

How to run the `rodeo` examples

Updated April 27, 2016 by david.kneis@tu-dresden.de

Contents

1	Installation of R and auxiliary tools	1
1.1	Install R	1
1.2	Install auxiliary tools	1
2	After the installation	2
3	Installation of R-packages	3

1 Installation of R and auxiliary tools

Extensive help on the installation of R and related utilities can be found on <https://cran.r-project.org/doc/manuals/r-release/R-admin.html>. Please consult this document if the following short instructions are insufficient.

1.1 Install R

Make sure that you really need to install R. If this is the case (first install or upgrade from an old version), go to <https://cloud.r-project.org/> and follow the download links in the box at the top of the page. Linux users probably want to use a package manager instead.

1.2 Install auxiliary tools

Although the `rodeo` package itself does not need compilation, it requires the respective compile/build tools when it is used. This is because of `rodeo`'s built-in Fortran code generator. Before installing any tools, check whether a current version is already installed.

On a Linux system, one typically needs to install the GNU compiler collection, including `gfortran`. See the section 'Essential and useful other programs under a Unix-alike' on <https://cran.r-project.org/doc/manuals/r-release/R-admin.html>.

Windows users need to install the so-called `Rtools` from <https://cran.r-project.org/bin/windows/Rtools>. Chose the version that is compatible with the installed R version. Please read the section 'The-Windows-toolset' (currently appendix D) on <https://cran.r-project.org/doc/manuals/r-release/R-admin.html> to circumvent typical pitfalls during and after installation. In particular, I recommend to

- install into a directory whose name does not contain blanks.
- let the automatic installer edit the `PATH` environment variable, if the option is there. This may fail, however, if you don't have administrator privileges (Sect. to 2 describes a solution for that case).

2 After the installation

On Windows, make sure that the directories where R and the Rtools were installed are **actually** included in your PATH environment variable. In addition, the order in which the directories are listed is **essential** as pointed out in section 'The-Windows-toolset' of <https://cran.r-project.org/doc/manuals/r-release/R-admin.html> (currently appendix D).

There are several ways to check the contents of the PATH variable on Windows:

- type `Sys.getenv("PATH")` at the R prompt.
- type `echo %PATH%` into a CMD terminal (the DOS-like black box).
- find the menu item to view/edit environment variables. The place differs between versions of Windows. Even worse, the menu can be accessed in different ways and only one way appears to work without administrator privileges. As a normal user on Windows 7 go to: **Control panel** → **User accounts** → **User accounts** → **Change my environment variables**.

Note that only the last option (i.e. the menu) can be used to **permanently edit** the PATH variable. As a normal user, you can only create/edit the user-variable PATH (and not the system-variable of the same name). Also note that changes to the PATH variable will not instantly be visible/active in other programs (e.g. R, CMD). Before you can see/use the altered PATH settings, you need to restart those programs.

Below you can find two examples for the contents of the PATH variable. The shown version numbers were up-to-date on 2016-04-26.

*Example on Windows 10, installed **with** administrator privileges*

```
C:\myPrograms\Rtools\bin;  
C:\myPrograms\Rtools\gcc-4.6.3\bin;  
C:\myPrograms\R\R-3.2.5\bin
```

*Example on Windows 7, installed **without** administrator privileges*

```
C:\Windows\System32;  
C:\Windows;  
  <several other system direcories>  
C:\Users\myName\Documents\Rtools\bin;  
C:\Users\myName\Documents\Rtools\gcc-4.6.3\bin;  
C:\Users\myName\Documents\R\R-3.2.5\bin
```

In the latter example, the system-variables seem to precede the user-variables. Nevertheless, things work fine.

If everything was set up properly, the two commands

```
R CMD SHLIB --help  
gfortran --help
```

should show some usage info when entered in a CMD terminal on Windows (or bash on Linux).

3 Installation of R-packages

The following packages are required to run the examples:

<code>rodeo</code>	The code generator
<code>deSolve</code>	Numerical solvers for differential equations
<code>readxl</code>	Reads spreadsheet data (.xlsx files)
<code>lhs</code>	Latin hypercube sampling methods

The packages are all available on CRAN (<https://cran.r-project.org/>) and installation is most conveniently done from within R, using

```
install.packages(c("rodeo", "deSolve", "readxl", "lhs"))
```

The installation may take some time because dependent packages are installed along with the above-mentioned ones.

The latest development version of `rodeo` can also be installed directly from the source code repository using the `devtools` package.

```
library("devtools")
install_github("dkneis/rodeo")
```