

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	2

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 this is necessary. For example, you could enter the command

```
gfortran --version
```

at a terminal. If this succeeds, i.e. shows a version info, the required tools may already be there. Otherwise, see below:

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).

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). For example, on a Windows 8 system, the first 3 entries of the PATH variable should be (in that order):

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

assuming that you both **R** and **Rtools** were installed to a custom folder **myPrograms**. The shown version numbers are/were up-to-date on 2016-04-26.

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).
- navigate to the jungle of control settings until you hopefully find the menu item where you can view/edit environment variables. The place differs between versions of Windows.

In order to **permanently** edit the PATH variable, you probably need to find and use the respective menu. 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.

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:

rodeo	The code generator
deSolve	Numerical solvers for differential equations
readxl	Reads spreadsheet data (.xlsx files)
lhs	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")
```