

Package: ICCDesign (via r-universe)

June 8, 2026

Title Intraclass Correlation Coefficient (ICC) Design, Calculation and Interactive 'shiny' Toolkit

Version 0.1.1

Description A comprehensive toolkit for intraclass correlation coefficient (ICC) analysis, integrating three core functionalities: (1) Closed-form sample size calculation for ICC estimation with assurance probability, based on Zou (2012) <doi:10.1002/sim.5466>; (2) Full implementation of all 10 ICC types (6 common + 4 supplementary) for point estimation, exact confidence interval calculation, and formal hypothesis testing, following the methods of McGraw & Wong (1996) <doi:10.1037/1082-989X.1.1.30> and the standard decision framework; (3) An interactive 'shiny' application that guides users through ICC type selection, performs calculations, and provides reliability evaluation based on the Koo & Li (2016) <doi:10.1016/j.jcm.2016.02.012> criteria. Compared to existing packages, it provides a unified decision workflow and supports all less common ICC variants.

License GPL (>= 3)

Depends R (>= 4.1.0)

Imports stats, shiny

Suggests testthat, knitr, rmarkdown

Encoding UTF-8

LazyData false

URL <https://github.com/KlariZhang/ICCDesign>

BugReports <https://github.com/KlariZhang/ICCDesign/issues>

Config/roxygen2/version 8.0.0

RoxygenNote 8.0.0

Config/pak/sysreqs cmake make libuv1-dev zlib1g-dev

Repository <https://klarizhang.r-universe.dev>

Date/Publication 2026-06-02 10:22:18 UTC

RemoteUrl <https://github.com/klarizhang/iccdesign>

RemoteRef HEAD

RemoteSha 14765e59838f7a632197ee55135b024f300a09cd

Contents

ICCDesign-package	2
icc_calc	3
icc_data	5
icc_power	6
icc_preprocess_data	7
icc_sample_size	8
icc_sample_size_lower	9
icc_sample_size_width	10
run_icc_app	11
Index	12

ICCDesign-package	<i>ICCDesign: Intraclass Correlation Coefficient Analysis and Study Planning</i>
-------------------	--

Description

A comprehensive, user-friendly R package for intraclass correlation coefficient (ICC) analysis and sample size planning. Implements the full McGraw & Wong (1996) framework supporting all 10 ICC types, with an intuitive 4-question decision system that eliminates the need for users to memorize ICC type codes.

The package provides both a command-line interface for advanced users and a fully interactive Shiny web application for point-and-click analysis. It also includes automated reliability evaluation based on Koo & Li (2016) criteria, publication-ready report generation, and rigorous sample size and power calculation functions.

Key Features

- Full support for all 10 ICC types from McGraw & Wong (1996)
- Intuitive 4-question decision framework for ICC type selection
- Automated reliability rating and interpretation
- Publication-ready text, Markdown, and HTML reports
- Sample size calculation based on confidence interval lower bound or width
- Power analysis for existing study designs
- Interactive Shiny web application with data upload and visualization
- Comprehensive data validation and error handling

Main Functions

- [icc_calc](#) Top-level function for complete ICC analysis
- [icc_sample_size](#) Unified interface for sample size and power calculation
- [run_icc_app](#) Launch the interactive Shiny application
- [icc_preprocess_data](#) Data preprocessing and validation utility

Author(s)

Maintainer: Ziyu Liu <1755454769@qq.com>

Authors:

- Ziyu Liu <1755454769@qq.com>
- Ruilin Ma
- Chenge Gao
- Yundan Zhang

References

McGraw, K. O., & Wong, S. P. (1996). Forming inferences about some intraclass correlation coefficients. *Psychological Methods*, 1(1), 30-46.

Koo, T. K., & Li, M. Y. (2016). A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *Journal of Chiropractic Medicine*, 15(2), 155-163.

See Also

Useful links:

- <https://github.com/KlariZhang/ICCDesign>
- Report bugs at <https://github.com/KlariZhang/ICCDesign/issues>

icc_calc

Calculate Intraclass Correlation Coefficient (ICC)

Description

Top-level main function for complete ICC analysis. Users only need to provide raw data and answer 4 design questions. The function automatically handles data preprocessing, parameter validation, ICC type mapping, core calculation, reliability evaluation, and report generation.

Usage

```
icc_calc(
  data,
  same_raters,
  rater_effect = NULL,
  rating_type,
  agreement_type = NULL,
  alpha = 0.05,
  rho0 = NULL,
  interaction = TRUE,
  na.rm = TRUE,
  verbose = TRUE
)
```

Arguments

<code>data</code>	Data frame or matrix. Raw data where rows = subjects, columns = raters/measurements.
<code>same_raters</code>	Logical. Are all subjects measured by the same group of raters?
<code>rater_effect</code>	Character. "random" or "fixed". Ignored if <code>same_raters = FALSE</code> .
<code>rating_type</code>	Character. "single" (single rating) or "average" (average of k ratings).
<code>agreement_type</code>	Character. "absolute" (absolute agreement) or "consistency" (consistency). Ignored if <code>same_raters = FALSE</code> .
<code>alpha</code>	Numeric. Significance level for confidence interval, default 0.05.
<code>rho0</code>	Numeric. Optional null hypothesis value for non-zero test, default NULL.
<code>interaction</code>	Logical. Whether to include interaction term in two-way models, default TRUE.
<code>na.rm</code>	Logical. Whether to automatically remove rows with missing values, default TRUE.
<code>verbose</code>	Logical. Whether to emit warnings and tips, default TRUE.

Value

A named list containing:

data_summary List. Data preprocessing summary.

icc_result List. Full ICC calculation results.

evaluation List. Reliability evaluation results.

report Character. Standardized report text.

warning_msg Character or NULL. Warning message.

tip_msg Character or NULL. Tip message.

References

McGraw, K. O., & Wong, S. P. (1996). Forming inferences about some intraclass correlation coefficients. *Psychological Methods*, 1(1), 30-46.

Koo, T. K., & Li, M. Y. (2016). A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *Journal of Chiropractic Medicine*, 15(2), 155-163.

Examples

```

# Example 0: Use built-in example dataset
data(icc_data)
result <- icc_calc(icc_data, same_raters = TRUE, rater_effect = "random",
                  rating_type = "single", agreement_type = "absolute")
# Example 1: One-way random effects, single rating (ICC(1,1))
data <- matrix(rnorm(100), nrow = 20, ncol = 5)
result <- icc_calc(data, same_raters = FALSE, rating_type = "single")

# Example 2: Two-way random effects, average rating, absolute agreement (ICC(2,k))
result <- icc_calc(data, same_raters = TRUE, rater_effect = "random",
                  rating_type = "average", agreement_type = "absolute")

# Example 3: Two-way mixed effects, single rating, consistency (ICC(3,1))
result <- icc_calc(data, same_raters = TRUE, rater_effect = "fixed",
                  rating_type = "single", agreement_type = "consistency")

# Example 4: Special scenario - automatic mapping with tip
# Random effects + consistency (automatically mapped to ICC(3,1))
result <- icc_calc(data, same_raters = TRUE, rater_effect = "random",
                  rating_type = "single", agreement_type = "consistency")

# Example 5: Special scenario - not recommended combination with warning
# Fixed effects + absolute agreement (NOT RECOMMENDED)
result <- icc_calc(data, same_raters = TRUE, rater_effect = "fixed",
                  rating_type = "average", agreement_type = "absolute")

# Example 6: Advanced parameters - custom alpha and non-zero test
result <- icc_calc(data, same_raters = TRUE, rater_effect = "random",
                  rating_type = "single", agreement_type = "absolute",
                  alpha = 0.1, rho0 = 0.6, verbose = FALSE)

# Example 7: Extract specific results
result$icc_result$point_est # ICC point estimate
result$evaluation$rating_en # Reliability rating
result$report               # Full text report

```

 icc_data

Example ICC Dataset

Description

A small, carefully constructed example dataset containing ratings from 4 raters on 5 subjects. Designed to demonstrate all package functionality with fast execution and predictable results.

Format

A numeric matrix with 5 rows (subjects) and 4 columns (raters). Row names are "Subject1" to "Subject5", column names are "Rater1" to "Rater4".

Details

This dataset can be used to calculate all 10 ICC types supported by the package, simply by changing the design parameters in `icc_calc`.

The dataset was simulated to have an approximate ICC of 0.8, which falls into the "Good" reliability category according to Koo & Li (2016).

Source

Simulated data for demonstration purposes.

Examples

```
data(icc_data)
head(icc_data)
```

icc_power

Power Calculation for ICC Study Design

Description

Power Calculation for ICC Study Design

Usage

```
icc_power(
  n,
  rho,
  rho0 = NULL,
  omega = NULL,
  k = 3,
  same_raters,
  rater_effect = NULL,
  rating_type,
  agreement_type = NULL,
  alpha = 0.05,
  method = c("lower", "width"),
  verbose = TRUE
)
```

Arguments

n	Number of subjects.
rho	Anticipated ICC value.
rho0	Lower bound for method="lower".
omega	Half-width for method="width".
k	Number of observations. Default 3.

same_raters	Logical.
rater_effect	"random" or "fixed".
rating_type	"single" or "average".
agreement_type	"absolute" or "consistency".
alpha	Significance level. Default 0.05.
method	"lower" or "width".
verbose	Print messages. Default TRUE.

Value

Power.

Examples

```
# Note: It is recommended to use the unified interface icc_sample_size()
icc_power(n = 30, rho = 0.7, rho0 = 0.5, k = 3,
          same_raters = TRUE, rater_effect = "fixed",
          rating_type = "single", agreement_type = "consistency")
```

icc_preprocess_data *Preprocess and Validate Data for ICC Analysis*

Description

Performs standardized data cleaning, format conversion, and legality validation for raw input data. Provides a unified valid data input for all ICC calculation functions to avoid repetitive validation code.

Usage

```
icc_preprocess_data(data, na.rm = TRUE)
```

Arguments

data	A data frame or matrix. Rows represent subjects, columns represent raters/ repeated measurements. Must contain only numeric values.
na.rm	Logical. Default is TRUE. If TRUE, remove rows with missing values; if FALSE, retain missing values and return a warning.

Value

A named list containing:

- data_matrix** Numeric matrix. Standardized numeric matrix (no missing values if `na.rm = TRUE`).
- n** Integer. Number of valid subjects (rows).
- k** Integer. Number of raters/repeated measurements (columns).
- warning_msg** Character or NULL. Warning message for missing values, NULL if no missing values.
- error_msg** Character or NULL. Error message for invalid data, NULL if data is valid.

Examples

```
# Preprocess the built-in example dataset
data(icc_data)
processed <- icc_preprocess_data(icc_data)
str(processed)
```

icc_sample_size *Unified ICC Sample Size & Power Interface*

Description

Unified ICC Sample Size & Power Interface

Usage

```
icc_sample_size(method = c("lower", "width", "power"), ...)
```

Arguments

method "lower", "width", "power".
... Arguments passed to underlying functions.

Value

Sample size or power.

Examples

```
# Method 1: Sample size based on lower confidence limit (most recommended)
# Ensure 95% CI lower bound >= 0.75 (good reliability)
n1 <- icc_sample_size(
  method = "lower",
  rho = 0.85,
  rating_target = "good",
  k = 3,
  same_raters = TRUE,
  rater_effect = "random",
  rating_type = "single",
  agreement_type = "absolute"
)

# Method 2: Sample size based on confidence interval width
# Ensure 95% CI half-width <= 0.1
n2 <- icc_sample_size(
  method = "width",
  rho = 0.7,
  omega = 0.1,
  k = 3,
  same_raters = FALSE,
```

```

    rating_type = "average"
  )

# Method 3: Power calculation for existing study design
power <- icc_sample_size(
  method = "power",
  n = 30,
  rho = 0.7,
  rho0 = 0.5,
  k = 3,
  same_raters = TRUE,
  rater_effect = "fixed",
  rating_type = "single",
  agreement_type = "consistency"
)

```

icc_sample_size_lower *Sample Size for ICC based on Lower Confidence Limit*

Description

Sample Size for ICC based on Lower Confidence Limit

Usage

```

icc_sample_size_lower(
  rho,
  rho0 = NULL,
  k = 3,
  same_raters,
  rater_effect = NULL,
  rating_type,
  agreement_type = NULL,
  alpha = 0.05,
  assurance = 0.8,
  rating_target = NULL,
  verbose = TRUE
)

```

Arguments

rho	Anticipated ICC value.
rho0	Desired lower bound.
k	Number of observations per subject. Default 3.
same_raters	Logical.
rater_effect	"random" or "fixed".
rating_type	"single" or "average".

agreement_type "absolute" or "consistency".
 alpha Significance level. Default 0.05.
 assurance Assurance probability. Default 0.8.
 rating_target Shortcut for rho0.
 verbose Print messages. Default TRUE.

Value

Required sample size.

Examples

```
# Note: It is recommended to use the unified interface icc_sample_size()
icc_sample_size_lower(rho = 0.8, rho0 = 0.6, k = 3, same_raters = FALSE, rating_type = "single")
```

icc_sample_size_width *Sample Size for ICC based on Confidence Interval Width*

Description

Sample Size for ICC based on Confidence Interval Width

Usage

```
icc_sample_size_width(
  rho,
  omega,
  k = 3,
  same_raters,
  rater_effect = NULL,
  rating_type,
  agreement_type = NULL,
  alpha = 0.05,
  assurance = 0.8,
  verbose = TRUE
)
```

Arguments

rho Anticipated ICC value.
 omega Desired half-width.
 k Number of observations. Default 3.
 same_raters Logical.
 rater_effect "random" or "fixed".
 rating_type "single" or "average".

agreement_type "absolute" or "consistency".
 alpha Significance level. Default 0.05.
 assurance Assurance probability. Default 0.8.
 verbose Print messages. Default TRUE.

Value

Required sample size.

Examples

```
# Note: It is recommended to use the unified interface icc_sample_size()
icc_sample_size_width(rho = 0.7, omega = 0.1, k = 3, same_raters = FALSE, rating_type = "average")
```

run_icc_app	<i>Launch the ICCDesign Shiny Application</i>
-------------	---

Description

Starts an interactive Shiny application for ICC analysis, reliability reporting, and sample size or power planning.

Usage

```
run_icc_app(
  host = "127.0.0.1",
  port = NULL,
  launch.browser = interactive(),
  ...
)
```

Arguments

host Host address passed to shiny::runApp. Default is "127.0.0.1".
 port Optional port passed to shiny::runApp. If NULL, Shiny selects an available port.
 launch.browser Logical. Whether to open the app in a browser. Default is interactive().
 ... Additional arguments passed to shiny::runApp.

Value

Runs the Shiny application.

Examples

```
if (interactive()) {
  # Launch the interactive Shiny application
  run_icc_app()
}
```

Index

* utilities

- icc_preprocess_data, 7

- icc_calc, 3, 3, 6
- icc_data, 5
- icc_power, 6
- icc_preprocess_data, 3, 7
- icc_sample_size, 3, 8
- icc_sample_size_lower, 9
- icc_sample_size_width, 10
- ICCDesign (ICCDesign-package), 2
- ICCDesign-package, 2

- run_icc_app, 3, 11