

Sweave를 이용한 문서작성 튜토리얼

Chel Hee Lee

November 22, 2013

Abstract

이 문서는 Sweave를 이용한 통계 보고서의 생성과 L^AT_EX를 활용한 문서의 편집에 관련된 팁들을 담고 있습니다. 이 문서의 소스 Sweave-ko.Rnw는 <http://ihelp.r-forge.r-project.org/>에서 찾을 수 있습니다. 본 문서의 최초작성일은 2008년 11월 01일이며, 이를 복구하여 새로운 내용을 업데이트 하는데 도움을 주신 정우준 <muoe78@gmail.com>님께 감사드립니다. 문서에 대한 질문, 제안 및 수정, 그리고 요청사항은 gnustats@gmail.com으로 보내주세요.

1 기본 작동원리와 실행방법

.Rnw 또는 .Snw 라는 확장자를 가지는 파일들이 R을 이용한 통계보고서를 작성하는데 이용되는 소스파일입니다. 이 소스파일은 .tex라는 확장자를 가지는 L^AT_EX문서를 생성하게 됩니다. 이렇게 얻어진 L^AT_EX문서를 컴파일하여 .pdf 형식의 최종적인 문서를 얻게 됩니다.

1.1 준비물

문서 작성에 필요한 최소한의 소프트웨어는 L^AT_EX사용을 위한 시스템인 텍스라이브 (texlive) 풀버전과 한글을 이용할 수 있도록 해주는 kotex, 그리고 통계소프트웨어인 R¹ 입니다. 이는 우분투 13.10 (November 22, 2013)를 기준으로 작성되었습니다.

```
$ sudo apt-get install texlive-full
$ sudo apt-get install ko.tex
$ sudo apt-get build-dep r-base
```

1.2 간단한 예제

이 문서는 독자가 이미 L^AT_EX문법에 익숙하다는 것을 가정합니다 (만약 L^AT_EX에 익숙하지 않다면, 다음 문서를 참조해주세요). 먼저 ex-ko.Rnw 이라는 파일을 자신이 익숙한 텍스트 에디터 (Emacs 또는 Geany)를 이용하여 다음의 코드를 붙여넣고 생성하도록 합니다.

¹주로 우분투에 처음으로 R을 apt-get install을 이용하여 처음 설치하거나, apt-get install build-essentials을 수행하지 않았던 경우에는 X11() 관련 라이브러리를 찾을 수 없다는 메시지를 보게 되는 경우가 있습니다. 이러한 경우에 sudo apt-get install libx11-dev를 사용하게 됩니다. apt-get build-dep를 이용한 R의 설치의 가장 단순하면서도 안정적인 방법이라고 할 수 있습니다.

```

\documentclass{article}
\usepackage{amsmath}
\usepackage[utf8]{inputenc}
\usepackage{kotex}
\title{나의 첫번째 Sweave 파일}
\author{Chel Hee Lee}
\begin{document}
\maketitle
<<>>=
rnorm(10)
@
\end{document}

```

이제 위의 예제 코드를 읽어보도록 합니다. `\maketitle` 바로 다음 라인부터 `\end{document}` 바로 이전 라인에 해당하는 부분을 제외한 모든 문서작성법이 \LaTeX 와 동일하다는 것을 알 수 있습니다. 그리고, `rnorm(10)`이라는 R 코드가 `<<>>=`와 `@` 사이에 위치함을 알 수 있습니다. 이 코드 블록을 R 코드청크 (R code chunk)라고 하며, `<<>>=` 기호와 `@` 기호는 코드청크의 시작과 끝을 알려주는 것입니다. `<<>>=` 기호내에 다양한 옵션 줌으로서 R 코드청크내의 입력과 출력물을 조절하게 되는데, 이는 다음 섹션 2에서 자세히 설명하도록 하고, 먼저 이 예제 파일을 컴파일하여 pdf 파일을 먼저 생성해 보도록 합니다.

1.3 PDF 문서 생성과 색상변경

다음의 명령어를 예제 소스파일 `ex-ko.Rnw`이 위치하고 있는 디렉토리에서 입력하게 되면 `ex-ko.tex`라는 \LaTeX 소스파일로 변경되어 집니다.

```
SHELL> R CMD Sweave ex-ko.Rnw
```

이것을 R 콘솔상에서 수행하고 싶다면 다음과 같이 입력합니다.

```
R> Sweave("ex-ko.Rnw")
```

실제로 어느 부분이 변경되었는지를 꼭 확인해보세요.

`ex-ko.Rnw` 내의 R 코드청크 부분이 생성된 `ex-ko.tex`내에서 `\begin{Schunk}`와 `\end{Schunk}`라는 인바이런먼트 (environment)내에 새로이 놓여지게 되었다는 것을 알 수 있을 것입니다. 또한, `Schunk` 인바이런먼트 내에서 `\begin{Sinput}`와 `\end{Sinput}`로 R 코드청크의 명령어들을 받아들이는 부분을 처리하는 입력과 관계된 부분 인바이런먼트와 R 코드청크의 명령어를 입력받아 처리한 뒤 결과를 출력해주는 `\begin{Soutput}`와 `\end{Soutput}`이라는 또 다른 부분 인바이런먼트가 존재함을 알 수 있습니다.

```

\begin{Schunk}
\begin{Sinput}

```

```

> rnorm(10)
\end{Sinput}
\begin{Soutput}
[1] -0.01586924 -1.26184744  0.12126180 -1.89415261 -0.32959513 -1.33359802
[7] -0.32337990  0.36252665  1.58149043  1.28022107
\end{Soutput}
\end{Schunk}

```

이렇게 생성된 `ex-ko.tex` 파일은 `pdflatex`를 이용하여 pdf 문서를 생성하게 되는데, 이를 한 번에 하자 한다면 다음과 같이 `-pdf` 옵션을 함께 줍니다.

```
SHELL> R CMD Sweave --pdf Sweave-ko.Rnw
```

간혹, 문서의 인코딩으로 인하여 위의 명령이 실행되지 않을 경우가 있습니다. 다음과 같이 인코딩 옵션 `--encoding=utf8`을 함께 줄 수 있습니다.

```
SHELL> R CMD Sweave --pdf --encoding=utf8 Sweave-ko.Rnw
```

1.4 Schunk 수정하기

간단히 요약하면 Sweave는 기본적인 tex 문서에 Schunk라는 부분을 출력하는 역할을 하는 것입니다. 따라서, 꼭 R 코드창크를 포함하는 .Rnw 소스파일을 생성하지 않아도 일반적인 tex 문서에 다음과 같은 형식으로 Schunk 인바이런먼트를 넣어주기만 하면 Rnw로부터 생성된 것과 동일한 포맷의 문서를 얻을 수 있습니다.

```

\begin{Schunk}
\begin{Sinput}
R input
\end{Sinput}
\begin{Soutput}
R oupput
\end{Soutput}
\end{Schunk}

```

코드 창크를 보다 확실히 알 수 있도록 Schunk 인바이런먼트의 속성을 살짝 변경해 보도록 합니다. 다음의 코드를 프리앰블에 위치시키도록 합니다.

```

\definecolor{SinputColor}{rgb}{0.56,0,0}
\DefineVerbatimEnvironment{Sinput}{Verbatim}{
  formatcom={\color{SinputColor}},
  fontsize=\normalsize,
  fontshape=sl,
  baselinestretch=0.75,

```

```
xleftmargin=2em
}
```

이 코드가 의미하는 것은 Schunk 내의 Sinput의 부분 인바이런먼트를 다음과 같은 옵션을 가지고 수정을 하는 것입니다. Sinput에 보여지게 되는 색상은 RGB의 값이 0.56,0,0으로 정의된 SinputColor를 이용하며, 글씨의 크기는 표준이고, 글씨 모양은 슬랜티드 (slanted) 글꼴으로 이용하고, 라인간의 간격은 0.75로 조정 한 뒤, 마지막으로 왼쪽으로부터의 마진은 2em 만큼 떠서 보여주라는 것입니다. 어떤 독자는 코드블락을 어떤 박스로 둘러 싸고 싶을 수도 있는데, 이는 frame=single 이라는 옵션으로 조정할 수 있습니다. 이와 마찬가지로 Schunk 의 출력부분을 다음과 같이 수정하고 난 뒤 다시 PDF 문서를 생성해 보세요.

```
\definecolor{SoutputColor}{rgb}{0,0,0.56}
\DefineVerbatimEnvironment{Soutput}{Verbatim}{
  formatcom={\color{SoutputColor}},
  fontsize=\normalsize,
  fontshape=sl,
  baselinestretch=0.75,
  xleftmargin=2em
}
```

이제부터 코드 청크는 다음과 같은 형식으로 보여지게 될 것입니다.

```
> rnorm(10)
```

```
[1]  0.41304169  0.27134405  0.61472428 -0.21269816 -0.04126072  0.11233879
[7] -0.34236071 -0.63745509 -0.82817447 -0.99905499
```

어떤 독자 분께서 R에서 사용하는 글꼴이 무엇인지 문의하셨는데, 이는 모노스페이스 (monospace)라는 폰트페이스를 이용하는 것입니다. 즉, 일반문서에 사용되는 로만 (roman)글꼴의 R은 \texttt{R}라는 명령어를 이용하여 R와 같이 보여지게 됩니다.

2 알아야 할 기본 문법

간단한 예제를 통하여 문서를 생성해 보았기 때문에 이번 섹션부터는 R 코드청크의 입출력을 조절해주는 옵션들에 대해서 알아보도록 합니다. 코드청크의 옵션은 일반적으로 아래와 같은 형식과 기본값을 가지고 있습니다.

```
<<label=문자열, echo=TRUE, eval=TRUE, fig=FALSE, include=FALSE, results=verbatim>>=
# R codes
@
```

옵션을 조절하는 방법은 다음과 같이 기억하면 사용이 쉽습니다.

- `label` – “내가 R 코드청크를 넣은 결과물을 문서의 다른 부분에서도 사용할 수 있도록 이름을 붙여야겠어”
- `echo` – “그런데, 코드청크의 결과를 바로 이 위치에서 보여줄까?” 기본값은 `TRUE` 입니다.
- `eval` – “결과를 보여주려면 연산을 해야겠지?” 기본값은 `TRUE`입니다. 만약, 코드청크내의 연산이 버거워서 단순히 코드만 보여주되, 실행하고 싶지 않다면 `FALSE`로 정해주면 됩니다.
- `result` – “R을 통하여 얻은 결과니까 R을 이용했다는 것을 보다 분명히 보여주기 위해서 R의 출력물 형태를 유지해야지”. 기본값은 `verbatim`입니다. R은 `verbatim` 인바이런먼트를 이용하여 결과물을 그대로 전달하기 때문입니다. 만약 `LaTeX`문서형식으로 통일성을 갖춘 문서를 생성하고자 한다면 이 옵션을 `tex`로 변경하면 됩니다. 이 부분은 다소 추가적인 조정이 필요한데 섹션 ??에서 다루도록 합니다.
- `fig` – “가만히 보니까 숫자만으로 내가 보여주고자 하는 분석결과가 설명이 되는가? 꼭 비주얼라이제이션이 필요할까?” 기본값은 `FALSE`입니다. 이 옵션은 특히 바로 아래에 설명하는 `include`라는 옵션과 명시적으로 사용되게 됩니다.
- `include` – “수치적으로 보여주던 그래픽적으로 보여주던 일단 결과물이 현재 작성한 코드청크 부분에 딱 맞추어서 위치해야겠어” 기본값은 `TRUE`입니다. 만약, `fig`의 옵션이 `TRUE`일 때, `include`의 옵션을 `FALSE`로 설정한다면 보다 유연한 문서작성이 가능한데, 이는 섹션 ??에서 설명합니다.

이 옵션들을 `SweaveOpts`라는 명령어를 통하여 문서전체의 통일성을 맞추기 위해서 `.Rnw` 문서의 프리앰블에 위치시켜 사용할 수도 있습니다. 예를들면, 아래와 같이 사용이 가능합니다.

```
\SweaveOpts{echo=TRUE, eval=TRUE, fig=FALSE, eps=FALSE, pdf=TRUE,
             keep.source=FALSE, concordance=TRUE, prefix.string=fig}
```

옵션을 이용한 예를 보여주기 위하여 몇 가지의 경우를 테스트 해 보겠습니다.

`> citation()`

To cite R in publications use:

*R Core Team (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.
URL <http://www.R-project.org/>.*

A BibTeX entry for LaTeX users is

```
@Manual{,
  title = {R: A Language and Environment for Statistical Computing},
  author = {{R Core Team}},
  organization = {R Foundation for Statistical Computing},
```

```

address = {Vienna, Austria},
year = {2013},
url = {http://www.R-project.org/},
}

```

We have invested a lot of time and effort in creating R, please cite it when using it for data analysis. See also 'citation("pkgname")' for citing R packages.

```
> rnorm(10)
```

```

[1]  1.35556764  0.78035690 -0.03286862 -0.30739324 -0.17582204  1.06206926
[7] -1.27323183 -0.93225822  0.88395348  0.38877208

```

```
> options(width=60)
```

```
> rnorm(10)
```

```

[1]  1.20587391 -0.03524565  0.81590084  0.84794971
[5] -0.68020308 -1.36535675 -0.71971676 -0.56484245
[9]  0.35752527  0.95521211

```

이외에도 몇 가지 옵션들이 더 존재합니다. 완전한 내용을 확인하고 싶으시다면 아래의 명령어를 이용하여 도움말 페이지를 참고해주세요.

```
> help("Sweave", package="utils")
```

3 Sweave에서 그래프 Figure 조정하기

Sweave에서 Figure는 아래와 같은 방법으로 할 수 있습니다. 여기에서 `library(cairoDevice)`는 한글을 사용하기 위함이며, 한글글꼴은 `par(family="Korea1deb")`로 선택되었습니다. 간단한 예를 들어봅니다. 첫 번째 랜덤확률변수 X_1 는 평균이 0.5이고 분산이 1²인 정규분포 $f(x_1) \sim N(\mu_1 = 0.5, \sigma_1^2 = 1)$ 를 따르고, 두 번째 랜덤확률변수 X_2 는 평균이 -0.5이고 분산이 2²인 정규분포 $f(x_2) \sim N(\mu_2 = -0.5, \sigma_2^2 = 2^2)$ 를 따른다고 할 때, 이 두 정규분포 곡선이 서로 겹치는 부분을 도식화 해보도록 합니다. 이는 Figure 1에 보여지는 바와 같습니다.

```

> rm(list=ls())
> library(cairoDevice)
> library(scales)
> par(family="Korea1deb")
> x <- seq(-3,3,0.05)
> fx1 <- dnorm(x, 0.5, 1)
> fx2 <- dnorm(x, -0.5, 2)
> plot(range(x), range(c(fx1,fx2)), type="n",
+      main="히스토그램", xlab="X", ylab="확률밀도")
> polygon(c(x,x[1]), c(pmin(fx1,fx2),0), col = alpha("azure2", 0.8))
> lines(x, fx1, col="red", lwd=3)
> lines(x, fx2, col="blue", lwd=3)
> text(x=0, y=0.05, labels="겹치는 부분")
> legend("topleft", legend=c(expression(f(x[1])), expression(f(x[2]))),
+      col=c("red", "blue"), lty=c(1,1), lwd=c(3,3))

```

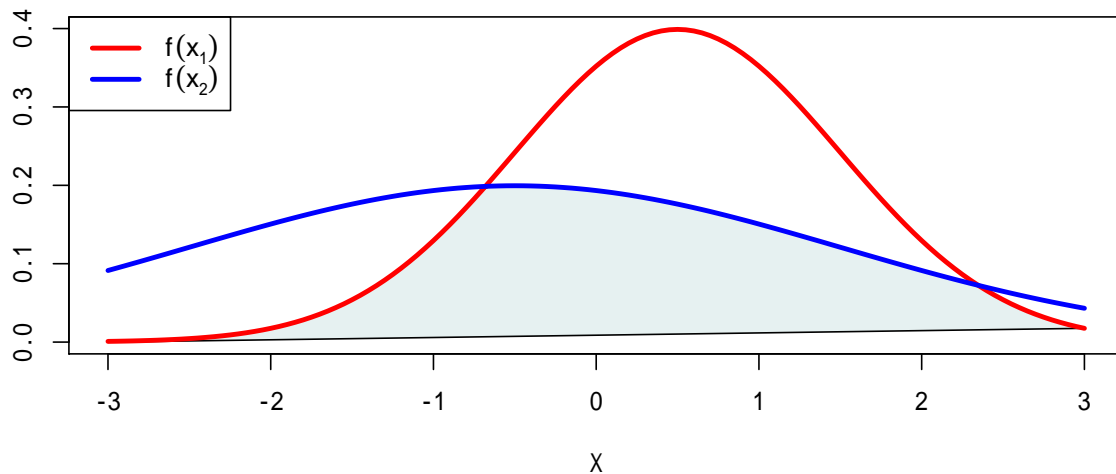


Figure 1: 두 개의 정규확률분포 곡선 $f(x_1)$ 과 $f(x_2)$ 가 서로 겹치는 부분

4 Sweave에서 테이블 조정하기

먼저 일반적인 R의 출력물에 대한 기본환경인 `verbatim`에서 다음 코드창크의 결과를 확인해 봅니다.

```
> tb <- do.call(rbind, .Platform)
> tb
```

```
      [,1]
OS.type  "unix"
file.sep "/"
dynlib.ext ".so"
GUI      "X11"
endian   "little"
pkgType  "source"
path.sep ":"
r_arch   ""
```

이제 \LaTeX 에서 보다 보기 좋은 테이블을 `xtable`이라는 패키지를 활용하여 생성해 보도록 합니다. 결과는 Table ??에서 보는 바와 같습니다.

```
> library(xtable)
> tb <- do.call(rbind, .Platform)
> rownames(tb) <- c("운영체제", "파일경로 구분자", "다이나믹 확장자", "그래픽환경",
+                  "endian", "패키지 유형", "경로구분자", "R 아키텍처")
> colnames(tb) <- c("변수값")
> print(xtable(tb, caption="\texttt{.Platform()} 결과 출력", label="tab:example"))
```

	변수값
운영체제	unix
파일경로 구분자	/
다이나믹 확장자	.so
그래픽환경	X11
endian	little
패키지 유형	source
경로구분자	:
R 아키텍처	

Table 1: .Platform() 결과 출력

5 여러 개의 Sweave 파일을 한데 묶기

L^AT_EX로 일반적으로 대규모의 프로젝트를 진행하면서 많은 부서로부터 올라온 내용들을 각각의 챕터로 구성하게 되는 경우에, 이들을 한데 모아 묶어주는 기능을 하게 되는 `main.tex`이 필요하게 됩니다. 즉, 하나의 문서파일에 모든 것을 다 넣어 관리할 수 없으므로 여러개의 파일들에 쪼개는 것이 더 명료하게 관리를 할 수 있다는 것입니다. 이러한 작업을 하기 위해서 `\input`이라는 L^AT_EX명령어가 있습니다. 이와 동일한 기능을 하는 명령어가 `\SweaveInput{filename}`입니다.

6 그 밖에 알아두면 좋은 패키지들

Sweave를 이용한 문서작성 프로그램에는 `knitr` (<http://cran.r-project.org/web/packages/knitr/index.html>) 와 `markdown` (<http://cran.r-project.org/web/packages/markdown/index.html>)이 많이 애용되고 있습니다. `markdown`과 관련하여 에드워드 강 <ewkang@hotmail.com>님께서 http://ihelp.r-forge.r-project.org/shared_rguide.html에 정리해 두신 한글자료가 있습니다.

6.1 Sweave로 HTML 문서 생성하기

7 업데이트 기록

1. 2008-11-01 문서생성 (2009년 상반기 관리부재로 인한 문서 손실)
2. 2013-11-21 정우준에 의하여 문서복구 후 이철희에게 전달

References

- Leisch, F. (2002a). Sweave: Dynamic Generation of Statistical Reports Using Literate Data Analysis. In Härdle, W. and Rönz, B., editors, Compstat 2002 — Proceedings in Computational Statistics, pages 575–580. Physica Verlag, Heidelberg. ISBN 3-7908-1517-9.
- Leisch, F. (2002b). Sweave, Part I: Mixing R and L^AT_EX. R News, 2(3):28–31.
- Leisch, F. and Rossini, A. J. (2003). Reproducible Statistical Research. Chance, 16(2):46–50.
- R Core Team (2013). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.