**Steps for submitting a package to CRAN, modified from** [**http://r-pkgs.had.co.nz/release.html**](http://r-pkgs.had.co.nz/release.html)

1. Add a new public repository on GitHub with desired package name. Make sure there is a GitHub ReadMe.
2. In RStudio, create a new project/package.
3. Make sure to update to the newest R, RStudio (Help>Check for updates), and RTools (<https://cran.rstudio.com/bin/windows/Rtools/>) builds, and then move/update all packages. (RStudio demands the latest builds.)
4. Make sure to (1) notate the inst > NEWS.Rd file and (2) update the version number in DESCRIPTION file (if needed) and update the R version dependency. (For the R version, make sure the 3rd-digit “patch” version ends in a zero (e.g., better to use R version 3.4.0 instead of 3.4.3, which would cause a rejection from CRAN).
5. In RStudio, CTRL+Shift+L to load all
6. In RStudio, CTRL+Shift+B to build package.
7. In RStudio, CTRL+Shift+E to troubleshoot / check the package.
8. If building a vignette, use CTRL+Shift+K to preview it (with the vignette source .Rmd file open), and then run devtools::build\_vignettes() to get it properly built into the package (in the /inst folder where vignette() looks). Note Vignette file is the .Rmd file in Vignettes folder (not the one in inst/doc folder!)
	1. Reminder that CTRL-Shift-B does not update Vignettes to save time. To force vignette to be built within package docs, use devtools::build() instead
9. Run devtools::build\_win() to confirm package will run on Windows and to build a Windows binary bundle. Will take up to 30 minutes to run behind the scenes, with e-mails sent confirming success (or failure)
	1. If does not run, try following:
	2. Is .libPaths() set to personal library?
	3. Are R, RStudio, and RTools updated to latest versions?
	4. In Build > Configure Build Tools, make sure "Generate documentation with Roxygen" is checked
	5. In Tools > Global Options > Packages, make sure "View Rcheck directory after failed R CMD check" is checked
10. Log into GitHub and have RStudio access it:
11. Use RStudio Tools > Shell to sync using > git init
12. If need to confirm ssh in R, modify path to: file.exists("C:/Users/YourUsername/.ssh/id\_rsa.pub")
13. When creating new repo at https://github.com/YourUsername, do not add a README or license. Just make a blank repo with a title and short description (longer title of package).
14. This should reveal instructions to ... or push an existing repository from the command line
15. If asks for username, submit Username and/or passphrase.
16. Enter > exit to exit out of Shell
17. If have Travis CI errors, make sure to log into Travis (via GitHub login, top right) to see details.
	1. If the error is caused by Travis having an older version of R, ignore the error and re-submit later.

1. Fix anything needing fixing. When both OK, commit and push to GitHub (https://github.com/YourUsername/PackageName).

1. If need to have RStudio add a file (such as this one) to .Rbuildignore, use:
2. devtools::use\_build\_ignore("Submitting to CRAN.docx"), or use GUI GitHub interface.

1. If needed, check reverse dependencies using following R commands:
2. Confirm search path is set: options(devtools.revdep.libpath = "C:/Users/YourUsername/Documents/R/R\_tmp\_libraries\_for\_revdep")
3. devtools::revdep\_check()
4. devtools::use\_revdep() # to set up the revdep folder
5. Open Check.R (in revdep folder) and copy and paste commands. Results are placed in revdep folder.

1. Make sure NEWS.md is updated (and commit/push to GitHub).

BEST TO DO CLEAN RESTART NOW AND THEN CTRL+Shift+B, CTRL-Shift+E, RESUBMIT TO WINDOWS TO SEE IF ANY FINAL ERRORS. NEXT STEP SUBMITS TO R-CRAN!!

1. Now ready to release with devtools::release() and follow prompts.
2. If get a git error, make sure all files are committed and pushed
3. Also make sure that RStudio's git page is set to the 'master' where the repo is located. (top-right pulldown)

1. See Hadley's section "On failure" for details if rejected and on how to resubmit.