Package 'DACF'

July 16, 2026

Title Data Analysis with Ceiling and/or Floor Data

Version 1.1.0

Description An implementation of data analytic methods in R for analyses for data with ceiling/floor effects. The package currently includes functions for mean/variance estimation and mean comparison tests. Implemented methods are from Aitkin (1964) <doi:10.1007/BF02289723> and Liu & Wang (2021) <doi:10.3758/s13428-020-01407-2>.

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Imports mvtnorm, qgraph, Matrix, psych, utils

Suggests knitr, rmarkdown, MASS, ggplot2

VignetteBuilder knitr

NeedsCompilation no

Maintainer Qimin Liu <Qliuacademia@gmail.com>

Repository CRAN

Depends R (>= 3.5)

Author Qimin Liu [aut, cre],
Lijuan Wang [aut],
Lauren Trichtinger [aut],
Grace Murray [ctb]

Date/Publication 2026-07-16 21:00:45 UTC

Contents

`cf_correction` *cf_correction*

Description

Extends Tukey's HSD and Bonferroni procedure to account for ceiling and floor effects

Usage

```
cf_correction(
  x,
  tests = "all",
  df.adjustment = "trunc",
  gh.correction = "no_gh",
  alpha = 0.05,
  flr,
  ceil
)
```

Arguments

`x`	a dataframe of data with ceiling/floor effects and corresponding group variables or aov results
`tests`	a character string specifying the desired multiple-comparison procedure: "all" (default, Tukey's HSD and Bonferroni), "tukey" (Tukey's HSD), or "bonf" (Bonferroni)
`df.adjustment`	a character string specifying the desired method for adjusting the degree of freedom: "trunc" (default, Liu and Wang's truncated-normal corrections) or "unadj" (unadjusted)
`gh.correction`	a character string specifying if the Welch & Games-Howell correction for heteroscedasticity should be included: "no_gh"(default) or "yes_gh"
`alpha`	a (non-empty) numeric value of desired of alpha level
`flr`	a (non-empty) numeric value of the floor threshold (e.g., the minimum score of the measurement scale)
`ceil`	a (non-empty) numeric value of the ceiling threshold (e.g., the maximum score of the measurement scale)

Value

a matrix containing pairwise comparison results. Columns depend on the `tests` and `gh.correction` arguments:

- `Comparison_i`: First group in the comparison
- `Comparison_j`: Second group in the comparison
- `mean_i`: The adjusted mean of the first group
- `mean_j`: The adjusted mean of the second group
- `diff_in_means`: The difference in adjusted means
- `tukey.CI_lwr`: Lower bound of Tukey's HSD confidence interval (tests = "tukey" or "all")
- `tukey.CI_upr`: Upper bound of Tukey's HSD confidence interval (tests = "tukey" or "all")
- `hedges_g`: Hedges' g effect size (tests = "tukey" or "all")
- `Q`: Adjusted Tukey's Q statistic (tests = "tukey" or "all")
- `p`: p-value for Tukey's Q statistic (tests = "tukey" or "all")
- `t`: t-test statistic (tests = "bonf" or "all")
- `t_p`: p-value for the t-test statistic (tests = "bonf" or "all")
- `p.bonferroni`: Bonferroni-adjusted p-value (tests = "bonf" or "all")
- `bonf.CI_lwr`: Lower bound of Bonferroni confidence interval (tests = "bonf" or "all")
- `bonf.CI_upr`: Upper bound of Bonferroni confidence interval (tests = "bonf" or "all")
- `gh.CI_lwr`: Lower bound of Games-Howell confidence interval (gh.correction = "yes_gh", tests = "tukey" or "all")
- `gh.CI_upr`: Upper bound of Games-Howell confidence interval (gh.correction = "yes_gh", tests = "tukey" or "all")
- `games.howell.t`: Games-Howell t statistic (gh.correction = "yes_gh", tests = "tukey" or "all")
- `games.howell.p`: Games-Howell p-value (gh.correction = "yes_gh", tests = "tukey" or "all")
- `games.howell.g`: Games-Howell Hedges' g (gh.correction = "yes_gh", tests = "tukey" or "all")
- `welch.satterthwaite.df`: Welch-Satterthwaite degrees of freedom (gh.correction = "yes_gh")
- `welch.bonf.CI_lwr`: Lower bound of Welch-Bonferroni confidence interval (gh.correction = "yes_gh", tests = "bonf" or "all")
- `welch.bonf.CI_upr`: Upper bound of Welch-Bonferroni confidence interval (gh.correction = "yes_gh", tests = "bonf" or "all")
- `t_Welch`: Welch t statistic (gh.correction = "yes_gh", tests = "bonf" or "all")
- `p_Welch`: Welch p-value (gh.correction = "yes_gh", tests = "bonf" or "all")
- `p_Welch.bonferroni`: Welch Bonferroni-adjusted p-value (gh.correction = "yes_gh", tests = "bonf" or "all")

References

Aitkin MA. Correlation in a Singly Truncated Bivariate Normal Distribution. *Psychometrika*. 1964;29(3):263-270. doi:10.1007/BF02289723

Cohen, A. C. (1959). Simplified Estimators for the Normal Distribution When Samples Are Singly Censored or Truncated. *AnnalsTechnometrics*, 1(3), 217–237. doi:10.1080/00401706.1959.10489859

Greene, W. H. (2002). Econometric Analysis. *In Econometric Analysis*.

Liu, Q., Wang, L. (2020) t-Test and ANOVA for data with ceiling and/or floor effects. *Behav Res* 53, 264–277 . doi:10.3758/s13428020014072

Qi, H., Dong, Z., Wei, Q., Chen, X., & Luo, Y. (2025). Are Gamers Happier? Multidimensional Well-Being Differences in Risk Groups for Problematic Internet Use and Internet Gaming Disorder. *Personality and Individual Differences* 246, 113334. doi:10.2139/ssrn.5148047

Examples

```
data("Qi_data_males") # (Qi et al., 2025)
cf_correction(Qi_data_males, tests = "all", df.adjustment = "trunc",
              gh.correction = "yes_gh", alpha = .05, flr = 0, ceil = 9)
cf_correction(Qi_data_males, tests = "tukey", df.adjustment = "unadj",
              gh.correction = "no_gh", alpha = .05, flr = 0, ceil = 9)
cf_correction(Qi_data_males, tests = "all", df.adjustment = "unadj",
              gh.correction = "no_gh", alpha = .05, flr = 0, ceil = 9)
```

`f.star.test` *f.star.test*

Description

conduct a Brown-Forsythe F star test

Usage

```
f.star.test(means, variances, ns)
```

Arguments

`means`	a (non-empty) numeric vector of the group means
`variances`	a (non-empty) numeric vector of the group variances
`ns`	a (non-empty) numeric vector of sample sizes per group

Value

`statistic`	the value of the adjusted Brown-Forsythe F star statistic
`p.value`	the p-value for the test
`est.f.squared`	effect size estimate as in Cohen's f squared

Examples

```
# a f star test for three-group mean comparison
f.star.test(c(-.2,0,.2),c(1,1,1),c(100,100,100))
f.star.test(c(0,0,1),c(2,1,3),c(100,100,100))
```

---

ggm_cfe *Gaussian Graphical Model with Ceiling/Floor Effect Correction*

Description

Estimates a psychological network (GGM) from data with ceiling and/or floor effects using a two-step pairwise censored-normal correction. Step 1 corrects item means and variances via truncated-normal moment matching (Liu & Wang, 2021). Step 2 estimates pairwise latent correlations by solving the moment equation E[cov*(rho)] = observed censored covariance, evaluated using 50-point Gauss-Hermite quadrature. The corrected covariance matrix is supplied as input to EBIC-glasso or significance-based edge selection, preserving the standard qgraph/bootnet workflow.

Usage

```
ggm_cfe(
  data,
  floor = NULL,
  ceiling = NULL,
  method = "EBICglasso",
  gamma = 0.5,
  nlambda = 100,
  lambda.min.ratio = 0.01,
  threshold = FALSE,
  alpha = 0.05,
  verbose = TRUE
)
```

Arguments

`data`	A data frame or numeric matrix (n x p). Rows = observations, columns = variables (items/nodes).
`floor`	Floor threshold(s). Either a single value applied to all variables, or a named numeric vector with one value per column. Use NULL to indicate no floor effect. Default: NULL.
`ceiling`	Ceiling threshold(s). Same format as `floor`. Default: NULL.
`method`	Estimation method(s): "EBICglasso" (default), "FDR", "Bonferroni", or "all" to run all three.
`gamma`	EBIC hyperparameter (0 = BIC, 0.5 = default). Only used when method includes "EBICglasso".
`nlambda`	Number of tuning parameters searched by EBICglasso. Default: 100.

`lambda.min.ratio`
: Smallest lambda searched, as a fraction of the largest lambda. Passed to `qgraph::EBICglasso`. Default: 0.01.

`threshold`
: Logical. Passed to `qgraph::EBICglasso`; `TRUE` enforces higher specificity at the cost of sensitivity. Default: FALSE.

`alpha`
: Significance level for p-value methods. Default: 0.05.

`verbose`
: Logical. Print progress and diagnostics. Default: TRUE.

Value

An object of class `"ggm_cfe"` containing:

`network`
: Partial correlation matrix from the corrected method (primary estimator: EBICglasso or first specified method).

`network_corrected`
: Partial correlation matrix from naive EBICglasso (ignoring ceiling/floor effects).

`network_naive`
: Partial correlation matrix estimated from the raw (uncorrected) sample covariance using the primary method.

`networks`
: Named list of partial correlation matrices for all requested methods.

`networks_corrected`
: Partial correlation matrix from naive EBICglasso (ignoring ceiling/floor effects).

`networks_naive`
: Named list of naive partial correlation matrices (one per requested method), estimated without ceiling/floor correction.

`Sigma_corrected`
: The corrected covariance matrix (p x p).

`Sigma_naive`
: The raw sample covariance matrix (p x p).

`diagnostics`
: List with per-variable censoring statistics, nearPD correction magnitude, and floor/ceiling thresholds.

  censoring Data frame (p rows) with per-variable censoring statistics: variable, `n_floor`, `n_ceil`, `pct_floor`, `pct_ceil`, naive and corrected means (`mu_naive`, `mu_corrected`), and standard deviations (`sd_naive`, `sd_corrected`).

  npd_correction Relative Frobenius-norm magnitude of the nearest positive-definite projection applied to the corrected covariance matrix. Values above 0.05 suggest instability and should be interpreted cautiously.

  a_vec Length-p numeric vector of effective floor thresholds used internally (with `NULL` inputs replaced by data-driven lower bounds).

  b_vec Length-p numeric vector of effective ceiling thresholds (with `NULL` inputs replaced by data-driven upper bounds).

`method`
: Character vector of estimation method(s) run.

`gamma`
: EBIC hyperparameter value used.

`nlambda`
: Number of regularization parameters searched.

`lambda.min.ratio`
: Smallest lambda as a fraction of the largest.

`threshold`
: Logical; whether EBICglasso hard-thresholding was applied.

alpha	Significance level for p-value methods.
n	The number of observations.
p	The number of variables.
varnames	The character vector of variable (column) names.
data	Original data matrix.
call	Matched call.

Examples

```
# Simulate data with ceiling effects
set.seed(42)
Y <- MASS::mvrnorm(200, rep(0,5), diag(5) + 0.3)
Y_cens <- pmin(Y, 1.0)   # ceiling at 1.0 SD
fit <- ggm_cfe(Y_cens, floor=NULL, ceiling=1.0)
print(fit)
plot(fit)

# With psych::bfi personality data
# library(psych)
# fit <- ggm_cfe(bfi[,1:25], floor=1, ceiling=6)
```

`induce.cfe` *induce.cfe*

Description

inducing ceiling/floor effects in data

Usage

```
induce.cfe(floor.perc, ceiling.perc, y)
```

Arguments

floor.perc	a (non-empty) numeric value from 0 to 1 denoting the desired percentage of floor effects
ceiling.perc	a (non-empty) numeric value from 0 to 1 denoting the desired percentage of ceiling effects
y	a (non-empty) numeric vector of data

Value

y scores with induced ceiling/floor effects

Examples

```
x=rnorm(1000,0,1) #simulate "healthy data"
x.c20=induce.cfe(0,.2,x) #induce 20% ceiling effects into the data
sum(x.c20==max(x.c20))/length(x.c20) #check ceiling percentage
x.f20=induce.cfe(.2,0,x) #induce 20% floor effects into the data
sum(x.f20==min(x.f20))/length(x.f20) #check ceiling percentage
```

`lw.f.star` *lw.f.star*

Description

conduct an F star test for data with ceiling/floor effects

Usage

```
lw.f.star(data, formula, flr, ceil, method_type)
```

Arguments

`data`	a dataframe of data with ceiling/floor effects and corresponding group variables in wide format
`formula`	a formula denoting the dependent and independent variable, e.g., y~group
`flr`	a (non-empty) numeric value of the floor threshold (e.g., the minimum score of the measurement scale)
`ceil`	a (non-empty) numeric value of the ceiling threshold (e.g., the maximum score of the measurement scale)
`method_type`	a character string specifying the preferred method type. "a" uses the original sample size and "b" uses after-truncation sample size.

Value

`statistic`	the value of the Brown-Forsythe F star statistics
`p.value`	the p-value for the test
`est.f.squared`	effect size estimate in Cohen's f squared

Examples

```
dat=threeganova.sim(1000,.16,1)
dat[dat$group==1,3]=induce.cfe(0,.15,dat[dat$group==1,3])
lw.f.star(dat, y~group, flr=min(dat$y), ceil=max(dat$y), "a") #using truncated n
lw.f.star(dat, y~group, flr=min(dat$y), ceil=max(dat$y), "b") #using original n
```

lw.t.test *lw.t.test*

Description

conduct a t test adjusting for ceiling and/or floor effects

Usage

```
lw.t.test(x1, x2, flr1, ceil1, flr2, ceil2, method_type)
```

Arguments

x1	a (non-empty) numeric vector of data values for group 1 with floor/ceiling effects
x2	a (non-empty) numeric vector of data values for group 2 with floor/ceiling effects
flr1	a (non-empty) numeric value of the floor threshold for group 1
ceil1	a (non-empty) numeric value of the ceiling threshold for group 1
flr2	a (non-empty) numeric value of the floor threshold for group 2
ceil2	a (non-empty) numeric value of the ceiling threshold for group 2
method_type	a character string specifying the preferred method type. "a" uses the original sample size and "b" uses after-truncation sample size.

Value

statistic	the value of the adjusted t test statistics
p.value	the p-value for the test
est.d	effect size estimate as in Cohen's d
conf.int	95% confidence interval

Examples

```
x1.c=induce.cfe(0,.3,rnorm(1000,20,5)) #group 1 scores with 30% ceiling data
x2.c=induce.cfe(.15,0,rnorm(1000,30,5)) #group 2 scores with 15% floor data
lw.t.test(x1.c, x2.c, flr1=min(x1.c), ceil1=max(x1.c), flr2=min(x2.c), ceil2=max(x2.c), "a")
lw.t.test(x1.c, x2.c, flr1=min(x1.c), ceil1=max(x1.c), flr2=min(x2.c), ceil2=max(x2.c), "b")
```

plot_censoring *Censoring Profile Plot for ggm_cfe Objects*

Description

Plots the floor and ceiling proportions for each variable in a `ggm_cfe` object. Uses **ggplot2** if available, otherwise falls back to base graphics.

Usage

```
plot_censoring(x)
```

Arguments

`x`	An object of class `"ggm_cfe"` as returned by `ggm_cfe`.

Value

Invisibly returns `x`.

Examples

```
set.seed(42)
Y <- MASS::mvrnorm(200, rep(0, 5), diag(5) + 0.3)
Y_cens <- pmin(Y, 1.0)
fit <- ggm_cfe(Y_cens, floor = NULL, ceiling = 1.0)
plot_censoring(fit)
```

Qi_data_males *Test dataset for DACF package*

Description

A dataset used to demonstrate ceiling/floor effect corrections.

Usage

```
Qi_data_males
```

Format

A data frame with 1731 rows and 2 columns:

group The values of the grouping variable. The participants were divided into three groups: moderate Problematic Internet Use (PIU) and Internet Gaming Disorder (IGD) risk, high PIU risk, and high IGD risk.

value The observed data of the dependent variable, IGD severity from a scale from 0 to 9.

Source

Qi, H., Dong, Z., Wei, Q., Chen, X., & Luo, Y. (2025). Are Gamers Happier? Multidimensional Well-Being Differences in Risk Groups for Problematic Internet Use and Internet Gaming Disorder. *Personality and Individual Differences* 246, 113334. doi:10.2139/ssrn.5148047

`rec.mean.var` *rec.mean.var*

Description

recover mean and variance of the data with ceiling/floor effects

Usage

```
rec.mean.var(y, flr, ceil)
```

Arguments

`y`	a (non-empty) numeric vector of data with ceiling/floor effects
`flr`	a (non-empty) numeric value of the floor threshold (e.g., the minimum score of the measurement scale)
`ceil`	a (non-empty) numeric value of the ceiling threshold (e.g., the maximum score of the measurement scale)

Value

`ceiling.percentage`	the percentage of ceiling values in the data
`floor.percentage`	the percentage of floor values in the data
`est.mean`	estimated mean of the true scores
`est.var`	estimated variance of the true scores

Examples

```
# simulate normally distributed true scores
x=rnorm(1000,2,4)
mean(x); var(x)
# induce 20% floor effects
# and estimate the true mean variance from the floor data
x.f=induce.cfe(.2,0,x)
rec.mean.var(x.f, flr=min(x.f), ceil=max(x.f))
# induce 20% ceiling effects
# and estimate the true mean and variance from the ceiling data
x.c=induce.cfe(0,.2,x)
rec.mean.var(x.c, flr=min(x.c), ceil=max(x.c))
# induce 20% and 10% of floor and ceiling effects, respectively
```

```
# and estimate the true mean and variance from the data with floor and ceiling effects
x.cf=induce.cfe(.2,.1,x)
rec.mean.var(x.cf, flr=min(x.cf), ceil=max(x.cf))
```

---

threeganova.sim *threeganova.sim*

Description

simulate three-group anova data

Usage

```
threeganova.sim(group_n, f_sqr, sd.1)
```

Arguments

group_n	a (non-empty) numeric value of desired sample size per group
f_sqr	a (non-empty) numeric value of desired Cohen's f squared value
sd.1	a (non-empty) numeric value of desired standard deviation ratio

Value

a dataframe containing scores "y", grouping factor "group", and residual errors.

Examples

```
sample.3g=threeganova.sim(1000,.16,5) #data of n=1000, sd1=sd3=1 and sd2=5, and f^2=.16
colnames(sample.3g) #examine the column names
dim(sample.3g) #examine the data structure
aggregate(sample.3g$y,sd,by=list(sample.3g$group)) #check group standard deviations
```

Index