

# Package ‘centrifugeR’

May 8, 2026

**Type** Package

**Title** Non-Trivial Balance of Centrifuge Rotors

**Version** 0.1.7

**Description** Find the numbers of test tubes that can be balanced in centrifuge rotors and show various ways to load them. Refer to Pham (2020)  [<doi:10.31224/osf.io/4xs38>](https://doi.org/10.31224/osf.io/4xs38) for more information on package functionality.

**Depends** R ( $\geq$  4.2.0)

**Imports** pracma ( $\geq$  2.4.2), shiny ( $\geq$  1.7.4), shinythemes ( $\geq$  1.2.0), grDevices, graphics, utils

**URL** <https://phamdn.shinyapps.io/centrifugeR/>

**BugReports** <https://github.com/phamdn/centrifugeR/issues>

**License** GPL-3

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**NeedsCompilation** no

**Author** Duy Nghia Pham [aut, cre] (ORCID:  [<https://orcid.org/0000-0003-1349-1710>](https://orcid.org/0000-0003-1349-1710))

**Maintainer** Duy Nghia Pham  [<nghiapham@yandex.com>](mailto:nghiapham@yandex.com)

**Repository** CRAN

**Date/Publication** 2022-12-15 22:40:07 UTC

## Contents

centrifugeR-package . . . . .	2
launch . . . . .	3
p2c . . . . .	3
pie2 . . . . .	3
rotor . . . . .	5
rotorCheck . . . . .	6
rotorEqual . . . . .	7

rotorSpeed . . . . .	8
rotorUnequal . . . . .	8
rotorVerify . . . . .	10
<b>Index</b>	<b>11</b>

---

centrifugeR-package    *centrifugeR: Non-Trivial Balance of Centrifuge Rotors*

---

## Description

Find the numbers of test tubes that can be balanced in centrifuge rotors and show various ways to load them. Refer to Pham (2020) [doi:10.31224/osf.io/4xs38](https://doi.org/10.31224/osf.io/4xs38) for more information on package functionality.

## Guidelines

centrifugeR helps obtain the perfect centrifuge balance. Call `rotor` to know how many tubes can be loaded into the rotor and see different ways to place these tubes. Alternatively, call `launch` to run the Shiny app. Also, call `rotorVerify` to verify the balance of pre-existing tube configurations. Call `rotorSpeed` for RPM/RCF conversion.

## Copyright

centrifugeR: Non-Trivial Balance of Centrifuge Rotors. Copyright (C) 2020-2022 Duy Nghia Pham

centrifugeR is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

centrifugeR is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with centrifugeR. If not, see <https://www.gnu.org/licenses/>.

## Author(s)

Duy Nghia Pham <[nghiapham@yandex.com](mailto:nghiapham@yandex.com)>

---

launch	<i>Run the Shiny App</i>
--------	--------------------------

---

**Description**

Run the Shiny App

**Usage**

launch()

---

p2c	<i>Convert Hole Positions to Coordinates</i>
-----	--

---

**Description**

p2c returns the coordinates of rotor holes. This is not meant to be called directly.

**Usage**

p2c(x, y)

**Arguments**

x	an integer or a vector, the hole position(s).
y	an integer, the number of rotor holes.

**Value**

p2c returns a vector of coordinates.

---

pie2	<i>Draw a Rotor</i>
------	---------------------

---

**Description**

pie2 was modified after [pie](#). This is not meant to be called directly.

**Usage**

```
pie2(  
  x,  
  labels = names(x),  
  edges = 200,  
  radius = 0.8,  
  clockwise = FALSE,  
  init.angle = if (clockwise) 90 else 0,  
  density = NULL,  
  angle = 45,  
  col = NULL,  
  border = NULL,  
  lty = NULL,  
  main = NULL,  
  panel = NULL,  
  ...  
)
```

**Arguments**

x	See <a href="#">pie</a> for more details.
labels	See <a href="#">pie</a> for more details.
edges	See <a href="#">pie</a> for more details.
radius	See <a href="#">pie</a> for more details.
clockwise	See <a href="#">pie</a> for more details.
init.angle	See <a href="#">pie</a> for more details.
density	See <a href="#">pie</a> for more details.
angle	See <a href="#">pie</a> for more details.
col	See <a href="#">pie</a> for more details.
border	See <a href="#">pie</a> for more details.
lty	See <a href="#">pie</a> for more details.
main	See <a href="#">pie</a> for more details.
panel	a letter for labeling panel.
...	See <a href="#">pie</a> for more details.

**Value**

pie2 returns a figure.

rotor

*Balance Tubes Using Higher-Order Symmetrical Configurations***Description**

rotor returns the numbers of tubes that can and cannot be loaded in a centrifuge rotor and optionally shows various ways to balance a certain number of tubes.

**Usage**

```
rotor(n, k = NULL, seed = 1, elapse = 1)
```

**Arguments**

n an integer, the number of rotor holes.  
 k an integer, the number of tubes (optional).  
 seed an integer, the seed for random number generation. Setting a seed ensures the reproducibility of the result. See [set.seed](#) for more details.  
 elapse an integer, the constrained time in seconds for random sampling.

**Details**

The number of rotor holes n ranges from 4 to 48.

**Value**

rotor returns a list with two components:

check a list with three components:  
 n the number of rotor holes.  
 valid a vector containing the numbers of tubes that can be loaded.  
 invalid a vector containing the numbers of tubes that cannot be loaded.  
 load a list with three components:  
 k the number of tubes.  
 decompose a data frame showing different ways to decompose k.  
 hole a data frame showing hole positions to load tubes.  
 visual a list of rotor images showing hole positions to load tubes.

**References**

Sivek G. On vanishing sums of distinct roots of unity. *Integers*. 2010;10(3):365-8.

Peil O, Hauryliuk V. A new spin on spinning your samples: balancing rotors in a non-trivial manner. arXiv preprint arXiv:1004.3671. 2010 Apr 21.

**See Also**

[rotorVerify](#) for verifying the balance of pre-existing tube configurations.

**Examples**

```
rotor(30, 7)
```

---

rotorCheck

*Check Centrifuge Rotors*


---

**Description**

rotorCheck returns the numbers of tubes that can and cannot be loaded in a single operation.

**Usage**

```
rotorCheck(n, k = NULL)
```

**Arguments**

n                    an integer, the number of rotor buckets.  
k                    an integer, the number of tubes.

**Details**

The number of rotor buckets n ranges from 4 to 48.

If k is specified, rotorCheck will check whether the input number of tubes can be loaded or not.

**Value**

rotorCheck returns a list with two components:

valid                a vector containing the numbers of tubes that can be loaded.  
invalid              a vector containing the numbers of tubes that cannot be loaded.

**References**

Sivek G. On vanishing sums of distinct roots of unity. *Integers*. 2010;10(3):365-8.

**See Also**

[rotorEqual](#) for balancing tubes of equal mass and [rotorUnequal](#) for balancing tubes of unequal mass.

---

`rotorEqual`*Balance Tubes of Equal Mass*

---

### Description

`rotorEqual` returns the positions of rotor buckets that must be loaded or empty to balance tubes of equal mass.

### Usage

```
rotorEqual(n, k, seed = 2019)
```

### Arguments

<code>n</code>	an integer, the number of rotor buckets.
<code>k</code>	an integer, the number of tubes.
<code>seed</code>	an integer, the seed for random number generation. Setting a seed ensures the reproducibility of the result. See <a href="#">set.seed</a> for more details.

### Details

The number of rotor buckets `n` ranges from 4 to 48. The number of tubes `k` must be greater than 0 and smaller than the number of rotor buckets `n`.

### Value

`rotorEqual` returns a list with two components:

<code>loaded</code>	a vector containing the positions of rotor buckets that must be loaded.
<code>empty</code>	a vector containing the positions of rotor buckets that must be empty.

`rotorEqual` also plots a schematic diagram of the centrifuge rotor.

### References

Sivek G. On vanishing sums of distinct roots of unity. *Integers*. 2010;10(3):365-8.

Peil O, Hauryliuk V. A new spin on spinning your samples: balancing rotors in a non-trivial manner. arXiv preprint arXiv:1004.3671. 2010 Apr 21.

### See Also

[rotorCheck](#) for checking centrifuge rotors and [rotorUnequal](#) for balancing tubes of unequal mass.

---

rotorSpeed	<i>Calculate RPM/RCF</i>
------------	--------------------------

---

**Description**

rotorSpeed converts rotational speed to relative centrifuge force and vice versa.

**Usage**

```
rotorSpeed(radius, value, type)
```

**Arguments**

radius	a numeric, the centrifugal radius in millimeters.
value	a numeric, the rotational speed in revolutions per minute or the relative centrifuge force in $\times$ g.
type	the type of the above-mentioned value, "rpm" for rotational speed or "rcf" for relative centrifuge force.

**Value**

rotorSpeed returns a numeric that is the rotational speed in revolutions per minute or the relative centrifuge force in  $\times$  g.

**References**

Rickwood D, editor. Centrifugation: a practical approach. London: Information Retrieval Ltd; 1978. 224 p.

**Examples**

```
rotorSpeed(100, 12000, "rpm")  
rotorSpeed(100, 6000, "rcf")
```

---

rotorUnequal	<i>Balance Tubes of Unequal Mass</i>
--------------	--------------------------------------

---

**Description**

rotorUnequal returns the required masses and the positions of tubes of unequal initial mass.

**Usage**

```
rotorUnequal(n, mass = NULL, seed = 2019)
```

**Arguments**

n	an integer, the number of rotor buckets.
mass	a numeric vector with optional names attribute, the masses (and optional names) of tubes.
seed	an integer, the seed for random number generation. Setting a seed ensures the reproducibility of the result. See <a href="#">set.seed</a> for more details.

**Details**

The number of rotor buckets n ranges from 4 to 48. The number of tubes (i.e. `length(mass)`) should not be greater than the number of rotor buckets n.

If mass is not specified, the names and the masses of tubes must then be taken from the keyboard. In case mass has no names attribute, tubes will be named automatically (i.e. S1, S2, S3, etc.).

**Value**

rotorUnequal returns a data frame with three columns:

initial	a vector containing the initial masses of tubes.
required	a vector containing the required masses of tubes.
position	a vector containing the bucket positions of tubes.

rotorUnequal also plots a schematic diagram of the centrifuge rotor.

**References**

Sivek G. On vanishing sums of distinct roots of unity. *Integers*. 2010;10(3):365-8.

Peil O, Hauryliuk V. A new spin on spinning your samples: balancing rotors in a non-trivial manner. arXiv preprint arXiv:1004.3671. 2010 Apr 21.

**See Also**

[rotorCheck](#) for checking centrifuge rotors and [rotorEqual](#) for balancing tubes of equal mass.

**Examples**

```
# Call the function then input the names and the masses of tubes
rotorUnequal(30)
liver
10.05
gill
9.68
muscle
9.88

# Prepare the masses of tubes then call the function
samples <- round(rnorm(19, mean = 10, sd = 0.5), 2)
```

```
rotorUnequal(30, samples)

# Prepare the masses and the names of tubes then call the function
small.samples <- c(10.05, 9.68, 9.88)
names(small.samples) <- c("liver", "gill", "muscle")
rotorUnequal(30, small.samples)
```

---

rotorVerify

*Verify Centrifuge Balance*

---

### Description

rotorVerify checks whether the rotor is balanced given the positions of tubes of equal mass.

### Usage

```
rotorVerify(n, pos)
```

### Arguments

n	an integer, the number of rotor buckets.
pos	an integer vector, the positions of tubes.

### Details

The number of rotor buckets n ranges from 4 to 48. The positions of tubes pos ranges from 1 to n.

### Value

rotorVerify returns 1 if the rotor is balanced and 0 if the rotor is unbalanced.

### References

Johnsson M. Balancing a centrifuge. R-bloggers. 2016. Available from: <https://www.r-bloggers.com/2016/06/balancing-a-centrifuge/>.

### See Also

[rotorCheck](#) for checking centrifuge rotors.

### Examples

```
rotorVerify(30, c(10, 20, 30))
rotorVerify(30, c(1, 11, 21, 4, 28))
```

# Index

centrifugeR-package, [2](#)

launch, [2](#), [3](#)

p2c, [3](#)

pie, [3](#), [4](#)

pie2, [3](#)

rotor, [2](#), [5](#)

rotorCheck, [6](#), [7](#), [9](#), [10](#)

rotorEqual, [6](#), [7](#), [9](#)

rotorSpeed, [2](#), [8](#)

rotorUnequal, [6](#), [7](#), [8](#)

rotorVerify, [2](#), [6](#), [10](#)

set.seed, [5](#), [7](#), [9](#)