R COMPILER TOOLS FOR RCPP ON MACOS

April 24, 2020 r (https://thecoatlessprofessor.com/tags/r), rcpp (https://thecoatlessprofessor.com/tags/rcpp), compilers (https://thecoatlessprofessor.com/tags/compilers), os x (https://thecoatlessprofessor.com/tags/os-x)

Editor's Note: This is an updated post that covers how to install the macOS toolchain for versions of *R* starting at 4.y.z.

For *R* versions between *R*:

- **3.0.0 - 3.3.3**, please see R Compiler Tools for Rcpp on OS X before R 3.4.0 (https://thecoatlessprofessor.com/programming/r-compiler-tools-for-rcpp-on-os-x-before-r-3.4.0/).
- **3.4.0 - 3.4.4**, please see R Compiler Tools for Rcpp on OS X before R 3.5.0 (https://thecoatlessprofessor.com/programming/r-compiler-tools-for-rcpp-on-macos-before-r-3.5.0/).
- **3.5.0 - 3.5.3**, please see R Compiler Tools for Rcpp on OS X before R 3.6.0 (https://thecoatlessprofessor.com/programming/r-compiler-tools-for-rcpp-on-macos-before-r-3.6.0/).
- **3.6.0 - 3.6.3**, please see R Compiler Tools for Rcpp on OS X before R 4.0.0 (https://thecoatlessprofessor.com/programming/r-compiler-tools-for-rcpp-on-macos-before-r-4.0.0/).

INTRO

The objective behind this post is to provide users with information on how to setup the macOS toolchain for compiling used in the 4.y.z series of R (r-project.org). The post is structured primarily for macOS Catalina (10.15.z) users.

REMOVING OLD INSTALLATION FILES

If you previously used either the `clang4`, `clang6`, `clang7`, `clang8`, or the `macos-rtools` installer, please consider deleting the old components that were installed.

Instructions for uninstallation can be found here:

Uninstalling the R development toolchain on macOS (https://thecoatlessprofessor.com/programming/r/uninstalling-the-r-development-toolchain-on-macos)

That said, please remove both `~/.R/Makevars` and `~/.Renviron` files prior to continuing as they were set in a prior iteration. You can achieve this by using:

```
unlink("~/.R/Makevars")
unlink("~/.Renviron")
```

Alternatively, if there is contents inside of the `~/.Renviron` file that must be retained, please look for where the `PATH` variable is listed and remove any part with `clang`.

Open the file in *R* using:

```
file.edit("~/.Renviron")
```

Next, find and remove:

```
PATH="/usr/local/clang7/bin:${PATH}"
```

Then, save and close the file. You may need to restart *R* for the changes to take effect.

INSTALLATION INSTRUCTIONS

One of the primary ways to setup the *R* toolchain for compiled code on macOS is to individually installing each element yourself.

There are **two** components to the R 4.y.z toolchain based on Section: C.3 macOS (https://cran.r-project.org/doc/manuals/r-release/R-admin.html#macOS) of R Installation and Administration (https://cran.r-project.org/doc/manuals/r-release/R-admin.html). These components are:

1. Xcode Command Line Tools ("Xcode CLI")
2. gfortran
   - **gfortran 6.3**: Sierra (10.12) and High Sierra (10.13)
   - **gfortran 8.2**: Mojave (10.14) and Catalina (10.15)

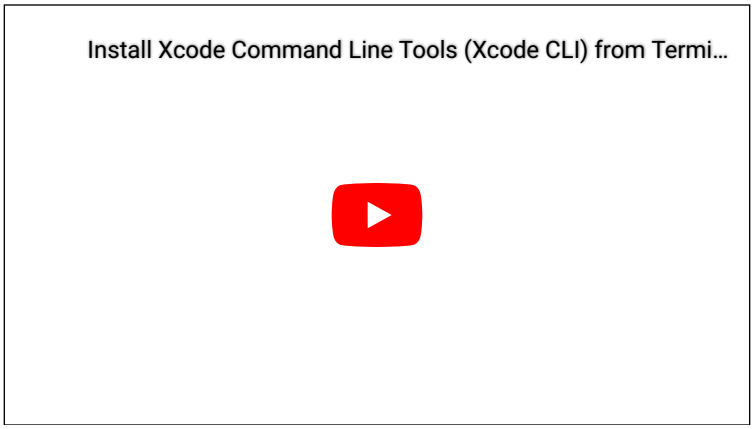
The `gfortran` component is dependent on the version of macOS being used.

For historical information about the toolchain, please see: R macOS toolchain evolution (https://thecoatlessprofessor.com/programming/cpp/r-macos-toolchain-evolution)

INSTALL GUIDE

This guide provides a step-by-step breakdown of the actions required to setup the toolchain. As a result, this guide will use both installers and script commands ia `Terminal.app` found in `/Applications/Utilities/`. Terminal is macOS' equivalent to Linux's shell and Window's command line. From Terminal, we will install only the XCode Command Line Tools ("Xcode CLI") (https://developer.apple.com/library/ios/technotes/tn2339/_index.html). These provide the system headers used to build the official CRAN binary for *R*.

XCODE COMMAND LINE TOOLS



1. Open the `Terminal` from `/Applications/Utilities/`
2. Type the following into `Terminal`

   - **Note:** Anytime the Xcode CLI toolchain updates, you *must* re-run this command.

   ```
   xcode-select --install
   ```

3. Press "Install"

4. Verify installation by typing into terminal:

   ```
   gcc --version
   ```

INSTALL OS-SPECIFIC GFORTRAN BINARY

Determine the version of macOS the computer is being run on.

1. Click the Apple menu in the corner of your screen
2. Choose "About This Mac"
3. Look for Version `10.1*` or the name of the operating system.

For additional help in determining this, please see Apple's support page on Find out which macOS your Mac is using (https://support.apple.com/en-us/HT201260).

Then, download and install the appropriate `gfortran` binary from the https://github.com/fxcoudert/gfortran-for-macOS (https://github.com/fxcoudert/gfortran-for-macOS)

- Mojave (10.14) - Catalina (10.15) uses `gfortran8.2`
  - Download Link

    - https://github.com/fxcoudert/gfortran-for-macOS/releases/download/8.2/gfortran-8.2-Mojave.dmg (https://github.com/fxcoudert/gfortran-for-macOS/releases/download/8.2/gfortran-8.2-Mojave.dmg)
  - GitHub Release Page Overview (https://github.com/fxcoudert/gfortran-for-macOS/releases/tag/8.2)
- Sierra (10.12) - High Sierra (10.13) uses `gfortran6.3`
  - Download Link
    - https://github.com/fxcoudert/gfortran-for-macOS/releases/download/6.3/gfortran-6.3-Sierra.dmg (https://github.com/fxcoudert/gfortran-for-macOS/releases/download/6.3/gfortran-6.3-Sierra.dmg)
  - GitHub Release Page Overview (https://github.com/fxcoudert/gfortran-for-macOS/releases/tag/6.3)

Both of the installers will place the `gfortran` binary into `/usr/local/gfortran`. This will be picked up by the default implicit variable set by *R* during compilation.

QUICK CHECK

To verify that everything is working appropriately, let's do a quick C++ program using Rcpp (http://www.rcpp.org/) and Armadillo (http://arma.sourceforge.net/docs.html).

First, let's install `Rcpp` and `RcppArmadillo` within R.

```
install.packages(c('Rcpp', 'RcppArmadillo'))
```

Create a new file, name the follow: `helloworld.cpp`

By adding the `.cpp` extension, the file is viewed as being C++ code.

Within the file write:

```
#include <RcppArmadillo.h>

// [[Rcpp::depends(RcppArmadillo)]]

// [[Rcpp::export]]
void hello_world() {
  Rcpp::Rcout << "Hello World!" << std::endl;
}

// After compile, this function will be immediately called using
// the below snippet and results will be sent to the R console.

/*** R
hello_world()
*/
```

Compile the function using:

```
Rcpp::sourceCpp('path/to/file/helloworld.cpp')
```

where `'path/to/file/'` is the location containing `helloworld.cpp`

If everything is installed appropriately, then you should see the following in the console:

```
> hello_world()

Hello World!
```

In addition, you should have a new function within the global environment scope called "hello_world". You can call this function like a normal R function via:

```
hello_world()
```

COMMON ERRORS

The following are debugged errors that you *may* run into.

```
clang: warning: argument unused during compilation: '-fopenmp'
fatal error: 'omp.h' file not found
```

Unfortunately, with *R* 4.0.0 the CRAN distributed version of *R* loses the ability to use OpenMP without a custom setup.

ACKNOWLEDGEMENTS

Thanks to dr.ing. MPH Verouden (https://www.vcard.wur.nl/Views/Profile/View.aspx?id=61838&ln=eng) for catching a couple of errors in the post and letting me know!

« *Previous*: R Compiler Tools for Rcpp on macOS before R 4.0.0 (https://thecoatlessprofessor.com/programming/cpp/r-compiler-tools-for-rcpp-on-macos-before-r-4.0.0/)

helloworld.cpp:18:17: warning: character constant too long for its type
helloworld.cpp:18:56: error: expected ';' after top level declarator
Rcpp::sourceCpp('/Users/nikitaabalakin/helloworld.cpp')
^
;
2 warnings and 3 errors generated.
make: *** [helloworld.o] Error 1
clang++ -arch arm64 -std=gnu++14 -
I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG -
I../inst/include -I"/Library/Frameworks/R.framework/Versions/4.1-
arm64/Resources/library/Rcpp/include" -
I"/Library/Frameworks/R.framework/Versions/4.1-
arm64/Resources/library/RcppArmadillo/include" -I"/Users/nikitaabalakin" -
I/opt/R/arm64/include -fPIC -falign-functions=64 -Wall -g -O2 -c
helloworld.cpp -o helloworld.o
Error in Rcpp::sourceCpp("helloworld.cpp") :
Error 1 occurred building shared library.

After getting my new Mac with m1 chip ran into this problem, the whole thing worked fine on with Monterey 12.1 with intel, but it just won't work with m1 pro and Monterey 12.1. And yes, I did download the proper gfortran. Any suggestions?

Libory • a year ago

I ran the helloworld.cpp example, and I got this error which I really do not understand:

Error in
dyn.load("/private/var/folders/46/1tz_54_n3glfmgftvqsspwrr0000gn/T/Rtmp
x86_64-apple-darwin17.0-1.0.6/sourcecpp_2b511a42ea8/sourceCpp_2.so") :
unable to load shared object
'/private/var/folders/46/1tz_54_n3glfmgftvqsspwrr0000gn/T/RtmpDghQ3E/
x86_64-apple-darwin17.0-1.0.6/sourcecpp_2b511a42ea8/sourceCpp_2.so':
dlopen(/private/var/folders/46/1tz_54_n3glfmgftvqsspwrr0000gn/T/RtmpDg
x86_64-apple-darwin17.0-1.0.6/sourcecpp_2b511a42ea8/sourceCpp_2.so, 6):
Symbol not found: ___addtf3
Referenced from: /usr/local/lib/libquadmath.0.dylib
Expected in: /usr/local/lib/libgcc_s_x86_64.1.dylib
in /usr/local/lib/libquadmath.0.dylib

But, if I run something like

Rcpp::evalCpp("1 + 1")

then it works.

Does anyone now how to solve this problem?
Many thanks

이희승 • a year ago • edited

When I ran "helloworld.cpp", I failed and got the error message.

>
Rcpp::sourceCpp('/Volumes/Data_CSNL/people/LHS/classes/Computational\
Modeling/HW3/helloworld.cpp')

ld: warning: directory not found for option '-
L/usr/local/gfortran/lib/gcc/x86_64-apple-darwin18/8.2.0'
ld: library not found for -lquadmath
clang: error: linker command failed with exit code 1 (use -v to see invocation)
make: *** [sourceCpp_2.so] Error 1
clang++ -mmacosx-version-min=10.13 -std=gnu++11 -
I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG -
I../inst/include -
I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/Rcpp/i
-
I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppA
-I"/Volumes/Data_CSNL/people/LHS/classes/Computational
Modeling/HW3" -I/usr/local/include -fPIC -Wall -g -O2 -c helloworld.cpp -o
helloworld.o
clang++ -mmacosx-version-min=10.13 -std=gnu++11 -dynamiclib -Wl,-
headerpad_max_install_names -undefined dynamic_lookup -single_module -
multiply_defined suppress -

multiply_defined suppress
L/Library/Frameworks/R.framework/Resources/lib -L/usr/local/lib -o sourceCpp_2.so helloworld.o -L/Library/Frameworks/R.framework/Resources/lib -lRlapack -L/Library/Frameworks/R.framework/Resources/lib -lRblas -L/usr/local/gfortran/lib/gcc/x86_64-apple-darwin18/8.2.0 -L/usr/local/gfortran/lib -lgfortran -lquadmath -lm -F/Library/Frameworks/R.framework/.. -framework R -Wl,-framework -Wl,CoreFoundation
Error in Rcpp::sourceCpp("/Volumes/Data_CSNL/people/LHS/classes/Computational Modeling/HW3/helloworld.cpp") :
Error 1 occurred building shared library.

Is there anyone who can help me?

^ | v • Reply • Share ›

James Balamuta Mod ➚ 이희승 • a year ago

Missing gfortran per `-lquadmath`

Pick one:

Mojave (10.14) - Catalina (10.15) uses gfortran8.2
https://github.com/fxcouder...

Sierra (10.12) - High Sierra (10.13) uses gfortran6.3
https://github.com/fxcouder...

^ | v • Reply • Share ›

이희승 ➚ James Balamuta • a year ago

Hi James!
Thanks for your advice!
I am using macOS Big Sur 11.2.3 and installed gfortran11 following https://github.com/fxcouder....
Should I uninstall it and choose one of gfortran 8.2 or 6.3?

^ | v • Reply • Share ›

jerry ➚ 이희승 • 9 months ago

How do you uninstall gfortran in this case? I have the same problem and I'd like to install 10.2.

I have 8.2 currently. Thanks in advance to both

^ | v • Reply • Share ›

jerry ➚ jerry • 9 months ago

Nvm about the uninstall, I was able to follow instructions from elsewhere. But now, I can't get to install the fortran version from the GitHub site you pointed out. I downloaded 10.2 for catalina (the .dmg file) but when firing it up I get an apple warning saying it can't be verified and won't open it. Any thoughts?

^ | v • Reply • Share ›

James Balamuta Mod ➚ 이희승 • a year ago

Uninstall and use gfortran 8.2 as you are on an Intel mac, e.g. x86_64-apple-darwin18.

^ | v • Reply • Share ›

이희승 ➚ James Balamuta • a year ago • edited

The problem was resolved by uninstalling gfortran11 and installing 8.2.
Thanks soooooooooo much James for your absolutely quick and precise prescription!!!!!

^ | v • Reply • Share ›

Chhaya Werner • 2 years ago

Thank you so much for this detailed tutorial--unfortunately I'm still running into issues. Using Catalina 10.15.7 and R 4.0.2. I've followed all the instructions to remove old versions, and have confirmed that I have XCode 12.1 and Command Line Tools installed and working by running a test script in XCode directly.

However, when I try to run the test Rcpp script, I get the error:
Error in Rcpp::sourceCpp("~/Documents/testRcpp.cpp") :

Error 1 occurred building shared library.
WARNING: The tools required to build C++ code for R were not found.
Please install Command Line Tools for XCode (or equivalent).

Any suggestions?

Reply • Share ›

James Balamuta Mod → Chhaya Werner • 2 years ago

This indicates that Xcode CLI was not installed. Try downloading Xcode CLI from Apple at:

https://developer.apple.com...

Reply • Share ›

Chhaya Werner → James Balamuta • 2 years ago

I downloaded and installed Xcode CLI from the Apple developer website and am still getting the same error, plus a new one. The full error (below) includes clang++ error output:

clang++ -mmacosx-version-min=10.13 -std=gnu++11 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG -I../inst/include -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/l -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/l -I"/Users/chhaya/Documents" -I/usr/local/include -fPIC -Wall -g -O2 -c testRcpp.cpp -o testRcpp.o

clang++ -mmacosx-version-min=10.13 -std=gnu++11 -dynamiclib -Wl,-headerpad_max_install_names -undefined dynamic_lookup -single_module -multiply_defined suppress -L/Library/Frameworks/R.framework/Resources/lib -L/usr/local/lib -o sourceCpp_2.so testRcpp.o -L/Library/Frameworks/R.framework/Resources/lib -lRlapack -L/Library/Frameworks/R.framework/Resources/lib -lRblas -L/usr/local/gfortran/lib/gcc/x86_64-apple-darwin18/8.2.0 -L/usr/local/gfortran/lib -lgfortran -lquadmath -lm -F/Library/Frameworks/R.framework/.. -framework R -Wl,-framework -Wl,CoreFoundation

Error in Rcpp::sourceCpp("~/Documents/testRcpp.cpp") :
Error 1 occurred building shared library.
WARNING: The tools required to build C++ code for R were not found.

Please install Command Line Tools for XCode (or equivalent).

ld: unsupported tapi file type '!tapi-tbd' in YAML file '/Library/Developer/CommandLineTools/SDKs/MacOSX10.15.s for architecture x86_64
clang: error: linker command failed with exit code 1 (use -v to see invocation)
make: *** [sourceCpp_2.so] Error 1

Reply • Share ›

James Balamuta Mod → Chhaya Werner • 2 years ago

Looks like you installed Xcode 12 CLI on macOS Catalina. Xcode 12 seems to be keyed only for Big Sur (macOS 11).

```
ld: unsupported tapi file type '!tapi-tbd' in YAML file
```

Please uninstall Xcode CLI by typing in terminal:

```
sudo rm -rf /Library/Developer/CommandLineTools
```

Download the Xcode 11.* CLI version from the above link.

Regarding the "WARNING: The tools required to build C++ code for R were not found." This is from RStudio, there might need to be a reset issued to the command path after this is done.

Reply • Share ›

Chhaya Werner ➚ James Balamuta • 2 years ago

Thank you, that did it! Works in R and RStudio now. For some reason the Software Update panel in System Preferenceskeeps suggesting an update to XCode 12.* but I'll just make sure to not allow it

▲ | ▼ • Reply • Share ›

Anna-Leigh Brown • 2 years ago

Have followed this tutorial about a dozen times

Always the same errors. Running on Catalina

```
/Library/Developer/CommandLineTools/usr/bin/../include/c++/v1/cmath:646:1
error: no template named 'numeric_limits'
static_assert(numeric_limits<_FloatT>::radix == 2, "FloatT has incorrect
radix");
^
fatal error: too many errors emitted, stopping now [-ferror-limit=]
20 errors generated.
make: *** [helloworld.o] Error 1
clang++ -mmacosx-version-min=10.13 -std=gnu++11 -
I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG -
I../inst/include -
I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/Rcpp/inc
-
I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppArma
-I"/Users/annaleigh/Desktop" -I/usr/local/include -fPIC -Wall -g -O2 -c
helloworld.cpp -o helloworld.o
Error in Rcpp::sourceCpp("Desktop/helloworld.cpp") :
```

At my wits end, as I can't run any code needing compilation on my mac anymore

▲ | ▼ • Reply • Share ›

James Balamuta Mod ➚ Anna-Leigh Brown • 2 years ago

@Anna-Leigh Brown thanks for writing. Are you on macOS Catalina?

In R, could you run:

```
sessionInfo()
```

And share the output here?

▲ | ▼ • Reply • Share ›

Anna-Leigh Brown ➚ James Balamuta • 2 years ago

Yep I'm on catalina!

I have a feeling there's a general issue with the c++ header locations as I'm not able to compile correctly with python either.

```
R version 4.0.2 (2020-06-22)
Platform: x86_64-apple-darwin17.0 (64-bit)
Running under: macOS Catalina 10.15.5


Matrix products: default
BLAS:
/System/Library/Frameworks/Accelerate.framework/Versions/A/F
LAPACK:
/Library/Frameworks/R.framework/Versions/4.0/Resources/lib/l


locale:
[1] en_GB.UTF-8/en_GB.UTF-8/en_GB.UTF-8/C/en_GB.UTF-
8/en_GB.UTF-8


attached base packages:
[1] stats graphics grDevices utils datasets methods base


loaded via a namespace (and not attached):
```

loaded via a namespace (and not attached):
[1] compiler_4.0.2 htmltools_0.5.0 tools_4.0.2 yaml_2.2.1
tinytex_0.26 rmarkdown_2.5 knitr_1.30 digest_0.6.27
xfun_0.18
[10] rlang_0.4.8 evaluate_0.14

Reply • Share ›

James Balamuta Mod ➚ Anna-Leigh Brown
• 2 years ago • edited

Let's try a reset on the Xcode CLI tools as I'm not sure if Xcode itself is installed on your system.

In Terminal, can you type:

```
sudo xcode-select --reset
sudo xcode-select -s
/Library/Developer/CommandLineTools
```

Another approach is removing and reinstalling Xcode CLI:

```
rm -rf /Library/Developer/CommandLineTools
sudo xcode-select --install
```

One other question: What version of Rcpp are you using? It wasn't listed in the `sessionInfo()`.

Reply • Share ›

Anna-Leigh Brown ➚ James Balamuta • 2 years ago

My bad, had a ton of packages loaded into namespace so did that from a freshly opened R session

```
other attached packages:
[1] Rcpp_1.0.5
```

There are different problems now, which definitely did not occur the last time I tried `sudo xcode-select --install`

Can't install the software because it is not currently available from

the Software Update server.

There are solutions to this online, but I'd rather not have to manuall install x-code if it's not neccessary? I'll go down that route if you think it would be helpful though

https://stackoverflow.com/q...

Reply • Share ›

Anna-Leigh Brown ➚ Anna-Leigh Brown
• 2 years ago • edited

Using the solution from Rstan question
https://discourse.mc-stan.o...

To be clear though I am on R 4.0

adding this as my makevars at least is compiling hello.world

```
# Fill with appropriate flag statements
sudo cat <<- EOF > ~/.R/Makevars
# clang: start
CFLAGS=-isysroot
/Library/Developer/CommandLineTools/SDKs/MacOSX
CCFLAGS=-isysroot
/Library/Developer/CommandLineTools/SDKs/MacOSX
CXXFLAGS=-isysroot
/Library/Developer/CommandLineTools/SDKs/MacOSX
CPPFLAGS=-isysroot
/Library/Developer/CommandLineTools/SDKs/MacOSX
# clang: end
EOF
```

1 Reply • Share ›

James Balamuta Mod ➚ Anna-Leigh Brown • 2 years ago

That's interesting that *FLAGS needed to be pointed at the macOSX.sdk as default flags for > R 4.0 set this already in the site makevars file. Thanks for reporting back with your results!

1 ^ | v • Reply • Share ›

Derek Beach • 2 years ago

This does not work on my machine. I keep getting :

make: *** No rule to make target `world.o', needed by `sourceCpp_5.so'. Stop.
Error in Rcpp::sourceCpp("~/Desktop/R/Hello world.cpp") :
Error 1 occurred building shared library.

I have set the working directory etc correctly. Just nothing happens. I have tried these instructions about 10 times + followed supplemental guidance. Nothing works. Guidance?

^ | v • Reply • Share ›

James Balamuta Mod ➚ Derek Beach • 2 years ago

> make: *** No rule to make target `world.o', needed by `sourceCpp_5.so'. Stop.

What is the context with `world.o`? Is this being run as the hello world example above?
Did you remove ~/.R/Makevars?

What happens when you try:

Rcpp::evalCpp("1 + 1")

^ | v • Reply • Share ›

Alexander Toenges • 2 years ago • edited

With only this tutorial I could not get OpenMP enabled. After removing anaconda completely, and inspired by the manual of the data.table package I then added the following to ~/.R/Makevars:
The third line is important.

```
LOC = /usr/local/gfortran/
CC=$(LOC)/bin/gcc -fopenmp
CXX=$(LOC)/bin/g++ -fopenmp
# -O3 should be faster than -O2 (default) level optimisation ..
CFLAGS=-g -O3 -Wall -pedantic -std=gnu99 -mtune=native -pipe
CXXFLAGS=-g -O3 -Wall -pedantic -std=c++11 -mtune=native -pipe
LDFLAGS=-L/usr/local/opt/gettext/lib -L$(LOC)/lib -Wl,-rpath,$(LOC)/lib
CPPFLAGS=-I/usr/local/opt/gettext/include -I$(LOC)/include -
I/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include
```

With this I could get OpenMP support. One quick test would e.g. be to compile RcppArmadillo which typically complains if the compiler is not OMP compatible.

^ | v • Reply • Share ›

James Balamuta Mod ➚ Alexander Toenges • 2 years ago

The suggested approach by data.table as depicted above changes the underlying compiler from clang to gcc. This is not supported for the official macOS binary from CRAN. Do not use this approach.

There is a better approach for enabling OpenMP. Please see the macOS developer site:
https://mac.r-project.org/o...

^ | v • Reply • Share ›

Justin Silverman • 2 years ago

Can you say more about OpenMP support? Any idea if this is a temporary thing or I should switch to no longer using OpenMP for development?

^ | v • Reply • Share ›

Dave Armstrong • 2 years ago

Thanks; it worked! I was having trouble with compiling rstan. I had recently followed your instructions on the Stan discussion list for installing the 3.6.X toolchain on Catalina, which was also very helpful. After upgrading to 4.0.0 and following your advice here, I was able to re-install rstan from source and get it to work.

get it to work.

Reply • Share ›

ABOUT

TheCoatlessProfessor is a website that strives to bring statistical prowess to the masses through useful articles for the stumbleuponer and googler. Our goal is to make readily available helpful tips, tutorials, and resources that the students of Statistics and Computer Science will appreciate.

WHERE

- Department of Statistics, Illini Hall Rm 101, University of Illinois at Urbana-Champaign, Champaign, Illinois 61820.
- balamut2@illinois.edu (mailto:balamut2@illinois.edu?subject=[TCP]%20your%20inquiry&body=Thanks%20for%20writing%20an%20e-mail,%20please%20note%20that%20I%27m%20a%20bit%20busy%20right%20now.%20If%20you%20need%20technical%20help,%20it%
- @axiomsofxyz (https://twitter.com/axiomsofxyz) on Twitter
- coatless (https://github.com/coatless) on GitHub
- ResearchGate (https://www.researchgate.net/profile/James_Balamuta)
- Google Scholar (https://scholar.google.com/citations?user=q82k-JsAAAAJ)

SUBSCRIBE

To receive updates, you can subscribe to the RSS feed (https://thecoatlessprofessor.com/index.xml) of all updates to the site in an RSS feed reader.

SEARCH

Search with DuckDuckGo