Installation of R-Extension (v1.3) on Windows 7 - 64 bit (and really use 64 bit) for NetLogo (v5.0.4) and R (v3.0.1)
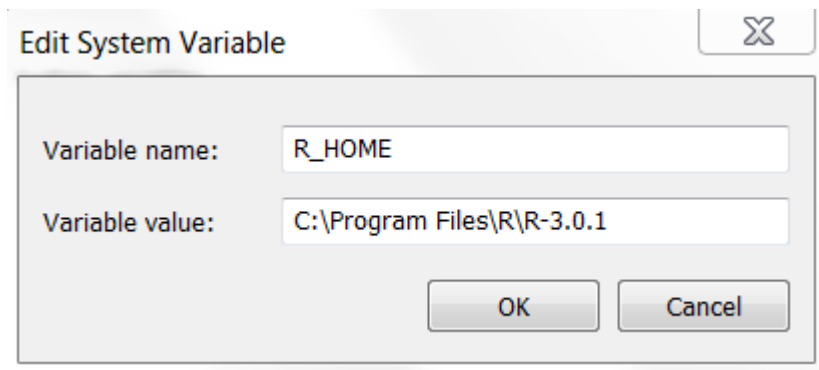
1. **Download and install R** (http://cran.r-project.org/bin/windows/base/), target directory of the installation: C:\Program Files\R\R-3.0.1 (or other version number), Selected components: User installation, Startup options: No
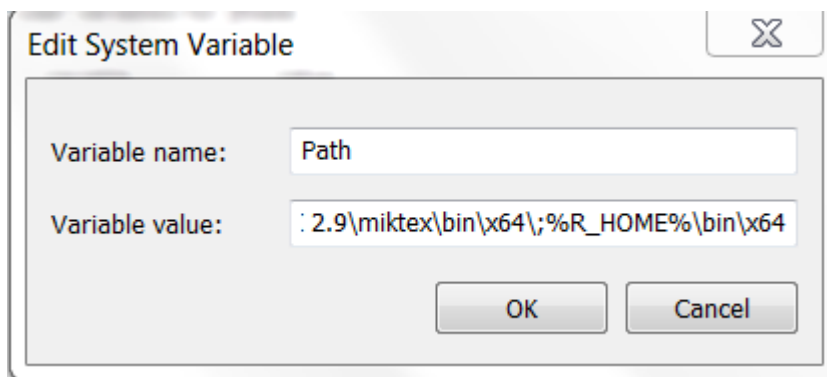
2. **Download and install NetLogo 5.0.4** (standard 32-bit bundled version with Java Runtime Environment) (http://ccl.northwestern.edu/netlogo/download.shtml), Destination Directory of the installation: C:\Program Files (x86)\NetLogo 5.0.4

3. **Set environment variables for R**: an R.dll and R.exe should be found in C:\Program Files\R\R-3.0.1\bin\x64 (depends on the path in step 1),

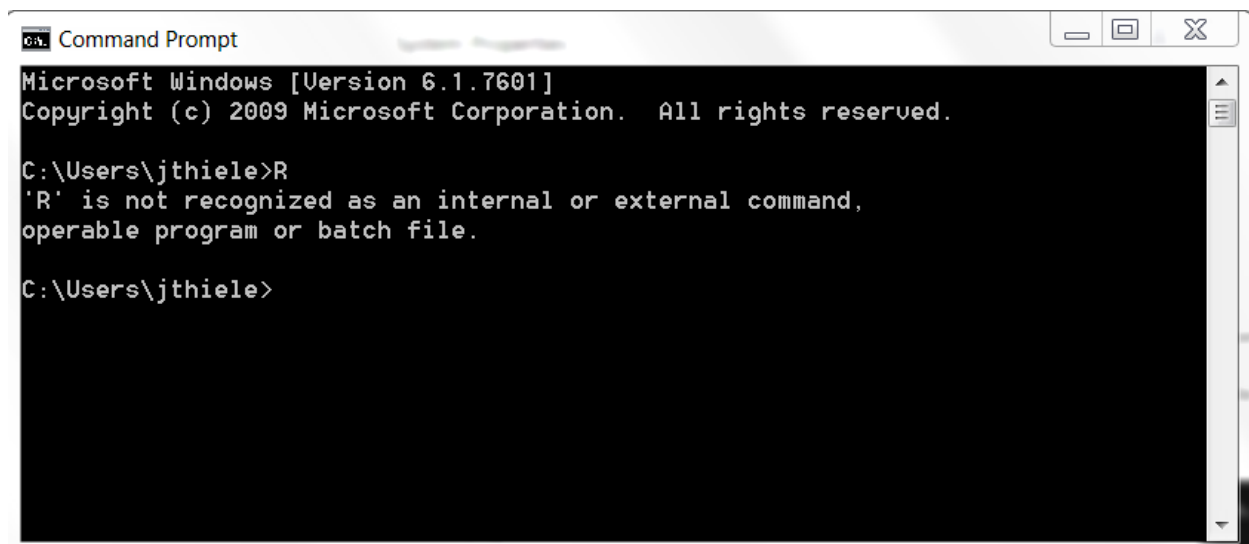Set at first the R_HOME variable to C:\Program Files\R\R-3.0.1.

Edit System Variable   ✕

Variable name: R_HOME

Variable value: C:\Program Files\R\R-3.0.1

                                        OK   Cancel

Now, append the PATH variable by %R_HOME%/bin/x64 (or directly by C:\Program Files\R\R-3.0.1\bin\x64).

Edit System Variable   ✕

Variable name: Path

Variable value: : 2.9\miktex\bin\x64\;%R_HOME%\bin\x64

                                        OK   Cancel

4. **Test the new environment variables** by opening a MS DOS prompt and type R.

If you see this, your Path environment variable is wrong:

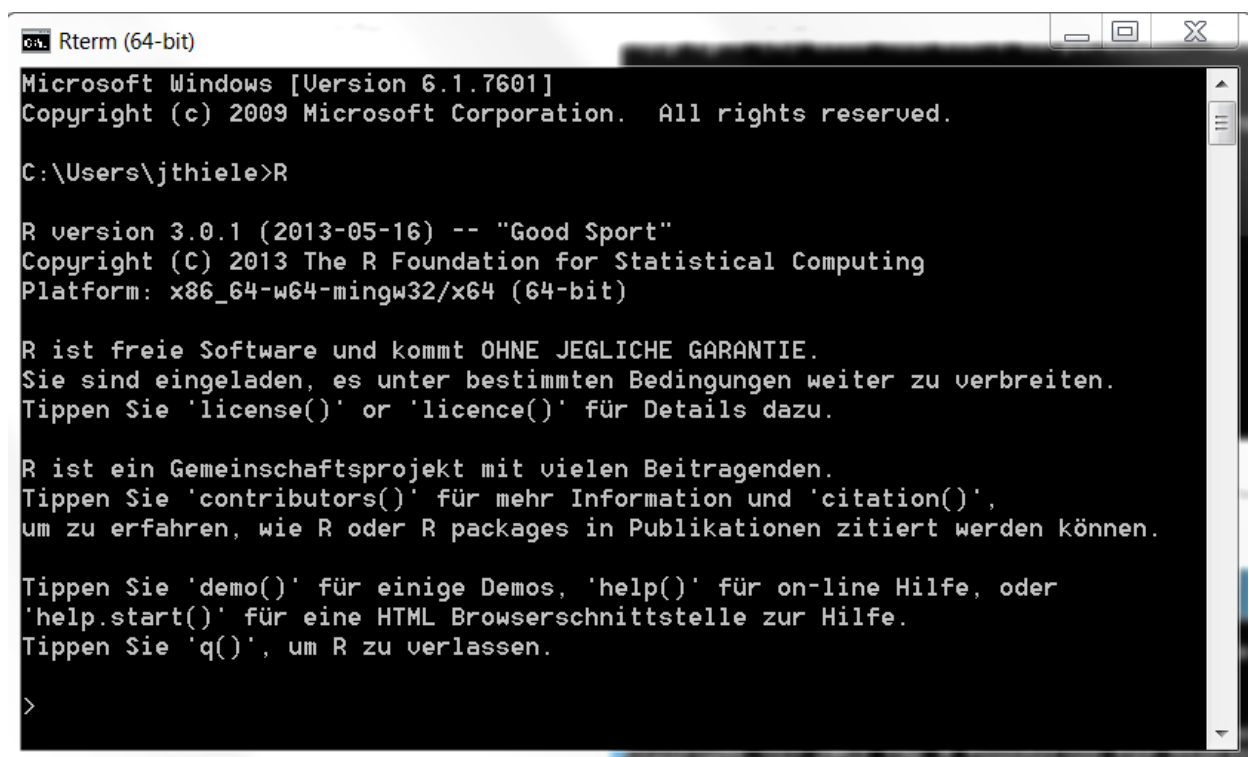If the Rterm opens, everything until now went ok (don't forget to open a new MS DOS prompt after editing the environment variables):


5**. Install rJava**.

Type into the open Rterm command line (make sure in the start up message of the Rterm there was written: *Platform: ... (64-bit)*):

install.packages("rJava")

If you already installed the rJava package in 32-bit mode, please delete it first.
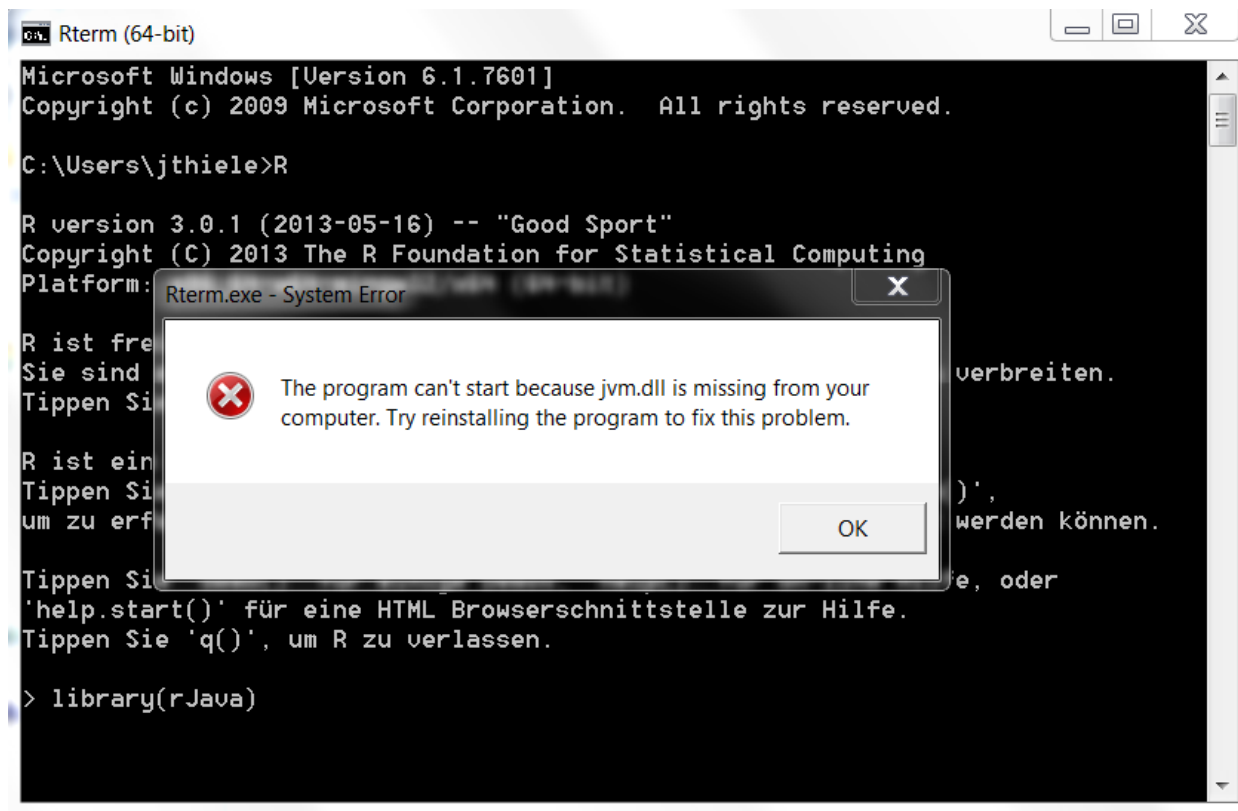
If you haven't started the MS DOS prompt as administrator you will be asked to create a personal library. Click "Yes" and keep the path given in mind. We will need it later (here it is C:\Users\jthiele\Documents/R/win-library/3.0):
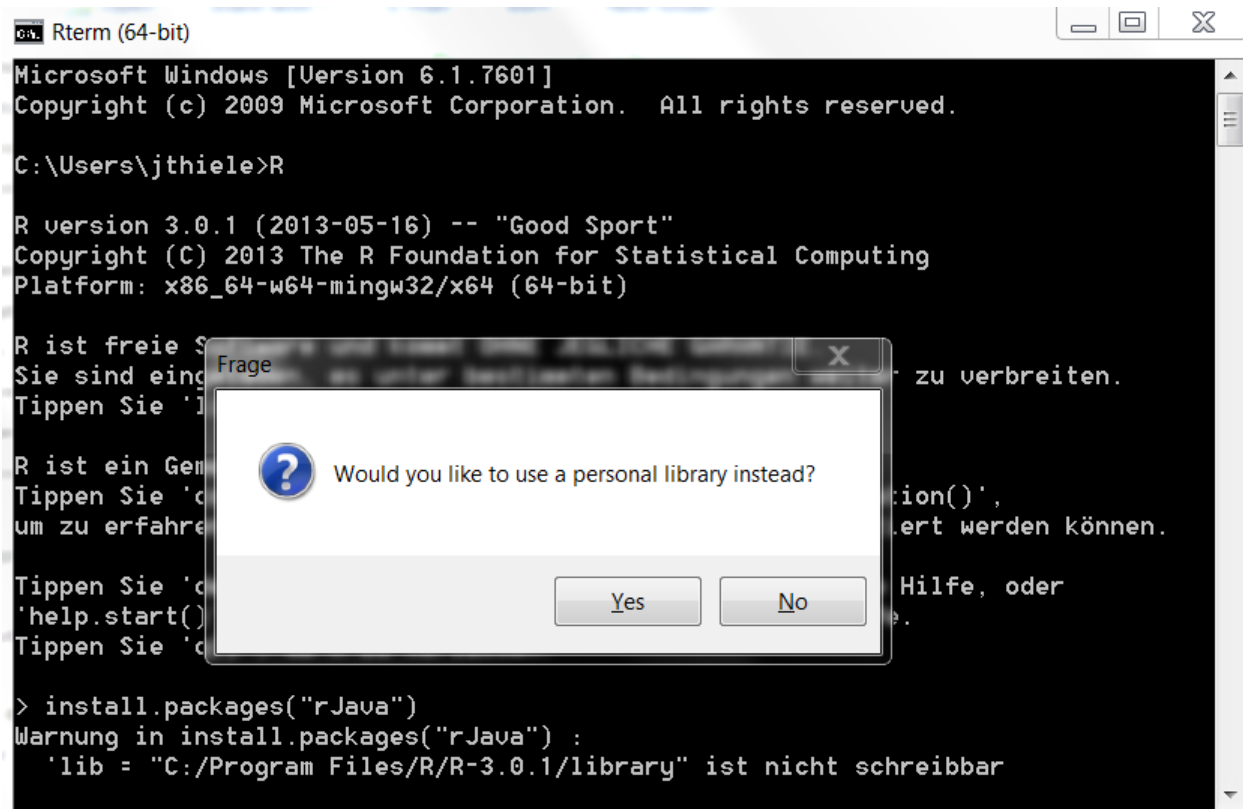
6. **Try to load rJava**.

If you type in the Rterm *library(rJava)* and getting an error message that the *jvm.dll* cannot be found, go to step 7, otherwise go to step 9.

7. In case that **rJava cannot be loaded**:

The reason could be that the *JAVA_HOME* environment variable refers to a 32-bit Java (or no Java was found).

First, check if there is a *JAVA_HOME* environment variable. Then check, if you have a 64-bit Java installed (check your *C:\Program Files* folder for Java installations). If so, go to step 8.

If you don't have a 64-bit Java installed install a new 64-bit Java.

Get a 64-bit Java for example from here:

http://www.oracle.com/technetwork/java/javase/downloads/index.html

Choose the Windows x64 (64-bit) Online or Offline version.

At the beginning of the installation check the box "Change Destination folder" (or similar) to check, where Java will be installed.

It should be something like the following (*C:\Program Files\Java\jre7*):
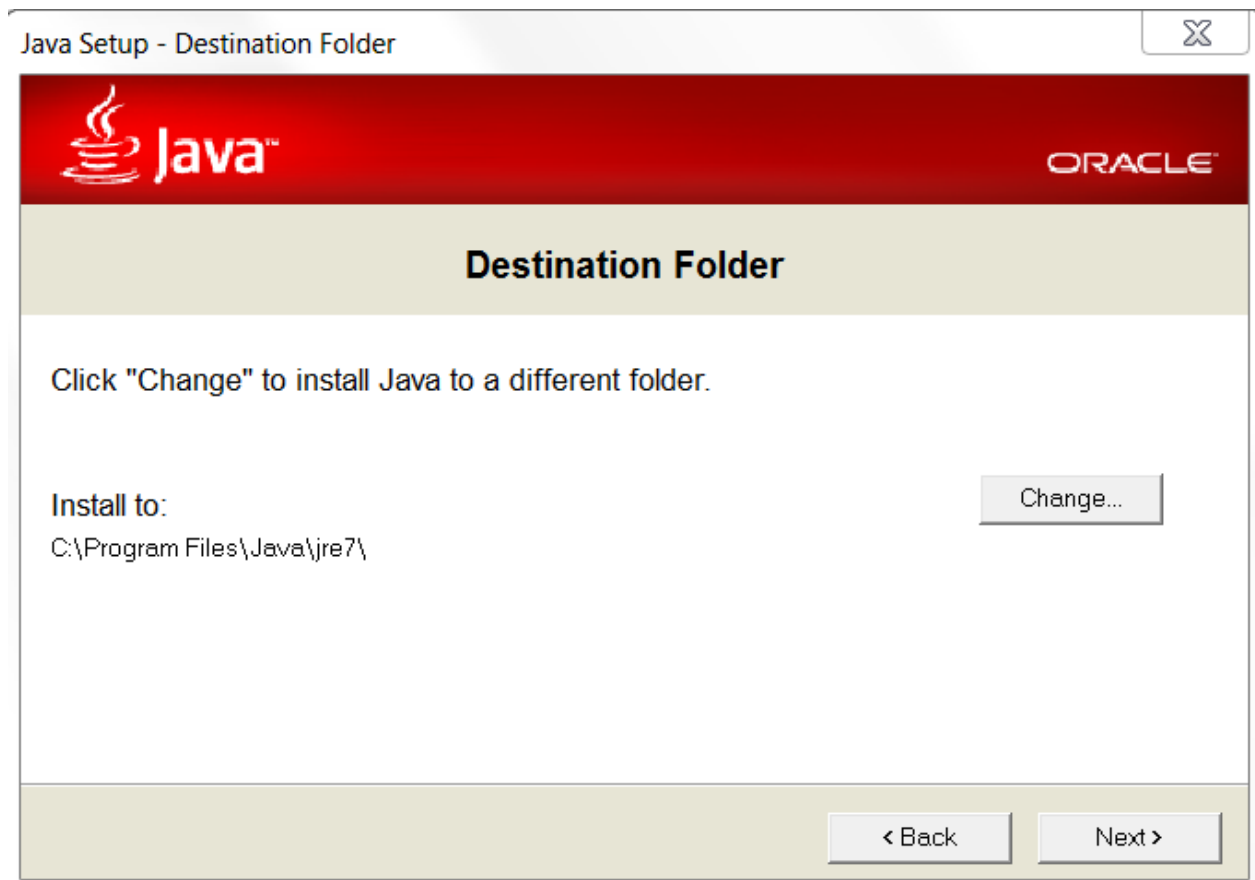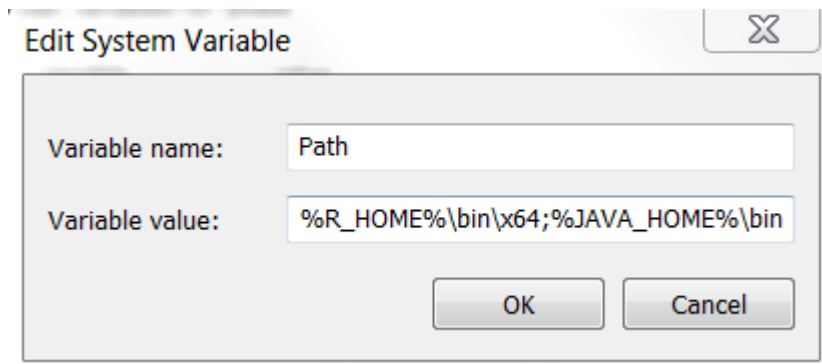
8. Add/Change the *JAVA_HOME* environment variable with the path to the 64-bit Java (maybe not necessary but cleaner).
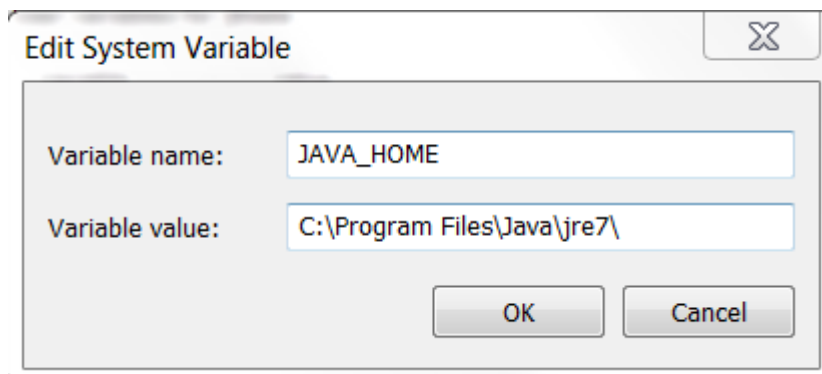
Now, append the *PATH* environment variable by *;%JAVA_HOME%/bin*.

Edit System Variable

Variable name:	Path
Variable value:	%R_HOME%\bin\x64;%JAVA_HOME%\bin

OK   Cancel

Then, open a new MS DOS prompt and start the Rterm again. Type within the Rterm command line again *library(rJava)*. Should work now (means no error message is shown)!

9. See **which Java version is running with rJava**.

Type in the Rterm (after loading rJava: *library(rJava)*):

.jinit()

.jcall("java/lang/System", "S", "getProperty", "os.name")

.jcall("java/lang/System", "S", "getProperty", "os.arch")

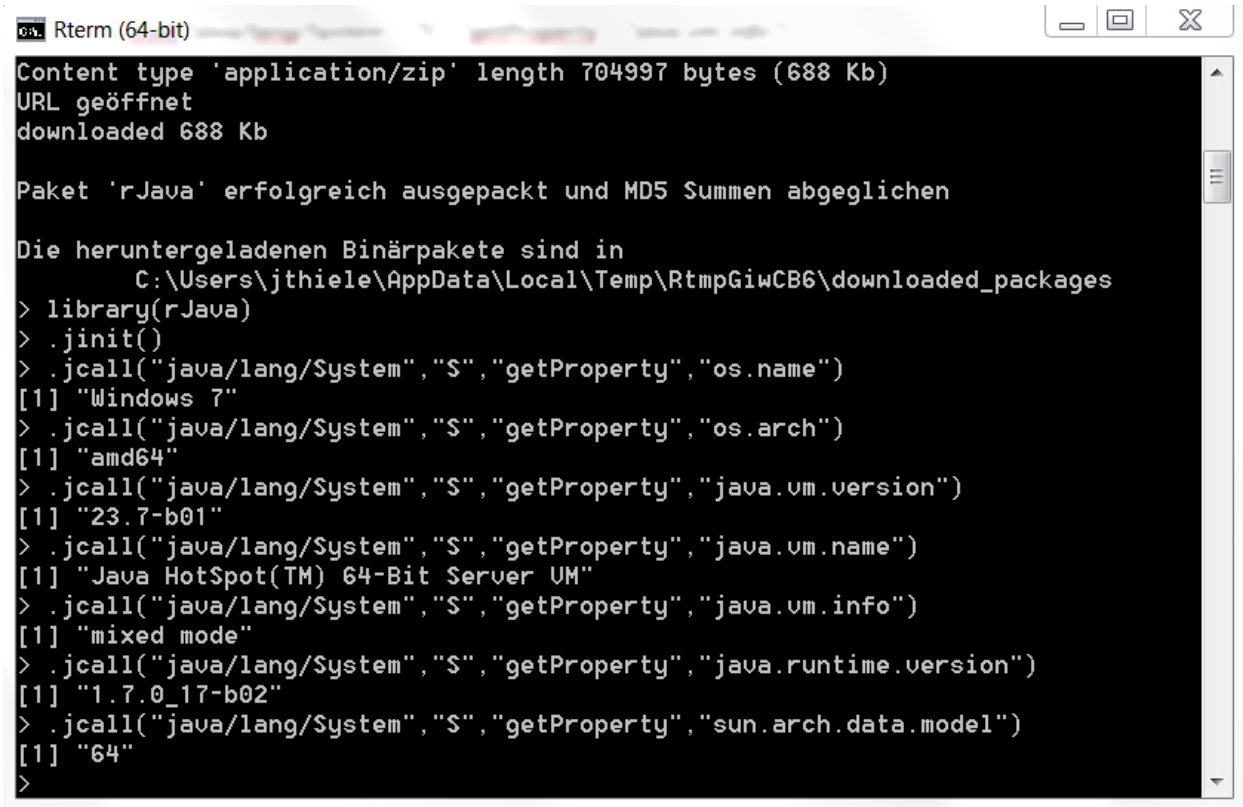*.jcall("java/lang/System", "S", "getProperty", "java.vm.version")*

.jcall("java/lang/System", "S", "getProperty", "java.vm.name")

.jcall("java/lang/System", "S", "getProperty", "java.vm.info")

.jcall("java/lang/System", "S", "getProperty", "java.runtime.version")

.jcall("java/lang/System", "S", "getProperty", "sun.arch.data.model")

```
Rterm (64-bit)
Content type 'application/zip' length 704997 bytes (688 Kb)
URL geöffnet
downloaded 688 Kb

Paket 'rJava' erfolgreich ausgepackt und MD5 Summen abgeglichen

Die heruntergeladenen Binärpakete sind in
        C:\Users\jthiele\AppData\Local\Temp\RtmpGiwCB6\downloaded_packages
> library(rJava)
> .jinit()
> .jcall("java/lang/System","S","getProperty","os.name")
[1] "Windows 7"
> .jcall("java/lang/System","S","getProperty","os.arch")
[1] "amd64"
> .jcall("java/lang/System","S","getProperty","java.vm.version")
[1] "23.7-b01"
> .jcall("java/lang/System","S","getProperty","java.vm.name")
[1] "Java HotSpot(TM) 64-Bit Server VM"
> .jcall("java/lang/System","S","getProperty","java.vm.info")
[1] "mixed mode"
> .jcall("java/lang/System","S","getProperty","java.runtime.version")
[1] "1.7.0_17-b02"
> .jcall("java/lang/System","S","getProperty","sun.arch.data.model")
[1] "64"
>
```
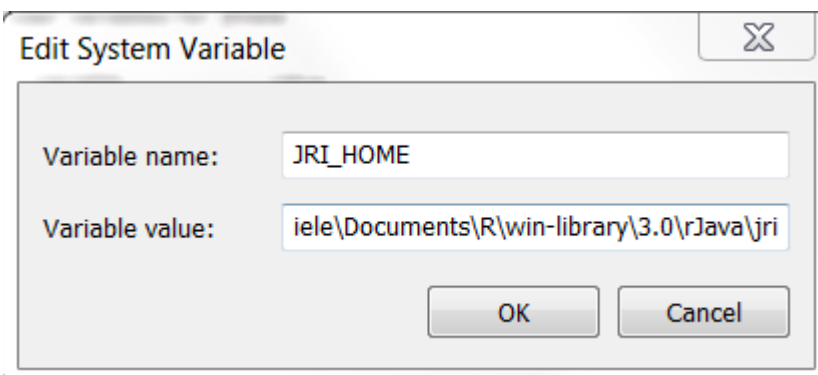
10. **Set JRI_HOME environment variable.**

Now, come back to the path you have in mind from step 5, the installation path of the rJava package.

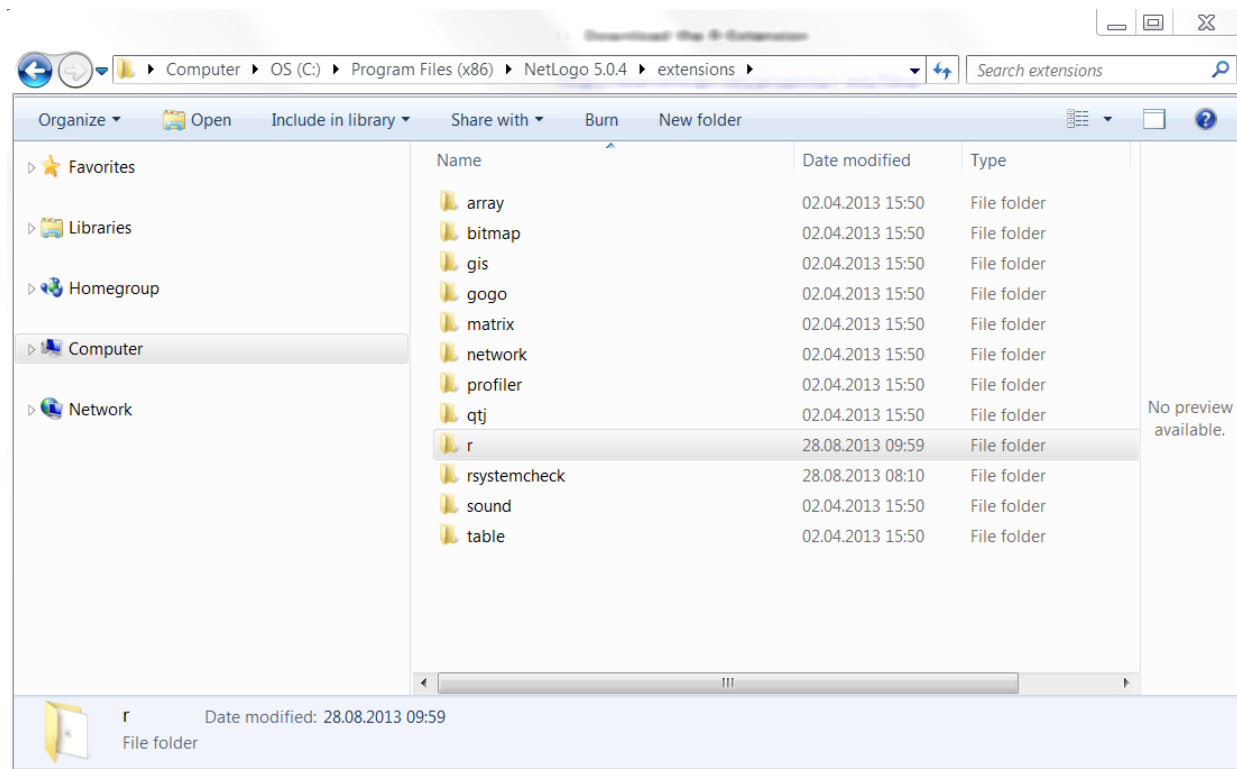Here, it was *C:\Users\jthiele\Documents/R/win-library/3.0*.

We will create a new environment variable with name *JRI_HOME* and value
C:\Users\jthiele\Documents\R\win-library\3.0\rJava\jri.

11. **Download the R-Extension**

(http://sourceforge.net/projects/r-ext/files/).

Extract the zip folder and copy only the folder *r* into the extensions folder of the NetLogo installation
(here: *C:\Program Files (x86)\NetLogo 5.0.4\extensions*).

12. Test it all together!

Start NetLogo 5.0.4 with 64-bit Java, e.g. by starting it from the MS-DOS-Command prompt:

Write into the Procedures Tab:

extensions [r]

and press the Check button (If NetLogo closes immidetly something with the *R_HOME* and *PATH* variable

is wrong). If there is a problem with rJava/JRI you should see an error message.

Otherwise go to the Interface Tab and type into the Command Center:

print r:get "rnorm(10)"

You should see a list of 10 random values.