# Submitting to CRAN and Continuous Integration

John Muschelli
http://johnmuschelli.com/smi_hackathon_2018.html
Johns Hopkins Bloomberg School of Public Health

# Unit Testing

- Runs specific code for test cases.
- `testthat` and `RUnit` packages
- `usethis::use_testthat()`

```r
context("All my data tests")
test_that("Data is installed", {
    expect_is(sri24_image_df(), "data.frame")
})
```

- see `expect_equal`, `expect_identical`, etc.

# Unit Testing: Any issue/error/edge case deserves a unit test

# Code Coverage: `covr` package

- Calculates the code coverage: % of lines tested when package is tested
  - excludes white space
  - `covr::package_coverage()` - defaults to tests unit tests only
  - `covr::package_coverage(type = "all")` - runs tests, vignettes, examples.

# Code Coverage: `covr` package

Online services that allow you to view the coverage online:

- Coveralls: https://coveralls.io/
- Codecov: https://codecov.io/

# Checking Packages Locally

- Check the Package: Build → Check Package.
  - Make sure Build → Configure Build Tools. Put `--as-cran` in `Check Package` area.

```
devtools::check(args = c('--as-cran'))
```

- Runs examples, tests, and check imports/suggests/etc.

# Checking Packages Locally

```
devtools::check(args = c('--as-cran'))
```

Before submitting to CRAN:

- Run `available::available("PACKAGE_NAME")`: checks package name
- Shows `notes`: non-global visible things are a by-product of `tidyverse` non-standard evaluation

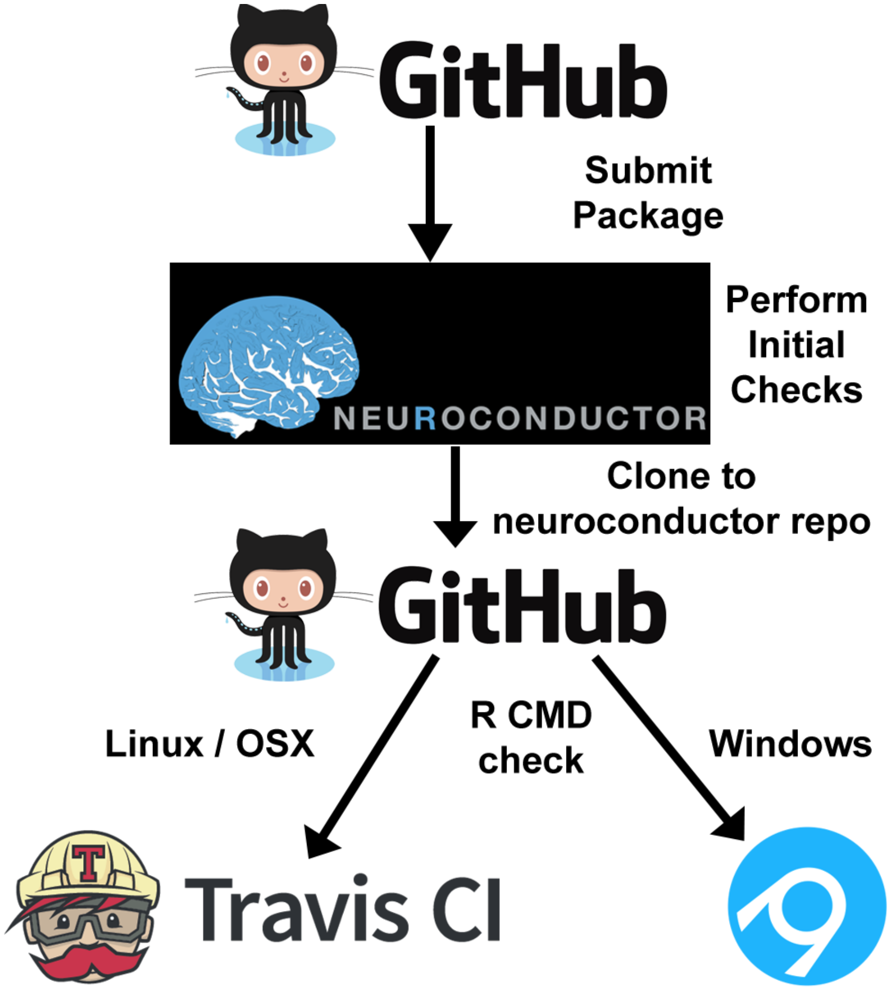
# Continuous Integration: Travis and Appveyor

- Builds and checks R packages on Windows (Appveyor) and Linux/OS X (Travis CI)
- Sign up with GitHub, works well with GitHub
- Installs `R`, runs `R CMD check --as-cran`, and provides result

# Development Pipeline:

Check the package for stability

- uses current Neuroconductor Packages



# Using CI and Coverage: Badges

- `usethis::use_travis()` - creates `.travis.yml`
    - add `warnings_are_errors: true`

Add coverage: `usethis::use_coverage()` - pick `coveralls`/`codecov`:

```
after_success:
  - Rscript -e 'covr::coveralls(type = "all")'
  - Rscript -e 'covr::codecov(type = "all")'
```

- `usethis::use_appveyor()` - creates `appveyor.yml`

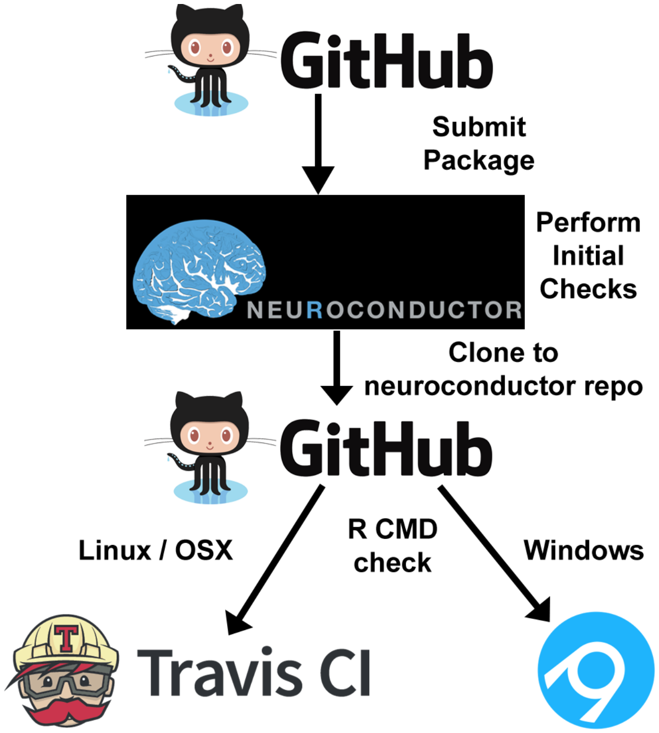
# Submitting to CRAN

- Writing R Extensions - all things for a package
- CRAN policy (long): https://cran.r-project.org/web/packages/policies.html
- `devtools::release()` (needs `NEWS.md` and `cran-comments.md`)
- RStudio → Build → Build Source Package
  - builds vignettes, creates a `.tar.gz`, upload to https://cran.r-project.org/submit.html

# Submitting to Neuroconductor

https://neuroconductor.org/submit-package

- Must be a GitHub repo
- Verify via maintainer email
- See https://neuroconductor.org/list-packages/all



# Solutions to Common Problems

- Clear out `man` folder
- Replace `NAMESPACE` with blank file with `# Generated by roxygen2: do not edit by hand` at the top
- Check your `Imports`/`Suggests`/`Depends`

# Recommendations

- Use `Imports`. Only in very rare cases, use `Depends`, unless depending on `R` version.
- Use `@importFrom` directives instead of `@import`
  - helps you identify problems (moving functions)
- Use the “Go to file/function” search in RStudio
- Separate files for each function (personal preference)