

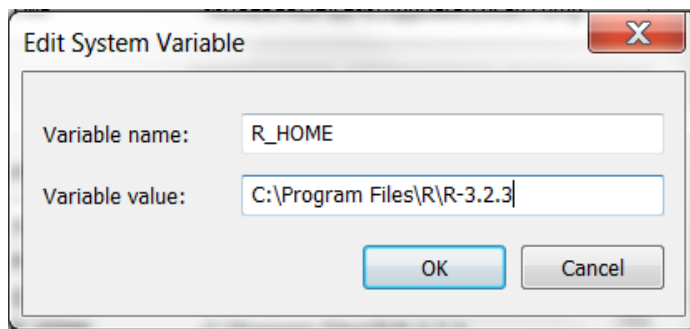
## Installation of R-Extension (v1.4) on Windows - 64 bit for NetLogo 5.3 and R 3.2.3

1. **Download and install R 3.2.3** (<http://cran.r-project.org/bin/windows/base/R-3.2.3-win.exe>), target directory of the installation: C:\Program Files\R\R-3.2.3, Selected components: User installation, Startup options: No

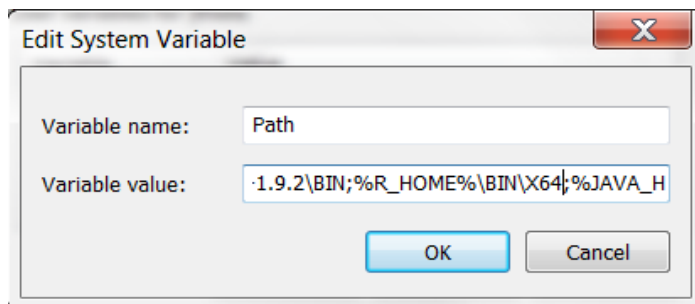
2. **Download and install NetLogo 5.3** (standard 64-bit bundled version with Java Runtime Environment) (<http://ccl.northwestern.edu/netlogo/5.3/NetLogo-5.3-64.msi>), Destination Directory of the installation: C:\Program Files\NetLogo 5.3

3. **Set environment variables for R:** an R.dll and R.exe should be found in C:\Program Files\R\R-3.2.3\bin\x64 (depends on the path in step 1),

Set at first the R\_HOME variable to C:\Program Files\R\R-3.2.3.

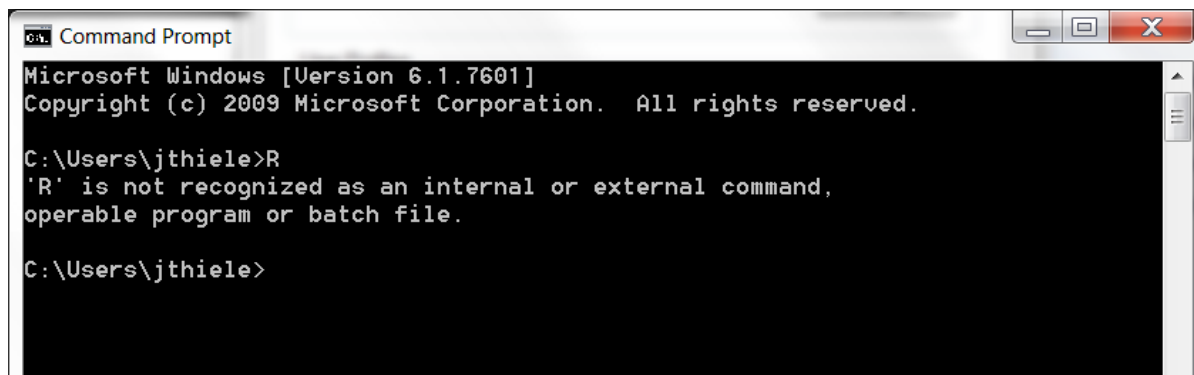


Now, append the PATH variable by %R\_HOME%\bin\x64 (or directly by C:\Program Files\R\R-3.2.3\bin\x64).



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

If you this, your Path environment variable is wrong:

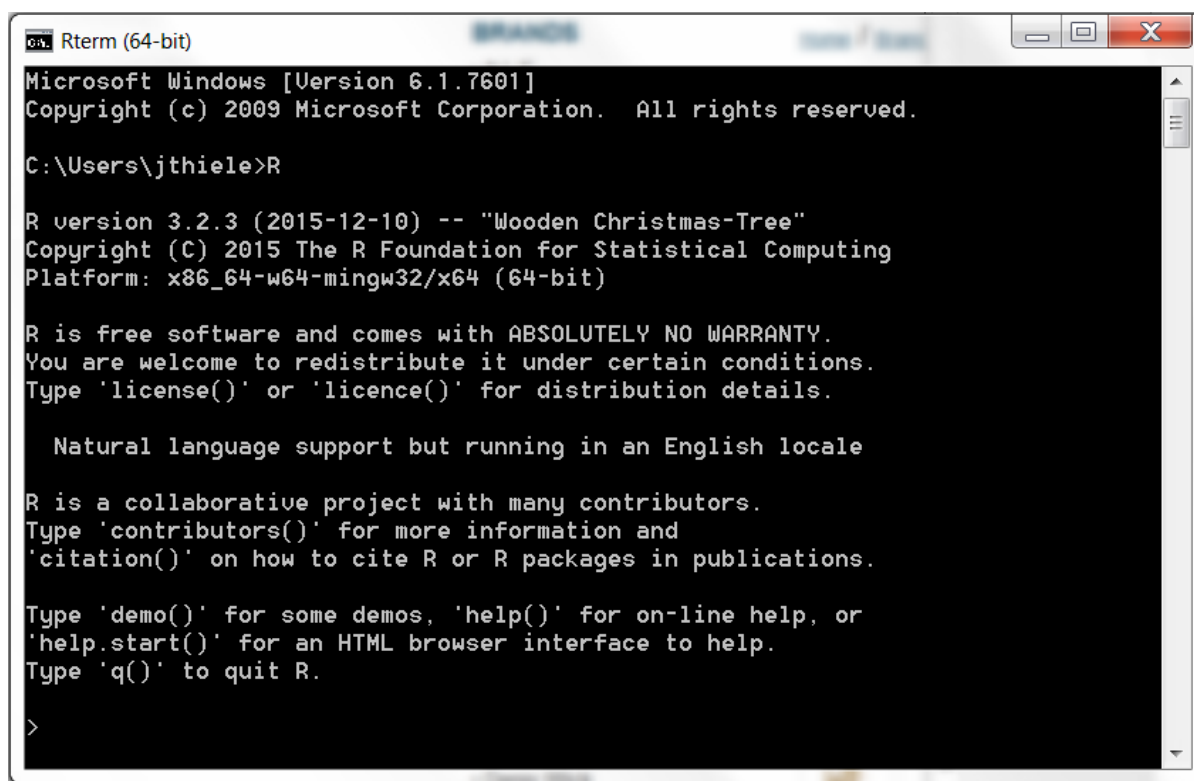


```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\jthiele>R
'R' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\jthiele>
```

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



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\jthiele>R

R version 3.2.3 (2015-12-10) -- "Wooden Christmas-Tree"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

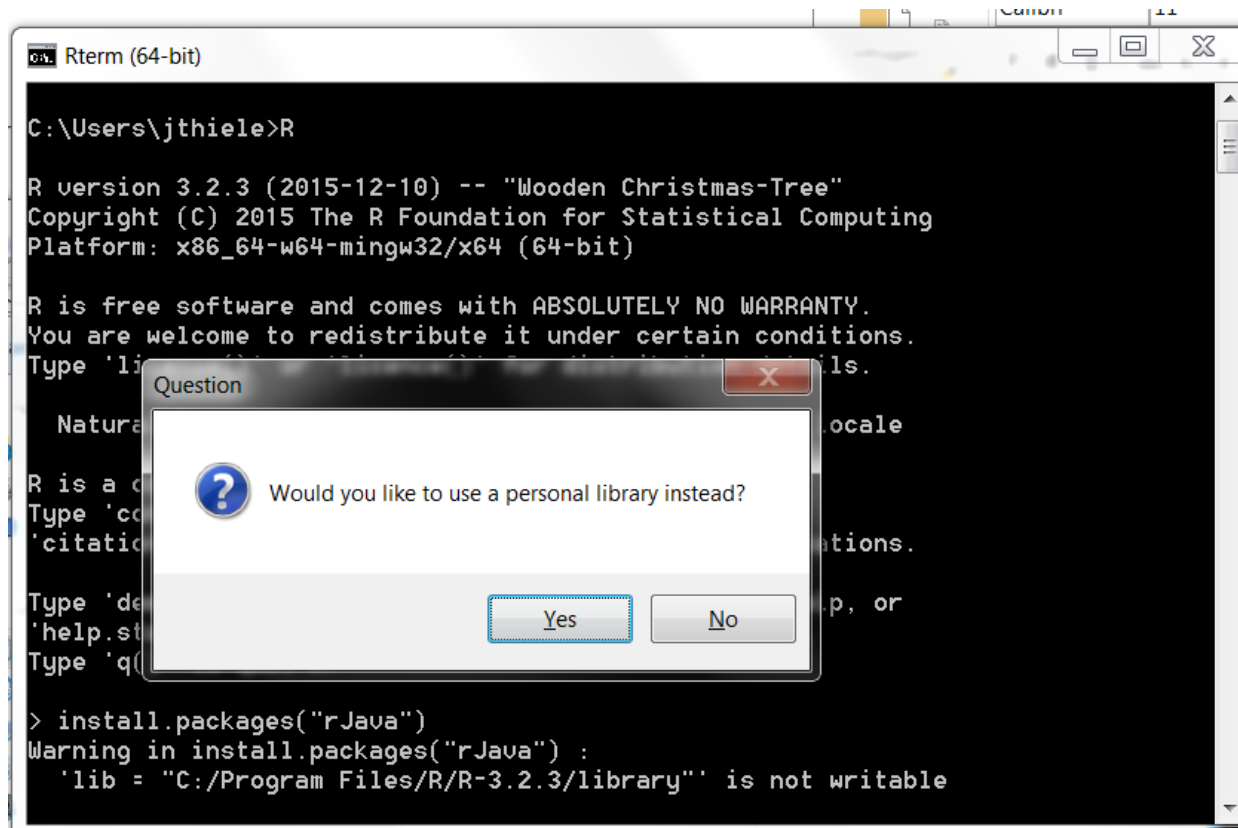
## 5. Install rJava.

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")
```

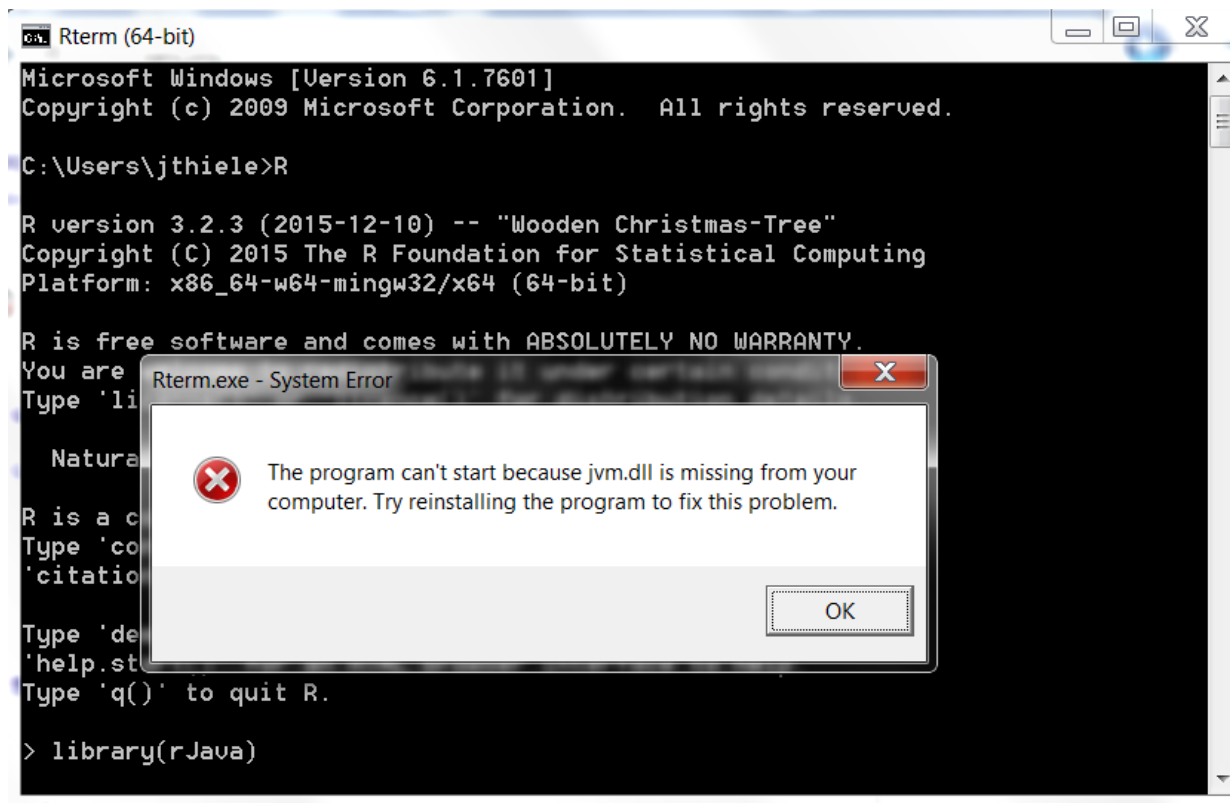
If you already installed the rJava package in 32-bit mode, please delete it first.

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\3.2):



## 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.

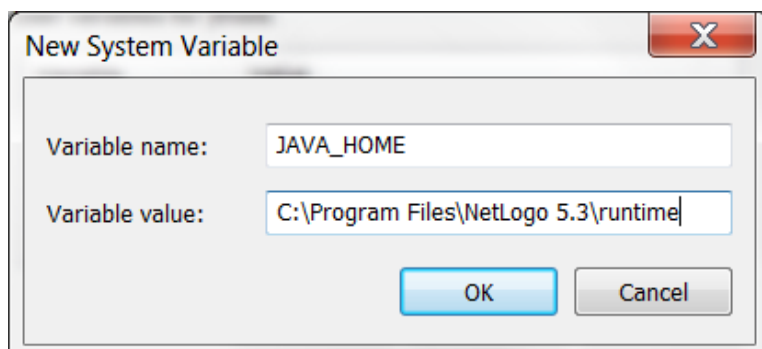


#### 7. In case that **rJava cannot be loaded**:

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 or use the Java bundled with NetLogo). If so, go to step 8.

8. Add/Change the `JAVA_HOME` environment variable with the path to the 64-bit Java (e.g. bundled with NetLogo).



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")

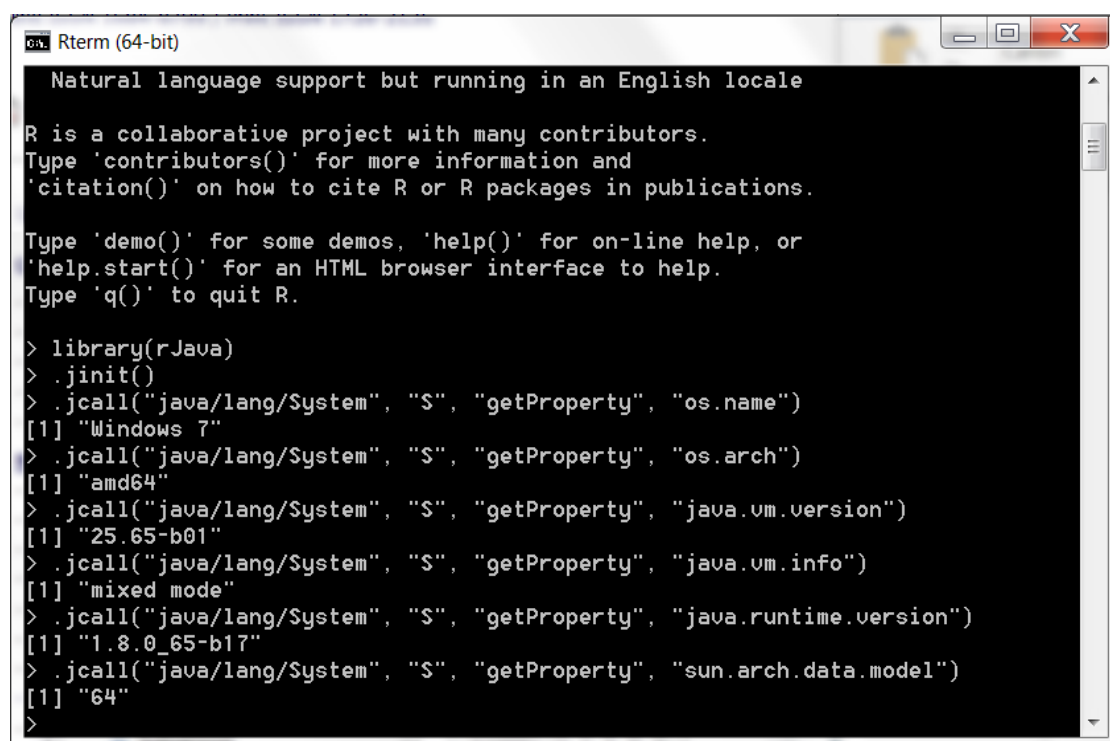
.jcall("java/lang/System", "S", "getProperty", "java.vm.version")

.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")
```

A screenshot of an Rterm window titled "Rterm (64-bit)". The window has a black background with white text. At the top, it says "Natural language support but running in an English locale". Below that, it provides information about R: "R is a collaborative project with many contributors. Type 'contributors()' for more information and 'citation()' on how to cite R or R packages in publications. Type 'demo()' for some demos, 'help()' for on-line help, or 'help.start()' for an HTML browser interface to help. Type 'q()' to quit R." The user has entered several commands: `> 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] "25.65-b01"`, `> .jcall("java/lang/System", "S", "getProperty", "java.vm.info")`, `[1] "mixed mode"`, `> .jcall("java/lang/System", "S", "getProperty", "java.runtime.version")`, `[1] "1.8.0_65-b17"`, `> .jcall("java/lang/System", "S", "getProperty", "sun.arch.data.model")`, `[1] "64"`, and `>`.

```
Rterm (64-bit)
Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 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] "25.65-b01"
> .jcall("java/lang/System", "S", "getProperty", "java.vm.info")
[1] "mixed mode"
> .jcall("java/lang/System", "S", "getProperty", "java.runtime.version")
[1] "1.8.0_65-b17"
> .jcall("java/lang/System", "S", "getProperty", "sun.arch.data.model")
[1] "64"
>
```

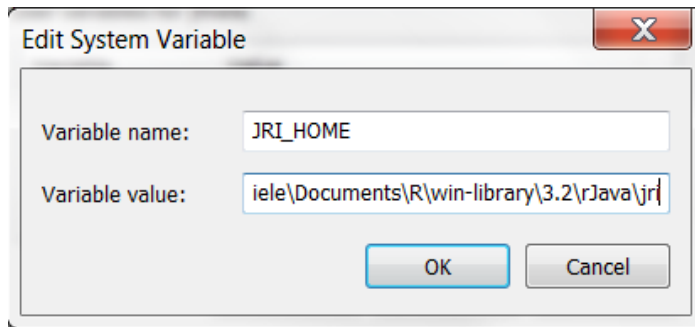
#### 10. Set `JRI_HOME` environment variable.

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\3.2`.

We will create a new environment variable with name `JRI_HOME` and value

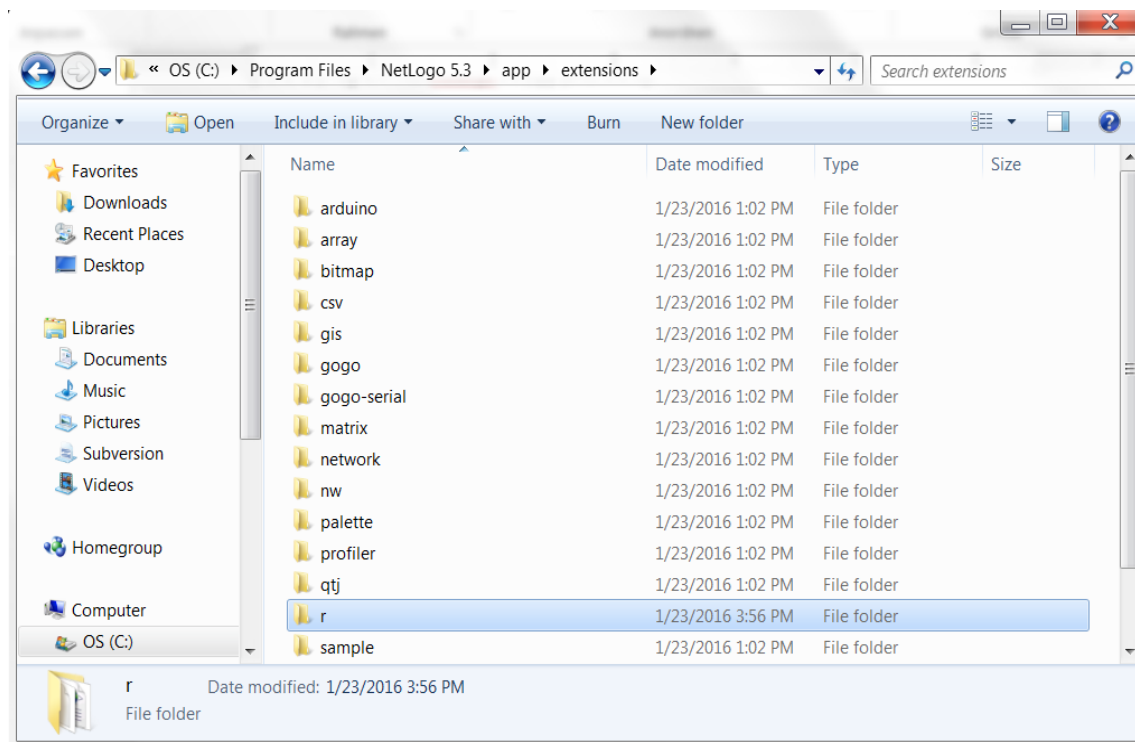
*C:\Users\jthiele\Documents\R\win-library\3.2\rJava\jri.*



### 11. Download the R-Extension

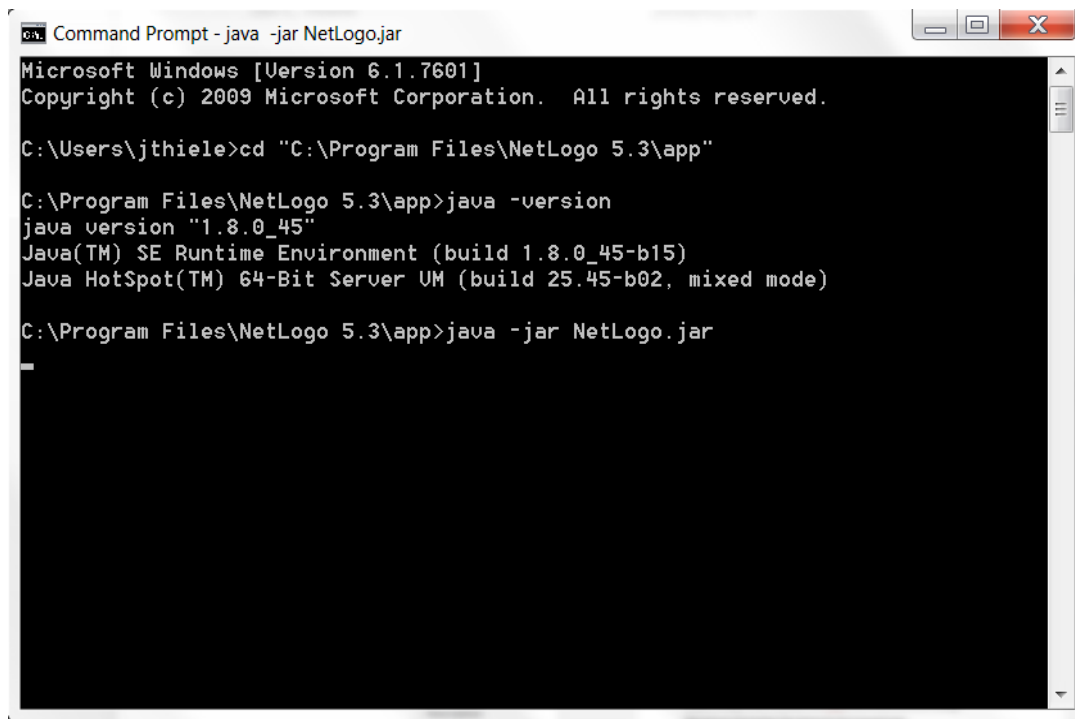
(<https://sourceforge.net/projects/r-ext/files/v1.4%20for%20NetLogo%205.3%20and%20R%203.2%20and%20later/>).

Extract the zip folder and copy only the folder *r* into the extensions folder of the NetLogo installation (here: *C:\Program Files\NetLogo 5.3\app\extensions*).



### 12. Test it all together!

Start NetLogo 5.3 (with 64-bit Java), e.g. by click on it in start menu or by starting it from the MS-DOS-Command prompt (if Java is installed):



```
Command Prompt - java -jar NetLogo.jar
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\jthiele>cd "C:\Program Files\NetLogo 5.3\app"

C:\Program Files\NetLogo 5.3\app>java -version
java version "1.8.0_45"
Java(TM) SE Runtime Environment (build 1.8.0_45-b15)
Java HotSpot(TM) 64-Bit Server VM (build 25.45-b02, mixed mode)

C:\Program Files\NetLogo 5.3\app>java -jar NetLogo.jar
```

Write into the Procedures Tab:

```
extensions [r]
```

and press the Check button (If NetLogo closes immediately 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.

