



# A fresh look at unit testing with **tinytest**

Mark van der Loo  
Statistics Netherlands



*useR!*2021  
Zürich (virtually)

[@markvdloo](https://twitter.com/markvdloo)

[github.com/markvanderloo](https://github.com/markvanderloo)

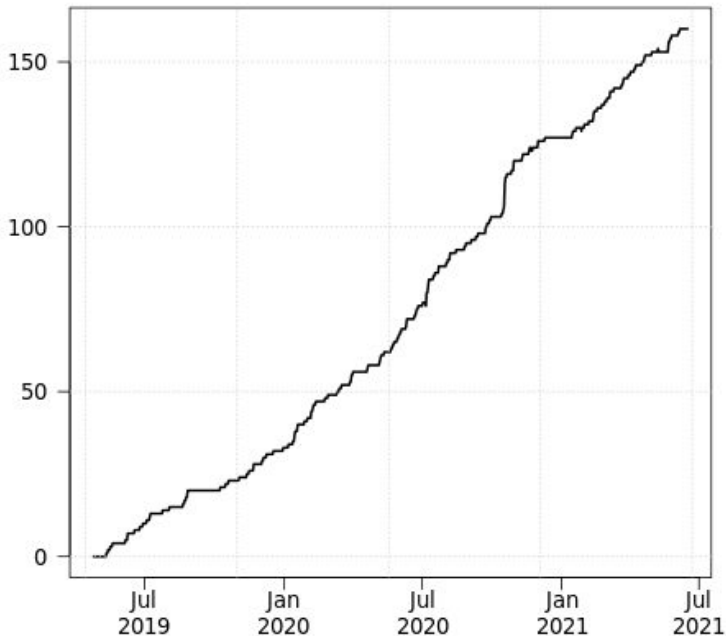
**Materials:** [github.com/markvanderloo/useR2021](https://github.com/markvanderloo/useR2021)

# Unit testing

```
> add_one <- function(x) x + 2
>
> expect_equal(add_one(1), 2)
----- FAILED[data]: <-->
call| expect_equal(add_one(1), 2)
diff| Expected '2', got '3'
```



# Packages suggesting tinytest



Including CRAN and bioconductor, last updated Jun 16, 2021



**Supported by**  
 - littler  
 - pkgKitten

```

✓ Add 'tinytest' to list of test-suite packages.
git-vsn-1d: https://svn.r-project.org/R/trunk@79900 00b4003-00f-0310-0c12-ca700c1e041
└─ master
└─ horrik committed 17 hours ago 1 parent d31525e com11 8026909d5426647d346a0228045f4b063d4a

Showing 1 changed file with 1 addition and 1 deletion.
└─ 2 src/library/tools/R/check.R
└─ 00-217.7-217.7_00 src/libs <-
217 |         sug                               217 |         sug
218 |     }                                       218 |     }
219 | }                                           219 | }
220 | If(tests) do we need the test-suite package 220 | If(tests) do we need the test-suite package
221 | available                                 221 | available
222 |     c("RUnit", "testthat")                222 |     c(sug, intersect(names(p$Suggests),
223 |         + "tinytest"))                      c("RUnit", "testthat", "tinytest"))
224 | }                                           224 | }
225 | deps <- unique(c(names(p$Depends),         225 |     + unique(c(names(p$Depends),
226 |         names(p$Imports)))                   names(p$Imports)))
    
```

## test suite package in tools

2 vignettes

**Method:**  
[arxiv.org/abs/2002.07472](https://arxiv.org/abs/2002.07472)



- [tinytest\\_0.9.2.tar.gz](#) 2019-04-25 18:20 178K
- [tinytest\\_0.9.3.tar.gz](#) 2019-05-10 15:40 193K
- [tinytest\\_0.9.4.tar.gz](#) 2019-06-05 14:00 215K
- [tinytest\\_0.9.6.tar.gz](#) 2019-06-28 15:20 217K
- [tinytest\\_1.0.0.tar.gz](#) 2019-08-23 23:00 411K
- [tinytest\\_1.1.0.tar.gz](#) 2019-09-26 17:20 414K
- [tinytest\\_1.2.0.tar.gz](#) 2020-05-06 15:00 424K
- [tinytest\\_1.2.1.tar.gz](#) 2020-05-18 14:10 425K
- [tinytest\\_1.2.2.tar.gz](#) 2020-06-18 21:30 426K
- [tinytest\\_1.2.3.tar.gz](#) 2020-10-07 18:30 430K





An Chu  
Ott Toomet  
Jon Clayden  
Michel Lang  
Brian Ripley  
Thomas Fuchs  
Jan Wijffels  
Bryan Hanson  
Lionel Henry  
John Blischak  
Jozef Hajnala  
Maja Zaloznik  
James Thompson  
Sebastian Meyer  
Patrick Breheny  
GH user Billy34  
Matthijs Berends  
GH user salim-b  
Henrik Bengtsson  
Dirk Edelbuettel  
George G Vega Yon  
Bart-Jan van Rossum  
Jonas Kristoffer Lindeloef



mypkg/

DESCRIPTION

inst

└─ tinytest

└─ test\_csort.R

└─ test\_quad.R

← Unit tests

man

└─ csort.Rd

└─ quad.Rd

NAMESPACE

R

└─ mycode.R

`tinytest::test_package("mypkg")`

tests

└─ tinytest.R

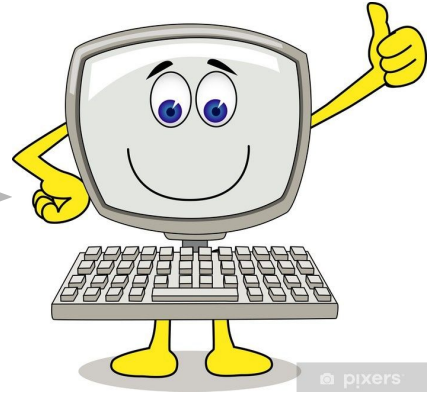


# Tests travel with the package



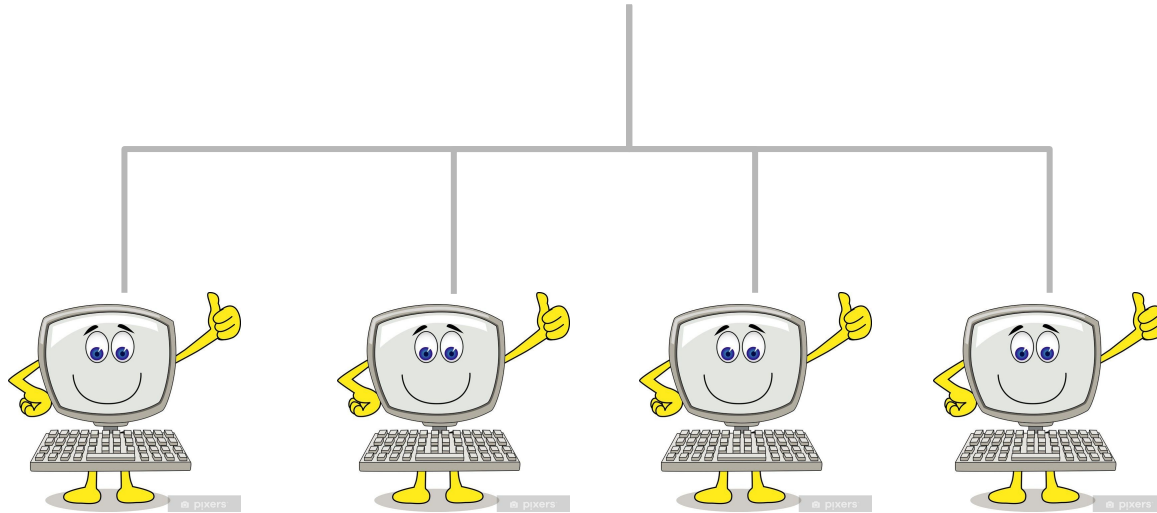
The Comprehensive R  
Archive Network

`expect_equal(1+1, 2)`



# Testing in parallel

```
R> run_test_dir(ncpu=4)
```



# Reporting side effects

```
R> run_test_file(side_effects=TRUE)  
report_side_effects()
```



LC\_COLLATE="C"





# tinytest

- Small, **dependency-free** package
- **Easy to set up** and use
- Support for **pkg building cycle**
- **Parallel testing**, as easy as `ncpu=n`
- **Track side effects** with ease
- **Tests travel** with the package



Using tinytest  
Mark van der Loo  
June 18, 2021 | Package version 1.2.4.11

Contents	
1. Purpose of this package: unit testing	2
2. Expressing tests	3
2.1. Test functions	3
2.2. Alternative syntax	4
2.3. Interpreting the output and print options	4
3. Test files	5
3.1. Surrounding test results, getting the data	5
3.2. Programming over tests, ignoring test results, testing only	6
3.3. Running order and side effects	7
3.4. Monitoring side effects	7
4. Testing packages	8
4.1. Build test-able test intentionally	8
4.2. Testing functions that are not exported via <code>TEST</code>	8
4.3. Using data stored in files	10
4.4. Skipping tests via <code>CRAN</code>	10
4.5. Testing your package after installation	11
4.6. Using extension packages	11
4.7. Mocking dependencies	11
5. Testing with environmental variables	12
6. Running tests in parallel	12
7. At the file or package and unit testing	13
7.1. Make your package testable	13
7.2. Test the surface, not the internals	13
7.3. How many tests did I write?	14
7.4. It's not a bug, it's a feature!	15

Tinytest by example  
Mark van der Loo  
June 20, 2021 | Package version 1.2.4.11

Contents	
1. <code>expect_equal</code>	2
2. <code>expect_equal2</code>	3
3. <code>expect_identical</code>	4
4. <code>expect_null</code>	4
5. <code>expect_true</code> , <code>expect_false</code>	6
6. <code>expect_message</code>	7
7. <code>expect_warning</code>	8
8. <code>expect_error</code>	9
9. <code>expect_output</code>	10
10. <code>expect</code>	11
Introduction	
This document provides a number of real-life examples on how <code>tinytest</code> is used by other packages. The examples aim to illustrate the general idea of writing functions that are easy to test in the test file documentation and the using <code>tinytest</code> package. There is a section for each function. Each section starts with a short example that demonstrates the core purpose of the function. Next, one or more examples from packages that are published on <code>CRAN</code> are shown and explained.	
Sometimes a few lines of code were modified or added for brevity. This is indicated with comment blocks between square brackets, e.g.	
[1] <code>test</code> in the <code>expect_message</code> only for this report.	
This document is probably not interesting to read from to back. It is more aimed to browse since it is able to get an idea on how <code>tinytest</code> can be used in practice.	
Package authors are invited to contribute new use cases or new users can learn from them. Please contact the author of this package either by email or via the GitHub repository.	

