Running function tests

|  |
| --- |
| Document development, review and version history |
| ***Development and Review***  Authored/revised by:   | Name | Date | | --- | --- | | Alan Haynes,[[1]](#footnote-23) | 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.

## 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](https://github.com/SwissClinicalTrialOrganisation/validation_platform/issues/new?assignees=&labels=test%2C%3Aalarm_clock%3A+triage+%3Aalarm_clock%3A&projects=&template=new_test_internal.yml&title=%5BPackage+selftest%5D%3A+XXX+version+YYY). Fill out the form with the appropriate information.

## 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 ]  
  
Way to go!

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?   
2025-03-27   
  
### 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] praise\_1.0.0 stringi\_1.8.4 hms\_1.1.3   
 [7] digest\_0.6.37 magrittr\_2.0.3 grid\_4.3.0   
[10] evaluate\_1.0.3 timechange\_0.3.0 pkgload\_1.4.0   
[13] fastmap\_1.2.0 rprojroot\_2.0.4 jsonlite\_2.0.0   
[16] sessioninfo\_1.2.3 cranlogs\_2.1.1 brio\_1.1.5   
[19] conflicted\_1.2.0 SCTORvalidation\_0.4.2 httr\_1.4.7   
[22] purrr\_1.0.4 scales\_1.3.0 httr2\_1.1.2   
[25] cli\_3.6.4 rlang\_1.1.5 crayon\_1.5.3   
[28] gitcreds\_0.1.2 munsell\_0.5.1 withr\_3.0.2   
[31] cachem\_1.1.0 yaml\_2.3.10 tools\_4.3.0   
[34] accrualPlot\_1.0.7 tzdb\_0.5.0 memoise\_2.0.1   
[37] dplyr\_1.1.4 colorspace\_2.1-1 ggplot2\_3.5.1   
[40] curl\_6.2.2 vctrs\_0.6.5 R6\_2.6.1   
[43] lifecycle\_1.0.4 stringr\_1.5.1 waldo\_0.6.1   
[46] pkgconfig\_2.0.3 desc\_1.4.3 gtable\_0.3.6   
[49] pillar\_1.10.1 glue\_1.8.0 gh\_1.4.1   
[52] xfun\_0.51 tibble\_3.2.1 tidyselect\_1.2.1   
[55] knitr\_1.50 htmltools\_0.5.8.1 rmarkdown\_2.29   
[58] readr\_2.1.5 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

|  |
| --- |
| 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](https://github.com/SwissClinicalTrialOrganisation/pkg_validation/issues/new?assignees=&labels=test%2C%3Aalarm_clock%3A+triage+%3Aalarm_clock%3A&projects=&template=new_test.yml&title=%5BPackage+test%5D%3A+XXX+version+YYY). Copy the information returned by R to the appropriate field on the report form.

1. Senior Statistician, Department of Clinical Research (DCR), University of Bern [↑](#footnote-ref-23)