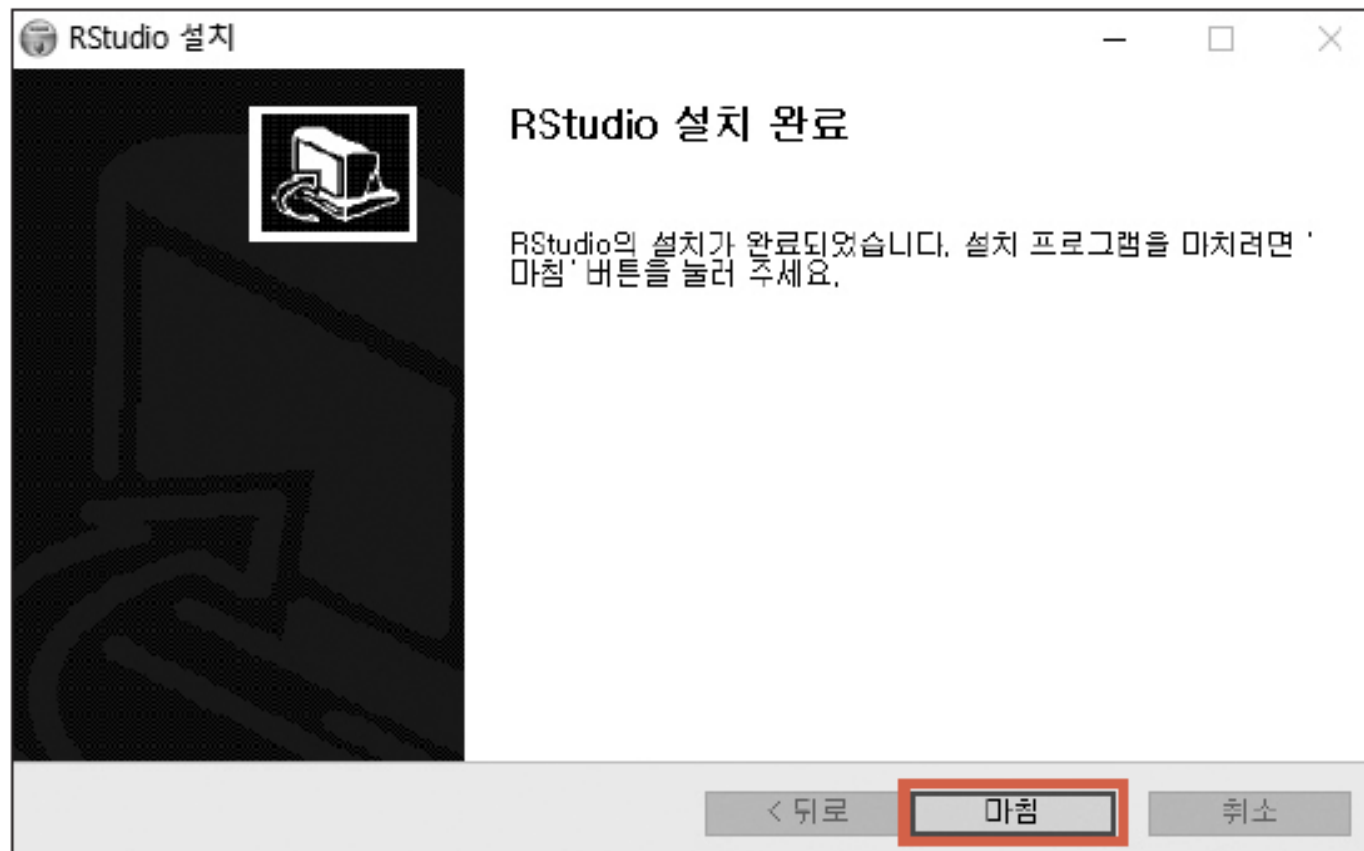
# 1.4 RStudio

▼ 그림 1-27 바로 가기 선택



# 1.4 RStudio

▼ 그림 1-28 RStudio 설치 완료



# 1.4 RStudio

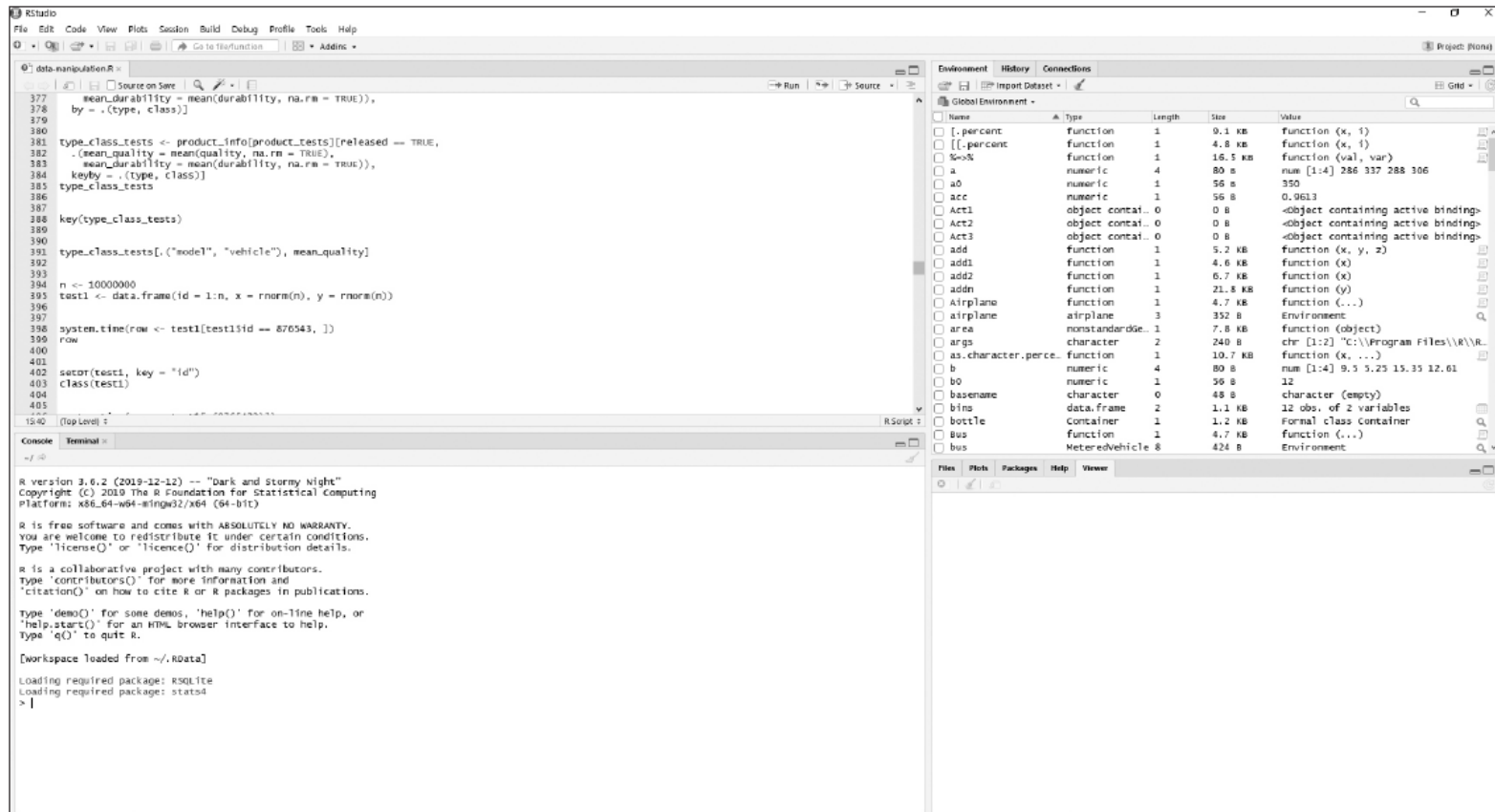
## » RStudio의 사용자 인터페이스

- 다음은 윈도우 운영 체제에서 실행한 RStudio 모습
- macOS나 리눅스를 사용한다고 해도 전체적인 모습은 거의 차이가 없을 것



# 1.4 RStudio

▼ 그림 1-29 실행한 RStudio



# 1.4 RStudio

## » RStudio의 사용자 인터페이스

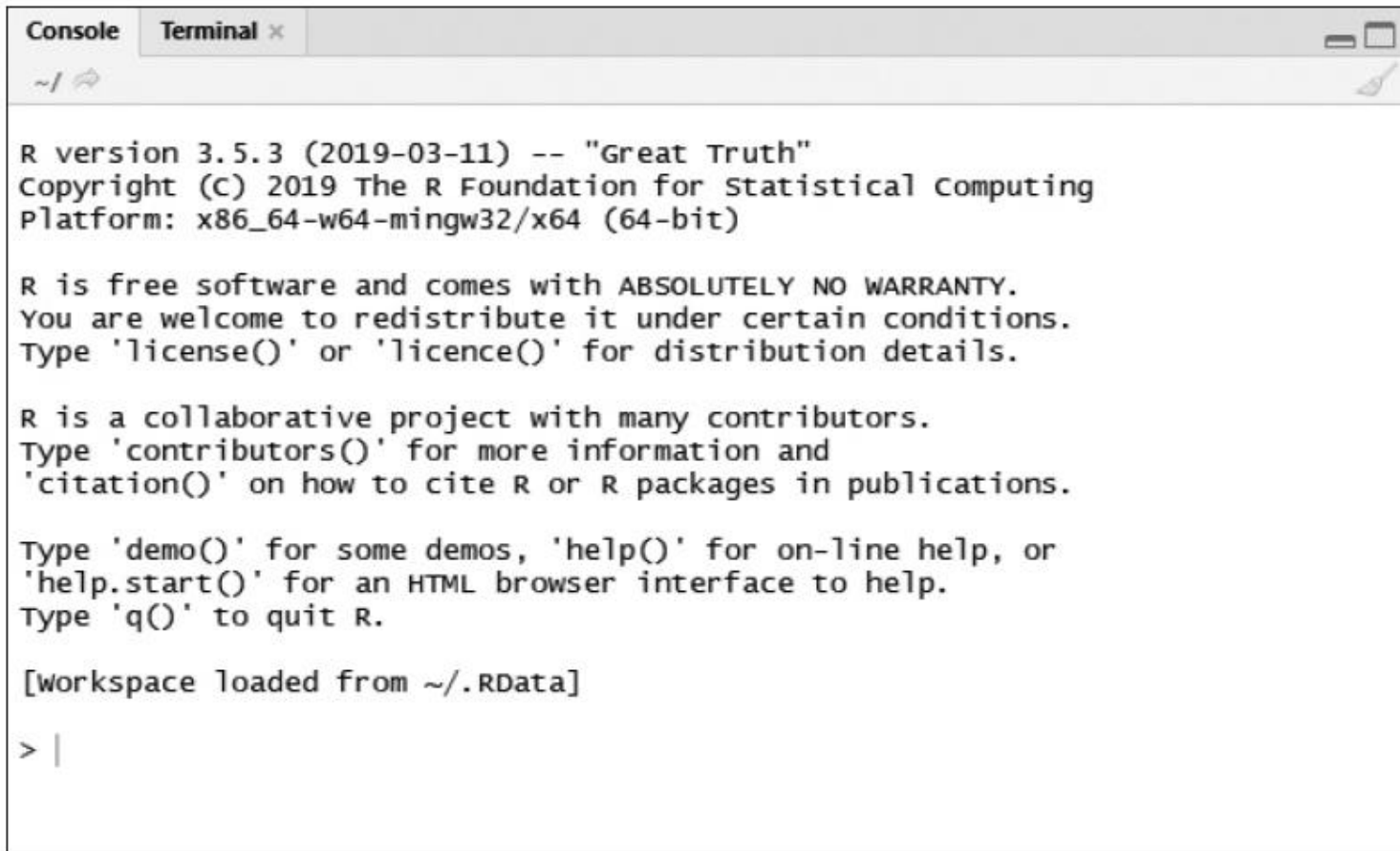
- 얼핏 보아도 알 수 있듯이 메인 창은 여러 부분으로 구성됨
- 각 부분을 분할 창(pane)이라고 하며, 각각 다른 기능을 수행함
- 각 창은 데이터 분석에 맞게 디자인함

### 콘솔 창

- 다음은 RStudio에 탑재된 R 콘솔 창을 보여 줌
- 콘솔 창은 대부분 명령 프롬프트나 터미널과 동일하게 동작함
- 사실 콘솔 창에 명령어를 입력하면 RStudio는 명령어 요청을 R 엔진으로 보냄
- 다시 말해 R 엔진에서 모든 명령어를 처리함
- RStudio는 사용자에게서 입력을 받아 R 엔진에 이것을 전달하고, 그 결과를 다시 출력하는 중간 역할을 함

# 1.4 RStudio

▼ 그림 1-30 콘솔 창



```
Console Terminal x
~/
R version 3.5.3 (2019-03-11) -- "Great Truth"
Copyright (c) 2019 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.

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]

> |
```

# 1.4 RStudio

## » RStudio의 사용자 인터페이스

- 콘솔 창에서 명령어를 수행하거나, 변수를 설정하거나, 통계량을 계산하려고 어떤 표현식의 값을 출력함
- 데이터를 변환하고 차트를 만드는 일을 아주 손쉽게 할 수 있음

# 1.4 RStudio

## » RStudio의 사용자 인터페이스

### 편집 창

- 콘솔 창에 직접 명령어를 입력하는 것이 데이터 작업에서 일반적인 방법은 아님
- 그 대신 우리는 스크립트를 작성함
- 이것은 일련의 논리 흐름에 맞게 R 엔진이 수행할 명령어들을 파일에 적어 놓은 것임
- 이 편집기는 R 스크립트, 마크다운 문서, 웹 페이지, 다양한 유형의 구성 파일 및 C++ 소스 코드를 편집하는 데 유용함

# 1.4 RStudio

▼ 그림 1-31 편집 창

```

fig8-6.R x
Source on Save
Run Source
1 library(ggplot2)
2
3 d <- read.csv('input/data-salary-3.txt')
4 KIDGID <- unique(d[,3:4])
5
6 N <- nrow(d)
7 K <- 30
8 G <- 3
9 coefs <- as.data.frame(t(sapply(1:K, function(k) {
10   d_sub <- subset(d, KID==k)
11   as.numeric(lm(Y ~ X, data=d_sub)$coefficients)
12 })))
13 colnames(coefs) <- c('a', 'b')
14 d_plot <- data.frame(coefs, KIDGID)
15
16
17 bw <- diff(range(d_plot$a))/20
18 p <- ggplot(data=d_plot, aes(x=a))
19 p <- p + theme_bw(base_size=18)
20 p <- p + theme(plot.margin=unit(c(9,15,9,9), "pt"))
21 p <- p + facet_wrap(~KID, nrow=3)
22 p <- p + geom_histogram(binwidth=bw, color='black', fill='white')
23 p <- p + geom_density(geom="area", aes(y=..count../(bw))), alpha=0.2, color='black', fill='gray20')
24 p <- p + labs(x='a', y='count')
25 ggsave(file='output/fig8-6-left.png', plot=p, dpi=300, w=4, h=6)
26
27
28 bw <- diff(range(d_plot$b))/20
29 p <- ggplot(data=d_plot, aes(x=b))
30 p <- p + theme_bw(base_size=18)
31 p <- p + facet_wrap(~KID, nrow=3)
32
33
1:1 (Top Level)
R Script

```

# 1.4 RStudio

## » RStudio의 사용자 인터페이스

- 이 코드 편집기는 일반적인 텍스트 편집기보다 훨씬 많은 기능을 제공함
  - 구문 강조, R 코드 자동 완성, 중단점을 사용한 디버깅 등 기능을 지원함
  - R 스크립트를 작성할 때 다음 단축키를 사용할 수 있음
- 
- 선택한 행을 실행하려면 **Ctrl** + **Enter** 를 누름
  - **Ctrl** + **Shift** + **S** 를 눌러 현재 문서를 소스 실행함  
즉, 현재 문서의 모든 표현식을 순차적으로 실행함
  - **Tab** 또는 **Ctrl** + **Space** 를 눌러 현재 입력과 일치하는 변수나 함수의 자동 완성 목록을 표시함
  - 줄 번호의 왼쪽 여백을 클릭하고 중단점을 설정함  
다음에 이 줄을 실행하면 이제 프로그램은 이 부분에서 일시 중지되고 사용자가 확인하길 기다림

# 1.4 RStudio

## » RStudio의 사용자 인터페이스

### 환경 창

- 환경 창은 새로 만든 사용 가능한 변수와 함수를 보여줌
- 기본적으로 현재 작업 중인 사용자 작업 영역의 전역 환경에 있는 변수들을 보여줌



# 1.4 RStudio

▼ 그림 1-32 환경 창

Environment History Connections					
Global Environment					
<input type="checkbox"/>	Name	Type	Length	Size	Value
<input type="checkbox"/>	a	numeric	4	80 B	num [1:4] 286 337 288 306
<input type="checkbox"/>	a0	numeric	1	56 B	350
<input type="checkbox"/>	args	character	2	240 B	chr [1:2] "C:\\Program Files\\R\\R...
<input type="checkbox"/>	b	numeric	4	80 B	num [1:4] 9.5 5.25 15.35 12.61
<input type="checkbox"/>	b0	numeric	1	56 B	12
<input type="checkbox"/>	basename	character	0	48 B	character (empty)
<input type="checkbox"/>	d	data.frame	5	3.1 KB	40 obs. of 5 variables
<input type="checkbox"/>	e	numeric	7	112 B	num [1:7] 0.2 0.1 0.2 0.2 0.2 0.2 ...
<input type="checkbox"/>	f	function	1	52.6 KB	function ()
<input type="checkbox"/>	g	function	1	9.7 KB	function (x)
<input type="checkbox"/>	h	function	1	10.3 KB	function (x)
<input type="checkbox"/>	h1	histogram	6	2.9 KB	List of 6
<input type="checkbox"/>	h2	histogram	6	2.9 KB	List of 6
<input type="checkbox"/>	K	numeric	1	56 B	4
<input type="checkbox"/>	KID	integer	40	208 B	int [1:40] 1 1 1 1 1 1 1 1 1 1 ...
<input type="checkbox"/>	lambda	numeric	1	56 B	20
<input type="checkbox"/>	lambda2	numeric	501	4 KB	num [1:501] 3.09 3.08 3.07 3.06 3...
<input type="checkbox"/>	m	numeric	1	56 B	0.154109403753684
<input type="checkbox"/>	N	numeric	1	56 B	40
<input type="checkbox"/>	N_k	numeric	4	80 B	num [1:4] 15 12 10 3

# 1.4 RStudio


## » RStudio의 사용자 인터페이스

- 새 객체(변수 또는 함수)를 만들 때마다 환경 창에 새로운 항목이 나타남
- 항목에는 변수 이름과 해당 값의 간단한 설명을 표시함
- 변수 값을 변경하거나 해당 변수를 제거할 때는 실제로 환경이 바뀌었으므로 환경 창에 해당 변경 사항을 반영함

# 1.4 RStudio

## » RStudio의 사용자 인터페이스

### 히스토리 창

- 히스토리 창은 콘솔 창에서 실행된 이전 표현식들을 표시함
- 콘솔 창에서 를 누르면 이전에 수행한 작업을 반복할 수 있음

# 1.4 RStudio

▼ 그림 1-33 히스토리 창



```

Environment History Connections
To Console To Source
p <- p + geom_point(position=position_jitter(w=0.4, h=0), size=1)
} else {
p <- p + geom_point(size=1)
}
p <- p + scale_color_gradient(low='grey65', high='grey5')
}
ggp <- putPlot(ggp, p, i, j)
}
}
png(file='D:/006966_wonderful2/02. 원고개발/실습/code/ch.11/output/fig11-4.png', w=1600, h=1600, ..
print(ggp, left=0.3, bottom=0.3)
dev.off()
help stan
stan
parameters {
real<lower=0, upper=100> sigma;
}
model {
sigma ~ uniform(0, 1);
}
SAMPLING FOR MODEL 'model1' NOW (CHAIN 1).
d <- read.csv(file='D:/006966_wonderful2/02. 원고개발/실습/code/ch.4/input/data-salary.txt')
res_lm <- lm(Y ~X, data=d)
res_lm
    
```

# 1.4 RStudio

## » RStudio의 사용자 인터페이스

### 파일 창

- 파일 창에는 폴더에 있는 파일들을 표시함
- 폴더 간 이동, 새 폴더 만들기, 폴더 혹은 파일 삭제, 폴더 혹은 파일 바꾸기 등을 수행할 수 있음

▼ 그림 1-34 파일 창



# 1.4 RStudio

## » RStudio의 사용자 인터페이스

- RStudio 프로젝트에서 작업할 때는 파일 창에서 프로젝트 파일들을 확인하고 구성하는 것이 편리함

# 1.4 RStudio

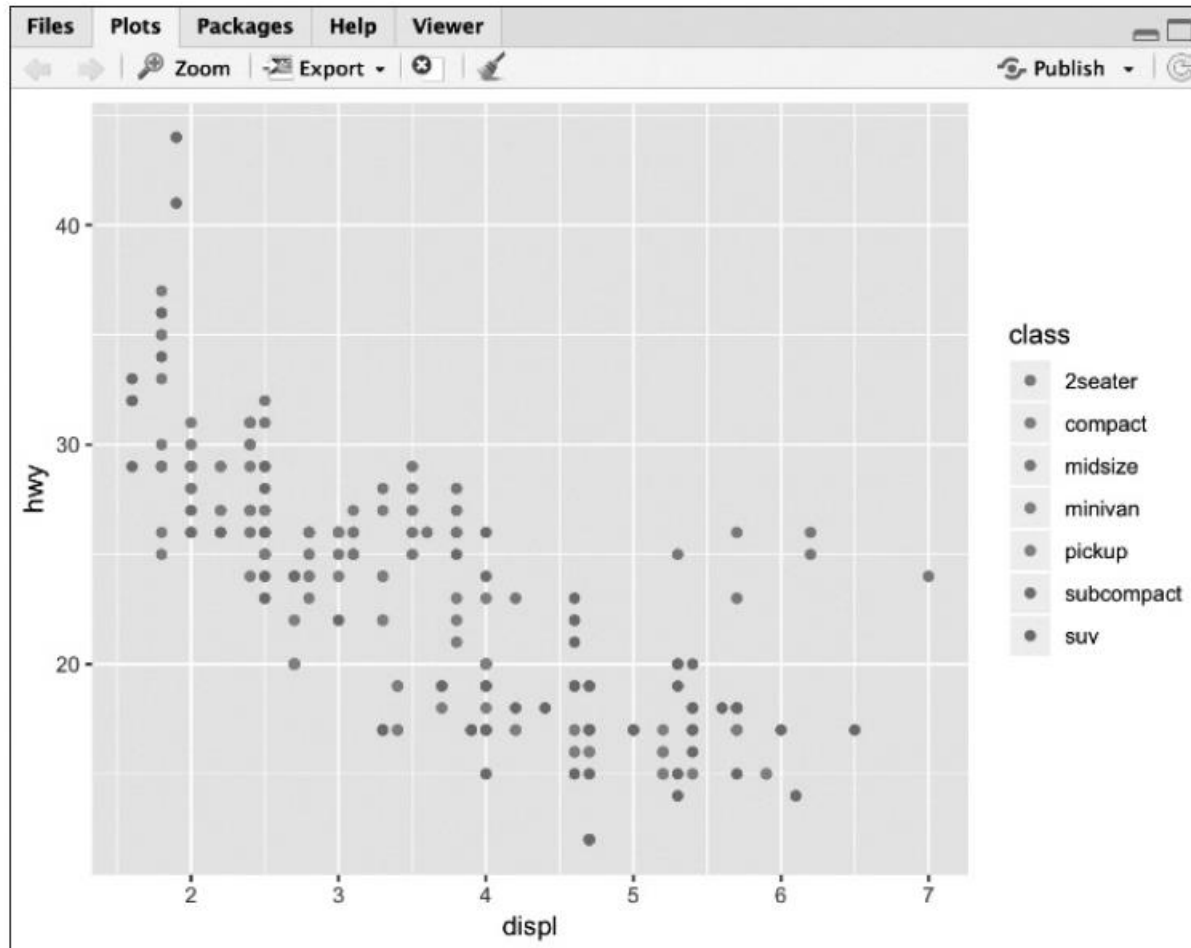
## » RStudio의 사용자 인터페이스

### 플롯 창

- 플롯 창은 R 코드로 만든 그림을 표시하는 데 사용함
- 둘 이상의 그림을 생성할 때 이전 것은 자동으로 저장되며, 사용자가 지우기 전까지는 모든 그림을 앞뒤로 탐색할 수 있음

# 1.4 RStudio

▼ 그림 1-35 플롯 창





# 1.4 RStudio

## » RStudio의 사용자 인터페이스

- 플롯 창의 크기를 조정하더라도 그래픽이 크기에 맞게 조정되어, 조정 전과 같이 멋진 그림을 얻을 수 있음
- 나중에 사용할 수 있게 그림을 파일로도 저장할 수 있음

# 1.4 RStudio

## » RStudio의 사용자 인터페이스

### 패키지 창

- R의 강력함은 대부분 패키지에서 나옴
- 패키지 창은 설치된 모든 패키지를 표시함
- CRAN에서 패키지를 쉽게 설치 혹은 업데이트하고, 라이브러리에서 기존 패키지를 제거할 수 있음

# 1.4 RStudio

▼ 그림 1-36 패키지 창

Files Plots Packages Help Viewer			
Install Update		Q	
Name	Description	Version	
<b>User Library</b>			
<input type="checkbox"/> assertthat	Easy Pre and Post Assertions	0.2.0	⊗
<input type="checkbox"/> base64enc	Tools for base64 encoding	0.1-3	⊗
<input type="checkbox"/> bayesm	Bayesian Inference for Marketing/Micro-Econometrics	3.1-0.1	⊗
<input type="checkbox"/> bayesplot	Plotting for Bayesian Models	1.5.0	⊗
<input type="checkbox"/> bda	Density Estimation for Grouped Data	10.1.9	⊗
<input type="checkbox"/> BH	Boost C++ Header Files	1.66.0-1	⊗
<input type="checkbox"/> bindr	Parametrized Active Bindings	0.1.1	⊗
<input type="checkbox"/> bindrcpp	An 'Rcpp' Interface to Active Bindings	0.2.2	⊗
<input type="checkbox"/> bitops	Bitwise Operations	1.0-6	⊗
<input type="checkbox"/> classInt	Choose Univariate Class Intervals	0.2-3	⊗
<input type="checkbox"/> cli	Helpers for Developing Command Line Interfaces	1.0.0	⊗
<input type="checkbox"/> cmprsk	Subdistribution Analysis of Competing Risks	2.2-7	⊗
<input type="checkbox"/> coda	Output Analysis and Diagnostics for MCMC	0.19-2	⊗
<input type="checkbox"/> coin	Conditional Inference Procedures in a Permutation Test Framework	1.2-2	⊗
<input type="checkbox"/> colorspace	Color Space Manipulation	1.3-2	⊗
<input type="checkbox"/> colourpicker	A Colour Picker Tool for Shiny and for Selecting Colours in Plots	1.0	⊗
<input type="checkbox"/> compositions	Compositional Data Analysis	1.40-2	⊗

# 1.4 RStudio

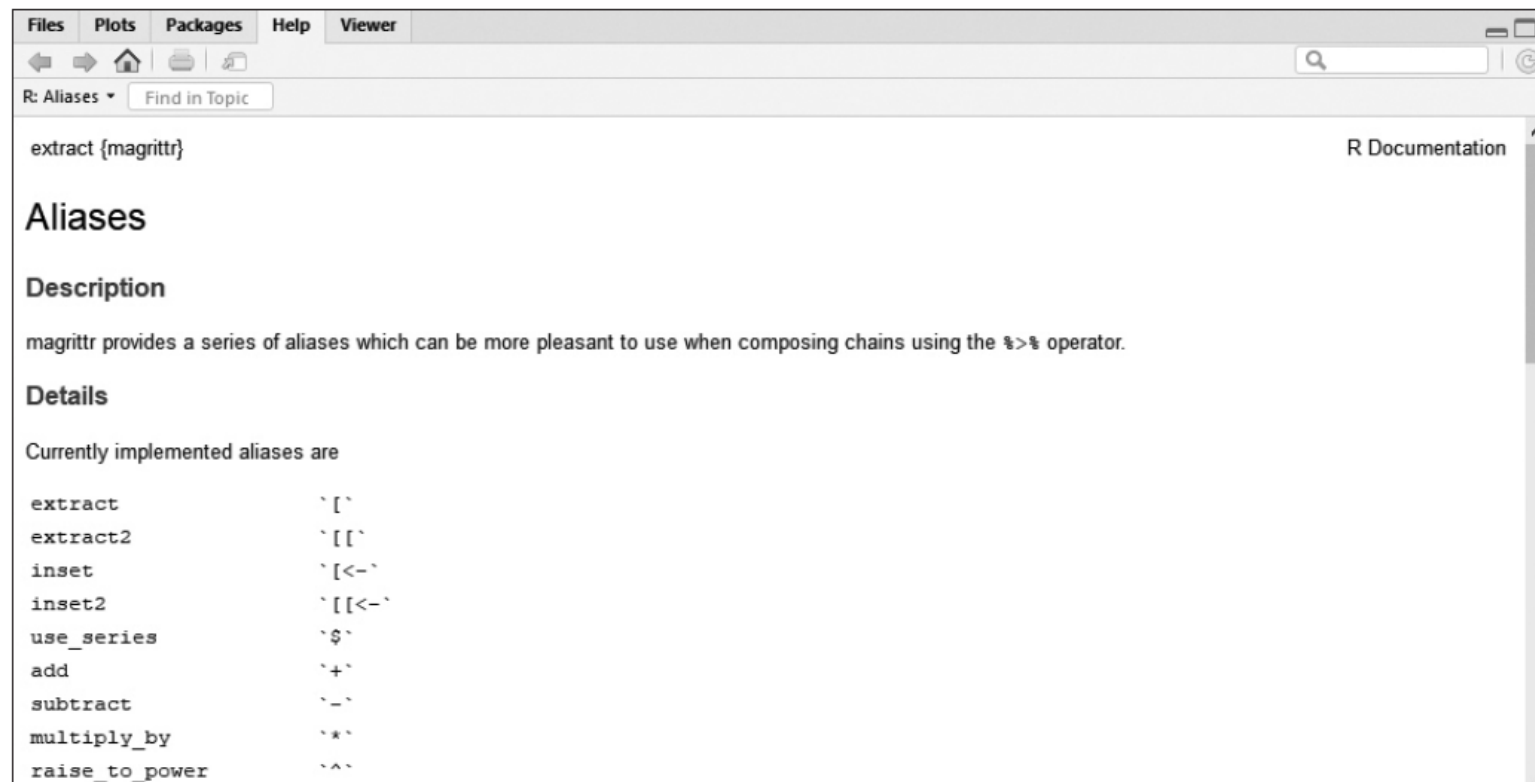
## » RStudio의 사용자 인터페이스

### 도움말 창

- R의 강력함은 또한 상세한 문서화에서 나옴
- 도움말 창에 나오는 설명서로 함수 사용법을 쉽게 배울 수 있음

# 1.4 RStudio

▼ 그림 1-37 도움말 창



# 1.4 RStudio

## » RStudio의 사용자 인터페이스

- 함수 설명서를 보는 다양한 방법이 있음
  - 검색 창에 함수 이름을 직접 입력해서 찾아봄
  - 콘솔 창에 함수 이름을 입력하고 **F1** 을 누름
  - 함수 이름 앞에 물음표(?)를 넣고 실행함
- 실제로 R의 모든 함수를 일일이 기억할 필요는 없음
- 잘 모르는 함수가 나왔을 때 여기에서 도움말을 얻는 방법만 기억함

# 1.4 RStudio

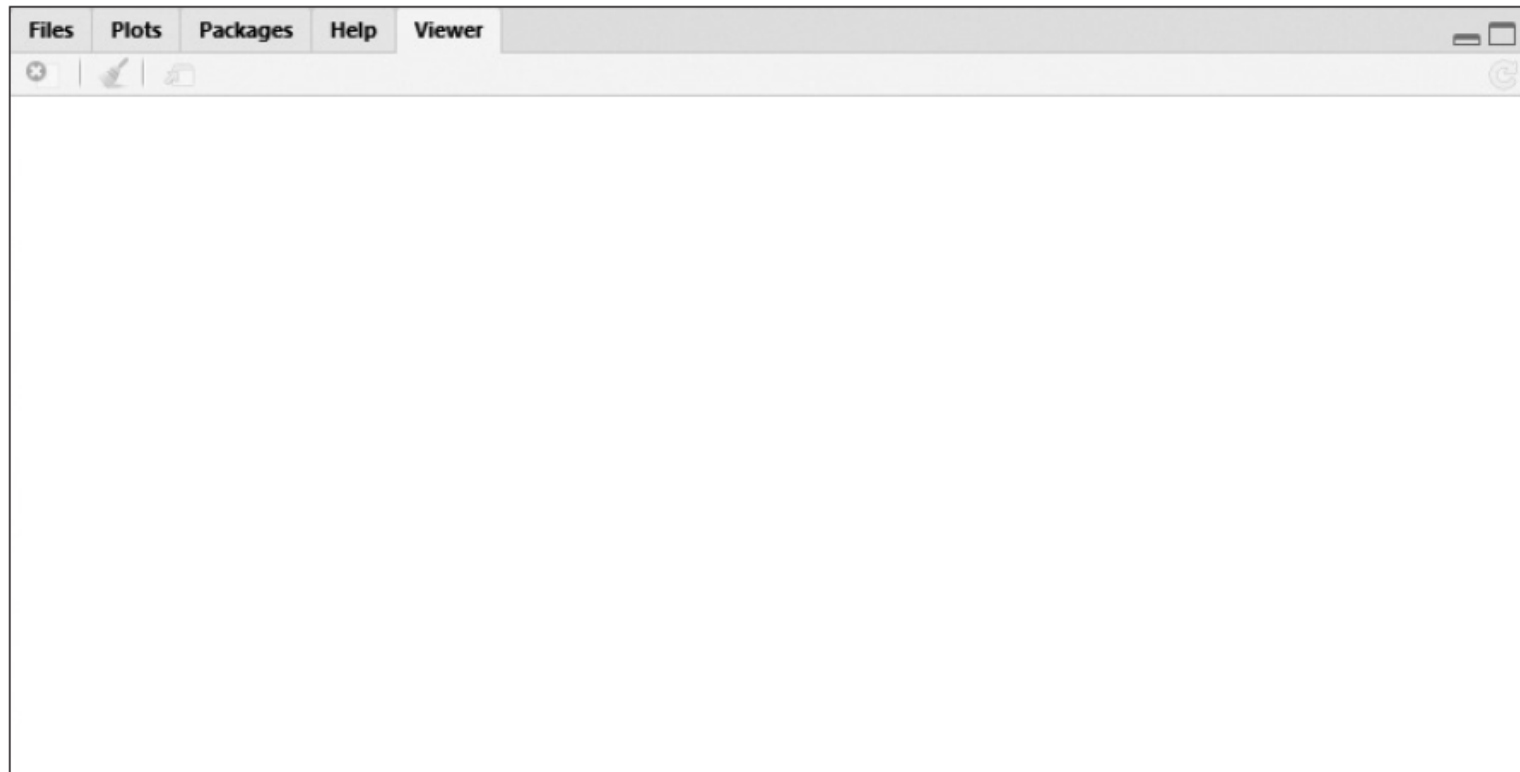
## » RStudio의 사용자 인터페이스

### 뷰어 창

- 뷰어 창은 새로 추가된 기능
- 기존 데이터를 사용자와 상호 작용하여 보여 주는 자바스크립트 라이브러리와 R의 기능을 통합한 패키지 숫자가 점점 많아져서 도입함
- 다음은 필자가 개발한 `formattable`(<https://renkun-ken.github.io/formattable/>) 패키지의 예로, R의 데이터 프레임에 엑셀의 조건부 서식을 간단히 구현한 것

# 1.4 RStudio

▼ 그림 1-38 뷰어 창





# 1.4 RStudio

## » RStudio 서버

- R을 지원하는 리눅스 버전을 사용한다면 RStudio 서버를 좀 더 쉽게 설정할 수 있음
- 그것은 웬만한 노트북보다 더 성능이 좋음
- 더욱 안정적으로 호스트 서버에서 주로 실행됨
- 웹 브라우저를 사용하여 RStudio의 R 세션을 실행할 수 있음
- 사용자 인터페이스는 거의 동일함
- 마치 로컬 컴퓨터를 사용하는 것처럼 서버의 컴퓨팅 리소스와 메모리에 접근할 수 있음

# 1.5 간단한 예

## » 간단한 예

- 콘솔 창에서 명령을 입력하여 연산을 수행함
- 모델 피팅과 그래프를 생성하는 간단한 예제를 실습함
- 먼저 정상 분포에서 추출한 난수 100개로 된 x 벡터를 생성함
- 그런 다음 숫자 100개로 구성된 또 다른 y 벡터를 만듦
- 각 원소는 x에 3배한 값에 2를 더하고 임의의 노이즈가 섞인 값을 가짐
- <-는 대입 연산자로, 나중에 자세히 다룰 것
- str()을 사용하여 벡터 구조를 출력함

```
> x <- rnorm(100)
```

```
> y <- 2 + 3 * x + rnorm(100) * 0.5
```

```
> str(x)
```

```
num [1:100] -0.4458 -1.2059 0.0411 0.6394 -0.7866 ...
```

```
> str(y)
```

```
num [1:100] -0.022 -1.536 2.067 4.348 -0.295 ...
```

# 1.5 간단한 예

## » 간단한 예

- 이미  $x$ 와  $y$  사이에  $y = 3x + 2 + e$ 라는 관계가 있다는 사실을 알고 있음
- 두 샘플  $x$ 와  $y$  사이에 간단한 선형 회귀 모델을 적용할 수 있음
- 이때 선형 모델의 파라미터(2와 3)는 어떻게 찾는지 알아보자
- 이를 위해  $\text{lm}(y \sim x)$  함수를 사용할 것

```
> model1 <- lm(y ~ x)
```

# 1.5 간단한 예

## » 간단한 예

- 모델 피팅의 결과를 model1 객체에 저장함
- 이렇게 찾은 모델은 간단히 model1을 입력하거나 print(model1)을 입력하여 출력할 수 있음

```
> model1
```

```
Call:
```

```
lm(formula = y ~ x)
```

```
Coefficients:
```

(Intercept)	x
2.051	2.973

## 1.5 간단한 예

### » 간단한 예

- 더 자세한 내용을 알고 싶다면 model1에 대해 summary() 함수를 사용함

```
> summary(model1)
```

Call:

```
lm(formula = y ~ x)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.88242	-0.31700	-0.08221	0.25683	1.14234

# 1.5 간단한 예

## » 간단한 예

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	2.00448	0.04723	42.45	<2e-16 ***
x	3.00067	0.04687	64.02	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4721 on 98 degrees of freedom

Multiple R-squared: 0.9767, Adjusted R-squared: 0.9764

F-statistic: 4099 on 1 and 98 DF, p-value: < 2.2e-16

## 1.5 간단한 예

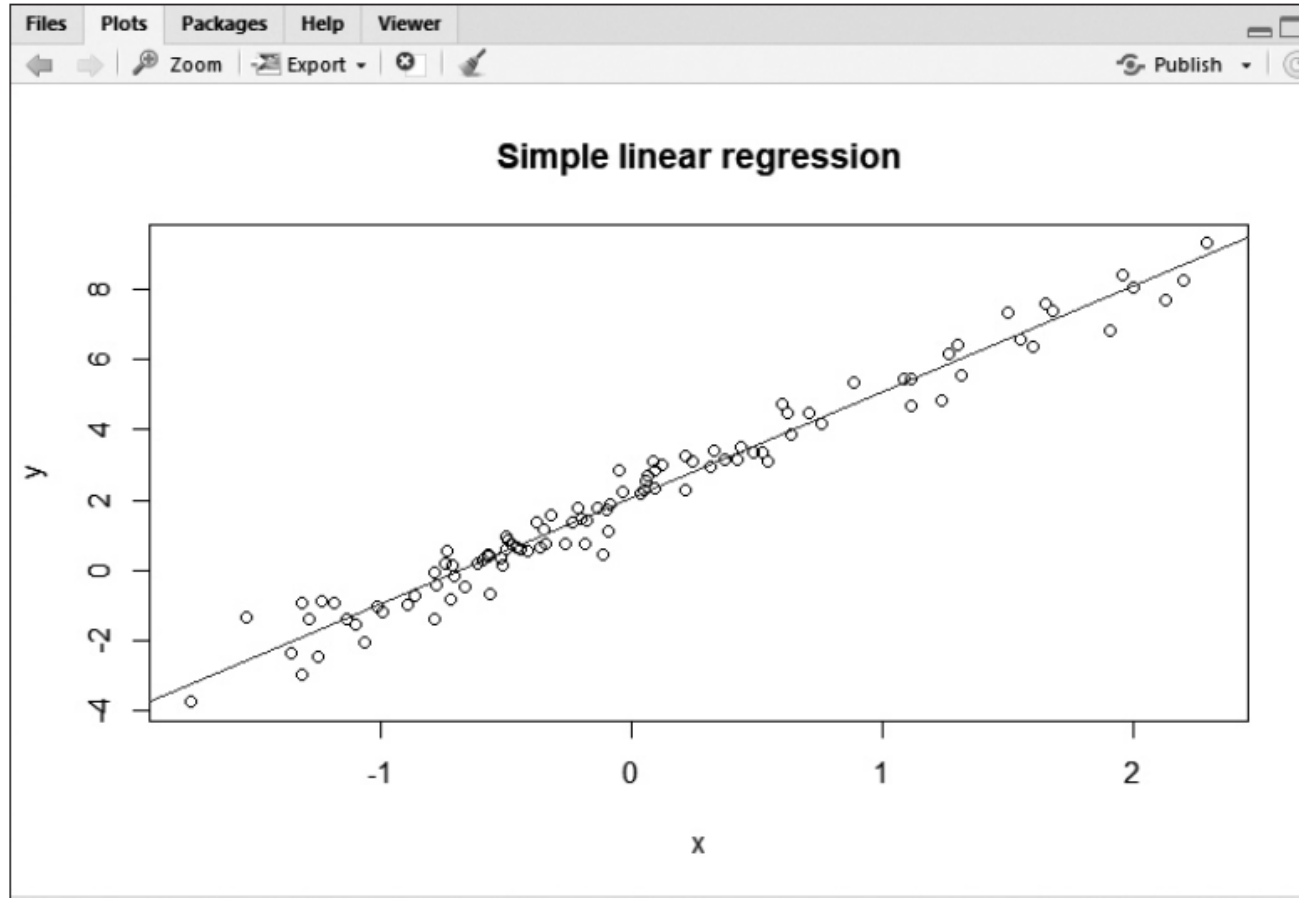
### » 간단한 예

- 물론 이 점들과 피팅된 모델을 함께 그래프로 나타낼 수도 있음

```
> plot(x, y, main = "Simple linear regression")  
> abline(model1$coefficients, col = "blue")
```

# 1.5 간단한 예

▼ 그림 1-39 간단한 선형 회귀 그래프





# 1.6 마치며

## » 마치며

- 이 장에서는 R의 주요 장점에서 몇 가지 기본적인 사실들을 살펴봄
- 윈도 운영 체제에서 R을 설치하는 방법도 배움
- R 프로그래밍을 더욱 손쉽게 해 주는 RStudio를 알아봄
- RStudio의 사용자 인터페이스를 살펴보고 메인 창에 있는 각 창의 기능도 다루었음
- 마지막으로 데이터를 활용하여 모델을 구하고, 간단한 그래프를 그릴 수 있게 몇 가지 R 명령을 실행해서 R을 활용하는 작업에 대한 첫인상을 얻음