

# The Friendly Beginners' R Course

## 초보자를 위한 친근한 R 강좌

저자 : Toby Marthews(파나마의 BCI Research Centre(<http://www.stri.org>))

번역 : 박 종호 (순천대학교)

2010년 7월 27일

This document is the Korean translation of The Friendly Beginners' R Course written by Toby Marthews. The translator is Jongho Park (schrs@scnu.ac.kr). It might contain typos and errata. Comments and corrections are always welcome. Please feel free to email correspondence to schrs@scnu.ac.kr.

이 문서는 Toby Marthews가 쓴 The Friendly Beginners' R Course written의 한글 번역본이다. 번역은 박 종호 (schrs@scnu.ac.kr)가 담당하였다. 이 문서에 타이핑 오류와 잘못된 부분이 있을 것이다. 비평과 오류 교정은 언제든지 환영하며 schrs@scnu.ac.kr로 메일을 보내주시면 고맙겠다.

이 강좌는 그림을 포함하여 단지 15쪽에 불과하여, 독자가 시간이 있을 때 살펴볼 수 있는, 현재 이용할 수 있는 R에 대한 가장 쉬운 입문서일 것이다. 이 교재에는 예제 파일("first.r", "mystery.r", "quadrats.r", "quadratdata")이 포함되어 있다. 이들 예를 통해서 많은 것을 이해할 수 있을 것이다. 별표로 된 줄 아래부터 읽어라. (필요하다면) 처음에 R을 설치하는 방법을 포함하여 모두를 쉽게 이해할 수 있을 것이다.

2005년 8월 Toby, (최종 업데이트 2010년 1월)

\*\*\*\*\*

독자는 R 언어와 환경을 배우기로 결정하였는가? 혹시, 음... a) 네 사장/감독/자문관이 너에게 전체 아이디어에 대해 감정이 나쁘다고 말하였거나, b) 내가 분석할 것이 있는데 친구가 네 상식에 반하여- R이 쓰기 쉽고 도움이 될 것이라고 장담하였거나, c) 통계자료를 읽거나 책에 나온 모형을 모델링하려다 포기하고, R이 이를 우회하는 길이라는 필사적인 희망을 갖거나 d) 네 오타쿠 등급을 올리거나 다른 사람들에게 그런 인상을 남기려 한다고 말하는 것이 더 정확할지도 모르겠다.

이유가 무엇이든, 이 패키지가 무엇을 할 수 있는가를 안다면, R을 배우려는 것은 좋은 생각이라고 믿는다. R은 많은 재치 있는 것들을 할 수 있으며, 자료 분석을 많이 해야 한다면 네 삶을 더 편하게 해 줄 것이다. 오다쿠 또한 좋은 지적이다. 내가 이력서에 R에 대해 안다고 써넣으면 모든 사람이 실제보다 더 영리하다고 믿기 때문이다.

R을 '그냥 또 다른' 통계 패키지라고 생각하는 사람들에게 할 얘기가 있다. 글썄, R은 프로그래밍 언어이며 동시에 통계 분석을 할 수 있는 수단이다. 이것은 내가 R이 현재 다른 것들 보다 앞선다고

생각하는 이유 중 하나이다. R을 배워서 프로그래밍 (모델링 수업에서 배우거나 (배워야만 하는) 것의 70-80%가 이러한 기술이다.)과 컴퓨터를 이용하여 통계학을 할 수 있는 기술을 얻을 것이다. 그래서, 둘 다를 함께 배움으로써, 하나의 가격으로 두 가지 기술을 얻을 수 있다. (나는 이 단락을 2005년에 썼지만, 2009년에 Owen Petchey과 Andrew Beckerman, Dylan Childs은 Bulletin of the British Ecological Society 40에 쓴 기사 “통계 소프트웨어가 주는 충격과 공포 - 왜 R인가?”에서 비슷한 이야기를 하였고, R이 Sigmaplot과 MS Excel, SAS, Genstat, Mathematica를 모두 대체할 것이라 한 것을 읽었다!)

**R 교과서를 사지마라.** (적어도 이 교재를 읽기 전에는.) 첫째, R 웹사이트 <http://www.R-project.org>에서 공짜로 내려 받을 수 있는 2300쪽의 R 설명서가 있다. 둘째, R은 ‘새로운 통계학’이 아니라 표준 통계학을 보다 빨리 하는 수단이다. 따라서 독자는 표준 교과서를 사용할 수 있으며, 또 사용해야 한다. 필요하다면 이 책에 주를 추가하여. R은 MATLAB(<http://www.mathworks.com/products/matlab>)과 비슷하며 이것의 공짜 대안이다. (둘을 비교한 것으로는 <http://germain.umemat.maine.edu/faculty/hiebeler/comp/matlabR.pdf>을 보라.) SAS, SPSS, Stata, Systat 사용자에게는, “Quick-R” (<http://www.statmethods.net/index.html>)이 왜 R이 도움이 되는지를 설명한다.

나는 겨우 2005년부터 R을 사용해왔다. 이것은 내가 R의 장점과 단점을 정말로 알지 못한다는 것을 의미한다. 그러나 다음 몇 쪽에서 독자가 자신의 R 스크립트를 작성하고, R 함수를 사용하고, 멋진 그림을 그릴 수 있도록 하고, R에 익숙해지는 데 충분한 빠른 안내서를 제공할 것이다. 이 안내서는 전에 컴퓨터를 사용해봤지만 프로그램을 해본 경험이 없는 사람들을 위해 썼다. (프로그래밍의 경험이 있다면 아래 어떤 부분을 건너뛰어야 할지 알 것이다.)

독자가 이 안내서를 훑어보는 데 얼마나 걸릴지 모른다. (사람마다 다를 것이다.) 그러나 단지 6개 도전 과제가 있어 너무 길지는 않을 것이다. 컴퓨터 앞에 앉아서, 이 안내서를 인쇄하고, 커피 한잔 (아니면 녹차라도)을 준비하여 각 절을 하나씩 시작해 보자.

## 1 R을 설치하고 실행하기

독자는 컴퓨터와 컴퓨터가 어떻게 작동하는지에 대한 일반적인 지식이 조금 필요하다. 독자가 이미 컴퓨터 언어와 작업공간 디렉토리에 대해 알고 컴퓨터에 R을 설치하였다면 다음 절로 가라.

컴퓨터 프로그램은 항상 어떤 컴퓨터 언어로 작성된다. 컴퓨터 언어는 (BASIC, JavaScript, R)과 같은 스크립트 언어와 (FORTRAN, PASCAL, C, C++, Java)와 같은 컴파일 언어가 있으며 프로그래머가 어떤 것을 사용하든 간에 (1과 0으로 구성된) 기계어로 번역되어 컴퓨터에서 실제로 실행된다. 차이는 다음과 같다. 스크립트 언어로는 컴퓨터가 프로그램을 한 줄씩 번역하여 실행한다. 그러나 컴파일 언어는 전체 프로그램을 한꺼번에 번역하여 하드디스크에 실행 가능한 기계어로 저장해 놓고 (윈도우에서는 보통 확장자가 exe이다.) 직접 이 실행 파일을 실행한다.

일반적으로 말해서, 스크립트 언어는 느리지만 좀 더 사용자 친화적이다. (특히 예러 알림에서.) 그리고 컴파일 언어는 훨씬 빠르지만 사용하기가 훨씬 불편하다. 그래서 R로 프로그램을 만든다면 C나 FORTRAN으로 만든 같은 프로그램보다 훨씬 느리다. 그러나 독자는 다음을 알아야 한다. a) 실제로 많은 계산을 한다면 그 차이를 느낄 수 있다. b) 전에 컴퓨터 언어를 사용해보지 않았다면, FORTRAN과 같은 것으로 시작한다면 머리를 쥐어뜯을 것이다. c) R의 경우, 삶을 훨씬 쉽게 만들어 주는 통계 함수와 그래프 그리는 것과 같은 추가적인 기능이 있다. 예를 들어, FORTRAN은 IDL과

같은 특별한 추가 설치를 해야 이것들을 할 수 있다. d) R과 같은 언어를 사용하여 프로그램 하는 방법을 배운다면 나중에 다른 컴퓨터 언어를 쉽게 배울 수 있다는 것을 알게 된다. 왜냐하면 모든 언어는 비슷한 구조를 가지고 있기 때문이다. (repeat loop, for loop, if 문, 등등).

이제 실제로 해보자. 컴퓨터에 R을 설치하는 방법이 아래 있다. 윈도우와 리눅스에서 설치 방법이 있다. (애플 맥에 대해서도 R이 있지만 내가 맥에 대해 알지 못한다.) **만약 윈도우를 사용한다면 이미 R을 설치했다 하더라도 아래 2-3 단계를 꼭 해보라.**

## 1.1 윈도우에서 설치

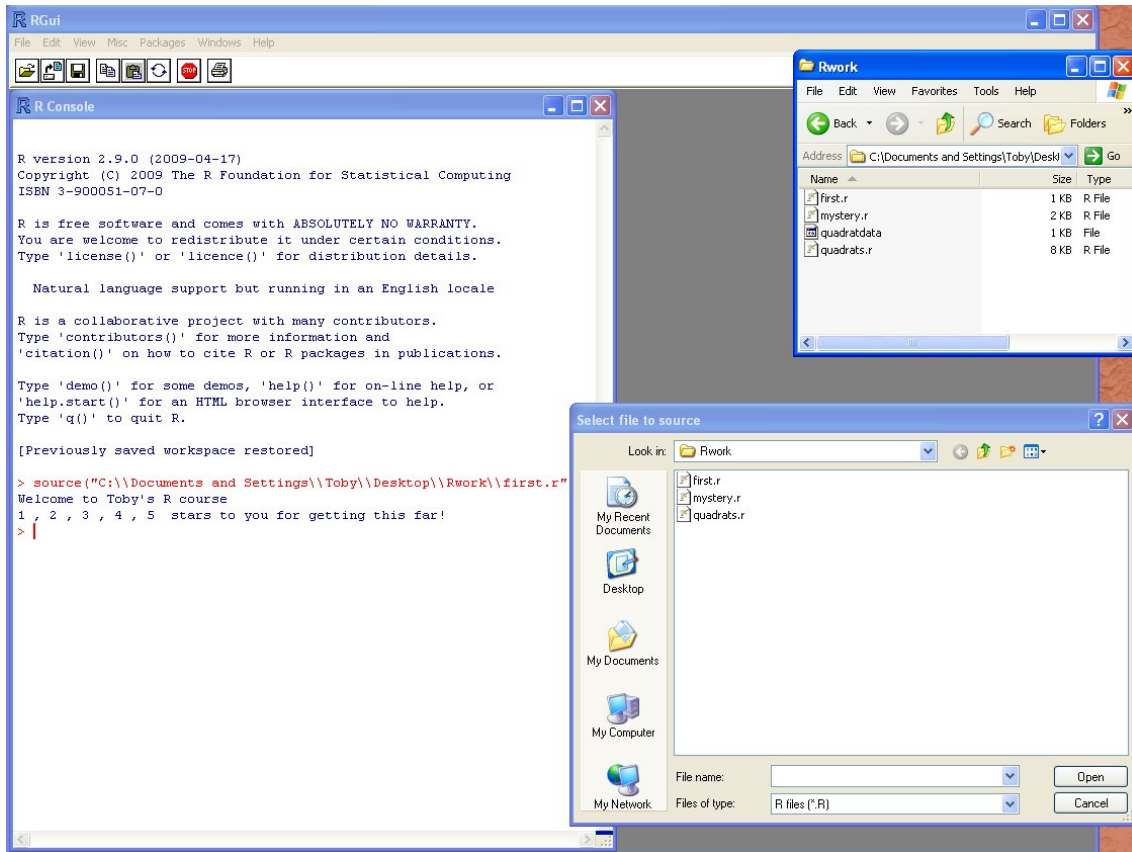
1. R 웹사이트 <http://www.R-project.org>로 가서 왼쪽의 Download/CRAN를 클릭하여 (내려받기 시간을 줄이기 위해) 독자에게 지리적으로 가까운 미러 사이트를 선택하라. Windows를 선택하고, “base”를 클릭하여, 설치 실행파일을 내려 받아 (“Download R x.x.x for Windows”를 클릭하라. 여기서 “x”는 숫자이다.) 바탕화면에 저장하라(“exe” 파일). 이것을 더블 클릭하여 설치를 시작하라. (모든 Online PDF Manuals를 기본 설치 옵션으로 선택하라.) 시작 메뉴 폴더 선택에서 ‘다음’을, 바탕화면에 아이콘 생성에 ‘예’를, 빠른 실행에 아이콘 생성에 ‘아니오’를 선택하라. (단계 3을 보라.) R이 이제 컴퓨터에 설치되었다. (데스크톱에 “R-x.x.x-win32.exe” 파일은 지워도 된다.)
2. R을 사용하기 위해 바탕화면에 (또는 원하는 어느 곳이든) 작업 폴더를 만들어라. (바탕화면에서 오른쪽 클릭하여 새 폴더 만들기를 선택하고 이름을 붙여라.) 그리고 (이 교재에 딸린) “first.r”을 이 폴더에 복사하라. R은 이 폴더에 (“RData”라는 파일에) 변수와 함수의 정의를 저장한다. 따라서 반드시 작업 폴더가 하나 있어야 한다. (아, “A->B”는 “메뉴 A에 가서 B를 선택하라.”라고 말하는 나의 방식이다.) 주의하라. 짜증나는 방식이지만 어떤 윈도우 시스템은 첨부된 이메일 파일을 저장할 때 자동으로 “xxx.r”을 “xxx.r.txt” 또는 “XXX.R.TXT”로 이름을 바꾼다. 이름을 다시 “xxx.r”로 바꿔야 한다.
3. 설치하는 동안 나타나는 바탕화면 바로가기 오른쪽 클릭하여, 속성을 선택하라. “대상”은 그대로 두고, 시작 위치 상자를 수정하여 단계 2에서 만든 작업 폴더의 위치를 넣어라. 그리고 적용을 클릭하라. 다음으로, 바탕화면 바로가기 오른쪽 클릭하여 “RGui”를 열라. (“Gui” = “Graphical User Interface”) 메뉴에서 ‘파일’->‘R 코드의 소스..’를 보고, R이 올바른 작업 폴더를 여는가를 확인하라. (나타나는 창이 단계 2의 폴더여야 한다. 맞으면 파일을 열지 말고 취소 버튼을 눌러라. 그러나 맞지 않으면 단계 2로 다시 가라.) 태스크 바에 빠른 실행 아이콘을 두고 싶다면 바탕화면의 바로가기 마우스로 끌어 태스크 바로 (보통 시작 메뉴의 시작 오른쪽으로) 끌어 놓아라.
4. 이제 R을 시작하라. R이 간단한 프로그램을 실행시킬 수 있는가를 보자. 파일 메뉴에서 R 코드의 소스를 열어 작업 폴더에서 first.r을 찾아서 열라. R이 이 프로그램을 실행시켜 환영 메시지를 보여줄 것이다. (first.r 파일은 단순히 텍스트 파일이다. 텍스트 에디터로 열어 내용을 볼 수 있다.)
5. 아직 완전히 끝나지 않았다. 파일 메뉴에서 스크립트 열기를 눌러 first.r를 선택하라. R 에디터 창이 이 프로그램을 열어 변경할 수 있도록 한다. (이것도 역시 할 수 있는지 확인하라.) 줄 6

에서 5를 찾아서 10으로 바꿔라. 파일 메뉴에서 다른 이름으로 저장을 눌러 이것을 “first2.r”로 저장하라. (에디터 창을 닫아라.)

6. 이제 단계 4와 같은 방식으로 first2.r을 실행하라. 제대로 실행된다면 1부터 10까지 볼 것이고, 별 10개를 받을 자격이 있다.
7. 빨간 X를 누르거나 q()을 입력하여 R을 끝마칠 수 있다. 지금은 작업공간 이미지를 저장할 필요가 없다. (사실 이 초보자 강좌에서는 작업공간을 저장할 것인가에 대해 항상 ‘아니오’를 눌러도 된다.)

## 1.2 리눅스에서 설치

1. R 웹사이트 <http://www.R-project.org>로 가서 왼쪽의 Download/CRAN를 클릭하여 (내려받기 시간을 줄이기 위해) 독자에게 지리적으로 가까운 미리 사이트를 선택하라.
2. 리눅스를 선택하고 자신의 리눅스 버전에 맞는 파일을 찾아서 설치하라. (YAST와 같은 것을 통해서나 더블클릭하여 리눅스 버전에 맞는 방법으로 설치하라. 자세한 내용은 <http://cran.r-project.org/doc/manuals/R-admin.pdf>를 참조하라.)
3. R을 사용하기 위해 바탕화면에 (또는 원하는 어느 곳이든) 작업 디렉토리를 만들어라. 그리고 (이 교재에 딸린) “first.r”을 이 디렉토리에 복사하라. R은 이 디렉토리에 (“RData”라는 파일에) 변수와 함수의 정의를 저장한다. 따라서 작업 디렉토리가 반드시 하나 있어야 한다.
4. 터미널을 열고, cd를 사용하여 작업 디렉토리로 디렉토리를 바꾸어라. R을 입력하여 R 언어로 들어가라. (프롬프트가 >로 변할 것이다.)
5. R이 간단한 프로그램을 실행시킬 수 있는가를 보자. “source(“first.r”)”를 입력하면 R이 이 프로그램을 실행하여 환영 메시지를 보여줄 것이다. (first.r 파일은 단순히 텍스트 파일이다. GNUemacs, kate, gedit, ue, pico, vi 같은 텍스트 에디터로 열어 내용을 볼 수 있다.)
6. 아직 완전히 끝나지 않았다. 텍스트 에디터로 first.r을 열어(터미널을 사용하지 말아라. 터미널을 열어 놓고 동시에 이것은 다른 창에서 하라.) 이 프로그램을 변경할 수 있도록 한다. (이것도 역시 할 수 있는지 확인하라.) 줄 6에서 5를 찾아서 10으로 바꿔라. 파일 메뉴에서 다른 이름으로 저장을 눌러 이것을 “first2.r”로 저장하라. (에디터 창을 닫아라.)
7. 이제 단계 5와 같은 방식으로 first2.r을 실행하라. 제대로 실행된다면 1부터 10까지 볼 것이고, 별 10개를 받을 자격이 있다.
8. q()을 입력하여 R을 끝마칠 수 있다. 지금은 작업공간 이미지를 저장할 필요가 없다. (사실 이 초보자 강좌에서는 작업공간을 저장할 것인가에 대해 항상 ‘아니오’를 눌러도 된다.)



## 2 두 창: 콘솔과 에디터

첫 번째 R 스크립트를 실행하는 데 성공하면 더 많은 것을 원할 것이다! 지금까지 한 것에 익숙해졌으면 원래 `first.r`을 에디터에 다시 열어서라. 다음 두 가지를 할 수 있는가를 보라.

Q1. FOR loop를 1에서 5로 올라가는 대신 5에서 1로 내려가도록 할 수 있는가?

Q2. 이것을 올라갔다 다시 내려가도록 할 수 있는가? (이것은 두 개의 FOR loop를 순차적으로 쓰는 것이 가장 쉽다.)

이 두 문제를 시도해 볼 때 (이들이 지루하게 느껴져도 독자는 뛰기 전에 걷는 것을 배워야 한다.) R 프로그래머가 동시에 창을 두 개 띄우는 방법에 익숙해져야 한다. 에디터 창에서 프로그램을 편집하고 저장하며, 콘솔 창(다른 말로 터미널이라 한다.)으로 건너가서 프로그램을 실행한다. (윈도우 버전에서는 어떤 창이 활성화되었는가에 따라 파일 메뉴가 달라진다.) 어쨌건, 이것이 다른 언어에서도 프로그램 하는 방식이다. 많은 사람이 두 창의 크기를 조정하고 움직여서 중복되지 않으면서 가능한 한 크게 설정한다.

R의 내장 텍스트 에디터(에디터 창)는 매우 단순하여 사용을 권하지 않는다. 공짜로 설치할 수 있는 더 좋은 많은 에디터들이 있다. (어떤 사람들은 MS 메모장과 MS 워드패드를 사용하지만 별로

좋지 않다.<sup>1</sup>) 나는 “Notepad++” (<http://notepad-plus.sourceforge.net/uk/site.htm>에서 공짜로 내려 받을 수 있다.)를 사용하고 있으며 이를 권장한다.<sup>2</sup> 이것은 매우 좋다.<sup>3</sup> 좋은 텍스트 에디터 없이 프로그램을 하는 것은 불필요하게 삶을 힘들게 만드는 것이다!

Q1과 Q2를 건너뛰지 마라. 편집하고, 저장하고, 실행하는 과정은 내가 설치한 R이 잘 작동하는가를 확인하기 위한 것이다. 이것이 제대로 작동하지 않는다면 지금까지 했던 것을 다시 확인하거나 도움을 청하라. (<http://www.R-project.org>에 있는 설치 FAQ를 보라.) 도중에 “syntax error”가 나오면 이는 내가 편집한 코드에 잘못된 것이 있다는 것을 의미한다. 타이핑 에러나 괄호의 쌍이 맞는가와 같은 것을 확인하라.

하지만 독자는 R을 설치할 때 이미 초보자를 위한 문서와 FAQ를 컴퓨터에 함께 설치했다. 언제든지 콘솔 창에 다음과 같이 입력하여 이들을 볼 수 있다.

```
help.start()
```

여기에는 수백 쪽의 정보가 있다. 그러나 지금은 알아야 할 모든 것을 알려주는 이 초보자 강좌를 읽고 있기 때문에 그것을 볼 필요는 없다! 단지 설명서가 설치되어 있고 내 짧은 초보자 강좌에서 얻을 수 없다면 그것을 찾아 볼 수 있기 때문에 이야기 한다. 그러나 독자는 이미 이 강좌에 상당히 들어왔는데 왜 끝까지 읽어보지 않고 그것을 찾아보겠는가?

### 3 R 설명서

독자는 앞의 질문들에 대답하려고 first.r에 집중하는 동안 각 줄이 무엇을 하는지 이해할 수 있었을 것이다. 거기 있는 코멘트(# 줄)에서 모든 것을 설명하지는 않았다. 왜냐하면 독자는 R의 광범위한 설명서 시스템을 사용하는 습관을 가져야 하기 때문이다. 다음은 이것의 사용 방법이다.

R 스크립트를 받아서, 이것이 어떻게 작동하는지를 알아보려고 텍스트 에디터로 열었다고 해보자. 첫 줄은 다음과 같다.

```
a=seq(-2,4,length.out=5)
```

---

<sup>1</sup>.txt 포맷으로 프로그램을 저장할 때 보통의 텍스트파일이 아니라 Windows-형식의 텍스트파일을 사용하는 것이다. ([http://en.wikipedia.org/wiki/Text\\_files#Standard\\_Windows\\_.txt\\_files](http://en.wikipedia.org/wiki/Text_files#Standard_Windows_.txt_files)를 보라.), 이것은 좀 더 복잡한 일을 하게 되면 문제가 될 수도 있지만 (예, UNIX scripting), 지금은 괜찮다.

<sup>2</sup>내가 Notepad++에서 사용하는 기본설정을 구하려면, Settings->Preference로 가서 Edit Components 탭에서 “Display line number margin”을 ON으로 하고, MISC 탭에서 “Auto-indent”를 OFF로 한다. 또한 “Don’t check at launch time”, “New Document/Open Save Directory” 탭에서 포맷을 (“Windows”가 아니라) “Unix”로 하고 “Global”에서 “Enable close button on each tab”으로 한다. 그리고 나서 Format 메뉴로 가서 encoding을 ANSI로 한다. 그 후 View 메뉴에서 “Word wrap”을 ON하고 “Show Symbol”->“Show White Space and TAB”과 “Show wrap symbol”. Notepad++와 함께 NppToR (<http://npptr.sourceforge.net/>)를 추가로 설치할 것을 강력히 권장한다. 이것은 R의 구문 하이라이트를 제공한다.

<sup>3</sup>나는 다른 R 사용자들이 “ConTEXT” (<http://www.contexteditor.org>), “TextPad” (<http://www.textpad.com>; \*\* 유료\*\*), “Tinn-R” (<http://www.sciviews.org/Tinn-R>) 또는 “Crimson Editor” (<http://www.crimsoneditor.com>)를 사용한다는 것을 안다. Tinn-R과 Crimson Editor가 R의 구문 하이라이트를 제공하고, TextPad는 이를 추가 설치 (<http://www.textpad.com/add-ons/files/syntax7r.zip>) 형태로 제공하지만, 나는 Notepad++와 NppToR의 조합을 좋아한다.



그러나 아직 이것이 무엇을 하는 것인지를 모른다. 여기 명령어는 seq (그 뒤 괄호는 이 명령어에 ‘전달된’ 인수를 담고 있다.) 따라서 첫 번째 할 일은 콘솔 창에 ?seq를 입력하여 seq에 대한 R 설명서를 여는 것이다. 설명서가 나타날 것이다. (윈도우에서는 새로운 창에 나타나며, 리눅스에서는 같은 창에 나타난다. q를 치면 처음으로 돌아간다.) 이 설명서는 일반적으로 매우 기술적인 방법으로 쓰였다. (이것에 익숙해질 것이다.) 그러나 보통 이것을 많이 읽을 필요는 없다. 설명을 무시하고 마지막으로 내려가서 예제를 보라. (seq 설명서의 첫 예는 “seq(0, 1, length.out=11)” 이다. 이 예들은 설명서에서 처음 보기에 가장 좋은 부분이다. 이것을 콘솔 창에 복사하여 어떤 일을 하는지를 볼 수 있다. (윈도우에서는 마우스로 원하는 예제를 선택하고, CTRL+c를 눌러 복사하고, 콘솔을 클릭하여 CTRL+v를 눌러 붙여 넣는다. 리눅스에서는 이것을 선택하고, Edit->Copy를 하고, 다시 q, 다시 Edit->Paste를 한다.) seq 첫 예제로 이를 해보라.

```
seq(0, 1, length.out=11)
[1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```

이것에서 seq 명령어가 연속된 수를 만들며, 프로그램의 잘 모르는 명령어가 -2에서 4까지 5개 순차적인 수를 만들어 이를 a라는 변수에 저장한다는 것이라는 것을 이해하는 데 천재성이 필요하지는 않다. a를 입력하여 이를 확인할 수 있다.

```
>a
[1] -2.0 -0.5 1.0 2.5 4.0
```

이제 다른 명령어를 보자.

Q3 “plot” 설명서의 “Discrete Distribution Plot” 예와 “matrix” 설명서에서 “setting row and column names” 예를 복사하여 실행해보라.

어떤 명령어의 설명서가 나타나지 않는다면, (예, “?for” 는 나타나지 않는다.) 사용할 수 있는 찾는 방법이 있다. “help.search(“for”)” 를 입력하면 결과 목록의 가장 위에 “Control(base)” 이 있다. 이것은 “?Control” 를 치면 나온다. (대문자 “C” 에 주의하라.) 아마 좀 더 사용자에게 쉬운 도움말 찾는 방법은 R 웹사이트 <http://www.R-project.org> 의 “Manuals” 에서 “R Reference Index” 를 내려받는 것이다. 이것은 PDF 형식으로 되어있고 그 파일 안에서 CTRL+f를 사용하여 단어를 찾을 수 있다. 만약 XXX 같은 특정 주제를 찾아야 한다면, R 웹사이트에서 찾기 위해 “RSiteSearch(“XXX”)” 를 콘솔 창에 입력하라.

이러한 찾기 기능은 R에서 무언가를 하는 방법을 찾는 데 매우 유용하다. 예를 들어 표준적인 통계 그림에 상자 그림(box plot)이 있다. 그러나 지금은 R에서 이것을 어떻게 하는지 모른다. ?box를 입력하면 올바른 설명서를 얻지 못한다. 그러나 콘솔 창에 “help.search(“box”)” 나 “??box” 를 입력하거나, 설명서 색인에서 box를 찾으면 키워드 boxplot이 나타날 것이다. 이것이 올바른 사용 방법이다. (두 방법 모두 해볼 수 있는 예제를 보여준다.) 가끔 예제가 어떻게 작동하는지 분명하지 않다. 그러나 일반적으로 매우 도움이 된다. (예를 들어 ?boxplot 설명서의 끝에 있는 예제는 R이 시작할 때 미리 읽어 들인 “InsectSprays” 와 “OrchardSprays”, “ToothGrowth” 라는 자료를 호출한다. 그렇다고 이것

이 콘솔 창에 예제를 복사하면 예제가 작동하지 않는다는 것을 의미하지는 않는다. 그러나 숫자가 어디에서 왔는지 분명하지 않으며, 콘솔 창에 자료의 이름을 입력해야만 자료에 담긴 내용을 볼 수 있다.)

그런데 모든 R 명령어가 같은 방식으로 작동한다는 것을 지적해야겠다. 이들은 많은 옵션을 가지며, (해당 설명서에서 “Arguments” 목록을 보라.) 원하는 결과를 정확히 얻기 위해 옵션을 변경한다. 어떤 명령어는 옵션이 매우 많으며, 이것이 R이 메뉴 방식이 아닌 이유라는 것을 알게 될 것이다. 이 많은 옵션을 갖는 메뉴를 만드는 것은 아주 불가능할 것이다. R과 같은 명령어 방식 소프트웨어가 메뉴 방식 소프트웨어보다 사용자 친화적이지 못하지만, 메뉴 방식은 훨씬 제한된 옵션을 가지며, 이것은 독자가 원하거나 해야 할 것을 할 수 없다는 것을 의미한다는 데 모든 사람이 동의할 것이다.

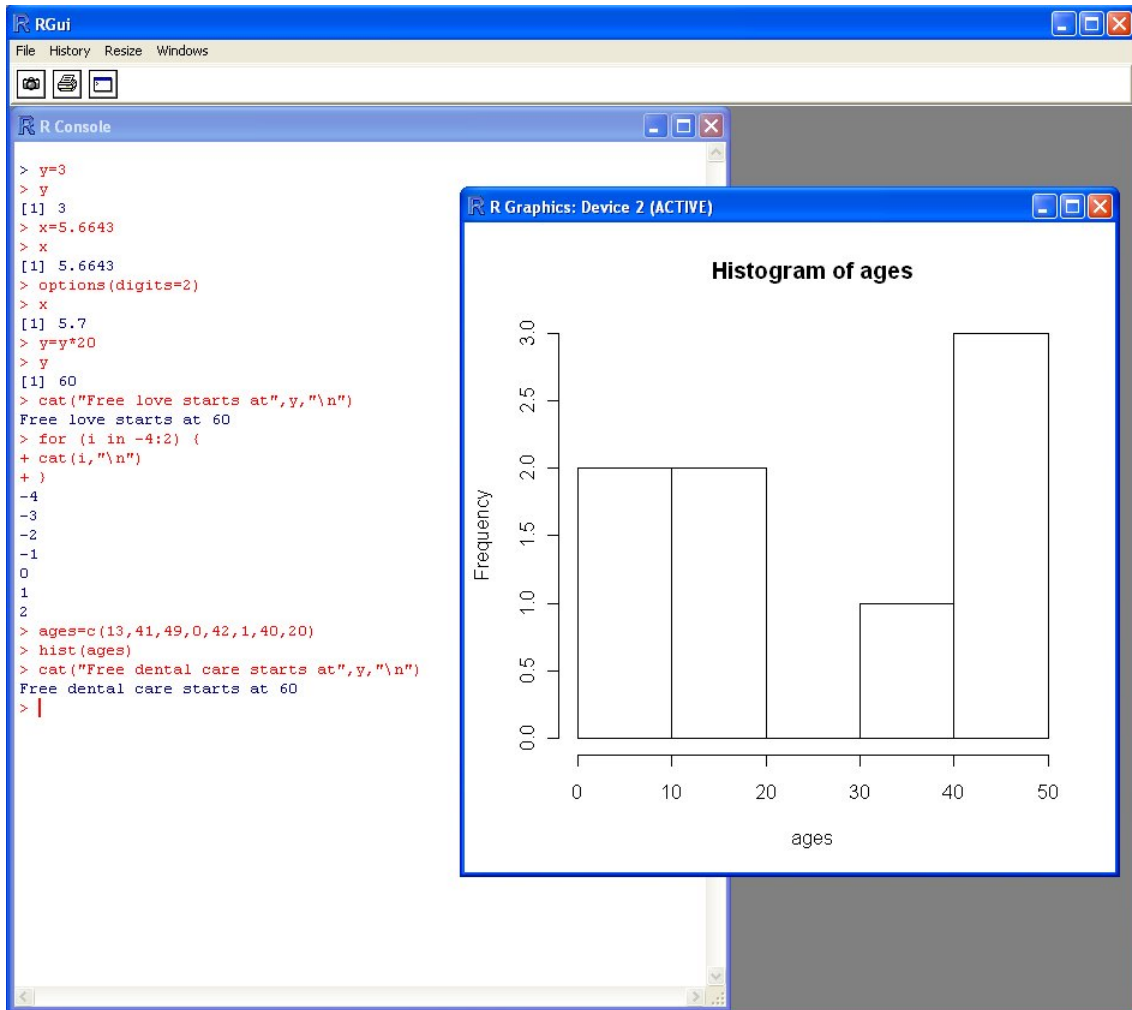
#### 4 콘솔 창에 직접 명령어 입력하기

그런데 독자가 이 짧고 읽기 쉬운 절을 좋아하기를 바란다. 나는 독자가 R을 사용하기 위해 알아야만 하는 것만 이야기하려고 한다. 모든 사람이 따를 것으로 확신하며 Q1과 Q2, Q3의 답을 이야기하는 것이 좋겠다. Q1에서는 단순히 “1:5”를 “5:1”로 바꾸는 것이다. Q2에서는 두 개의 맴돌이(loop)가 있어야 하는데 처음은 (1:5)로 올라가고, 다음은 (5를 두 번 세는 것을 피하기 위해 4:1)로 내려간다. Q3에서는 예쁜 그래프를 얻을 것이다. (“rpois(100,lambda=5)”는 a Poi(lambda=5) 분포에서 100개 추출을 의미한다. 나중에 이러한 것을 볼 것이다.) 와 윗 행이 1,2,3이고 아래 행이 11,12,13인 2x3 행렬이다. 이들 답을 얻은 사람은 별표 10개를 얻는다. (독자는 얻은 별의 수를 알고 있는가?)

다음은, 콘솔 창을 클릭하여 “y=3”를 입력하고 엔터를 쳐라. 이제 “y”를 입력하고 엔터를 쳐라. 또 “x=5.6643”와 “x”를, 다시 “options(digits=2)”와 “x”를, 다시 “y=y\*20”와 “y”를 입력하라. 어떻게 되는가 보이는가? 이와 같이 명령어를 바로 입력할 수 있다. 이제 “cat(“Free love starts at”,y,“\n”)”와 “for (i in -4:2)”를 입력하라. 프롬프트가 “>”에서 “+”로 변한다. 이것은 R이 불완전한 명령어를 발견했고 추가로 입력해야 한다는 것을 의미한다. 따라서 “cat(i,“\n”)”와 “”를 입력하라. 독자는 이해했을 것이다. “ages=c(13,41,49,0,42,1,40,20)”과 “hist(ages)”를 이어서 입력하여 R의 통계적인 면을 맛보자. 또한 “history” 기능이 있어 화살표를 위아래로 눌러서 이전 명령어를 다시 사용할 수 있다. 콘솔 창을 다시 클릭하고 위쪽 화살표를 몇 번 눌러 이 절의 첫 번째 명령어 “cat”이 나오도록 하라. “love”를 “dental care”로 바꾸고 엔터를 눌러 “Free dental care starts at 60”을 얻어라. 물론 이것이 내가 진짜로 말하려고 한 것이다. CTRL+L을 사용하여 콘솔 창을 지워라.

명령어를 직접 실행할 수 있는 기능은 매우 강력하며 스크립트 언어를 사용하는 주요 이유 중 하나이다. (컴파일 언어로는 이렇게 쉽게 할 수 없다.) 이해할 수 없는 많은 명령어 줄을 가진 프로그램을 받았다면 (다음 절에서 일어나는 것 같이), 독자는 콘솔 창에 이를 한 줄씩 복사하고 실행해서 무슨 일을 하는지 볼 수 있다.





## 5 미스터리 문제

다른 도전을 할 시간이다. 이 교재에 딸린 `mystery.r`을 보자. 독자의 임무는 이것이 무엇을 하는지 알아내는 것이다. 프로그램이 무엇을 하는지 알아내려고 할 때 처음 해야 할 일은 그것을 실행하는 것이다. (작업공간에 이것을 저장하고, R을 시작하고, 파일 메뉴에서 `source("...")`를 사용하는 것 등이다.) 두 번째 할 일은 이것을 텍스트 에디터로 열어서 코드와 코드 코멘트에서 이것이 무엇을 하려고 하는지를 알아보는 것이다. 알 수 있다면 별 15개를 추가로 얻게 될 것이다.

Q4 이 `mystery.r` 프로그램이 무엇을 계산하고 마지막에 인쇄물의 NND 열에 무엇을 인쇄하는가?

이 시점에서 많은 사람이 “whoa - what?”을 말할 것이다. 왜냐하면 우리는 갑자기 `sin`, `cos`, `abs`, `sqrt`와 `pi`(수학), 두 개의 제어 맵들이(control loops) (`for ( ... ) ...` 등)과 배열(뒤 따르는 `[]`에 어떤 것이 있는 변수들)이 있는 것으로 뛰어들었기 때문이다. (아직) 당황하지 마라. 그냥 익숙하지 않은 명령어를 찾아서 그 키워드가 무엇을 하려고 하는지를 설명서에서 확인하라. (“`?sqrt`”를 입력하고 마지막에

나온 예제를 콘솔 창에 복사하여 이것이 무엇을 하는가를 보라.) 그리고, 이것이 도움이 되지 않으면 확실히 알지 못하는 전체 프로그램 줄을 콘솔 창에 복사하여 작동하는지를 보라.

Q5 plot 명령어 바로 전에 “f=jpeg(file=“plot1.jpg”)” 를 삽입하고, abline 바로 뒤에 “dev.off()” 를 삽입하여, 그림이 스크린에 그려지는 대신 작업공간 폴더에 .jpg 그래픽 파일로 나타나게 한다. “?jpeg” 를 입력하여 .bmp 나 .png 파일로 내보낼 수 있는가를 알 수 있다. 이것은 매우 유용하다. 이것은 다른 프로그램에서 쓸 수 있도록 그래프를 저장할 수 있도록 해준다. 이것을 할 수 있다면 별 5개이다. 독자가 너무 똑똑하여 그래프 창에서 오른쪽 클릭하여, jpeg 명령어를 사용하지 않고, 오려붙이기로 이것을, 예를 들어 MS Word로 붙여넣기 한다면, 별을 받을 수 없어서 걱정이다! 다시 돌아가서 내가 말한 명령어로 해보라. 왜냐하면 그렇게 하는 방법을 아는 것이 좋기 때문이다! 놀랍게도 사람들은 종종 너무 영리하다... (:9).

이 멋진 프로그래밍 기술에 대해 어떻게 생각하는가? 머리가 아픈가 아니면 지나치게 쉬운가? 여기 까지 왔다면 지금까지 얻은 별의 개수를 세고 스스로 등을 두드려 격려하라. 도중에 다른 사람에게서 도움을 받을 때마다 별을 하나씩 없애라. 내가 보고 있다.

Q4에 대한 답은 이 교재 말미에 있으며 거꾸로 써져 있다. 그러나 지금 보지 마라. 먼저 해보라.

## 6 벡터

좋다. 잠깐 숨을 돌리자!

이 절은 “vector” 자료 유형에 대해 이야기하는 쉬운 절이다. 벡터 유형은 앞에서 이미 본 숫자의 c( ... ) 리스트이다. 콘솔 창을 클릭하고 “b=10;c=11;d=12” 를 입력하라. (세미콜론은 한 줄에 여러 명령어를 넣을 때 구분자이다.) 이제 “vec=c(3,b,b,8)” 와 “vec” 를 입력하면 vec 가 숫자로 구성된 리스트 (다른 말로 벡터)를 갖는다는 것을 볼 수 있다. 또한 이 숫자를 직접 볼 수 있다. “vec[2]” 와 “vec[4]”, “for (i in 1:4) cat(vec[i],“\n”)” 을 입력하라. (주의. 세 가지 다른 형태의 괄호가 있으며, 이들은 각각 올바른 장소에 있어야 한다.)

이제 “vec2=c(1,2,3,4)” 와 “vec3=vec+vec2”, “vec3”, “vec4=(vec\*vec2)+6”, “vec4”, “vec5=c(rep(3,times=20))”, “vec5” 를 입력하고 이들이 어떻게 되는지를 따라갈 수 있다면 벡터에 대해 알아야 할 매우 많은 것을 아는 것이다.

## 7 행렬

벡터 뒤에는 행렬이 온다. 이것은 세속적이고 전통적인 수학의 행렬이다. 즉 격자 또는 배열 또는 숫자 인 행렬이다. “mat=matrix(c(7,8,2,3,4,5,6,3,2,1,-2,-9),nrow=3,ncol=4)” 과 “mat” 을 입력하라. 이것은 행렬을 보여준다. 이제 “mat2=matrix(c(7,8,2,3,4,5,6,3,2,1,- 9),nrow=3,ncol=4,byrow=TRUE)” 와 “mat2”, “mat3=matrix(7,nrow=3,ncol=4)”, “mat3”, “mat4=matrix(3:14,nrow=3,ncol=4)”, “mat4”, “mat2+mat3”, “mat4\*2” 를 입력하라.

벡터와 행렬에 대한 이 모든 명령어는 콘솔 창에서 작동한 것과 같은 방식으로 프로그램에서 사용할 수 있다. 자발적으로 행렬로 무언가를 해보기에는 너무 어렵다고 느낀다면 그것으로 충분하다. 여기서는 더 이상 행렬을 다루지 않는다.

## 8 기타

해볼 만한 몇 가지 유용한 명령어들이 있다. 인쇄할 때 소수점 자리수를 조절하는 방법이 있다.

```
cat("x=",formatC(10.5,width=8,format="f",digits=3),"\n")
```

문자열을 조작하는 함수이다.

```
substring("abracadabra",3,8)           # 3 번째에서 8 번째 문자를 추출
nchar("abracadabra")                  # 문자의 수를 센다
strsplit("abracadabra","r")           # r 이 나타날 때마다 분리
strsplit("abcd",split="")             # 분리된 문자들로 분리
paste("ab","c","d",sep="")            # 둘을 결합
paste(c("a","b","c"),collapse="")     # 벡터 원소를 결합
```

그리고 날짜 형식을 배우려면 이 명령어들을 해보라.

```
dd=as.Date(c("2003-08-24","2003-11-23","2004-02-22","2004-05-03"))
diff(dd)
as.Date("1/1/19 60",format="%d/%m/%Y")
as.Date("1:12:1960",format="%d:%m:%Y")
as.Date("1960-12-1")-as.Date("1960-1-1")
as.Date("31/12/19 60",format="%d/%m/%Y")
as.integer(as.Date("1/1/197 0",format="%d/%m/%Y"))
as.integer(as.Date("1/1/2 000",format="%d/%m/%Y"))
dec1=as.Date("2004-12-1")
format(dec1,format="%b %d %Y")
format(dec1,format="%a %b %d %Y")
strptime("2005-02-07 14:00",format="%Y-%m-%d %H:%M") # Date+time 에는 strptime 를 사용하라.
```

```
times=strptime("07/02/2005 14:00",format="%d/%m/%Y %H:%M",tz="America/Lima")+1800*(0:16)
plot(x=times,y=runif(length(times),min=8,max=12),bty="l")
```

(여기서 tz 옵션은 표준 Linux/UNIX 이름 관행을 따라서 다른 시간대를 설정하도록 해준다. <http://www.vmware.com/support/developer/vc-sdk/visdk400pubs/ReferenceGuide/timezone.html>)

## 9 자료를 그래프로 그리기

R은 많은 재주 있는 plot 함수를 가지고 있다. 이것이 사용하는 방법을 배우는 중요한 이유 중 하나이다. 여기 두 예가 있다. 다음을 콘솔 창에 입력하여 기본적인 플롯을 그려라.

```
years=c(2004,2005,2006,2007,2008)
rainfall=c(1500,1300,1800,1350,1950)
plot(x=years,y=rainfall)
```

좋다. 그러나 plot 이 R의 기본설정으로 그려졌다. 이것은 대부분 과학 학술지가 고집하는 그림의 표준 지침에 맞지 않는다. (예를 들어 the Journal of Ecology 홈페이지의 “Figures” 을 보라. <http://www.blackwellpublishing.com/submit.asp?ref=0022-0477> 이것은 “그림은 박스로 둘러지지 않아야 한다...” 로 시작한다. “chartjunk” 에 대해 들어본 적이 있는가? <http://en.wikipedia.org/wiki/Chartjunk>). 설명서에서 ?plot 와 ?par 를 보면 포맷 옵션을 바꾸는 방법을 찾을 수 있다. (거의 항상 bty=“l” 과 las=1 이 필요하다.) 그리고 다음 예는 같은 자료를 다르게 보여준다.

```
thisdata=data.frame(years,rainfall)
bestfit=lm(rainfall~years,data=thisdata)
x11();plot(thisdata,main=“Annual Rainfall”,xlab=“recorded
years”,ylab=“mm”,bty=“l”,las=1,ylim=c(0,2000),pch=4,sub=paste(“Best Fit
line is y = (”,bestfit$coefficients[2],“) x +
(”,bestfit$coefficients[1],“)”)
lines(thisdata)
abline(bestfit,lty=2)
```

그리고 데이터 레이블이 있는 그림이 있다.

```
sp=c(“Alchornea costaricensis”,“Alseis blackiana”,“Annona spraguei”,“Apeiba
aspera”,“Cordia alliadora”,“Sapium caudatum”)
seedmass.mg=c(38.5,0.12,40.4,14.2,2.9,64)
leaflifetime.mo=c(5.3,10.2,3.9,5.9,8.5,5.3)
posns=c(4,3,2,1,1,1) #1=below, 2=left, 3=above, 4=right
x11();plot(x=leaflifetime.mo,y=seedmass.mg,xlim=c(0,12),xlab=“Leaf lifetime
in months”,ylab=“Seed air-dry mass in mg”,bty=“l”,las=1,log=“y”,type=“n”)
points(x=leaflifetime.mo,y=seedmass.mg,pch=4)
text(x=leaflifetime.mo,y=seedmass.mg,labels=sp,pos=posns) # “,offset=-0.305” 를 추가하고 points
명령어를 제거하면 데이터 레이블이 실제로 정확히 점을 나타낸다.
```

R 본부의 어떤 뛰어난 사람이 독자나 나와 같은 사람들을 위해 멋진 그래픽 예제를 만들기로 하였다. 독자는 이를 꼭 보라. “demo(graphics)” 를 입력하고 ENTER 나 RETURN 을 계속 눌러 전체를 돌아보라. 또한 R 그래픽의 웹사이트 <http://addictedtor.free.fr/graphiques/thumbs.php> 와 많은 기본적인 예제에 대한 <http://zoonek2.free.fr/UNIX/48R/04.html> 를 보라. 축 라벨에 수식을 넣는 방법은 demo(plotmath) 를 보라. R 은 trellis/lattice plot (패키지 “lattice”: 이 패키지가 어떤 것인지는 다음 절을 보라, <http://www.stat.auckland.ac.nz/~paul/RGraphics/chapter4.pdf>), 3D 그래픽 (패키지 “rgl”, <http://rgl.neoscientists.org/gallery.shtml>), 동적 시각화 (GGobi와 연결하는

패키지 “rggobi”, <http://www.ggobi.org/>), 등을 (<http://www.math.yorku.ca/SCS/Gallery/>도 볼 가치가 있다.) 포함한 많은 고급 plot 을 만들 수 있다. 그러나 이들은 지금으로서는 너무 나간 것이다.

이 예제들의 모든 다양한 plot 명령어의 설명서를 모두 보고 예제를 해보도록 하라. 이것이 이 모든 인상적인 plot 을 만드는 방법을 배우는 유일한 방법이다.

## 10 패키지

많은 통계 분석을 하려면 R의 ‘packages’ 시스템을 알아야 한다. 콘솔 창에 “library()” 를 입력하여 R 을 설치할 때 컴퓨터에 어떤 패키지가 설치되었는지 알아보라. 이제 “search()” 를 입력하여 이미 불러온 패키지의 목록을 얻을 수 있다. (설치된 패키지라고 해서 항상 읽어 들일 필요는 없다.)

“methods” 같은 패키지는 이미 설치되었으며 기본적으로 불러온다. 그러나 “date” 는 설치만 되며, 사용하고 싶을 때는 “library(date)” 또는 “require(date)” 를 입력하여 불러온다. 무슨 뜻인지 보려면 date 를 불러오기 전과 후에 “date.mdy(sdate=15000)” 명령어 (이는 25-JAN-2001 을 주어야 한다.) 를 입력해보라.

아직 설치되지 않은 패키지를 사용하고 싶으면, 음 .... 솔직히 말해서 독자는 지금 R의 초보자 강좌를 읽고 있기 때문에 설치되지 않은 패키지를 사용할 어떤 이유도 없을 것이다. 그렇지만 네가 고집을 피운다면, 좋다. 콘솔 창에 “install.packages(“nlme”)” 를 입력하여 지리적으로 가까운 미러 사이트를 선택하고 어떻게 되는지를 보라 (패키지를 설치하려면 Windows Vista 사용자는 R 을 닫고 관리자로서 다시 R 을 열어야 한다. ( R 아이콘을 오른쪽 클릭하여 이 옵션을 선택한다.)) (많은) 설치할 수 있는 패키지 목록을 보고 <http://www.R-project.org> 에 가서 왼쪽에 있는 CRAN 을 클릭하라. 독자의 근처에 있는 미러 사이트를 선택하고 왼쪽에 있는 Packages 를 클릭하라. (방금 설치한 것을 보려면 이 목록에서 “nlme” 을 찾을 수 있다.) 모든 패키지의 PDF 설명서는 여기 R 웹사이트에 올려있다.

R 웹사이트의 패키지 목록은 특정한 작업을 수행할 수 있는 패키지를 찾을 수 있도록 해준다. 예를 들어 독자가 netCDF 파일 (.nc) 을 다루어야 한다고 하자. netCDF 서식을 읽고 쓸 수 있는 명령어가 있는 R 패키지가 있다. 독자는 목록 페이지에서 이 패키지를 찾아서 설치하고 설명서를 내려받을 수 있는가? (이것은 간단한 문제가 아니다. 왜냐하면 “netcdf” 라는 패키지가 없으며 install.packages(“netcdf”) 는 아무 것도 하지 않기 때문이다.)

## 11 Quadrat-o-phenia

이것은 초보자 강좌이다. 내 생각에 충분히 길다고 생각한다. 따라서 R의 특성 (벡터 연산, read.table() 명령어를 이용하여 텍스트파일에서 자료를 읽기,  $\chi^2$  검증, 막대 그림, 함수의 정의와 포와송 분포에서 표본 추출) 을 보여주는 마지막 예를 들고 여기서 마치려고 한다. quadrats.r 을 보고, 앞에서 말한 설명서 자료를 사용하여 이것이 어떻게 작동하는지를 짐작해보라. (이 프로그램은 작업에 대한 설명을 포함하고 있다.) 이 프로그램을 실행해보고 내가 이 코드를 얼마나 명백하고 간결한 방식으로 작성했는가에 대해 존경을 표하는 것은 놔두고 (또는 한 쪽의 문서에 얼마나 많은 고통을 압축할 수 있는가 하는 의혹으로 바라보거나...), 독자가 마지막 도전으로 다음 것을 했으면 좋겠다.

Q6  $\chi^2$  검증의 다른 예를 찾아라. (전에 교과서에서 보았거나 <http://www.mste.uiuc.edu/patel/chisquare/intro.html> 에서 찾을 수 있다.) 이들은 모두 정확히 같은 계산을 해야 한다. 독자가 교

과서나 웹사이트에서 찾은 문제를 `quadrats.r`을 수정하고 계산해서 같은 답을 얻기를 바란다. 이것을 할 수 있으면 30개의 별이 있다!

`quadrats.r`을 이용해 작업하면서 익숙하지 않은 많은 명령어를 만났을 것이다 이것은 의도적이다. 가능한 한 많은 유용한 명령어를 이들 프로그램에 넣으려고 하였다. 그 이유는 독자가 이 프로그램을 간직하고 사용하기를 바라고 때문이다. 독자도 알다시피 아무도 R 프로그램을 즉흥적으로 써내려갈 수는 없다. 대부분 프로그래머는 유용한 프로그램으로 구성된 라이브러리를 유지하고, 새로운 프로그램을 작성할 때 이 라이브러리에 있는 프로그램으로 시작하여 필요한 일을 할 수 있을 때까지 수정한다. 여기에 나쁠 것은 없다. (C와 FORTRAN 프로그래머는 많은 표준 루틴을 발간하기도 한다. “Numerical Recipes”, <http://www.nr.com>를 보라.) 그리고 이 스크립트는 마찬가지로 독자 라이브러리의 핵심이 될 것이다.

## 12 강좌를 마치며

이 강좌의 끝이다. 독자는 이제 더 이상 R의 초보자가 아니다. 어디로 가야 하는가? 초보자 수준을 지난 사람들에게는 하면서 배우는 것을 대신할 것이 많지 않다. 이것은 힘들고 지루하다! 독자는 원하는 교과서를 살 수 있다. 그러나 사기 전에 R 웹사이트 <http://www.R-project.org>를 먼저 확인해보라. 왼쪽의 Documentation/Other를 클릭하여 화면 중앙의 “contributed documentation”을 보라. 여기에는 공짜로 내려 받을 수 있는 많은 자료가 있다. 예를 들어, 나는 Tom Short의 참조 카드(reference card) <http://cran.r-project.org/doc/contrib/Short-refcard.pdf>를 좋아한다.

요즘 온라인으로 많은 R 강좌가 있다. 예를 들면, (앞에서 언급한) `help.start()`를 보거나 Ruth Ripley의 강좌를 보라. 이 강좌는 초보자 수준에서 시작하여 직접 패키지를 작성하는 방법까지 포함한다. (<http://www.stats.ox.ac.uk/~ruth/Rcourse/APTS.html>), 또는 <http://www.r4all.group.shef.ac.uk/>를(이것은 공짜가 아니다.) 볼 수 있다. 더 찾아보려면 웹을 보라. 예를 들어, 구글에서 (<http://www.google.com>) “tutorial R functions”를 찾아보면 함수들을 매우 잘 설명한 웹페이지가 여러 쪽 있을 것이다. 또한 곤란에 처하면 웹을 찾아보라. 어떤 것을 하는 방법을 찾으려고 애쓴다면, 누군가가 이전에 같은 문제를 만나서 해결책을 웹에 올려놨을 가능성이 99%이다. 예를 들어, R 그림에서 축 레이블을 회전시키는 데 어려움이 있다면, 구글에서 <https://stat.ethz.ch/pipermail/r-help/2000-December/009507.html>를 찾는 데 단지 몇 초가 걸릴 것이다. 어쨌건 나는 여기서 이와 같이 웹사이트에 해결책을 제공한 R 개발자 모두에게 깊이 감사드린다. 그 덕에 여러 번 어려움을 무사히 벗어났다.

지금까지 왔고, R이 사용자 친화적이지 않아서 만족하지 못한다고 느끼는 독자에게는, 도와줄 패키지들이 몇 개 있다. R Commander <http://socserv.mcmaster.ca/jfox/Misc/Rcmdr>, Tinn-R <http://www.sciviews.org/Tinn-R>, JaguaR <http://rosuda.org/JGR>, Brodgar <http://www.brodgar.com>와 (리눅스 전용인) RKWard <http://rkwart.sourceforge.net>를 보라. 그러나 이 소프트웨어 모두가 R의 능력을 확장하는 것이 아니라는 점을 지적하고 싶다. 이들은 R을 사용할 때 좀 더 ‘point-and-click’ 하는 인터페이스를 제공한다. 독자가 R 콘솔 창과 설명서, 명령어를 직접 입력하는 것에 만족한다면 이들을 볼 필요가 없다.

더 많은 정보와 도움을 청할 메일링 리스트는 (“XXX는 어떻게 하나요?”와 같은 일반적인 질문을 메일로 보내지 마라. 대신 “내가 XXX을 하려고 합니다. AAA와 BBB (예제 프로그램 ...)를 해보았



지만 제대로 되지 않습니다...”와 같이 구체적이어야 한다.) R 웹사이트의 링크를 보라. (예를 들어, R의 역사에 대해서는 <http://cran.r-project.org/doc/Rnews/Rnews2001-1.pdf> 를 보라.)

거의 다 되었다. 별에 대해 설명해야겠다. 이 강좌에서 독자가 얻은 별 하나하나에 대해 (최대 70개) 이제 나가서 씨나 나무를 사서 정원에 심어라. 이 방식으로 R을 공부하는 것이 우리 주변의 세상을 푸르게 하거나 아름답게 하는 데 공헌할 수 있다.

행운이 있기를!

Toby Marthews.

추신. 대부분 R 사용자(와 대부분 R 설명서)는  $x$ 에 3을 대입하려면 “ $x=3$ ” 대신 “ $x <- 3$ ”를 입력하여야 한다고 한다. 여기에는 몇 가지 이유가 있다. 그러나 나는 “ $x=3$ ”가 훨씬 직관적이라고 생각한다. (누가 내 프로그램을 보더라도 설명하기가 쉽다) 어쨌든 둘 다 마찬가지로이다.

추추신. R 코드를 작성할 때 내가 저지르는 4가지 오류를 주의하라.

i) `mean(5.4,6.7,2.3)`가 5.4라는 값을 주면 이것을 평균이라고 생각하기 쉽다. 그러나 실제 평균은 (`mean(c(5.4,6.7,2.3))`에서) 4.8이다.

ii) “`log(x)`”가 `log10(x)`가 아니라 `ln(x)`라는 것을 잊지 마라.

iii) `if` 문에서 “`==`”를 사용하는 것을 잊지 마라. (예를 들어, “`if (x=3) { ...}`” 대신 “`if (x==3) { ...}`”라고 입력하라.)

iv) 음수와 비교하지 마라. (예, “`if (xi-3) { ...}`”는 작동하지 않는다: “`if (xi(-3)) { ...}`”로 해야 한다.)

추추추신. 거의 잊었다: Q4의 답은: “`ecnatid ruobhgien tseraen`” rof sdnats “DNN”이다.

추추추추신. 예. 물론 나무를 사야한다! 우린 너를 감시할 방법이 있다. OK, 적어도 괜찮은 화분을 사서 그것을 사랑하라...?