

RStudio and .Rnw-files.

The compilation of .Rnw-files requires that both L^AT_EX and R are installed on the computer,¹ and furthermore, the R-package `knitr` must be installed in order for the compilation to work. The example file `Rnw_Example.Rnw` (that you hopefully got a copy of together with this file), will in addition require that the R-package `xtable` is installed.

There are several editors available for R and L^AT_EX,² but few of these can deal properly with .Rnw-files. Among editors that can deal with the .Rnw-files, the optimal editor is Emacs, <http://www.gnu.org/software/emacs/> (with the packages Auctex, <http://www.gnu.org/software/auctex/> and Emacs Speaks Statistics, <http://ess.r-project.org>), but this editor requires some time to get used to (the time invested in learning emacs will pay off quickly, so it's worthwhile to learn how to use it).

For newbies it might be easier to try RStudio, <http://www.rstudio.com>, which have a very simple solution for dealing with .Rnw-files, i.e. it's "push a button". But it's important to be aware of some adjustments that must be made in the default settings before attempting to compile any .Rnw-file.

Please consult the following support-page (and the links found there) <https://support.rstudio.com/hc/en-us/articles/200552056-Using-Sweave-and-knitr>, in order to get up-to-date information about the present defaults in RStudio, since things might have changed after this document was written in January 2016.

The following explanation assumes that the required programs R, L^AT_EX and RStudio all have been installed on your operative system, check the links below if you need to download any of these.

First of all, open the .Rnw-file in RStudio (if necessary, right-click on the file and select RStudio as the program to open it with). RStudio opens up with four sub-windows, and the content of the file, i.e. the source code, should be visible at the left side, upper half. There should be a button at the top of this window with the text **Compile PDF**, but you should not touch this before the adjustments mentioned below as been made.

To properly compile the file `Rnw_Example.Rnw`, the two R-packages `knitr` and `xtable` must be installed. This can be done by using the **Install** button that you find at the top of the 'package window' (right side, bottom half).

After the installation of the above mentioned packages, the next step is the adjustment of some default settings. From the menu at the top, first select **Tools** and then select **Global options**. This triggers a pop-up window where you from the left hand column must select **Sweave**, and then the option after **Weave .Rnw files using:** must be changed from **Sweave** to **knitr**, and then check that the option after **Typeset LaTeX into PDF using:** is given as **pdfLaTeX**. Depending on your operative system (i.e. if you are using Windows), you might also want to modify the option for **Preview PDF after compiling using:** in order to use a previewer that allows an open .pdf-file to be overwritten (see details below).

The button **compile PDF** can now be used to compile the .Rnw-file into a .pdf-file, and a comparison of the .Rnw-file against the finished .pdf-file should hopefully give some insight into how stuff works.

Messed up? Warning: An error will occur if you use **compile PDF** before the adjustment of the default options, and if you after fixing the options tries once more – you will most likely encounter a pop-up window with the title **Sweave Noweb syntax?** stating that the file contains **Sweave**-specific syntax. This triggers an error, and the reason for this error is that the source-file became corrupted during the first attempt at compilation. The culprit for this error is the preposterous behaviour that RStudio uses when it compiles with the **Sweave**-setting, i.e. it adds (without any warning) the following line of code to the source-file `\SweaveOpts{concordance=TRUE}`. This line must be removed manually in order for the file to compile properly.

Change pdf-reader? Some pdf-readers write-protects the .pdf-files they look upon, which might come in conflict with the desire of RStudio to write a new version of the file. This might in particular be a problem for those that use Windows, but it can be rectified by installing (and using) the pdf-reader **Sumatra PDF** from <http://www.sumatrapdfreader.org/free-pdf-reader.html>

Some download links:

R: <https://cran.rstudio.com/>

RStudio: <https://www.rstudio.com/products/rstudio/download/>

L^AT_EX: <https://latex-project.org/ftp.html>

¹ L^AT_EX and R are open-source programs, available for all platforms. See <http://latex-project.org> and <http://cran.r-project.org> for further information, including explanations related to the download-and-install-procedures for the different operative systems.

²See e.g. http://en.wikipedia.org/wiki/Comparison_of_TeX_editors and [http://en.wikipedia.org/wiki/R_\(programming_language\)#interfaces](http://en.wikipedia.org/wiki/R_(programming_language)#interfaces).

Some additional comments related to R:

It's highly recommended that you store the (non-trivial) code you use in scripts. By doing this, you avoid starting from scratch every time you need to do an analysis, and it enables you to recycle previous code when new similar situations arise. Furthermore it's paramount to have your code available if you're asked to reproduce some analysis.

The importance of reproducibility (and other aspects related to computations based upon a set of data) can be found in Sandve GK, Nekrutenko A, Taylor J, Hovig E (2013) Ten Simple Rules for Reproducible Computational Research. PLoS Comput Biol 9(10): e1003285. doi:10.1371/journal.pcbi.1003285

A script file for R is simply a text-file that should end with ".R" or ".r", where the version ".R" (capital letter) is the one many advanced R-users recommend. It's possible to save a text-file directly from R, but when writing a script it will in general be simpler to use an editor like RStudio.³

Use the function `source(<filename>)` in order to get the content of a script into the workspace of R. Note that `<filename>` can either be a full path from the root directory or simply the name of the file. If only the name of the file is used, then `source()` will look in the current working directory. If you use R (or RStudio) to open a file, then the working directory will by default be set to the one containing this file. In RStudio it's possible to set the working directory by using the menu `Session → Set Working directory`. Otherwise the functions `getwd()` and `setwd()` might be nice to know about.

Note that the hash-symbol, i.e. "#", in R-code indicates the start of a comment, and this gives you the possibility to document your code. It's highly recommended to use comments, since the code that seems to be obvious the day you write it can look like incomprehensible gibberish for a future version of yourself.

Some computations might take quite a bit of time to perform, and if the resulting object is something you anticipate that you will need later on too, your efficiency will skyrocket if you use `save()` to store these objects into a .Rda file and `load()` to get them back into the workflow later on.

A few R-resources:

The two first sections of "A (very) short introduction to R", by Paul Torfs & Claudia Brauer might be helpful to check out if you worry about how to install R and RStudio on your computer.

A quick search on "R" and "tutorial" will return an abundance of hits, and the number steadily increases. If you're a complete newbie with regard to the R programming language, then the one at <http://tryr.codeschool.com/> might be a good place to start, since it gives a gentle introduction to some of the most basic concepts in R. Another source that can be nice to know about is "twotutorials" (<http://www.twotutorials.com/>), whose slogan is *how to do stuff in r. two minutes or less. for those of us who prefer to learn by watching and listening*.

At the other end of the scale, there's the following book "The R Inferno", whose abstract says *If you are using R and think you're in hell, this is a map for you.* This is a free book that can be downloaded from <http://www.burns-stat.com/documents/books/the-r-inferno/>, and it contains an enormous collection of tips that clarify why things work like they do. This might not be a book for the complete newbie, but it might be worthwhile to take a look upon it to get an idea about the quirks of the R programming language.

A L^AT_EX-resource:

As for R there's an enormous amount of hits if you search for tips on the use of L^AT_EX. The L^AT_EX-part of the file `Rnw_Example.Rnw` is rather simple, and only contains the most basic stuff needed in order to function as a basic reference when writing a report to hand in (e.g. a compulsory exercise). If you want to do some more advanced stuff, then the L^AT_EX-WikiBook, <https://en.wikibooks.org/wiki/LaTeX>, might be a good place to check out.

Alternative formats:

It's possible to use `knitr` to insert the output from R into other formats than L^AT_EX, e.g. Markdown or HTML. Please read <http://yihui.name/knitr/demo/minimal/> for further details.

³ If you want to use scripts with Norwegian characters like æ, ø, å – then you must ensure that the encoding UTF-8 is used. In RStudio, the encoding can be adjusted from the menu `T Tools → Global Options → General → Default text-encoding`: