Installing autodiffr

Changcheng Li, John Nash, Hans Werner Borchers

2018/7/27

Introduction

`autodiffr` is an **R** package to exploit the Automatic Differentiation facilities of the Julia computation system. Automatic Differentiation allows the computation of derivatives of a function to the accuracy of analytic expressions. Moreover, it allows the user to avoid the effort, difficulty and tedium of developing derivatives by hand.

Prerequisites

Whatever system you use, to use `autodiffr` you need to install Julia, and indeed a (possibly the) correct version of Julia.

Go to https://julialang.org/downloads and choose the correct download for your system, and the correct version (at the time of writing 0.6.4) and install as appropriate. Note that the version of Julia is important, and the version available in the package repository of some systems may be inappropriate. Particular instructions are given in the sections below for different OS families.

We also need a sufficiently up-to-date version of **R**, though at the time of writing, the most recent lacks some tools on some operating systems. For example, there is not yet an appropriate version of the `Rtools` software for Windows 7 to accompany R-3.5.1 at 2018-7-27. This collection of software is largely provided in most *nix systems, including Mac OS.

Linux

Installation on Linux has some dependencies that may cause trouble. Some distros may not have the necessary versions of libraries, and cannot run `autodiffr`. We will try to suggest what these may be, but since there are many distros, with variations on their library infrastructure, this list will be haphazard. Moreover, as with the version of Julia, we may need correct versions of some system libraries.

We have not tested if a 64 bit distribution is required. That is, 32 bit distros have not been tried.

We have verified that the following distros support `autodiffr`

- Linux Mint 18.3 Sylvia
- Ubuntu 18.04 (HWB? Did you test)
- Debian Stretch (9.5)
- Linux Mint 19 Mate "Tara" (To get `devtools` to install, it was necessary to install the Debian/Ubuntu "Bionic" packages libssl-dev and libcurl4-openssl-dev. Installing R package `codetools` was done following a note when Rcpp was installed as a devtools dependency. Then JuliaCall and autodiffr were installed with install_github().)

The following distros do not appear to work:

- Bunsenlabs Deuterium 64 bit. (libstdc++.so.6 does not have GLIBC in a sufficiently recent version). This is based on Debian Jessie, and it is likely that distros in that family will not support `autodiffr`.
- Elementary OS Freya 0.3.2 (same complaint about GLIBC)

Virtualbox special needs - Debian or Ubuntu based distros

If you install in virtualbox, you will want to install the Guest Additions. In a running Virtualbox guest OS you need to choose Devices / Insert Guest Additions CD This will likely auto-mount (if not, click on it in a file manager GUI). To install the guest additions, you will likely need to install the `build-essential` package (Debian and Ubuntu family). You will also need the right linux-headers. Use the command `uname -a` to get the linux kernel verion, then install (again Debian or Ubuntu) `linux-headers-version` where `version` is something like `4.10.0-38`. It may be necessary to reboot here so these headers are recognized. Certainly a new terminal will be needed. Then either `sudo` the following commands or `su -` first.

```
cd /media/cdrom
sh ./VBoxLinuxAdditions.run
```

You can also set up a shared folder to allow for easy recording and transfer of files between host and guest OS environments.

In Debian/Ubuntu systems, to install an up-to-date R, first get the signature key for Johannes Ranke

`apt-key adv --keyserver keys.gnupg.net --recv-key 'E19F5F87128899B192B1A2C2AD5F960A256A04AF'`

NOTE: you may need to install the `dirmngr` package first in some distros.

In debian or ubuntu, as root,

`apt install r-base-dev r-recommended`

will install R.

To install Julia, in a web browser, go to https://julialang.org and download the 0.6.4 version tarball.

Extract this in your home directory. Rename `julia-xxxxx`, where `xxxxx` is the suffix to the name on your syste, to `j`. Then as root

```
ln -s /home/(you)/j/bin/julia /usr/bin/julia
```

You can test julia with the command `julia` in a terminal window.

Getting `autodiffr` working

Start **R** in a terminal and run

`install.packages("devtools")`

On some systems, however, you may need to install some packages. Do this in a separate terminal window. Note that the messages from `install.packages` suggest the names of packages to install for major distros.

`autodiffr` needs a fairly recent version of the package `JuliaCall` (??ref by Changcheng). It can be installed by

```
devtools::install_github("Non-Contradiction/JuliaCall")
```

Once this succeeds, `autodiffr` can be installed by:

```
devtools::install_github("Non-Contradiction/autodiffr")
```

After the installation is finished, we can run the package. Note that the first time `ad_setup()` is run during a session, it will take considerable time, but the next time it should be better. ?? Is this only slow the first execution overall, or does each session start take the same time??

```
require(autodiffr)
ad_setup()
```

```
Error in juliacall_initialize(.julia$dll_file) :
C:\Users\john\AppData\Local\JULIA-~1.4\bin\libjulia.dll
- The specified procedure could not be found.
```

There is a msg box that gives the entry point that could not be found, with a reference to libstdc++-6.dll

Macintosh

On Macintosh (Hans W Borchers)

Any newer macOS operating system should be sufficient, at least version 10.8; here we will use macOS "High Sierra" (that is 10.13.6). We assume you have R version 3.4.1 or higher installed, but preferable R >= 3.5.1 as the 3.5 series is significantly faster (on Mac), I saw speed improvements of a factor of 2 and more in some of my applications.

From the 'downloads' page of Julia, https://julialang.org/downloads/, get the disk image file for Julia version 0.6.4 (works also with 0.6.3) and install it on your Mac computer in the usual way. You will find more platform-specific instructions on https://julialang.org/downloads/platform.html but in principle it is as easy as this.

If you did not install the Julia image in the `/Applications` folder, you need to add the path to the Julia binary to your PATH variable or – what I found easier – as superuser create a symbolic link in the `/usr/local/bin` directory with

```
sudo ln -s path/to/julia-0.6.app/Contents/Resources/julia/bin/julia
```

/usr/local/bin/julia

To see if Julia is installed correctly, open a Terminal window and type in the Julia command `julia`. You will see the usual prelude and the Julia prompt.

```
               _
   _       _ _(_)_     |  A fresh approach to technical computing
  (_)     | (_) (_)    |  Documentation: https://docs.julialang.org
   _ _   _| |_  __ _   |  Type "?help" for help.
  | | | | | | |/ _` |  |
  | | |_| | | | (_| |  |  Version 0.6.4 (2018-07-09 19:09 UTC)
 _/ |\__'_|_|_|\__'_|  |  Official http://julialang.org/ release
|__/                   |  x86_64-apple-darwin14.5.0
```

The *autodiffr* start-up procedure `ad_setup()` will install necessary Julia packages. Because this can take quite a long time, pre-install some of these packages directly from Julia, e.g. the important `RCall.jl` package.

```
$ julia
julia> Pkg.add("RCall")
```

After some minutes about 30 or more packages are installed on your computer. You can find them in your Home directory at `~/.julia/v0.6/`. If you like, add also the forward and backward automatic differentiation packages "ForwardDiff" and "ReverseDiff". Some other depending packages will be installed, too.

Now it's time to install the R packages *JuliaCall* and *autodiffr*. Both are available on CRAN, but it is better to get their newest versions from Github, utilizing the *devtools* package.

```
devtools::install_github("Non-Contradiction/JuliaCall")
devtools::install_github("Non-Contradiction/autodiffr")
```

When loading the library *autodiffr* it is still necessary to create a process running Julia, to connect to it, to check whether all needed Julia packages are installed resp. to install them in case they are missing. All this is done by the `ad_setup` function. If everything is in place, it will finish in short time, otherwise it may take several minutes (the first time it is called).

```
library(autodiffr)
## Attaching package: 'autodiffr'
## The following object is masked from 'package:stats':
##     deriv

ad_setup()
## Julia version 0.6.4 at location [...]/julia/bin will be used.
## Loading setup script for JuliaCall...
Finish loading setup script for JuliaCall.
INFO: Precompiling module DiffResults.
INFO: Precompiling module ForwardDiff.
INFO: Precompiling module ReverseDiff.
```

The first time `ad_setup()` is called on your system, these three Julia modules / packages are being precompiled, later on this will not happen again, except when the julia packages get installed anew (e.g., a new version has appeared). Please note that `ad_setup` is required whenever you load the *autodiffr* package into R.

Now automatic differentiation is ready to get applied. As a test, define an R function and take its gradient (or Jacobian or Hessian). For instance, the following function will have a quite well known derivative.

```
f <- function(x) exp(x)
exp(1.0); grad(f, 1.0)
## [1] 2.718281828459045
## [1] 2.718281828459045
```

We see the result is accurate to the last digit. Compare this with a numerical derivative, such as is provided through the *numDeriv* package.

```
numDeriv::grad(f, 1.0)
## [1] 2.718281828442632
```

Function names `grad`, `jacobian`, `hessian` are used in the *numDeriv* (and *pracma*) package as well, so have to be qualified when used in parallel with the *autodiffr* package.

IMPORTANT: Do *not* install the upcoming Julia 0.7 as there are breaking changes between the 0.6 and 0.7/1.0 versions. After Julia 0.7 has become official and all relevant Julia packages have been updated to the new version, *autodiffr* will be adapted and published on CRAN.

Windows

On some Windows systems, antivirus protection may need to be disabled so the install can be accomplished. For example, though the ultimate installation of autodiffr failed, on a Windows 7 32bit Virtual Machine under VirtualBox, AVG antivirus (Free version) had to be disabled (protection turned off) to enable the Julia install, which was to `C:\Users\john\AppData\Local\Julia-0.6.4`. In the `bin\` subdirectory, there is julia.exe and this can be run in a terminal, or double-clicked to open. Julia runs, and can be terminated by `exit()` or `quit()`. Remember to re-enable antivirus protection once the installation of Julia (or other necessary software) is complete.

Windows 10

Asus UX303 – worked. What were steps?

Install Julia.

Install R

Need Rtools.

Install `devtools` and use it to `install_github() JuliaCall` and `autodiffr`.

Note that PATH needs editing. (Search in Control Panel for Environmental Variables to get the edit panel). Path for R seems to be set automatically, but Julia path (to the "bin" subdirectory) should be added. In my case this was

```
C:\Users\J\AppData\Local\Julia-0.6.4\bin
```

Windows 7 - 64 bit

Asus EEE 1225B

Installed Julia 0.6.4 and tested in CMD. OK

Installed R-3.5.1, devtools.

Tried to install_github("Non-contradiction/Julia_Call") – FAILED

Needs Rtools. Installed Rtools35.exe. But when tried R said this version NOT compatible. (In Win10 found Rtools34.exe worked. However, in Win10, was able to install autodiff and devtools WITHOUT Rtools. Possibly don't need the underscore.)

Following is partial record of some things tried. Apologies for messiness.

```
        JuliaCall::julia_command("@suppress begin function Base.float{V,D,O}(t::ReverseDiff.TrackedReal
        apiFuncs()
        .AD$inited <- TRUE
    }
}
<environment: namespace:autodiffr>
> ad_setup()
Julia version 0.6.4 at location C:\Users\mary\AppData\Local\JULIA-~1.4\bin will be used.
Loading setup script for JuliaCall...

Warning message:
running command '"C:\Users\mary\AppData\Local\JULIA-~1.4\bin/julia" "C:/Users/mary/Documents/R/win-libra
> .AD$julia <- JuliaCall::julia_setup(...)
Error: '...' used in an incorrect context
> .AD$julia <- JuliaCall::julia_setup()
Julia version 0.6.4 at location C:\Users\mary\AppData\Local\JULIA-~1.4\bin will be used.
Loading setup script for JuliaCall...
Error in .julia$cmd(paste0("include(\"", system.file("julia/setup.jl",  :
  Error happens when you try to execute command include("C:/Users/mary/Documents/R/win-library/3.4/Julia
                       To have more helpful error messages,
                       you could considering running the command in Julia directly
In addition: Warning message:
running command '"C:\Users\mary\AppData\Local\JULIA-~1.4\bin/julia" "C:/Users/mary/Documents/R/win-libra
>

Run in Julia and try to build CodecZlib

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

C:\Users\mary>cd Documents
```

```
C:\Users\mary\Documents>cd R

C:\Users\mary\Documents\R>cd win-library

C:\Users\mary\Documents\R\win-library>cd 3.4

C:\Users\mary\Documents\R\win-library\3.4>cd JuliaCall

C:\Users\mary\Documents\R\win-library\3.4\JuliaCall>cd julia

C:\Users\mary\Documents\R\win-library\3.4\JuliaCall\julia>julia
               _
   _       _ _(_)_     |  A fresh approach to technical computing
  (_)     | (_) (_)    |  Documentation: https://docs.julialang.org
   _ _   _| |_  __ _   |  Type "?help" for help.
  | | | | | | |/ _` |  |
  | | |_| | | | (_| |  |  Version 0.6.4 (2018-07-09 19:09 UTC)
 _/ |\__'_|_|_|\__'_|  |  Official http://julialang.org/ release
|__/                   |  x86_64-w64-mingw32

julia> ./setup.jl
ERROR: syntax: "./" is not a unary operator

julia> setup.jl
ERROR: UndefVarError: setup not defined

julia> source("setup.jl")
ERROR: UndefVarError: source not defined

julia> include("./setup.jl")
INFO: Precompiling module RCall.
ERROR: LoadError: CodecZlib.jl is not installed properly, run Pkg.build("CodecZl
ib") and restart Julia.
Stacktrace:
 [1] error(::String) at .\error.jl:21
 [2] include_from_node1(::String) at .\loading.jl:576
 [3] include(::String) at .\sysimg.jl:14
 [4] anonymous at .\<missing>:2
while loading C:\Users\mary\.julia\v0.6\CodecZlib\src\CodecZlib.jl, in expressio
n starting on line 38
ERROR: LoadError: LoadError: Failed to precompile CodecZlib to C:\Users\mary\.ju
lia\lib\v0.6\CodecZlib.ji.
Stacktrace:
 [1] compilecache(::String) at .\loading.jl:710
 [2] _require(::Symbol) at .\loading.jl:463
 [3] require(::Symbol) at .\loading.jl:405
 [4] include_from_node1(::String) at .\loading.jl:576
 [5] include(::String) at .\sysimg.jl:14
 [6] include_from_node1(::String) at .\loading.jl:576
 [7] include(::String) at .\sysimg.jl:14
 [8] anonymous at .\<missing>:2
while loading C:\Users\mary\.julia\v0.6\DataFrames\src\deprecated.jl, in express
ion starting on line 26
while loading C:\Users\mary\.julia\v0.6\DataFrames\src\DataFrames.jl, in express
```

```
ion starting on line 106
ERROR: LoadError: Failed to precompile DataFrames to C:\Users\mary\.julia\lib\v0
.6\DataFrames.ji.
Stacktrace:
 [1] compilecache(::String) at .\loading.jl:710
 [2] _require(::Symbol) at .\loading.jl:463
 [3] require(::Symbol) at .\loading.jl:405
 [4] include_from_node1(::String) at .\loading.jl:576
 [5] include(::String) at .\sysimg.jl:14
 [6] anonymous at .\<missing>:2
while loading C:\Users\mary\.julia\v0.6\RCall\src\RCall.jl, in expression starti
ng on line 8
ERROR: LoadError: Failed to precompile RCall to C:\Users\mary\.julia\lib\v0.6\RC
all.ji.
Stacktrace:
 [1] compilecache(::String) at .\loading.jl:710
 [2] _require(::Symbol) at .\loading.jl:497
 [3] require(::Symbol) at .\loading.jl:405
 [4] include_from_node1(::String) at .\loading.jl:576
 [5] include(::String) at .\sysimg.jl:14
while loading C:\Users\mary\Documents\R\win-library\3.4\JuliaCall\julia\setup.jl
, in expression starting on line 19

julia>

julia> Pkg.build("CodecZlib")
INFO: Building CodecZlib
Info: Downloading https://github.com/bicycle1885/ZlibBuilder/releases/download/v
1.0.0/Zlib.x86_64-w64-mingw32.tar.gz to C:\Users\mary\.julia\v0.6\CodecZlib\deps
\usr\downloads\Zlib.x86_64-w64-mingw32.tar.gz...
Exception setting "SecurityProtocol": "Cannot convert null to type "System.Net.
SecurityProtocolType" due to invalid enumeration values. Specify one of the fol
lowing enumeration values and try again. The possible enumeration values are "S
sl3, Tls"."
At line:1 char:35
+ [System.Net.ServicePointManager]:: <<<< SecurityProtocol =
    + CategoryInfo          : InvalidOperation: (:) [], RuntimeException
    + FullyQualifiedErrorId : PropertyAssignmentException

Exception calling "DownloadFile" with "2" argument(s): "The underlying connecti
on was closed: An unexpected error occurred on a send."
At line:4 char:24
+ $webclient.DownloadFile <<<< ("https://github.com/bicycle1885/ZlibBuilder/rel
eases/download/v1.0.0/Zlib.x86_64-w64-mingw32.tar.gz", "C:\Users\mary\.julia\v0
.6\CodecZlib\deps\usr\downloads\Zlib.x86_64-w64-mingw32.tar.gz")
    + CategoryInfo          : NotSpecified: (:) [], MethodInvocationException
    + FullyQualifiedErrorId : DotNetMethodException

==============================[ ERROR: CodecZlib ]==============================


LoadError: Could not download https://github.com/bicycle1885/ZlibBuilder/release
s/download/v1.0.0/Zlib.x86_64-w64-mingw32.tar.gz to C:\Users\mary\.julia\v0.6\Co
decZlib\deps\usr\downloads\Zlib.x86_64-w64-mingw32.tar.gz
```

```
while loading C:\Users\mary\.julia\v0.6\CodecZlib\deps\build.jl, in expression s
tarting on line 92

================================================================================


================================[ BUILD ERRORS ]================================


WARNING: CodecZlib had build errors.

 - packages with build errors remain installed in C:\Users\mary\.julia\v0.6
 - build the package(s) and all dependencies with `Pkg.build("CodecZlib")`
 - build a single package by running its `deps/build.jl` script

================================================================================
```

Windows 7 - 32 bit

Tried on VM, but did not get satisfactory outcome.

```
> ad_setup()
Julia version 0.6.4 at location C:\Users\john\AppData\Local\JULIA-~1.4\bin will be used.
Loading setup script for JuliaCall...
Error in .julia$cmd(paste0("include(\"", system.file("julia/setup.jl",  :
  Error happens when you try to execute command include("C:/Users/john/Documents/R/win-library/3.5/Julia
                       To have more helpful error messages,
                       you could considering running the command in Julia directly
In addition: Warning message:
In system2(file.path(.julia$bin_dir, "julia"), shQuote(command),  :
  running command '"C:\Users\john\AppData\Local\JULIA-~1.4\bin/julia" "C:/Users/john/Documents/R/win-lib

In Command window:

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

C:\Users\john>cd Documents

C:\Users\john\Documents>cd R

C:\Users\john\Documents\R>cd win-library

C:\Users\john\Documents\R\win-library>cd 3.5

C:\Users\john\Documents\R\win-library\3.5>cd JuliaCall

C:\Users\john\Documents\R\win-library\3.5\JuliaCall>cd julia

C:\Users\john\Documents\R\win-library\3.5\JuliaCall\julia>ls
JuliaObject.jl  convert.jl   incomplete_console.jl  libjulia.jl
REPLhook.jl     dispatch.jl  install_dependency.jl  setup.jl
asR.jl          display      interface1.jl
```

```
C:\Users\john\Documents\R\win-library\3.5\JuliaCall\julia>julia
                 _
   _       _ _(_)_     |  A fresh approach to technical computing
  (_)     | (_) (_)    |  Documentation: https://docs.julialang.org
   _ _   _| |_  __ _   |  Type "?help" for help.
  | | | | | | |/ _` |  |
  | | |_| | | | (_| |  |  Version 0.6.4 (2018-07-09 19:09 UTC)
 _/ |\__'_|_|_|\__'_|  |  Official http://julialang.org/ release
|__/                   |  i686-w64-mingw32

julia> include("setup.jl")
INFO: Precompiling module RCall.
ERROR: LoadError: RCall not properly installed. Please run Pkg.build("RCall")
Stacktrace:
 [1] error at .\error.jl:21
 [2] include_from_node1 at .\loading.jl:576
 [3] include at .\sysimg.jl:14
while loading C:\Users\john\.julia\v0.6\RCall\src\RCall.jl, in expression starti
ng on line 40
ERROR: LoadError: Failed to precompile RCall to C:\Users\john\.julia\lib\v0.6\RC
all.ji.
Stacktrace:
 [1] compilecache(::String) at .\loading.jl:710
 [2] _require(::Symbol) at .\loading.jl:497
 [3] require(::Symbol) at .\loading.jl:405
 [4] include_from_node1 at .\loading.jl:576
 [5] include at .\sysimg.jl:14
while loading C:\Users\john\Documents\R\win-library\3.5\JuliaCall\julia\setup.jl
, in expression starting on line 19

julia> Pkg.build("RCall")
INFO: Building CodecZlib
INFO: Building SpecialFunctions
INFO: Building Conda
INFO: Building RCall
INFO: Using R at c:\Program Files\R\R-3.5.1 and libR at c:\Program Files\R\R-3.5
.1\bin\i386\R.dll.

julia> include("setup.jl")
WARNING: replacing module JuliaCall
INFO: Precompiling module RCall.
JuliaCall

julia>
--------------------------------
```

But running again in julia, after ad_setup() in R gives

```
julia> include("setup.jl")
WARNING: replacing module JuliaCall
WARNING: Method definition sexp(Tuple) in module JuliaCall at C:\Users\john\Docu
ments\R\win-library\3.5\JuliaCall\julia\convert.jl:8 overwritten in module Julia
Call at C:\Users\john\Documents\R\win-library\3.5\JuliaCall\julia\convert.jl:8.
```

```
WARNING: Method definition rcopy(Type{Tuple}, Ptr{RCall.VecSxp}) in module Julia
Call at C:\Users\john\Documents\R\win-library\3.5\JuliaCall\julia\convert.jl:14
overwritten in module JuliaCall at C:\Users\john\Documents\R\win-library\3.5\Jul
iaCall\julia\convert.jl:14.
WARNING: Method definition rcopytype(Type{RCall.RClass{:JuliaTuple}}, Ptr{RCall.
VecSxp}) in module JuliaCall at C:\Users\john\Documents\R\win-library\3.5\JuliaC
all\julia\convert.jl:17 overwritten in module JuliaCall at C:\Users\john\Documen
ts\R\win-library\3.5\JuliaCall\julia\convert.jl:17.
WARNING: Method definition rcopytype(Type{RCall.RClass{:JuliaObject}}, Ptr{RCall
.EnvSxp}) in module JuliaCall at C:\Users\john\Documents\R\win-library\3.5\Julia
Call\julia\JuliaObject.jl:92 overwritten in module JuliaCall at C:\Users\john\Do
cuments\R\win-library\3.5\JuliaCall\julia\JuliaObject.jl:92.
WARNING: Method definition sexp(Any) in module JuliaCall at C:\Users\john\Docume
nts\R\win-library\3.5\JuliaCall\julia\JuliaObject.jl:96 overwritten in module Ju
liaCall at C:\Users\john\Documents\R\win-library\3.5\JuliaCall\julia\JuliaObject
.jl:96.
WARNING: Method definition sexp(AbstractArray{Any, N} where N) in module JuliaCa
ll at C:\Users\john\Documents\R\win-library\3.5\JuliaCall\julia\JuliaObject.jl:1
05 overwritten in module JuliaCall at C:\Users\john\Documents\R\win-library\3.5\
JuliaCall\julia\JuliaObject.jl:105.
JuliaCall

julia>
```

Running in a fresh instance of julia gives the same "RCall not properly installed" error from before.

Testing the `autodiffr` installation

And we can use package `numDeriv` to compare with `autodiffr`.

```
require(numDeriv)
```

```
## Loading required package: numDeriv
```

extra material

Note that later we will probably need to adjust some or other PATH variables, and possibly set up links to ensure **R** can find Julia.

To get Julia to start in R, `C:\Users\john\AppData\Local\Julia-0.6.4\bin` needs to be in the PATH. Then a CMD box will start Julia with the command `julia`. However, after require(autodiffr), the ad_setup() call gives an error

```
Julia version 0.6.4 at location
C:\Users\john\AppData\Local\JULIA-~1.4\bin will be used.
Error in juliacall_initialize(.julia$dll_file) :
  C:\Users\john\AppData\Local\JULIA-~1.4\bin\libjulia.dll
  - The specified procedure could not be found.
```

Thus for Windows, there must be another setting somewhere needed.

May be due to R 3.4.4??

then download Helium 64 and install, Guest Addiaions download and try to install, todos, Allowed upgrade, added java support. Backports from testing, leafpad, build-essential sudo apt-get install linux-headers-$(uname

-r) ## to get right headers “Insert” VBoxGuestAdditions, cd /media/cdrom sudo sh ./VBoxLinuxAdditions.run Reboot. Could not get larger. xrandr shows 1024 is largest screen. Also had difficulty getting R 3.5.1 – some key problem. Set up Debian Stretch. installed build-essential, checked uname -a, then installed linux-headers for correct version. Then su - and install VBoxLinuxAdditions.run. Had to do ’xrandr –output VGA-1 –mode “1440x900” to get large display. Installed leafpad. Added ... /debian stretch-cran35/ to sources.list Note: needed to install dirmngr. But stretch-cran35 does not have a Release file. Turned out it was “cdrom” that was problem. – just comment it out in sources.list Downloaded julia 0.6.4 from julialang.org, unpacked in ~, renamed julia-xxxxx to j As root set symlink in /usr/bin to j/bin/julia. Tested it works. start R. install(“devtools”), but needed git2r, httr (which needed system libssl-dev) and libcurl-openssl-dev. devtools in. require(devtools) / install_github(“Non-contradiction/JuliaCall”) install_github(“Non-contradiction/autodiffr”) require(autodiffr) / ad_setup() ## takes a very long time and disk is flashing ## Is it trying to locate a library file?? Eventually finished (20 mins?) Then test <- function(x) { sum(x*x) } testg <- makeGradFunc(test) x0 <- c(1,3,5) test(x0) ## 35 testg(x0) ## 2 6 10

`Later re-installed mintupdate and virtualbox-guest-dkms, ...x11 and one other removed, then screen sized to max automatically.`

?? need to check path

?? symlink

Julia.exe is in `C:\Users\john\AppData\Local\Julia-0.6.4` after installation (32 bit – the extra steps were not necessary as far as I can tell) `C:\Users\john\AppData\Local\Julia-0.6.4\bin` was added to the PATH.

CMD box then runs julia fine.