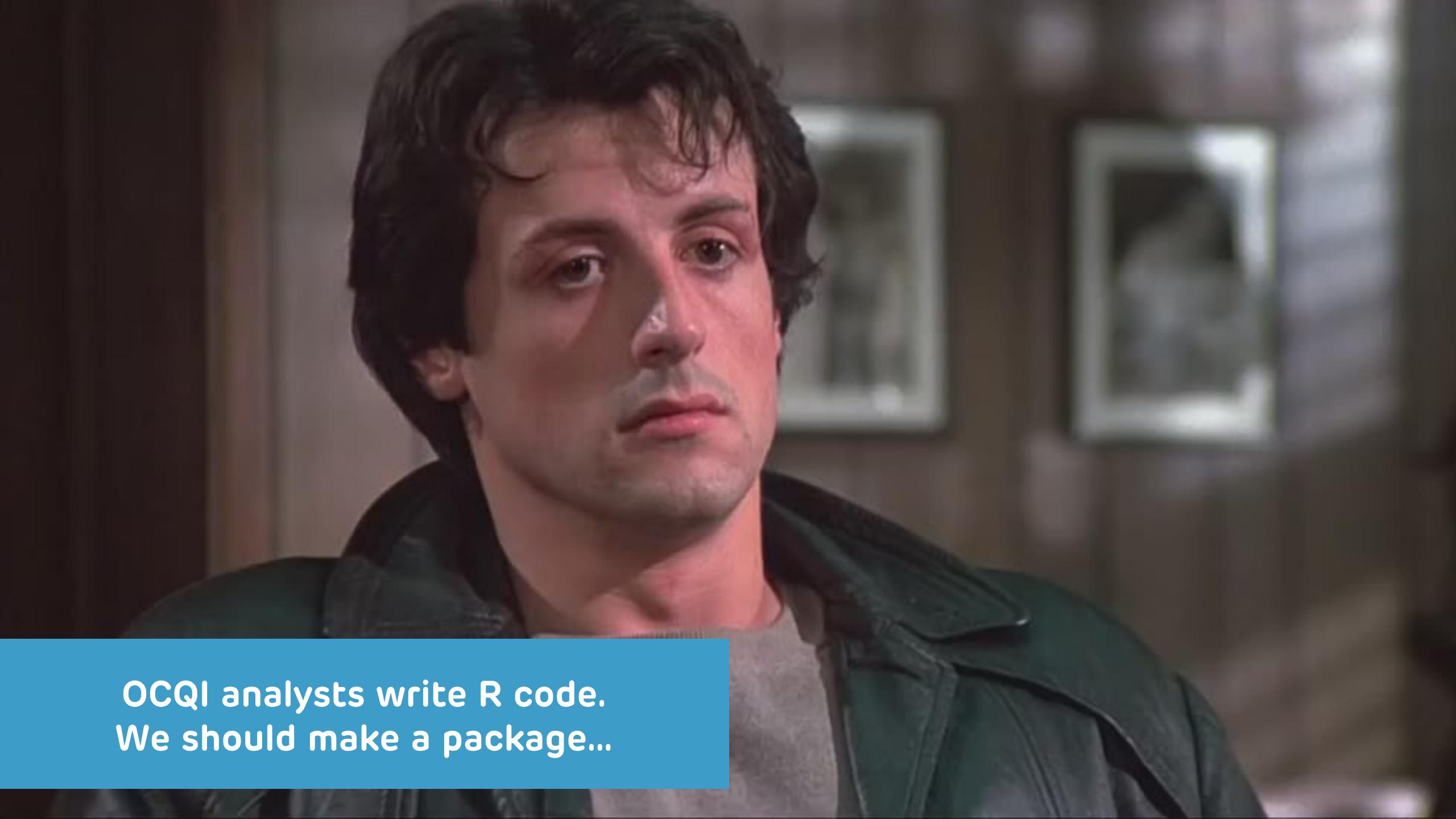


DEVELOPING R PACKAGES FOR STANDARDIZED DATA ANALYSIS

Developing, Distributing, & Maintaining

06/05/2019





OCQI analysts write R code.
We should make a package...

A boxer in a ring with a referee, used as a metaphor for data analysts.

25+ Data Analysts with
varying levels of R skills

R Standardization

New R libraries



!

Messy Repo

Inconsistent Styles

Few Tests

Breaking API
Changes



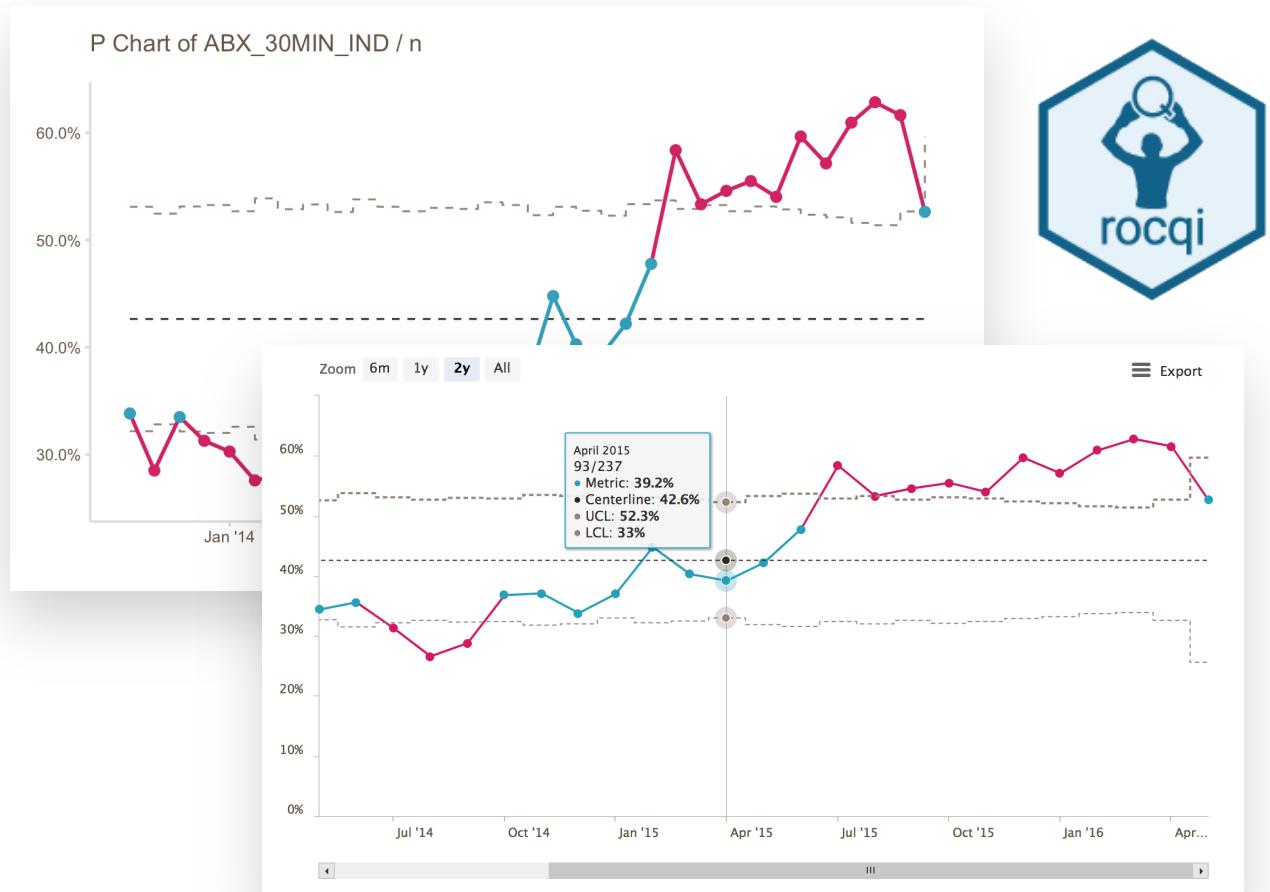
ROCQI II



PACKAGES • ROCQI

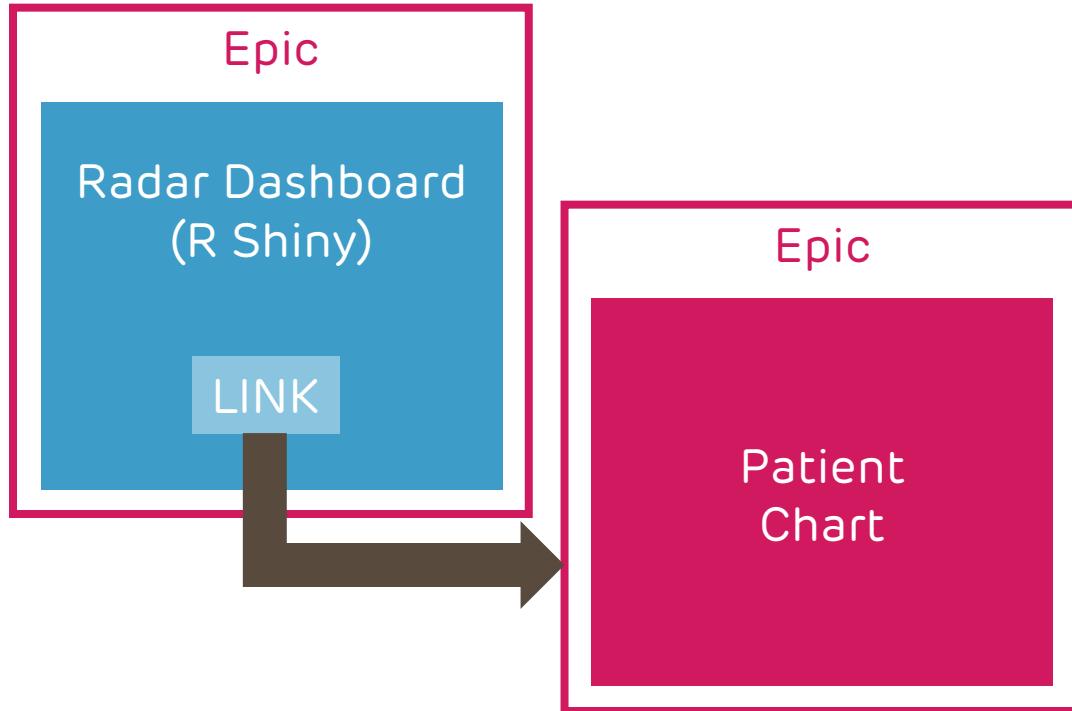
Gather, explore, analyze,
and present data for
CHOP.

```
● ● ●  
library("rocqi")  
  
# ggplot SPC  
spc(data = sepsis, chart = "p"  
    x = HOSP_ADMIT_DT,  
    y = ABX_30MIN_IND)  
  
# highcharter SPC  
hc_spc(data = sepsis, chart = "p"  
    x = HOSP_ADMIT_DT,  
    y = ABX_30MIN_IND)
```



Visualize static and interactive
Statistical Process Control charts

PACKAGES • EPICR



Launch Epic activities from
Radar Dashboard hyperlinks

Interface with Epic.

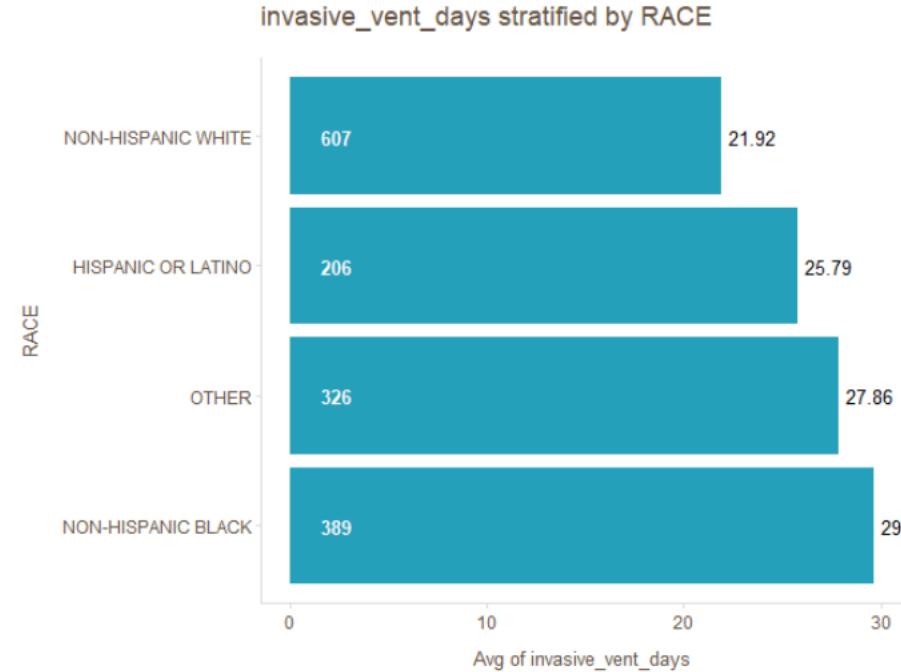
- **EpicACT**
 - `activity_uri()`
 - `chart_review_uri()`
 - `flowsheet_uri()`
- **Web Services**
 - `call_web_service()`
 - `set_smart_data_element()`

PACKAGES • DEMOGRAPHICS

```
library(demographics)

screen_demos(
  table  = "fact_aki_cohort",
  metric = "invasive_vent_days"
)
```

Visualize metrics across demographics.



Non-Hispanic Black patients had on average 30% more days on invasive ventilation than non-Hispanic White patients.

DEVELOPMENT TIPS



DOCUMENTATION

Document everything!



Generate from code

- `pkgdown`



Organize API

- Group by function



Create vignettes

- Guide for common use cases – ask users!

The screenshot shows the rocqi documentation website. The top navigation bar includes links for 'rocqi 1.7.0', 'Home', 'Reference', 'Articles', and 'Changelog'. The main content area has a sidebar titled 'Contents' with sections like 'SPC Basics' and 'Plots with spc()'. The main content area displays the 'Reference' page for the rocqi package, which is organized into several sections: CDW, Exploration, Reporting, and Visualization. Under the 'Visualization' section, there is a code snippet for creating a P Chart:

```
sepsis <- rocqi::sepsis
#> # A tibble: 6,860 x 2
#>   HOSP_ADMIT_DT ABX_30MIN_IND
#>   <date>          <int>
#> 1 2016-05-11      0
#> 2 2016-05-11      1
#> 3 2016-05-10      1
#> 4 2016-05-10      0
#> 5 2016-05-10      1
#> 6 2016-05-10      1
#> 7 2016-05-10      0
#> 8 2016-05-10      0
#> 9 2016-05-10      1
#> 10 2016-05-09     1
#> # ... with 6,850 more rows
```

Below the code is a P Chart titled 'P Chart of ABX_30MIN_IND / n'. The chart plots the percentage of ABX_30MIN_IND over time, showing a fluctuating trend with some significant peaks and troughs.

CODE STYLE

Maintain a consistent style throughout package.

👉 Adopt style guide

- tidyverse

⚾ Catch formatting mistakes

- lintr

abc Check spelling

- spelling

The screenshot shows the R Style Guide page from the tidyverse website. The left sidebar lists chapters 1 through 4 with their sub-sections. The main content area shows sections 2.6 Assignment, 2.7 Semicolons, 2.8 Quotes, and 2.9 Comments, each with examples of good and bad code practices.

2.6 Assignment
Use `<-`, not `=`, for assignment.

```
# Good
x <- 5

# Bad
x = 5
```

2.7 Semicolons
Don't put `;` at the end of a line, and don't use `;` to put multiple commands on one line.

2.8 Quotes
Use `"`, not `'`, for quoting text. The only exception is when the text already contains double quotes and no single quotes.

```
# Good
"Text"
'Text with "quotes"'
'<a href="http://style.tidyverse.org">A link</a>'

# Bad
'Text'
'Text with "double" and \'single\' quotes'
```

2.9 Comments
In data analysis code, use comments to record important findings and analysis decisions. If you need comments to explain what your code is doing, consider rewriting your code to be clearer. If you discover

Source: <https://style.tidyverse.org>

VERSION CONTROL WORKFLOW

Guarantee change transparency
& traceability.

Branch

- Name based on change

Commit

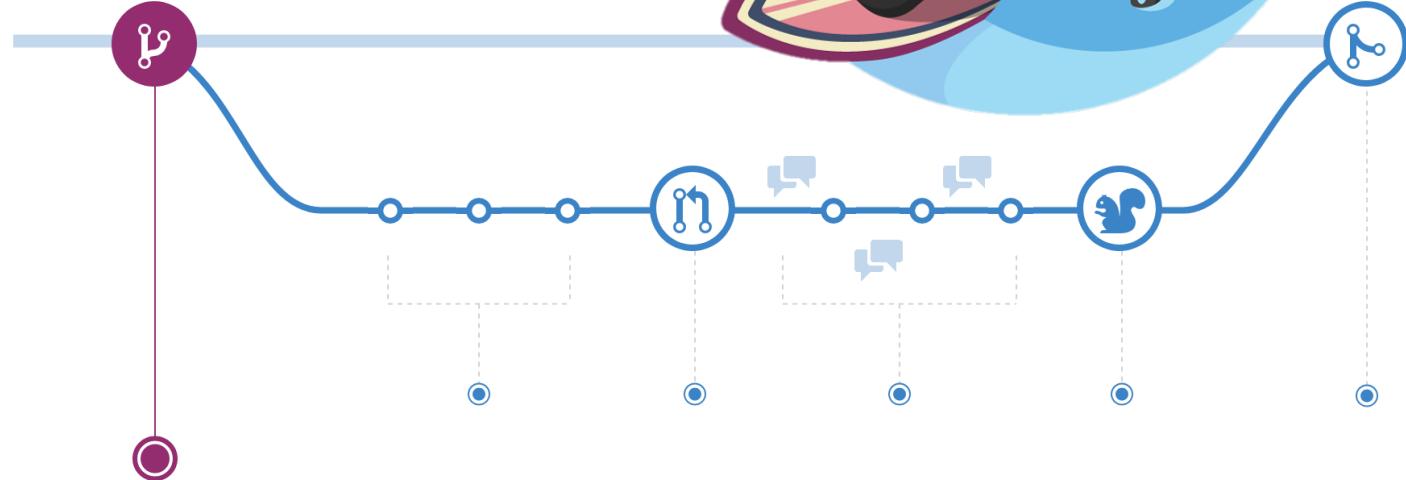
- Helpful commit messages

Pull Request

- Require code reviews

Merge

- Rebase when appropriate



GitHub Flow

Source: <https://guides.github.com/introduction/flow/>
<https://octodex.github.com/suftocat>

CHECKS, TESTS, & COVERAGE

Ensure your package is stable.



Keep CRAN-ready

- `devtools::check()`



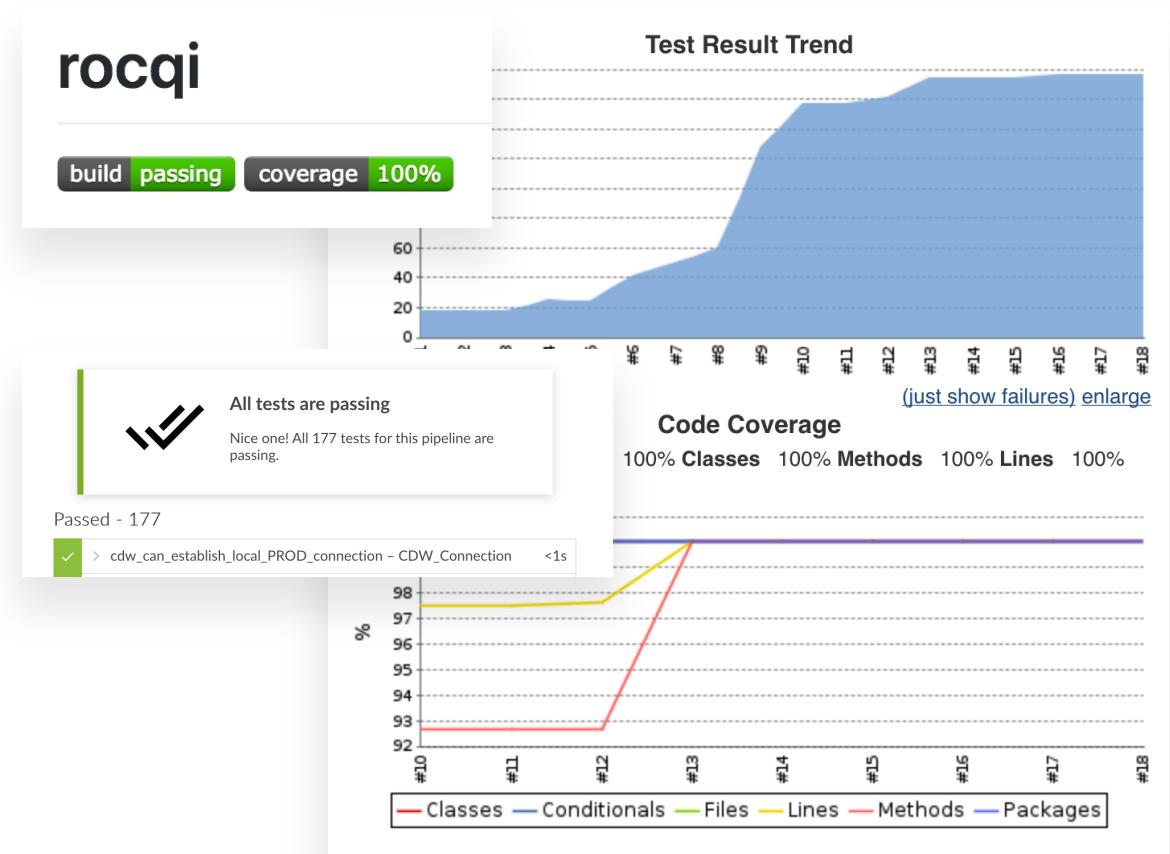
Test everything

- `testthat`



Measure test coverage

- `covr`



CONTINUOUS INTEGRATION

Confidence in new additions to the codebase.



Jenkins

- Pipelines = Configuration-as-code



GitHub Webhooks

- Branch & pull request

The screenshot shows the Jenkins Pipeline interface for the repository 'continuous-integration / rocqli'. The pipeline has 18 runs, all of which have passed. A specific pull request (PR-284) is highlighted, showing a comment from 'mirizioj' dated Mar 6: 'Add test cases to improve coverage score above 90%. This involves ignoring deprecated functions.' Below this comment, another message from 'mirizioj' says 'added some commits on Mar 6' with a list of commit messages: 'Add tracking tests', 'Remove deprecated functions from coverage', 'Fix line endings', 'Remove examples of deprecated functions to pass check', 'Fix line', and 'Add test'. A summary box at the bottom right of the commit list states 'All checks have passed' with '1 successful check'. The Jenkins logo, a cartoon character in a tuxedo holding a mug, is visible in the top right corner. The GitHub logo is also present on the right side of the interface.

ISSUES & PROJECTS

The screenshot shows a project management interface with two main sections. The top section is a Kanban board with four columns: Backlog, Planned, In Progress, and Done. The Backlog column has three items: 'Match formatting between hc_spc and spc: Control limit breaks' (bug), 'Matomo tracking no longer tracks user IDs' (bug), and 'Additional hyperlinks in roxygen documentation' (documentation, good first issue, help wanted). The Planned column has three items: 'Add other sensible defaults to keep_uppercase parameter in nice_display_names()' (enhancement), 'Add grey to chop_colors()' (enhancement), and 'Fix facet borders for theme_chop' (bug, good first issue). The In Progress column has one item: 'Automatic centerline shifting plan' (enhancement). The Done column is empty. The bottom section is a list of issues filtered by 'is:issue is:closed'. The list includes: 'Bump to Version 1.7.0 documentation' (closed, documentation), 'Adding a function to query clarity similarly to the ease at which we query CDW' (closed, enhancement), 'Fix glue dependency' (closed, bug), 'write_cdw_table` now fails to append to OCQI_TABLE_INFO metadata table' (closed, bug), 'Write a blog post about major package updates' (closed, documentation, help wanted), and 'Match formatting between hc_spc and spc' (closed, enhancement).

Transparently track project status and roadmap.

 Bugs & features as Issues

- Issue templates are 🤝

 Projects track progress

- Configure to auto-move issues and PRs

MILESTONES & RELEASES

Version 1.8.0

Due by June 03, 2019 0% complete

The screenshot shows a software interface for managing releases. At the top, there are buttons for 'Edit milestone' and 'New issue'. Below this, a summary shows 5 Open issues and 0 Closed issues. A list of issues follows, each with a checkbox, a title, a description, and a timestamp. The interface includes tabs for 'Releases' and 'Tags', and a section for 'Assets' containing source code links. A 'Draft a new release' button is also visible.

Issue Type	Title	Description	Opened By	Opened On	Comments
enhancement	Automatic centerline shifting plan	#285 opened 10 days ago by rileyj3	rileyj3	1 of 5	1
enhancement	Add grey to chop_colors()	#280 opened 16 days ago by rileyj3	rileyj3		1
bug	Match formatting between hc_spc and spc: Control limit breaks	#272 opened 25 days ago by schmuckern	schmuckern		2
enhancement	Fix facet borders for theme_color	#267 opened on Apr 10 by olsner1	olsner1		
enhancement	Add other sensible defaults to	#266 opened on Apr 10 by wildenhai	wildenhai		

Latest release: 1.7.0 (wildenhai released this 11 days ago)
Assets: Source code (zip), Source code (tar.gz)
Release Version 1.7.0

1.6.1 (mirizioj released this on Apr 2 - 9 commits to master since this release)
Assets: Source code (zip), Source code (tar.gz)
update jenkinsfile grep check

Track and manage releases.

Set date for next releases

- Get on a regular cycle (e.g. Monthly)

Tag releases

- Semantic Versioning
<https://semver.org/>

CRAN-LIKE REPOSITORY

Local R package repository.

💡 Installation instructions

- Add to .Rprofile



🔧 Configure with SCM

- GitHub Enterprise:
`ghentr`

```
👎 devtools::install_github(  
      "account/mypackage",  
      host = "github.research.chop.edu/api/v3"  
    )  
  
👍 install.packages("mypackage")
```

SUMMARY

- 📦 Packages help standardize code and artifacts across a team
- 👤 Adopting open source development practices is beneficial to everyone
- 🥊 Take it one “punch” at a time and see what works for your team

