SCTO Validation Platform

Running function tests

Release date: 2025-01-01

This document is an integral component of the SCTO Validation Platform

---

Document development, review and version history

Development and Review

Authored/revised by:

Name	Date
Alan Haynes,[1]	2025-01-01

Version History

Version	Date	Author(s)	Summary of Changes
1.0	2025-01-01	Alan Haynes,	Initial version

Although the unit of testing is the function, for organisational purposes, the tests are grouped by package. This is because the tests are run in the context of the package, and eases the process of running and reporting tests.

We distinguish between two types of tests:

- those that are incorporated into a package as unit tests, and
- those that have been written by members of the SCTO platform as part of this framework.

The two methods require slightly different approaches to running the tests.

1 Running tests from within a package

The easiest way to run tests from within a package is by using the same approach that the package developer uses. In the majority of cases (especially those using the testthat package), this involves downloading the package to your local machine and running the tests using the `devtools` package.

Downloading from GitHub

On the package GitHub page, click on the green “Code” button and select “Download ZIP”. Unpack the ZIP file.

If you have git installed, you could clone the repository to your computer instead of downloading the ZIP file.

Downloading from CRAN

In the downloads section of the package CRAN page, download the package source (`.tar.gz` file) and unpack it using 7zip. Open the file in 7zip and you should see a .tar folder, go into that and extract the contents to a folder.

Once you have the source code, in an R session, set the working directory to the package directory (e.g. open the .Rproj from the package, if it has one, or use `setwd`) if it does not.

Where packages use the testthat framework, all tests can be run via `devtools::test()`. Specific test files can be run using the filter argument, e.g. `devtools::test(filter = "...")`, where `...` depends on the test file name (tests are typically stored in the `tests/testthat` folder.

If they do not use testthat, you will need to explore the package to find how it’s tests are run.

Test results should be reported on the platform repository. Fill out the form with the appropriate information.

[1] Senior Statistician, Department of Clinical Research (DCR), University of Bern

2 Running tests from within the SCTO framework

Tests are run via the `test` function in the `validation` package:

```
SCTORvalidation::test("accrualPlot")
```

The function will download the testing files from GitHub, run the tests and format the results for easier copy/pasting into a reporting issue on GitHub.

```
Warning: package 'testthat' was built under R version 4.3.3
```

```
Loading required package: accrualPlot
```

```
Warning: package 'accrualPlot' was built under R version 4.3.3
```

```
Loading required package: lubridate
```

```
Warning: package 'lubridate' was built under R version 4.3.3
```

```

Attaching package: 'lubridate'
```

```
The following objects are masked from 'package:base':

    date, intersect, setdiff, union
```

```
✔ | F W  S  OK | Context

⠏ |          0 | accrual_create_df
⠋ |          1 | accrual_create_df
✔ |          8 | accrual_create_df

⠏ |          0 | summary
✔ |         10 | summary

══ Results ═════════════════════════════════════════════════════════════
[ FAIL 0 | WARN 0 | SKIP 0 | PASS 18 ]
```

```
Warning in rm(accrualdemo): object 'accrualdemo' not found
```

```
Error in gh(endpoint = "/user", .token = .token, .api_url = .api_url,  :
  GitHub API error (403): Resource not accessible by integration
ℹ Read more at
  <https://docs.github.com/rest/users/users#get-the-authenticated-user>
```

```
## Copy and paste the following output into the indicated sections of a new issue

ISSUE NAME:
[Package test]: accrualPlot version 1.0.7

### Name
runneradmin
```

```
### Name of the package you have validated
accrualPlot

### What version of the package have you validated?
1.0.7

### Where was the package from?
CRAN (R 4.3.3)

### Package repository version reference
NA

### When was this package tested?
2025-03-20

### What was tested?
Tests for package accrualPlot
 - `summary` produces expected results
 - `accrual_create_df` produces expected results

 These tests are primarily for testing the validation infrastructure. accrualPlot
 has extensive tests


### Test results
PASS

### Test output:


|file                    |context          |test                    | nb| passed|skipped |error |
warning|
|:-----------------------|:----------------|:-----------------------|--:|------:|:-------|:-----|-------:|
|test-accrual_create_df.R |accrual_create_df |monocentric             |  4|      4|FALSE   |FALSE |
0|
|test-accrual_create_df.R |accrual_create_df |multicentric            |  4|      4|FALSE   |FALSE |
0|
|test-summary.R          |summary          |monocentric as expected  |  5|      5|FALSE   |FALSE |
0|
|test-summary.R          |summary          |multicentric as expected |  5|      5|FALSE   |FALSE |
0|

### SessionInfo:
R version 4.3.0 (2023-04-21 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows Server 2022 x64 (build 20348)

Matrix products: default


locale:
[1] LC_COLLATE=English_United States.utf8
[2] LC_CTYPE=English_United States.utf8
[3] LC_MONETARY=English_United States.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.utf8

time zone: UTC
tzcode source: internal

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

other attached packages:
```

```
[1] lubridate_1.9.4 testthat_3.2.3

loaded via a namespace (and not attached):
 [1] rappdirs_0.3.3        generics_0.1.3        tidyr_1.3.1
 [4] stringi_1.8.4         hms_1.1.3             digest_0.6.37
 [7] magrittr_2.0.3        grid_4.3.0            evaluate_1.0.3
[10] timechange_0.3.0      pkgload_1.4.0         fastmap_1.2.0
[13] rprojroot_2.0.4       jsonlite_1.9.1        sessioninfo_1.2.3
[16] cranlogs_2.1.1        brio_1.1.5            conflicted_1.2.0
[19] SCTORvalidation_0.4.2 httr_1.4.7            purrr_1.0.4
[22] scales_1.3.0          httr2_1.1.1           cli_3.6.4
[25] rlang_1.1.5           crayon_1.5.3          gitcreds_0.1.2
[28] munsell_0.5.1         withr_3.0.2           cachem_1.1.0
[31] yaml_2.3.10           tools_4.3.0           accrualPlot_1.0.7
[34] tzdb_0.5.0            memoise_2.0.1         dplyr_1.1.4
[37] colorspace_2.1-1      ggplot2_3.5.1         curl_6.2.1
[40] vctrs_0.6.5           R6_2.6.1              lifecycle_1.0.4
[43] stringr_1.5.1         waldo_0.6.1           pkgconfig_2.0.3
[46] desc_1.4.3            gtable_0.3.6          pillar_1.10.1
[49] glue_1.8.0            gh_1.4.1              xfun_0.51
[52] tibble_3.2.1          tidyselect_1.2.1      knitr_1.50
[55] htmltools_0.5.8.1     rmarkdown_2.29        readr_2.1.5
[58] pkgsearch_3.1.4       compiler_4.3.0


### Where is the test code located for these tests?
SwissClinicalTrialOrganisation/validation_tests

### Where the test code is located in a git repository, add the git commit SHA
8109ba97752735e9949c347518ebc9234ca8edad
```

i Note

It may be that the package being tested, or indeed functions in the testing infrastructure (`testthat` and `waldo`) have been updated since the tests were authored. This may lead to tests failing that previously passed. In this case, the tests should be updated to reflect the new behaviour.

One particular case is a change in attributes. For example, the `coef` method for a model object returns a named vector, but we might compare the individual coeffients to a vector of expected values. There is thus a potential mismatch in the attributes of the two vectors, which may cause the test to fail with a `names for target but not for current` message. Use of the third edition of `testthat` (v3.0.0) should help to avoid this issue.

The results of the tests should be reported on the platform repository. Copy the information returned by R to the appropriate field on the report form.