

# Package ‘cyclocomp’

March 11, 2026

**Title** Cyclomatic Complexity of R Code

**Version** 1.1.2

**Description** Cyclomatic complexity is a software metric (measurement), used to indicate the complexity of a program. It is a quantitative measure of the number of linearly independent paths through a program's source code. It was developed by Thomas J. McCabe, Sr. in 1976.

**License** MIT + file LICENSE

**URL** <https://github.com/gaborcsardi/cyclocomp>

**BugReports** <https://github.com/gaborcsardi/cyclocomp/issues>

**Imports** callr, crayon, desc, remotes, withr

**Suggests** processx, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**NeedsCompilation** no

**Author** Gabor Csardi [aut, cre]

**Maintainer** Gabor Csardi <[csardi.gabor@gmail.com](mailto:csardi.gabor@gmail.com)>

**Repository** CRAN

**Date/Publication** 2026-03-11 10:10:02 UTC

## Contents

cyclocomp . . . . .	2
cyclocomp_package . . . . .	3
cyclocomp_package_dir . . . . .	4

<b>Index</b>	<b>5</b>
--------------	----------

---

`cyclocomp`*Cyclomatic Complexity of R Code*

---

### Description

Cyclomatic complexity is a software metric (measurement), used to indicate the complexity of a program. It is a quantitative measure of the number of linearly independent paths through a program's source code. It was developed by Thomas J. McCabe, Sr. in 1976.

Calculate the cyclomatic complexity of an R function or expression.

### Usage

```
cyclocomp(expr)
```

```
cyclocomp_q(expr)
```

### Arguments

`expr`            An R function or expression.

### Value

Integer scalar, the cyclomatic complexity of the expression.

### See Also

Useful links:

- <https://github.com/gaborcsardi/cyclocomp>
- Report bugs at <https://github.com/gaborcsardi/cyclocomp/issues>

Other cyclomatic complexity: [cyclocomp\\_package\(\)](#), [cyclocomp\\_package\\_dir\(\)](#)

### Examples

```
## Supply a function
cyclocomp(
  function(arg) { calculate(this); and(that) }
)
cyclocomp(ls)
cyclocomp(cyclocomp)

## Or a quoted expression
cyclocomp(quote( if (condition) "foo" else "bar" ))

## cyclocomp_q quotes the expression for you
cyclocomp_q(while (condition) { loop })

## Complexity of individual control flow constructs
```

```
cyclocomp(quote({
  if (condition) this
}))

cyclocomp(quote({
  if (condition) this else that
}))

cyclocomp(quote({
  for (var in seq) expr
}))

cyclocomp(quote({
  while (cond) expr
}))

cyclocomp(quote({
  repeat expr
}))

cyclocomp(quote({
  for (var in seq) {
    this
    break
    that
  }
}))

cyclocomp(quote({
  for (var in seq) {
    this
    next
    that
  }
}))
```

---

cyclocomp\_package      *Cyclomatic complexity of the objects in an installed package*

---

### **Description**

Note that the package must be installed.

### **Usage**

```
cyclocomp_package(package)
```

### **Arguments**

package      Package name, character scalar.

**Value**

Data frame with two columns: name and cyclocomp.

**See Also**

Other cyclomatic complexity: [cyclocomp\(\)](#), [cyclocomp\\_package\\_dir\(\)](#)

**Examples**

```
## They might take a while to run
## Not run:
cyclocomp_package("grDevices")
cyclocomp_package("methods")

## End(Not run)
```

---

cyclocomp\_package\_dir *Cyclomatic complexity of a local package*

---

**Description**

Automatically builds the package and installs it to a temporary directory.

**Usage**

```
cyclocomp_package_dir(path = ".", quiet = TRUE)
```

**Arguments**

path	Path to the root directory of the R package.
quiet	If 'FALSE', display package build information.

**Value**

Data frame with two columns: name and cyclocomp.

**See Also**

Other cyclomatic complexity: [cyclocomp\(\)](#), [cyclocomp\\_package\(\)](#)

# Index

## \* cyclomatic complexity

cyclocomp, [2](#)

cyclocomp\_package, [3](#)

cyclocomp\_package\_dir, [4](#)

cyclocomp, [2](#), [4](#)

cyclocomp-package (cyclocomp), [2](#)

cyclocomp\_package, [2](#), [3](#), [4](#)

cyclocomp\_package\_dir, [2](#), [4](#), [4](#)

cyclocomp\_q (cyclocomp), [2](#)