Package 'growthcleanr'

November 8, 2026

Title Data Cleaner for Anthropometric Measurements

Version 3.0.0

Description Flags implausible anthropometric (weight, height, and head circumference) measurements in longitudinal pediatric and adult datasets, such as those from electronic health records. Identifies other measurements that should often be excluded from analyses. The main function is cleangrowth(). Implements a modified version of the algorithm described in Daymont et al. (2017) <doi:10.1093/jamia/ocx037>.

URL https://growthcleanr.org/,
https://github.com/growth-data-tools/growthcleanr

BugReports https://github.com/growth-data-tools/growthcleanr/issues

Imports R.utils (>= 2.11.0), data.table (>= 1.13.0), foreach (>= 1.5.0), doParallel (>= 1.0.15), lubridate (>= 1.7.0), parallel, stats, utils, shiny, DT, shinyjs, shinyFiles

Depends R (>= 3.5.0)

License MIT + file LICENSE

Encoding UTF-8

Language en-US

LazyData true

Suggests argparser (>= 0.6), knitr (>= 1.29), rmarkdown (>= 2.3), testthat (>= 2.3.2)

VignetteBuilder knitr

RoxygenNote 7.3.3

NeedsCompilation no

Author Carrie Daymont [cre, aut, cph],
Robert W Grundmeier [aut],
Jeffrey Miller [aut],
Diego Campos [aut],
Dan Chudnov [aut],
Hannah De los Santos [aut],
David Freedman [aut],

Wen Hu [aut],
Chris Penney [aut],
Jesse Anderson [ctb],
Matt Boyas [ctb],
Lusha Cao [ctb],
Sara Couture [ctb],
Prashila Dullabh [ctb],
Jess Fernandez [ctb],
Andrew Gregorowicz [ctb],
Lyudmila Kompaniyets [ctb],
A Russell Localio [ctb],
Ruth Nwefo [ctb],
Max Olivier [ctb],
Samantha Pierce [ctb],
Michelle E Ross [ctb],
Steffani Silva [ctb],
Mithuna Srinivasan [ctb],
Marissa Sucosky [ctb],
Jennifer Wiltz [ctb],
Hanzhe Zhang [ctb],
Alexander G Fiks [sad],
Richard C (Mort) Wasserman [sad],
Andreas Achilleos [sad],
Nabeel Akhtar [sad],
Sarah Akhtar [sad],
Wilbert Beachy [sad],
Colin Bridgeman [sad],
Jessica Butts [sad],
Emily Despeaux [sad],
Christine Irvin [sad],
Madiha Jamil [sad],
Robert Marquis [sad],
Abigail Nelson [sad],
Sheila Nguyen [sad],
Taraneh Soleymani [sad],
Ronald Williams [sad]

Maintainer Carrie Daymont <cdaymont@pennstatehealth.psu.edu>

Repository CRAN

Date/Publication 2026-11-08 21:20:10 UTC

Contents

growthcleanr-package 3
CDCref_d 5
child_z3_reference 6
cleangrowth 6
cleangrowth_checkpoint 22

cleangrowth_checkpoint_combine . . . 23
ewma . . . 25
ext_bmiz . . . 26
fenton2025_ms_lookup_smoothed . . . 28
fent_foraga . . . 29
gc_cli_path . . . 29
gc_partial_run . . . 30
gc_preload_refs . . . 31
gc_runtrial . . . 32
growthcleanr_app . . . 34
growth_cdc_ext_infants . . . 35
growth_who_ext . . . 35
longwide . . . 36
permissiveness_presets . . . 38
rc-reference-medians . . . 39
read_anthro . . . 39
recode_sex . . . 40
run_gc_cli . . . 41
simple_bmi . . . 42
syngrowth . . . 43
tanner_ht_vel . . . 43
who_hc_maxvel . . . 44
who_hc_vel_3sd . . . 44
who_ht_maxvel . . . 45
who_ht_vel_3sd . . . 45

Index **46**

`growthcleanr-package` *growthcleanr: a data cleaner for anthropometric measurements*

Description

growthcleanr flags implausible anthropometric (height, weight, and head circumference) measurements in longitudinal datasets, such as those from electronic health records. It also identifies other measurements that should often be excluded from analyses.

Getting started

The main function is `cleangrowth`. Pass it a data frame with one row per measurement and columns `subjid`, `param`, `agedays`, `sex`, and `measurement` (plus an optional per-row `id`). Semi-wide and wide layouts, alternate column names, and date-of-birth / visit-date ages are also supported. growthcleanr will not delete or edit data: it returns your data with an added `exclude` column labeling each row `"Include"` or an exclusion code naming why it recommends exclusion.

The function `growthcleanr_app` opens a Shiny app that runs growthcleanr without writing code. A command-line driver is documented in `vignette("command-line")`.

New to growthcleanr? Start with `vignette("new-start-here")`. See `vignette("output")` for more information including the returned columns and `vignette("configuration")` for the full index of arguments and thresholds.

Author(s)

Maintainer: Carrie Daymont <cdaymont@pennstatehealth.psu.edu> [copyright holder]

Authors:

- Robert W Grundmeier
- Jeffrey Miller
- Diego Campos
- Dan Chudnov
- Hannah De los Santos
- David Freedman
- Wen Hu
- Chris Penney

Other contributors:

- Jesse Anderson [contributor]
- Matt Boyas [contributor]
- Lusha Cao [contributor]
- Sara Couture [contributor]
- Prashila Dullabh [contributor]
- Jess Fernandez [contributor]
- Andrew Gregorowicz [contributor]
- Lyudmila Kompaniyets [contributor]
- A Russell Localio [contributor]
- Ruth Nwefo [contributor]
- Max Olivier [contributor]
- Samantha Pierce [contributor]
- Michelle E Ross [contributor]
- Steffani Silva [contributor]
- Mithuna Srinivasan [contributor]
- Marissa Sucosky [contributor]
- Jennifer Wiltz [contributor]
- Hanzhe Zhang [contributor]
- Alexander G Fiks [scientific advisor]
- Richard C (Mort) Wasserman [scientific advisor]
- Andreas Achilleos [scientific advisor]

- Nabeel Akhtar [scientific advisor]
- Sarah Akhtar [scientific advisor]
- Wilbert Beachy [scientific advisor]
- Colin Bridgeman [scientific advisor]
- Jessica Butts [scientific advisor]
- Emily Despeaux [scientific advisor]
- Christine Irvin [scientific advisor]
- Madiha Jamil [scientific advisor]
- Robert Marquis [scientific advisor]
- Abigail Nelson [scientific advisor]
- Sheila Nguyen [scientific advisor]
- Taraneh Soleymani [scientific advisor]
- Ronald Williams [scientific advisor]

See Also

Useful links:

- https://growthcleanr.org/
- https://github.com/growth-data-tools/growthcleanr
- Report bugs at https://github.com/growth-data-tools/growthcleanr/issues

---

`CDCref_d` *CDC BMI reference data*

Description

Used for extended BMIz computation

CDCref_d.csv.gz

Used for extended BMI computation

child_z3_reference *Child unit-error-range reference values*

Description

Part of default reference tables

Details

Contains, for each age in days, sex, and param, the measurement values at recentered CSD z-scores of -3 (column zn3) and +3 (column zp3), computed with the same WHO/CDC blending and default recentering used by the child algorithm. Covers weight and height from birth to 20 years and head circumference from birth to 5 years (1826 days). Used to compute the informational `unit_error_range` output column and the unit-error check in Evil Twins (Child Step 9). See `vignette("algorithm-functions")`.

child_z3_reference.csv.gz

Used in function `cleangrowth()` (when `unit_error_range = TRUE`, the default)

cleangrowth *Clean growth measurements (main growthcleanr function)*

Description

growthcleanr flags implausible values in longitudinal anthropometric measurements – weight, height/length, and head circumference – for children and adults. growthcleanr is specifically designed to evaluate electronic health record data for outpatients. It does not delete or edit your data. It returns your data with an added `exclude` column labeling each row either `"Include"` or an exclusion code naming why it was set aside. Each subject is cleaned independently, using that subject's own measurement trajectory over time, so splitting, subsetting, or batching the data does not change any subject's result.

The default input is one row per measurement in **long** format with columns `subjid`, `param`, `agedays`, `sex`, and `measurement` (plus an optional per-row `id`); semi-wide and wide layouts are also accepted (see `input_format`). You can use date of birth and visit date instead of `agedays`. Most runs need only the input-column arguments; see "Arguments at a glance" under Details for how the rest are grouped.

New to growthcleanr? Start with `vignette("new-start-here")`. To check that growthcleanr is reading your columns, sex coding, and parameters correctly before a full run, use `gc_runtrial` on a sample of your data. See `vignette("output")` for the returned columns and `vignette("configuration")` for the full argument index. For ready-to-run scripts covering common setups, see `vignette("example-scripts")`.

Usage

```
cleangrowth(
  data = NULL,
  subjid = NULL,
  param = NULL,
  agedays = NULL,
  sex = NULL,
  measurement = NULL,
  piv.z.wt.high = 22,
  piv.z.ht.low.young = -25,
  piv.z.ht.low.old = -15,
  piv.z.ht.high = 8,
  piv.z.hc.low = -15,
  piv.z.hc.high = 15,
  error.load.mincount = 2,
  error.load.threshold = 0.5,
  sd.recenter = NA,
  cf_rescue = "standard",
  cf_detail = FALSE,
  corr_detail = FALSE,
  recenter_detail = FALSE,
  qm_detail = FALSE,
  ref.data.path = "",
  log.path = NA,
  parallel = FALSE,
  cores = NULL,
  quietly = FALSE,
  verbose = FALSE,
  adult_cutpoint = 20,
  adult_permissiveness = "looser",
  adult_scale_max_lbs = Inf,
  ewma_window = 15,
  adult_overrides = NULL,
  id = NULL,
  dob = NULL,
  visit_date = NULL,
  age_source = NULL,
  date_order = NULL,
  sex_coding = .gc_default_sex_coding,
  param_coding = NULL,
  carry_through = TRUE,
  input_format = "long",
  output_format = NULL,
  semiwide_cols = NULL,
  widened_emit = "exclude",
  wide_cols = NULL,
  max_blocks = 25,
  internal_detail = FALSE,
```

```
  ref_tables = NULL,
  tri_exclude = FALSE,
  debug = FALSE,
  full_detail = FALSE,
  unit_error_range = TRUE,
  display_gc_id = FALSE,
  exclude_detail = FALSE,
  mean_ht = TRUE,
  batch_size = 5000,
  length.adjust = FALSE,
  checkpoint = TRUE,
  checkpoint_dir = NULL,
  checkpoint_size = 5000,
  checkpoint_resume = FALSE,
  keep_checkpoints = TRUE,
  log_file = NULL,
  .checkpoint_driven = FALSE,
  ...
)
```

Arguments

data	Optional data.frame or data.table containing the input measurements. When supplied, subjid, param, agedays, sex, measurement, and id are interpreted as column-name strings (defaulting to the canonical names). When NULL (the default), the function uses the legacy vector-style API.
subjid	Subject identifier. In data-frame style, a column name (default "subjid"). In vector style, a vector of subject identifiers, one entry per measurement row (repeated across a subject's rows).
param	Measurement parameter. In data-frame style, a column name (default "param"); in vector style, a vector. Recognized values are "WEIGHTKG", "WEIGHTLBS", "HEIGHTCM", "HEIGHTIN", "LENGTHCM", "LENGTHIN", "HEADCM", and "HEADIN". Matching is case-insensitive and whitespace-trimmed, but the names must be spelled out: there are deliberately no built-in abbreviations, so a short label such as "wt", "ht", or "hc" is not assumed to be a growthcleanr parameter because it might be a statistical sample weight or something similar. Declare any non-canonical label with param_coding. Unrecognized params are set aside as Exclude-Not-GC-Param (with a warning); the value is preserved. Units and length/height handling: imperial parameters ("WEIGHTLBS", "HEIGHTIN", "LENGTHIN", "HEADIN") are converted to metric during calculation. By default, linear measurements under 2 years are evaluated as length and at 2 years (day 730) and older as height, regardless of whether they are input using a LENGTH or HEIGHT parameter; in children whose data are consistent with prematurity or small-for-gestational age (SGA) and who receive an assigned gestational age for cleaning purposes, that gestational age is used to determine an estimated corrected age, and this corrected age is used to determine default linear growth handling. This default behavior can be changed with length.adjust = TRUE. Full detail is in vignette("wrapper"), section "Length vs. height handling".

agedays — Age in days at each measurement. In data-frame style, a column name (default "agedays"). In vector style, a numeric vector. May be omitted if age is supplied instead as dob + visit_date (see age_source); when an agedays column is present it takes precedence over dates by default. Cleaning uses whole days: a fractional age is rounded down (1435.4 is day 1435; -0.5 is day -1) for all cleaning, but the submitted value is returned in the output agedays column (the whole-day value is available as gc_internal_agedays under internal_detail).

sex — Subject sex. In data-frame style, a column name (default "sex"). In vector style, a vector. By default 0, "m", "b", "male", or "boy" are read as male and 1, "f", "g", "female", or "girl" as female (case-insensitive, whitespace-trimmed); override with sex_coding. On pediatric rows, values matching neither list (and blank/NA) are set aside as Exclude-Missing-Info (with a warning). A sex column is always required, but the sex column value may be blank or NA for adults (ages at or above adult_cutpoint): the adult algorithm does not require a valid value in the sex column, but when available, it is used in the Evil Twins step and in calculating whether a value falls in the unit error range. A 1=male/2=female coding is detected and rejected with an informative error (growthcleanr reads 1 as female), so it cannot silently misclassify data; pass sex_coding = list(male = 1, female = 2) if that is your coding. If there is more than one subject and every value is 1, which could be all-female 0/1 data or all-male 1/2 data, it is read as female with a prominent warning (the run continues); pass an explicit sex_coding or code sex as F to remove the warning.

measurement — Recorded measurement. In data-frame style, a column name (default "measurement"). In vector style, a numeric vector. Weight must be in kilograms (kg), and linear measurements in centimeters (cm), unless param is "WEIGHTLBS", "HEIGHTIN", "LENGTHIN", or "HEADIN". Text or factor values are converted to numbers for cleaning (values that are not numbers are set aside as Exclude-Missing-Info) and are returned as submitted.

piv.z.wt.high — Upper unrecentered CSD z-score cutoff for WEIGHTKG (all ages). Weights with sd.orig > piv.z.wt.high are excluded as hard limits (Exclude-Hard-Limit). Defaults to 22.

piv.z.ht.low.young — Lower unrecentered CSD z-score cutoff for HEIGHTCM under one year of age. Heights with sd.orig < piv.z.ht.low.young are excluded as hard limits (Exclude-Hard-Limit). Defaults to -25.

piv.z.ht.low.old — Lower unrecentered CSD z-score cutoff for HEIGHTCM at or above one year. Heights with sd.orig < piv.z.ht.low.old are excluded as hard limits (Exclude-Hard-Limit). Defaults to -15.

piv.z.ht.high — Upper unrecentered CSD z-score cutoff for HEIGHTCM (all ages). Heights with sd.orig > piv.z.ht.high are excluded as hard limits (Exclude-Hard-Limit). Defaults to 8 (tighter than the weight upper limit).

piv.z.hc.low — Lower unrecentered CSD z-score cutoff for HEADCM (all ages). Head-circumference values with sd.orig < piv.z.hc.low are excluded as hard limits (Exclude-Hard-Limit). Defaults to -15.

piv.z.hc.high — Upper unrecentered CSD z-score cutoff for HEADCM (all ages). Head-circumference values with sd.orig > piv.z.hc.high are excluded as hard limits (Exclude-Hard-Limit).

Defaults to 15.

`error.load.mincount`

Minimum number of real errors (exclusions other than `Exclude-Identical`, `Exclude-Extraneous`, `Exclude-CF`, `Exclude-Missing-Info`, and `Exclude-HC-Out-of-Range`) required per (subject, param) group before Child Step 21 Error Load is evaluated for that group. Defaults to 2. Unlike 2.2.3, a high error load for one parameter can no longer result in the exclusion of the other parameter's values.

`error.load.threshold`

Error ratio threshold for Child Step 21. The ratio is `n_errors / (n_errors + n_includes)` computed per (subject, param) group, where `n_errors` and `n_includes` leave out the codes listed under `error.load.mincount`. If the ratio strictly exceeds this threshold (and at least `error.load.mincount` real errors are present), all remaining Include rows for that (subject, param) group are excluded with `Exclude-Traj-Uneval`. Defaults to 0.5.

`sd.recenter` Recentering medians. One of:

`NA` **(default)** Use the built-in reference file (`rcfile-resmoothed.csv.gz`). (In 2.x, `NA` derived medians from the input data for larger datasets; that option was removed.)

`"none"` No recentering: the recentered scores equal the reference scores (`tbc.sd = sd.orig`).

A data.table or data.frame Your own medians, with columns `param`, `sex`, `agedays`, and `rcmedian.sd` (a median column named `sd.median` is also accepted). The table must have exactly one median per `param`, `sex`, and `agedays`, and a median for every `param`, `sex`, and `agedays` of the child measurements to be cleaned; otherwise growthcleanr stops and reports the problem.

The 2.x options `"nhanes"` and `"derive"` were removed and stop with an error. See `vignette("recentering")`.

`cf_rescue` CF rescue mode for the child algorithm. One of:

"standard" (Default) Decide each identical-value string on its last copy in the string; the whole string is rescued or excluded together. When the last CF has a next included value, rescue if `cf_deltaz + cf_nextz <= 1` and `cf_deltaz <= 1.5`; when it has no next value, rescue by age (`cf_deltaz <= 0.5` if `agedays < 13*365.25`, else `cf_deltaz <= 1`). `cf_deltaz` is the absolute difference between the value's z-score before recentering and the z-score of the value that was copied. See `vignette("cf-rescue")`.

"none" All potential carried-forward values are excluded. A value is detected as carried forward when it exactly equals the measurement on the subject's previous measurement day and that day has only one measurement for the parameter (see `vignette("cf-rescue")`).

"all" No potential carried-forward values are excluded at this step; they are evaluated by the later steps like any other value.

`cf_detail` Logical. If TRUE, add six carried-forward (CF) columns to the output: `cf_status` (each row's CF role, e.g. "Orig", "CF-R" (rescued), "CF-NR" (not rescued), "Not-CF", or "Adult"), `cf_deltaz` and `cf_nextz` (the z-score differences used in the rescue decision), and three design aids for CF-logic development (`imperial`, `cf_dage_orig`, `cf_dage_next`). Each column and value is defined in `vignette("output")`, section 4, and the rescue rule in `vignette("cf-rescue")`. Defaults to FALSE.

corr_detail — Logical. If TRUE, include corr_val and corr_pt columns in the output. corr_val labels each value "Corrected" (its gestational-age-corrected z-score differs from the uncorrected one), "Uncorrected", or "Adult" (adult row). corr_pt labels each subject "Corrected" (the subject has at least one corrected value), "Uncorrected", or "Adult". Defaults to FALSE.

recenter_detail — Logical. If TRUE, include the recentered CSD z-score columns in the output: tbc.sd (the recentered z used throughout the child algorithm), ctbc.sd (its gestational-age-corrected counterpart), and rcmedian.sd (the per-row recentering median that was subtracted). This is specialized output for a related tool in development (see vignette("output"), section 5b). The three columns are a subset of what debug surfaces. Child rows only; adult rows receive NA. Defaults to FALSE. Also turned on by full_detail = TRUE.

qm_detail — Logical. If TRUE, emit the child-algorithm per-step eligibility flags (qm_elig_c_cf, qm_elig_c_hardlimit, qm_elig_c_eviltwins, qm_elig_c_trajextreme, qm_elig_extraneous, qm_elig_c_traj, qm_elig_c_absdiff, qm_elig_c_pair, qm_elig_c_single, qm_elig_c_trajuneval): one logical column per reported cleaning step marking the rows that were eligible for (in scope of) that step – the denominator against which that step's exclusions are measured. Child rows only; NA on adult rows. Used for calculations in a related package currently in development. Not used in cleaning; typically paired with exclude_detail = TRUE so downstream code can derive which step excluded each row. Additive, write-only (does not affect any exclusion decision), and off by default. Must be requested explicitly – it is NOT turned on by full_detail. Defaults to FALSE.

ref.data.path — Path to reference data. If not supplied, the reference files included with growthcleanr are used.

log.path — Deprecated; supplying it gives a warning. Use log_file instead. (It still writes per-batch log files in parallel, non-quiet runs.)

parallel — Determines if function runs in parallel. Defaults to FALSE. See vignette("large-datasets") and "Large datasets: run in parallel and save progress" in vignette("example-scripts").

cores — Number of parallel worker processes to use when parallel = TRUE. NULL (the default) auto-detects min(8, parallel::detectCores() - 1) – capped at 8 because growthcleanr's per-worker setup overhead makes more workers than that counterproductive. Pass a positive whole number to override; a value above the machine's core count is reduced to it (extra workers only oversubscribe, and an extreme value would stall the run spawning workers). Ignored when parallel = FALSE. (Replaces the former num.batches argument.)

quietly — Controls progress output. With the default quietly = FALSE, growthcleanr prints a minimal progress view: a "cleaning in progress" line, an X/Y batch (or checkpoint) counter, and the end-of-run cleaning summary. Set quietly = TRUE to turn off the progress lines and the summary. Warnings still appear, as does the one-time note shown when cleangrowth() is called with vectors (the 2.x style); that note is a message, so suppressMessages() will silence it. For the detailed per-step view, use verbose instead. Defaults to FALSE.

verbose — When TRUE, prints the detailed view: the milestone lines (preprocessing, per-batch cleaning child/adult) plus granular per-step diagnostics (pre-filter counts,

EWMA/SDE iteration counts, etc.). Implies `quietly = FALSE`. Intended for development and debugging. Defaults to FALSE.

`adult_cutpoint` Age in years, between 18 and 20, describing ages when the pediatric algorithm should not be applied (< adult_cutpoint), and the adult algorithm should apply (>= adult_cutpoint). Numbers outside this range are changed to the closest number within the range, with a warning. Defaults to 20. growthcleanr does not change this value based on your data (so each subject is cleaned independently of the rest of the dataset); if it is left at the default and every measurement with a known age is at age 18 years or older, a warning suggests setting `adult_cutpoint = 18` to clean those 18-to-20-year-olds with the adult algorithm instead of the pediatric one.

`adult_permissiveness`

Permissiveness level for the adult algorithm: "loosest", "looser" (default), "tighter", or "tightest" (case and surrounding spaces are ignored). Sets defaults for all adult PIV limits, EWMA caps, height bands, etc. To change one or a few of those settings without switching preset, use `adult_overrides`. For the intended use of each level and its preset values, see Section 1D of `vignette("configuration")`. For a ready-to-run script, see "Data that includes adults" in `vignette("example-scripts")`.

`adult_scale_max_lbs`

Physical scale upper limit in pounds for adult weight data (replaces 2.2.3's `weight_cap`, which was in kg). Weights within +/-0.1 kg of the cap are excluded if they are more than 50 kg different from their expected value, or if all of the subject's weights (at least two) are the same value at the cap. Defaults to Inf.

`ewma_window` Used by both the child and adult algorithms. Number of measurements on each side included in the exponentially weighted moving average (EWMA) weighting window. Defaults to 15.

`adult_overrides`

Optional named list overriding individual adult permissiveness settings (those set by `adult_permissiveness`), e.g. `list(overall_ht_min = 120, allow_ht_gain = FALSE)`. Each named value replaces the preset value for that setting; unnamed or unrecognized names are an error. Applies to adult rows only. The recognized names are the adult settings listed in `vignette("configuration")` (Section 1D). Defaults to NULL (use the preset for every setting).

`id` Row identifier. Optional but recommended. In data-frame style, a column name (default `"id"`); in vector style, a vector of row ids. If `id` is absent / `NULL` / contains `NA` / is not unique, a warning is issued and sequential integer ids (1:N in input row order) are auto-generated for internal use. The user's original `id` (whatever was supplied - even if `NA` or duplicated) is preserved in the output `id` column so callers can map results back to their data. To see the id that growthcleanr actually used internally, set `display_gc_id = TRUE` (or `full_detail = TRUE`); the output then includes a `gc_internal_id` column equal to the user's id when no regen was needed, or the auto-generated 1:N when a regen occurred. For stable row identifiers, supply a unique per-row `id`; a per-subject or per-encounter identifier is not a valid `id` for this purpose. For semi-wide/wide input with long output, the subject, sex and age columns are returned under the user's own names (for wide input, the age prefix), and the id identifies an input row,

not a measurement: it is returned under its own column name (repeated on each measurement from that row), and growthcleanr's per-measurement number is returned as gc_internal_id; with no id column there is no id column in that output. An id column named like a column growthcleanr returns (e.g. exclude) comes back as userinput_<name>, with a warning. In semi-wide/wide input the id need not be unique; in semi-wide input it orders a subject's visit rows when the per-measurement ids are assigned (see vignette("wrapper")), and in wide input it does not affect cleaning.

dob — Date of birth (data-frame style only). A column name; the column is class Date or text. Supplied with visit_date to compute agedays as floor(visit_date - dob) in whole days. A subject with more than one distinct dob value is set aside as Exclude-Missing-Info, with a warning. Text dates are accepted in any order-free format – year-first numeric ("2013-03-04", "2013/03/04", "20130304") or English spelled-month ("03mar2013", "15 Mar 2013", "March 15, 2024"); ambiguous numeric day/month dates ("3/4/2013") require date_order. All dob and visit_date values must share one format, the year must be four digits (1900-2099; two-digit years are rejected), and blank/NA yields a missing agedays. A value with no digits (e.g. "N/A", "NULL") is treated as blank, and a few values that are not real dates (e.g. "2013-02-30"; at most 1, or 5\ aside; both are reported in the input warning. More unreadable dates than that stop the run. Numeric date columns (e.g. software date integers from Stata, Excel, etc.) are rejected as ambiguous – convert to text or an R Date first. A lone dob or visit_date supplied alongside an agedays column is parsed and returned only (as gc_internal_dob/gc_internal_visit_date under internal_detail); ages still come from the agedays column unless you set age_source = "dates" to force date-based ages. Default NULL. See "Use dates instead of age in days" in vignette("example-scripts").

visit_date — Visit (measurement) date (data-frame style only). A column name; class Date or text in the same formats as dob. Supplied with dob to compute agedays, or on its own alongside an agedays column purely to be parsed and returned as gc_internal_visit_date (under internal_detail). See dob. Default NULL.

age_source — Which age source to use when resolving agedays: "agedays" (use the agedays column) or "dates" (compute from dob + visit_date). Default NULL auto-resolves: an agedays column is used whenever present, otherwise dob + visit_date are used. Set "dates" to force date-based ages even when an agedays column is also present; that column is then returned as submitted, and the ages computed from the dates are returned as gc_internal_agedays.

date_order — Day-vs-month order for ambiguous numeric dob / visit_date values, e.g. "3/4/2013" (data-frame style only). "dmy" reads day-month-year, "mdy" reads month-day-year; applied uniformly to the whole run. Default NULL accepts only the order-free formats (year-first numeric and English spelled-month) and errors on numeric day/month dates, so an ambiguous order is never guessed. growthcleanr never auto-detects the order.

sex_coding — Named list with elements male and female, each a vector of the codes (numbers or strings) that denote that sex in your data. Defaults to list(male = c(0, "m", "b", "male", "boy"), female = c(1, "f", "g", "female", "girl")). Matching is case-insensitive and whitespace-trimmed. Supplying your own table re-

places the defaults wholesale (include any standard codes you also use) and suppresses the 1/2 guard. Errors if a code is listed under both sexes. See "Set your sex coding (including 1 = male, 2 = female)" in `vignette("example-scripts")`.

`param_coding` Optional named list mapping your parameter labels to growthcleanr's canonical parameters, e.g. `list(WEIGHTKG = c("WT", "Weight"), HEIGHTIN = "HT")`. Names must be canonical parameters (`WEIGHTKG`, `WEIGHTLBS`, `HEIGHTCM`, `HEIGHTIN`, `LENGTHCM`, `LENGTHIN`, `HEADCM`, `HEADIN`); values are the labels to remap (case-insensitive, whitespace-trimmed). This is the only way to have a non-canonical label recognized (there are no built-in abbreviations), and also how you declare that a remapped column is imperial (map it to `WEIGHTLBS`/`HEIGHTIN`/`LENGTHIN`/`HEADIN`); otherwise a remapped label defaults to its metric target. Errors if a name is not canonical or a label maps to two parameters. Default `NULL`. See "Use your own measurement-type labels (WT / HT / HC)" in `vignette("example-scripts")`.

`carry_through` Logical (data-frame style only). If `TRUE` (default), any column in `data` that growthcleanr does not consume as a role column (i.e. anything other than the resolved `subjid`, `param`, `agedays`, `sex`, `measurement`, and `id` columns) is carried through and returned appended after the growthcleanr output columns, in its original input order and values. `dob`/`visit_date` are carried through this way (so submitted dates are echoed back). A carried column whose name matches a column growthcleanr returns in this run (for example `exclude`, when re-cleaning growthcleanr output) is returned under the name `userinput_<name>`, with a warning; names of growthcleanr columns not returned in the run are left alone. `FALSE` restores the prior behavior of dropping all non-role columns. With semi-wide or wide output the user's columns return structurally (the submitted table is the output base), so `TRUE` keeps every column and `FALSE` keeps only the role, id, and parameter value columns. With semi-wide or wide input and long output, the other columns (including `dob`/`visit_date`; in wide input, not the per-block columns) are repeated on each measurement from their row.

`input_format` Shape of `data`: `"long"` (default; one row per measurement), `"semi_wide"` (one row per visit: a `subjid`, an age source, a `sex`, and one column per parameter), or `"wide"` (one row per subject: a `subjid`, a `sex`, and repeated observation blocks marked by a `_N` suffix – `agedays_1`, `wt_1`, `ht_1`, `agedays_2`, ...). Semi-wide and wide are reshaped to long before cleaning, so results match submitting the same measurements in long format; both require the data-frame style. Map your own column names with `semiwide_cols`/`wide_cols`. For ready-to-run scripts, see "Semi-wide data (one row per visit)" and "Wide data (one row per subject)" in `vignette("example-scripts")`.

Length/height in non-long formats. Customized recumbent-length vs standing-height handling is a long-format-only feature: semi-wide and wide always use default handling (values under 2 years treated as recumbent length, 2 years and over as standing height, using estimated corrected age for children whose data are consistent with prematurity or small-for-gestational age (SGA) and who receive an assigned gestational age for cleaning purposes). So `length.adjust = TRUE` is rejected for these formats, and a `LENGTHCM`/`LENGTHIN` value at any age gets default handling, the same as in long input with `length.adjust = FALSE`. See `length.adjust` and `vignette("wrapper")`, section "Length vs. height handling".

output_format Shape of the returned results: "long", "semi_wide", or "wide". Default NULL mirrors input_format, so semi-wide input returns semi-wide output and wide input returns wide output. Re-widened output returns the submitted table itself – every column untouched, one row per input row – with the growthcleanr columns named in widened_emit inserted immediately right of the value column they describe (exclude_wt_kg for semi-wide, exclude_wt_1 for wide). It requires the matching input format (growthcleanr cannot infer visit rows or observation blocks from a different shape); checkpointed runs are supported. Choose output_format = "long" for the full per-measurement result table, which can be merged back onto the source data by subjid/param/agedays.

semiwide_cols Named list/vector mapping growthcleanr parameter names to the columns holding their values in semi-wide input, e.g. list(WEIGHTKG = "wt_kg", HEIGHTCM = "ht_cm", HEADCM = "head_cm"). Recognized names are WEIGHTKG, HEIGHTCM, LENGTHCM, HEADCM, WEIGHTLBS, HEIGHTIN, LENGTHIN, and HEADIN (imperial/length variants are converted downstream). If NULL (default), columns literally named as recognized parameters are used; at least one recognized parameter column must be present. The restriction rejecting LENGTHCM/LENGTHIN for children 2 years and older (described under input_format) applies. Only used when input_format = "semi_wide".

widened_emit Character vector naming the growthcleanr output columns to add to re-widened output (semi-wide or wide); default "exclude". Any per-measurement output column may be named (e.g. "bin_exclude", "exclude_detailed", "unit_error_range", "measurement", "agedays"), and a column whose own switch was not enabled is an error. Used when output_format is "semi_wide" or "wide". How the added columns are named (e.g. exclude_wt_kg) is described in vignette("wrapper"), sections "Output format: semi-wide" and "Output format: wide".

wide_cols Named list/vector mapping growthcleanr parameter names to the column-name *prefixes* holding their values in wide input, e.g. list(WEIGHTKG = "wt", HEIGHTCM = "ht", HEADCM = "hc") for columns wt_1, wt_2, ht_1, ... You choose the prefix; growthcleanr requires the _N block suffix and discovers the blocks from it. Recognized names are as for semiwide_cols, and the same length-at-2-years restriction (see input_format) applies. If NULL (default), prefixes literally named as recognized parameters are used (WEIGHTKG_1, ...). The age source is a prefix too: agedays names agedays_1, agedays_2, ... (or, on the date route, visit_date names visit_date_1, ... alongside a single patient-level dob column). subjid and sex are patient-level and unsuffixed. Subjects with fewer observations leave their trailing blocks empty; empty cells are dropped. Only used when input_format = "wide".

max_blocks Highest number of wide observation blocks expected; default 25. Finding more warns (with guidance to raise max_blocks or reshape to long or semi-wide) but still processes every block; an implausible count errors, since it usually means the _N suffixes parsed wrong. Only used when input_format = "wide".

internal_detail

Logical. If TRUE, adds growthcleanr's own working values for the key inputs the user supplied: gc_internal_meas (the measurement gc used – imperial converted to metric and LENGTHCM relabeled; the length.adjust offset is applied only to z-scores, not to this value), gc_internal_param (the

parameter label that goes with it), gc_internal_sex (the coerced 0/1 sex), gc_internal_id, and gc_internal_agedays (the whole-day age used for cleaning). The primary measurement, param, agedays, sex, and id columns always return the user's verbatim submission; these report what the algorithm worked with. The imperial conversion is reflected for every row (the adult algorithm applies the same factors), and LENGTHCM is relabeled HEIGHTCM at every age (adult length is cleaned as height); the length.adjust reconciliation is pediatric-only. If a dob and/or visit_date column was supplied, the parsed Date values are also added as gc_internal_dob / gc_internal_visit_date (whether or not the dates drove agedays), showing the interpretation growthcleanr used. Default FALSE. Enabled by full_detail.

ref_tables — Optional. Pre-loaded reference tables from gc_preload_refs. When provided, growthcleanr skips reading the reference files, saving about a second per call. Recommended for repeated calls (e.g., simulation loops). Defaults to NULL (reads files each call).

tri_exclude — Logical. If TRUE, include a tri_exclude column in the output that classifies each row as "Include", "Same-Day" (same-day extraneous values), or "Exclude" (all other exclusions). Defaults to FALSE.

debug — Logical. If TRUE, retain intermediate diagnostic columns in the output instead of trimming to the public columns. Intended for threshold refinement and development, not routine use. Currently surfaces the raw and recentered CSD z-scores captured before cleaning (sd.orig, sd.corr, tbc.sd, ctbc.sd) and the recentered plus/minus perturbation z-scores (tbc.p_plus, tbc.p_minus, ctbc.p_plus, ctbc.p_minus), plus gc_internal_sex (growthcleanr's interpreted 0/1 sex). Adult rows receive NA for the child-only z-score columns. Defaults to FALSE.

full_detail — Logical. Convenience switch: if TRUE, turns on all optional output groups at once (equivalent to cf_detail = TRUE, corr_detail = TRUE, debug = TRUE, tri_exclude = TRUE, display_gc_id = TRUE, recenter_detail = TRUE, internal_detail = TRUE, exclude_detail = TRUE, and mean_ht = TRUE, which is already on by default when the data has adult rows). Defaults to FALSE. Does NOT turn on qm_detail (the quality-measure eligibility flags must be requested explicitly), and does not affect unit_error_range, which is on by default independently.

unit_error_range — Logical. If TRUE (default), include a unit_error_range column: a direction-agnostic flag that is TRUE when a value's magnitude is consistent with a metric/imperial unit-conversion error and the value is not itself within the plausible z = -3 to +3 band. Purely informational – it never affects the exclude decision and is not an exclusion code. Computed only for evaluable rows (Exclude-Missing-Info, Exclude-Not-GC-Param, and Exclude-HC-Out-of-Range rows are FALSE). Real values from people with atypical growth (for example, extremely premature infants, severe obesity, some medical conditions) can fall in the unit-error range. Set FALSE to omit the column.

display_gc_id — Logical. If TRUE, add a gc_internal_id column to the output containing the id that growthcleanr actually used internally: the user's id when it was valid (no regen needed), or the auto-generated 1:N (in input row order) when the user's id was absent / NA / non-unique and growthcleanr had to regenerate. Lets callers

recover a reliable per-row key when their input id had problems (the output id column always preserves whatever the user supplied, even if invalid). Defaults to FALSE. Set via full_detail = TRUE alongside the other detail switches. For semi-wide/wide input with long output, gc_internal_id is always returned.

exclude_detail Logical. If TRUE, add an exclude_detailed column reporting each value's exclusion at the Detailed level: step-level codes that keep the child (-C-) / adult (-A-) algorithm prefix. The default exclude column always reports the consolidated Summary level. Preprocessing codes such as Exclude-Missing-Info are the same in both columns. Defaults to FALSE.

mean_ht Logical. Include a mean_ht column: the adult algorithm's calculated mean height (cm), computed only from the subject's finally-included heights but populated on all of the subject's adult height rows (an excluded row takes the value of the included height nearest in age). NA for child rows, adult weight rows, and subjects with no included adult height. The column can be used by researchers who want a consistent height for evaluation of body mass index. By default the column is returned when the data has adult rows and omitted otherwise; set TRUE to always return it or FALSE to omit it. For semi-wide or wide output, name it in widened_emit to add it beside the height values.

batch_size Number of subjects per processing batch. Subjects are never split across batches. Must be a single positive whole number. Defaults to 5000. This is the *in-memory* granularity (how many subjects are held in RAM at once within a run); it is independent of checkpoint_size, the persist granularity used under checkpointing.

length.adjust Logical (default FALSE). If TRUE, reconciles how each linear growth measurement was entered with what the reference at that age expects, in **both directions**: a LENGTHCM value at 2 years or older is z-scored 0.7 cm lower (recumbent to standing) and a HEIGHTCM value under 2 years 0.7 cm higher (standing to recumbent), where 2 years means day 730, the day the reference files switch from length to height; values already in the expected method, and rows with a missing age, are unchanged. The offset is applied only where a value is z-scored: the checks that work in raw centimeters (such as carried-forward detection) and gc_internal_meas use the entered value, and the returned measurement is always the user's verbatim submission. It is pediatric-only (rows at or above adult_cutpoint are not adjusted; adult LENGTHCM/LENGTHIN rows are relabeled as height, so adult length is cleaned as height). **Long input only:** rejected for input_format "semi_wide"/"wide", which always use default handling. For gestational-age-corrected children with chronological age 2 years or older but corrected age under 2 years, the default presumes recumbent length (the value is z-scored 0.7 cm lower against the standing-height reference), while length.adjust = TRUE uses the label. Full detail (the both-directions table, the exact cutoff, the gestational-age window, and where it runs in preprocessing) is in vignette("wrapper"), section "Length vs. height handling".

checkpoint Logical (default TRUE). Master switch for checkpointing – cleaning the subjects in checkpoints and writing each finished checkpoint to disk instead of holding the whole dataset in memory at once. Checkpointing substantially speeds up large datasets and prevents the out-of-memory crashes that large in-memory runs can hit (the memory-heavy steps scale super-linearly with the in-memory working set). When TRUE and no checkpoint_dir is given, checkpoints go

to an ephemeral per-session temporary directory and are deleted automatically when the run ends, including after an error or interrupt, though not if R itself crashes or is killed (keep_checkpoints is forced FALSE); this only engages when the data exceeds one checkpoint (more than checkpoint_size subjects), since a single checkpoint is the whole dataset in memory and would give no benefit – so small/typical data runs fully in memory with nothing written to disk. Set checkpoint = FALSE to disable checkpointing entirely and never write intermediate data to disk (the escape hatch for environments that forbid any data on disk); if a checkpoint_dir was also supplied it is ignored (with a warning). Supplying checkpoint_dir keeps the files at that location (see below). Very large datasets on a Linux system whose temporary directory is a RAM disk (tmpfs) should pass a checkpoint_dir on a real disk volume; macOS and Windows use disk-backed temporary directories by default.

checkpoint_dir Optional directory for *persistent* checkpointed runs. When set, cleangrowth splits subjects into checkpoints of checkpoint_size, cleans each checkpoint, saves its result to a checkpoint_NNNN.rds file in this directory, and combines the checkpoints at the end. Unlike the ephemeral checkpoint = TRUE default, these files persist after the run (subject to keep_checkpoints) and can be resumed: if a run is interrupted, re-running with checkpoint_resume = TRUE skips checkpoints that already have a checkpoint file. The directory is created if it does not exist; use a dedicated directory per dataset/run. When NULL (default), checkpointing – if it engages at all – writes to the ephemeral temporary directory described under checkpoint. Checkpointed results are byte-identical to a single-pass run (each subject's rows stay together in one checkpoint).

checkpoint_size

Number of subjects per checkpoint (one checkpoint_NNNN.rds file per checkpoint). Default 5000. Also the subject count above which the default checkpoint = TRUE engages (temporary folder) when checkpoint_dir is not set. This is the *persist* granularity and is independent of batch_size (the in-memory granularity within a checkpoint); batch_size < checkpoint_size is allowed. The number of checkpoints is capped at 9999: a checkpoint_size that would exceed the cap is increased to stay within it (with a warning when you set the size; results are not affected). When resuming, use the same checkpoint_size and dataset as the original run.

checkpoint_resume

Logical (default FALSE). Only used when checkpoint_dir is set. When FALSE, the run starts fresh: any checkpoint files already in checkpoint_dir from a prior run are deleted first. When TRUE, existing checkpoint files are reused – checkpoints that already have one are skipped – to resume an interrupted run. When TRUE, you are responsible for using the same dataset, checkpoint_size, and other cleaning arguments as the run that wrote those files; these are not checked, and a mismatch silently mixes results from different settings.

keep_checkpoints

Logical (default TRUE). Only used when checkpoint_dir is set. Controls what happens to the per-checkpoint checkpoint files *after* a successful run: if TRUE they are left in checkpoint_dir; if FALSE they are deleted once the combined result is returned. Independent of checkpoint_resume.

`log_file` Optional path to a log file. Used whenever checkpointing engages: when `checkpoint_dir` is set, or under the default `checkpoint = TRUE` when the data has more than `checkpoint_size` subjects. Progress messages are always written to the console (unless `quietly`); if provided, they are also appended here so a colleague can monitor a long run remotely. The log file is safe to read while the run is in progress.

`.checkpoint_driven` Internal. Set to TRUE by the internal checkpoint runner so that the per-checkpoint header, banner, and elapsed-time framing are printed by the checkpoint wrapper instead of here (this call then prints only its milestone lines). Not for direct use.

`...` Not used. Present so that an argument removed since growthcleanr 2.x (for example `num.batches` or `weight_cap`) stops with what to use instead; any other unrecognized argument stops as usual. The full list is in "Removed and renamed arguments" in `vignette("experienced-start-here")`.

Details

Calling styles:

1. **Data-frame style (recommended).** Pass a single data.frame or data.table as `data`. The `subjid`, `param`, `agedays`, `sex`, `measurement`, and `id` arguments are then column-name strings identifying which columns in `data` to use. Each defaults to the canonical name (e.g. `"subjid"`), so if your columns already use those names you can just call `cleangrowth(data = my_df)`. Use the column-name arguments only to remap (e.g. `subjid = "patid"`).
2. **Vector style (legacy).** Omit `data` and pass each of `subjid`, `param`, `agedays`, `sex`, and `measurement` as a vector. Retained for backward compatibility with existing code; new code should use the data-frame style.

All vectors (or the columns referenced in `data`) must have the same length and be aligned row-for-row. Pulling values from different datasets is not supported and not checked for.

Arguments at a glance. The many arguments fall into groups; most runs touch only the first one or two:

- **Core input column names** – `data`, `subjid`, `sex`, `agedays`, `param`, `measurement`, `id`.
- **Age from dates** (instead of `agedays`) – `dob`, `visit_date`, `age_source`, `date_order`.
- **Sex and parameter coding** – `sex_coding`, `param_coding`.
- **Input/output shape** – `input_format`, `output_format`, `semiwide_cols`, `wide_cols`, `max_blocks`, `widened_emit`, `carry_through`.
- **Run control and performance** – `parallel`, `cores`, `batch_size`, `quietly`, `verbose`, `ref_tables`, `ref.data.path`, `log.path`.
- **Checkpointed (resumable) runs** – `checkpoint`, `checkpoint_dir`, `checkpoint_size`, `checkpoint_resume`, `keep_checkpoints`, `log_file`.
- **Adult algorithm** – `adult_cutpoint`, `adult_permissiveness`, `adult_scale_max_lbs`, `adult_overrides`.
- **EWMA window (child and adult)** – `ewma_window`.
- **Length vs. height handling** – `length.adjust`.
- **Recentering** – `sd.recenter`.

- **Carried-forward rescue** – `cf_rescue`.
- **Optional output columns** – `exclude_detail`, `unit_error_range`, `tri_exclude`, `cf_detail`, `corr_detail`, `recenter_detail`, `internal_detail`, `display_gc_id`, `mean_ht`, `full_detail`, `qm_detail`, `debug`.
- **Advanced threshold overrides** (rarely changed) – `piv.z.*`, `error.load.*`. See `vignette("algorithm-functions"` for the CSD z-scores (`sd.orig`) the `piv.z.*` cutoffs apply to.

Value

By default, a `data.table` in long format (one row per input measurement, in the same order as the input); the exclusion codes are in its `exclude` column. (Prior versions returned a character vector of exclusion codes.) Eight columns are always present, plus `unit_error_range` unless you turn it off:

`id` User-provided row identifier (or the auto-generated 1:N if none was usable).

`subjid` Subject identifier, as submitted.

`param` Measurement parameter, as submitted.

`agedays` Age in days, as submitted.

`sex` Sex, as submitted.

`measurement` Measured value, as submitted.

`exclude` The result (a factor): `"Include"` or a Summary-level exclusion code naming why the value was set aside.

`bin_exclude` A two-way collapse of `exclude` (character): `"Include"` or `"Exclude"`.

`unit_error_range` **On by default** (set `unit_error_range = FALSE` to omit). A logical informational flag, `TRUE` for a value whose magnitude looks like a unit-conversion error; never affects `exclude`.

`mean_ht` **Returned by default when the data has adult rows** (set `mean_ht = FALSE` to omit). The adult algorithm's calculated mean height (cm), computed from included heights only and populated on all of a subject's adult height rows; `NA` for child rows, adult weight rows, and subjects with no included adult height.

Exclusion codes changed in version 3; for the mapping from 2.x codes, see "Mapping from prior exclusion codes" in `vignette("experienced-start-here")`.

Output fidelity. The five echoed input columns (`subjid`, `param`, `agedays`, `sex`, `measurement`) carry your *verbatim submitted values* – growthcleanr never silently alters or drops an input value. For example, imperial values are returned as submitted (not converted to metric), even though internally, growthcleanr converts them to metric. A value growthcleanr cannot use – an uninterpretable `sex`; a `0`, negative, or missing `measurement`; a missing `param`; or a negative/`NA` `agedays` – is flagged `Exclude-Missing-Info`; a `param` that is not one growthcleanr recognizes is flagged `Exclude-Not-GC-Param`; and a head-circumference value at 5 years or older (beyond the reference) is flagged `Exclude-HC-Out-of-Range`. In every case the value is still returned as submitted. The internal working values are available via `internal_detail`/`debug` (see below).

Column naming. The six data columns (`id`, `subjid`, `param`, `agedays`, `sex`, `measurement`) are renamed to match your original argument expressions when those resolve to a simple name – e.g. `cleangrowth(subjid = dt$patid, ...)` returns a column named `patid`. Complex expressions

fall back to the canonical name. In data-frame style, output columns keep your column names (e.g. `subjid = "patid"` returns `patid`).

Optional columns. Each of these switches adds columns; all default to off unless noted:

`exclude_detail` `exclude_detailed` – the step-level (Detailed) exclusion code behind each Summary `exclude` value.

`tri_exclude` `tri_exclude` – `"Include"`, `"Same-Day"`, or `"Exclude"`.

`cf_detail` `cf_status`, `cf_deltaz`, `cf_nextz`, and three carried-forward design aids (`imperial`, `cf_dage_orig`, `cf_dage_next`).

`corr_detail` `corr_val` (per value) and `corr_pt` (per subject) gestational-age correction status.

`recenter_detail` The recentered CSD z-score columns `tbc.sd`, `ctbc.sd`, and `rcmedian.sd` (child rows; NA for adults) – specialized output for a related tool in development.

`internal_detail` growthcleanr's working values for your inputs: `gc_internal_meas`, `gc_internal_param`, `gc_internal_sex`, `gc_internal_id`, `gc_internal_agedays` (the whole-day age used for cleaning) (and parsed `gc_internal_dob`/`gc_internal_visit_date` if dates were supplied).

`display_gc_id` `gc_internal_id` – the id growthcleanr actually used internally.

`debug` Intermediate z-score diagnostics for child rows (`sd.orig`, `sd.corr`, `tbc.sd`, `ctbc.sd`, and the plus/minus perturbation z-scores) plus `gc_internal_sex`; for development, not routine use.

`qm_detail` Ten per-step child eligibility flags (`qm_elig_*`); for eligibility-normalized quality measures used for a related package in development. Must be requested explicitly – **not** turned on by `full_detail`.

`full_detail` Convenience switch turning on all of the above *except* `qm_detail` (and independent of `unit_error_range`).

Non-long output. With `output_format = "semi_wide"` or `"wide"` the returned object is your submitted table with the columns named in `widened_emit` inserted beside the value they describe (see `output_format`).

The full catalog of exclusion codes (child + adult, Summary and Detailed) is in `vignette("exclusion-codes")`.

Examples

```
# Check your setup on a sample of your data first: gc_runtrial(syngrowth)

# Data-frame style (recommended): canonical column names, just pass the
# data.frame.
# syngrowth is a small example dataset bundled with growthcleanr.
df_stats <- as.data.frame(syngrowth)
# Use 5 subjects so the example runs quickly
df_stats <- df_stats[df_stats$subjid %in% unique(df_stats[, "subjid"])[1:5], ]
# We recommend including a unique per-row `id` column (syngrowth already
# does; this shows how to add one)
df_stats$id <- seq_len(nrow(df_stats))
clean_stats <- cleangrowth(data = df_stats)

# How many values were included, and why the others were excluded:
table(clean_stats$exclude)
```

```
# The result is your data with an added `exclude` column; keep only the
# measurements growthcleanr says to include:
clean_only <- clean_stats[clean_stats$exclude == "Include", ]

# Data-frame style with remapped column names
my_df <- df_stats
names(my_df)[names(my_df) == "subjid"] <- "patid"
clean_stats <- cleangrowth(data = my_df, subjid = "patid")

# Vector style (legacy)
clean_stats <- cleangrowth(subjid = df_stats$subjid,
                           param = df_stats$param,
                           agedays = df_stats$agedays,
                           sex = df_stats$sex,
                           measurement = df_stats$measurement,
                           id = df_stats$id)

# Parallel processing: run using 2 cores
clean_stats <- cleangrowth(data = df_stats,
                           parallel = TRUE,
                           cores = 2)
```

`cleangrowth_checkpoint` *Run cleangrowth() with automatic checkpointing (deprecated wrapper)*

Description

Deprecated. Checkpointing is built into `cleangrowth` via its `checkpoint_dir` argument – call `cleangrowth(data, checkpoint_dir = ...)` directly. This wrapper forwards to `cleangrowth()` and will be removed in a future version. The `use_existing_checkpoint_results` argument maps to `cleangrowth(checkpoint_resume = ...)`. Calling `cleangrowth()` directly also lets a checkpointed run use wide / semi-wide input and output (via its `input_format` / `output_format`).

Usage

```
cleangrowth_checkpoint(
  data,
  checkpoint_dir,
  checkpoint_size = 5000,
  log_file = NULL,
  keep_checkpoints = TRUE,
  use_existing_checkpoint_results = FALSE,
  quietly = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

`data`, `checkpoint_dir`, `checkpoint_size`, `log_file`, `keep_checkpoints`, `quietly`, `verbose`
: See `cleangrowth`. `quietly` defaults to `FALSE` here (this wrapper prints by default).

`use_existing_checkpoint_results`
: Deprecated alias for `cleangrowth(checkpoint_resume = ...)`: reuse existing checkpoint files (resume an interrupted run) instead of starting fresh.

`...`
: Additional arguments passed to `cleangrowth` (e.g. `cf_rescue`, `sd.recenter`, `adult_permissiveness`).

Value

A data.table – see `cleangrowth` for the output columns.

See Also

`cleangrowth`, `cleangrowth_checkpoint_combine`

Examples

```
## Not run:
library(data.table)
dt <- fread("large_dataset.csv")

# Preferred: call cleangrowth() directly with checkpoint_dir.
result <- cleangrowth(dt, checkpoint_dir = "results/gc_checkpoints",
                      checkpoint_size = 5000, log_file = "results/gc_run.log")

## End(Not run)
```

`cleangrowth_checkpoint_combine` *Combine saved checkpoint files from a checkpointed cleangrowth() run*

Description

Reads all checkpoint `.rds` files from a directory and combines them into a single data.table. Use this when a checkpointed `cleangrowth` run (`checkpoint_dir` set) completed all checkpoints but the combine step failed - you can re-combine without re-running the algorithm.

Usage

```
cleangrowth_checkpoint_combine(
  checkpoint_dir,
  id_order = NULL,
  log_file = NULL,
  quietly = FALSE,
  verbose = FALSE
)
```

Arguments

checkpoint_dir	Directory containing checkpoint .rds files named checkpoint_0001.rds, checkpoint_0002.rds, etc.
id_order	Optional vector of id values specifying the desired output row order (typically your_data$id). If NULL, rows are returned in checkpoint order (subjects grouped together but not in the original row order).
log_file	Optional path to append log messages.
quietly	Logical (default FALSE). When FALSE, prints a short start and completion line (and writes them to log_file if set). When TRUE, the console is silent; log_file still receives the log.
verbose	Logical (default FALSE). When TRUE, additionally prints the per-file read progress. Implies quietly = FALSE.

Details

All .rds files matching checkpoint_NNNN.rds in checkpoint_dir are read and combined in numeric order. Every matching file is included, so the directory should hold checkpoints from a single run only. (A checkpointed cleangrowth() run with checkpoint_resume = FALSE, the default, removes earlier checkpoint files from the directory at the start, so after a run only that run's files remain.)

Value

A long-format data.table with growthcleanr's standard column names (id, subjid, param, agedays, sex, measurement, plus result columns). Unlike cleangrowth(), it does not restore the user's column names, carried-through columns, or wide/semi-wide layout; if growthcleanr generated id values (because id was missing or not unique), id holds those generated values.

See Also

cleangrowth_checkpoint

Examples

```
## Not run:
# If the combine step failed at the end of a run, re-combine like this:
result <- cleangrowth_checkpoint_combine(
  checkpoint_dir = "results/gc_checkpoints",
```

```
  id_order       = dt$id   # original row order; needs a unique id column named id
)
saveRDS(result, "results/gc_output.rds")

## End(Not run)
```

---

ewma *Exponentially Weighted Moving Average (EWMA)*

Description

ewma calculates the exponentially weighted moving average (EWMA) for a set of numeric observations over time.

Usage

```
ewma(agedays, z, ewma.exp, ewma.adjacent = TRUE, window = 15, cache_env = NULL)
```

Arguments

agedays	Vector of age in days for each z score (potentially transformed to adjust weighting).
z	Input vector of numeric z-score data.
ewma.exp	Exponent used for age-gap weighting. May be a single scalar (recycled across all observations) or a per-observation vector matched to agedays / z; the algorithm's main internal callers pass a per-observation vector that varies the exponent by the widest neighbor age gap.
ewma.adjacent	Specify whether EWMA values excluding adjacent measurements should be calculated. Defaults to TRUE.
window	Maximum number of observations on each side for EWMA weighting. Default 15. Set to Inf to disable windowing.
cache_env	Optional environment for caching EWMA intermediate results. Used internally for performance. Defaults to NULL.

Value

A named list of numeric vectors, each the same length as agedays.

- ewma.all - EWMA at each observation's age, excluding only the observation itself.
- ewma.before - also excludes the immediate prior observation (sorted by agedays); equals ewma.all for the first observation and when n <= 2.
- ewma.after - also excludes the immediate subsequent observation; equals ewma.all for the last observation and when n <= 2.

When ewma.adjacent = FALSE, only ewma.all is returned (still wrapped in a one-element named list).

Examples

```
# Run on 1 subject, 1 type of parameter
df_stats <- as.data.frame(syngrowth)
df_stats <- df_stats[df_stats$subjid == df_stats$subjid[1] &
                       df_stats$param == "HEIGHTCM", ]

# Get the uncentered z-scores
measurement_to_z <- read_anthro(cdc.only = TRUE)
sd <- measurement_to_z(df_stats$param,
                       df_stats$agedays,
                       df_stats$sex,
                       df_stats$measurement,
                       TRUE)

# Calculate exponentially weighted moving average
e_df <- ewma(df_stats$agedays, sd, ewma.exp = -1.5)
```

`ext_bmiz` *Calculate extended BMI measures*

Description

`ext_bmiz` Calculates the sigma (scale parameter for the half-normal distribution), extended BMI percentile, extended BMIz, and the CDC LMS Z-scores for weight, height, and BMI for children between 2 and 19.9 years of age. Note that for BMIs <= 95th percentile of the CDC growth charts, the extended values for BMI are equal to the LMS values. The extended values differ only for children who have a BMI > 95th percentile.

Usage

```
ext_bmiz(
  data,
  age = "agem",
  wt = "wt",
  ht = "ht",
  bmi = "bmi",
  adjust.integer.age = TRUE,
  ref.data.path = ""
)
```

Arguments

`data`	Input data frame or data table
`age`	Name of input column containing subject age in months in quotes, default "agem"
`wt`	Name of input column containing weight (kg) value in quotes, default "wt"
`ht`	Name of input column containing height (cm) value in quotes, default "ht"
`bmi`	Name of input column containing calculated BMI in quotes, default "bmi"

`adjust.integer.age`
: If age inputs are all integer, add 0.5 if TRUE; default TRUE

`ref.data.path`
: Path to directory containing reference data

Details

This function produces output equivalent to CDC software and the cdcanthro R package.

The extended BMIz is the inverse cumulative distribution function (CDF) of the extended BMI percentile. If the extended percentile is very close to 100, the `qnorm` function in R produces an infinite value. This occurs only if the extended BMI percentile is > 99.99999999999999. This occurs infrequently, such as a 48-month-old with a BMI > 39, and it is likely that these BMIs represent data entry errors. For these cases, extended BMIz is set to 8.21, a value that is slightly greater than the largest value that can be calculated.

The output columns are listed under Value below.

`data` must have columns for at least age, sex, weight, height, and bmi.

`age` should be coded in months, using the most precise values available. To convert to months from age in years, multiply by 12. To convert to months from age in days, divide by 30.4375 (365.25 / 12).

`sex`: we recommend M and F, which is what `longwide()` returns. Male can be coded as boy, Boy, b, B, male, Male, m, or M, and female can be coded as girl, Girl, g, G, female, Female, f, or F. Numeric sex is accepted only if `sex` has both 1s and 2s and no other values, read as 1 = male, 2 = female; otherwise `ext_bmiz()` stops. This is because 0/1 is the numeric coding in `cleangrowth()`, and a column of only 1s could be either coding. The 1/2 coding matches the CDC reference data, CDC software, and the cdcanthro package. Other text values, and logical values, are not recognized: their z-scores are `NA`, with a warning.

`wt` should be in kilograms.

`ht` should be in centimeters.

Specify the input data column names for `age`, `wt`, `ht`, `bmi` using quotation marks. See example below.

`ext_bmiz()` does not change `data`. It returns a copy of `data` with the new columns added; your column names and values are kept as they are (the 0.5 added by `adjust.integer.age` is used only in the calculation). If `data` already has a column with the name of one of the columns `ext_bmiz()` adds (`bmiz`, `bmip`, `waz`, `wp`, `haz`, `hp`, `p95`, `p97`, `bmip95`, `mod_bmiz`, `mod_waz`, `mod_haz`, `sigma`, `original_bmip`, `original_bmiz`, `sev_obese`, `obese`), for example from an earlier run, it warns and the returned data has the newly calculated values in that column; rename the column first to keep both.

Values are calculated for ages 24 through 240 months, as in the cdcanthro package. Age 730 days is 23.98 months (730 / 30.4375), which is below 24 months, so a row at 730 days has `NA` values even though a child at that age may be exactly 2 years old; values begin at 731 days. A warning gives the number of rows at about 729-730 days.

If the argument `adjust.integer.age` is `TRUE` (the default), 0.5 will be added to all `age` values for the calculation if all input values are integers. Set to `FALSE` to disable.

By default, the reference data file `CDCref_d.csv`, made available at https://www.cdc.gov/growthchart-training/hcp/computer-programs/index.html, is included in this package for convenience. If you are developing this package, use `ref.data.path` to adjust the path to this file from your working directory if necessary.

Value

A data.table: a copy of `data` (same rows, row order, column names, and values) with these columns added: `bmiz`, `bmip`, `waz`, `wp`, `haz`, `hp`, `p95`, `p97`, `bmip95`, `mod_bmiz`, `mod_waz`, `mod_haz`, `sigma`, `original_bmip`, `original_bmiz`, `sev_obese`, `obese`.

References

Wei R, Ogden CL, Parsons VL, Freedman DS, Hales CM. A method for calculating BMI z-scores and percentiles above the 95th percentile of the CDC growth charts. Ann Hum Biol. 2020 Sep;47(6):514-521. doi:10.1080/03014460.2020.1808065.

Examples

```
# Run on a small subset of given data
df <- as.data.frame(syngrowth)
df <- df[df$subjid %in% unique(df[, "subjid"])[1:5], ]
# cleangrowth() returns the data with an 'exclude' column
clean_df <- cleangrowth(df)
df_wide <- longwide(clean_df) # convert to wide format for ext_bmiz
df_wide_bmi <- simple_bmi(df_wide) # compute simple BMI

# Calling the function with default column names
df_bmiz <- ext_bmiz(df_wide_bmi)

# Specifying different column names; note that quotes are used
dfc <- simple_bmi(df_wide)
colnames(dfc)[colnames(dfc) %in% c("agem", "wt", "ht")] <-
  c("agemos", "weightkg", "heightcm")
df_bmiz <- ext_bmiz(dfc, age="agemos", wt="weightkg", ht="heightcm")

# Disabling conversion of all-integer age in months to (age + 0.5)
dfc <- simple_bmi(df_wide)
df_bmiz <- ext_bmiz(dfc, adjust.integer.age=FALSE)
```

---

fenton2025_ms_lookup_smoothed

Approximated Fenton CSD Values (M, S_upper, S_lower)

Description

Approximated Fenton CSD values: approximated reference values derived from printed Fenton (2025) charts, for weight, length, and head circumference by sex and gestational age in days. Contains the median (M), S_upper, and S_lower, where S_upper x M is half the distance between the median and +2 SD and S_lower x M is half the distance between the median and -2 SD. Used with the CSD method for z-score calculation. These are approximations derived from the charts, not values published by the Fenton authors. See `vignette("wrapper")`, "Gestational-age reference values".

fenton2025_ms_lookup_smoothed.csv

Used in function `cleangrowth()` for GA correction (Step 2b).

References

Fenton TR, Elmrayed S, Alshaikh BN. Fenton Third-Generation Growth Charts of Preterm Infants Without Abnormal Fetal Growth: A Systematic Review and Meta-Analysis. Paediatr Perinat Epidemiol. 2025 Aug;39(6):543-555. doi:10.1111/ppe.70035. Epub 2025 Jun 19. PMID: 40534585; PMCID: PMC12391854.

`fent_foraga` *Approximated Fenton Median Weights by Gestational Age*

Description

A table of approximated Fenton median weights by gestational age, derived from printed Fenton (2025) charts. Maps sex and integer weight (grams, rounded to nearest 10) to the gestational age in days at which that weight is the median. These are approximations derived from the charts, not values published by the Fenton authors. See `vignette("wrapper")`, "Gestational-age reference values".

fent_foraga.csv.gz

Used in function `cleangrowth()` for GA correction (Step 2b).

References

Fenton TR, Elmrayed S, Alshaikh BN. Fenton Third-Generation Growth Charts of Preterm Infants Without Abnormal Fetal Growth: A Systematic Review and Meta-Analysis. Paediatr Perinat Epidemiol. 2025 Aug;39(6):543-555. doi:10.1111/ppe.70035. Epub 2025 Jun 19. PMID: 40534585; PMCID: PMC12391854.

`gc_cli_path` *Path to the growthcleanr command-line driver script*

Description

Returns the location of the installed `gcdriver.R` command-line driver, so it can be invoked with `Rscript` without hunting for the file:

```
DRIVER=$(Rscript -e 'cat(growthcleanr::gc_cli_path())')
Rscript "$DRIVER" input.csv output.csv --quiet
```

Usage

```
gc_cli_path()
```

Value

The absolute path to `gcdriver.R` in the installed package.

See Also

`run_gc_cli`; `vignette("command-line")`.

`gc_partial_run` *Incremental (partial) growthcleanr run*

Description

Re-clean only the subjects whose data changed since a previous `cleangrowth` run, and reuse the earlier cleaned results for everyone else. This is intended for workflows that repeatedly clean a mostly unchanging dataset – for example a simulation loop that perturbs a subset of subjects and re-cleans – where a full re-run would waste most of its work.

Usage

```
gc_partial_run(new_data, cached_output, changed_subjids, ...)
```

Arguments

`new_data`	A long-format `data.frame`/`data.table` of the current data, using growthcleanr's canonical column names (`subjid`, `param`, `agedays`, `sex`, `measurement`) plus a per-row `id`.
`cached_output`	The `data.table` returned by a previous `cleangrowth()` run on the earlier version of this dataset.
`changed_subjids`	Required. A vector of `subjid` values whose data changed since `cached_output` was produced and must be re-cleaned. Any subject present in `new_data` but not in `cached_output` (added) is re-cleaned automatically; any subject in `cached_output` but not in `new_data` (removed) is dropped. May be length 0 (re-clean only added subjects, if any). Values not present in `new_data` are ignored.
`...`	Passed verbatim to the inner `cleangrowth` call on the changed subset (e.g. `checkpoint_dir`, `checkpoint_size`, `batch_size`, `cf_rescue`, output switches). Must match the options used to produce `cached_output`.

Details

`gc_partial_run()` is a thin wrapper: it slices `new_data` down to the changed (and any newly added) subjects, hands that subset to a single ordinary `cleangrowth()` call, and row-binds the freshly cleaned rows to the retained rows of `cached_output`. All arguments in `...` – including `checkpoint_dir`, `checkpoint_size`, `batch_size`, and every output-column switch – are forwarded verbatim to that inner `cleangrowth()` call, so a large changed subset can still be checkpointed. The cache itself is never passed into `cleangrowth()`; the merge happens entirely at the output level.

Why the result is correct. After reference-based recentering, growthcleanr cleans each subject independently of every other subject, so cleaning only the changed subjects yields the exact same rows those subjects would receive in a full run. The combined result is therefore identical to `cleangrowth(new_data, ...)` for a given set of options (see Examples).

Caller responsibilities.

- `...` must specify the *same* options used to produce `cached_output` (same `cf_rescue`, permissiveness, output switches, etc.). If the requested output columns differ from those in `cached_output`, the function stops rather than silently produce `NA`-filled rows.
- `id` must be present and unique per row in both `new_data` and `cached_output`, and must be stable across runs (the same measurement keeps the same `id`). `id` is how retained cached rows and freshly cleaned rows are ordered in the output.
- Input must be long format (one measurement per row). Wide and semi-wide input are not supported by this wrapper.

Value

A `data.table` with the same columns as `cleangrowth()` would return under the same `...`, containing every subject in `new_data`, ordered by `id`.

See Also

`cleangrowth`

Examples

```
# Baseline clean of the full dataset
baseline <- cleangrowth(syngrowth)

# Suppose one subject's data was corrected; re-clean only that subject
changed <- syngrowth$subjid[1]
updated <- gc_partial_run(syngrowth, baseline, changed_subjids = changed)

# Identical to a full re-run (same options), for every subject
full <- cleangrowth(syngrowth)
identical(updated[order(id)]$exclude, full[order(id)]$exclude)
```

---

gc_preload_refs *Pre-load growthcleanr reference closures*

Description

`gc_preload_refs` loads all reference table closures used by `cleangrowth` once, avoiding repeated disk reads across repeated calls (e.g., simulation loops). Pass the result as the `ref_tables` argument to `cleangrowth`.

Usage

```
gc_preload_refs(path = "")
```

Arguments

path	Path to the reference data directory. Defaults to the package extdata directory. Pass an explicit path only if using custom reference files.

Value

A named list with two `read_anthro` closures:

mtz_cdc_prelim CDC-only (used by child algorithm)

mtz_who_prelim WHO+CDC (used by child algorithm)

Examples

```
refs <- gc_preload_refs()
result <- cleangrowth(
  subjid = syngrowth$subjid,
  param  = syngrowth$param,
  agedays = syngrowth$agedays,
  sex = syngrowth$sex,
  measurement = syngrowth$measurement,
  ref_tables = refs
)
```

---

gc_runtrial *Trial run of growthcleanr on a sample of your data*

Description

`gc_runtrial()` helps you confirm that your data are formatted correctly and that growthcleanr is reading your columns, sex coding, and parameters the way you intend – before you commit to a full `cleangrowth` run. It draws a small sample of patients, runs `cleangrowth()` on them with the exact configuration you pass, and prints a short diagnostic report: how the input was interpreted, how sex and parameters were mapped (the two highest-risk settings, shown prominently), anything growthcleanr flagged while reading, and a summary of the cleaning result. The cleaned sample is returned invisibly for inspection. For a ready-to-run script, see "Check your setup first" in `vignette("example-scripts")`.

Usage

```
gc_runtrial(
  data = NULL,
  n = 100,
  sampling = c("random", "first"),
  seed = NULL,
  ...
)
```

Arguments

data	A data.frame/data.table of growth measurements, in the same shape you would pass to `cleangrowth()` (long, semi-wide, or wide – set `input_format` via `...` for the latter two).
n	Number of patients (unique subjects) to sample. Default 100. If the data hold fewer subjects, all are used.
sampling	`"random"` (default) or `"first"` (the first `n` subjects in order of appearance).
seed	Optional integer seed for a reproducible random sample. The seed used is reported; your session's random-number state is left unchanged.
...	Any other `cleangrowth` argument (`subjid`, `param`, `agedays`, `sex`, `measurement`, `input_format`, `sex_coding`, `param_coding`, `semiwide_cols`, `wide_cols`, `dob`, `visit_date`, ...), passed straight through so the trial tests your real configuration.

Details

Sampling is by *patient*: all of a sampled subject's rows are kept, so trajectories stay intact. Random sampling (the default) is effectively free – the only real cost is running growthcleanr on the sampled patients, which is the same whether they were chosen at random or as the first `n`. Pass a `seed` for a reproducible random sample, or `sampling = "first"` for a deterministic peek that needs no seed.

Only the data-frame calling style is supported (pass your table as `data`); vector-style input is not. The trial always analyses and returns results in long format so the diagnostic is well defined, regardless of any `output_format` you set.

If your configuration makes `cleangrowth()` stop (for example a forgotten `input_format`, or a sex coding it will not guess), the trial reports that stop as the headline finding instead of re-throwing it – turning a hard error into setup guidance. The sex and parameter mapping sections still print, since they are derived from your configuration and data rather than the run.

Value

Invisibly, a data.table: the cleaned sample in long format on success (empty on a configuration error), with the structured diagnostic attached as `attr(x, "gc_trial")`. Prints the report as a side effect.

Examples

```
# quick check that a dataset is set up right
gc_runtrial(syngrowth)


## Not run:
# semi-wide data, custom sex coding, reproducible 50-patient sample
# (my_visits stands in for your own semi-wide dataset)
gc_runtrial(my_visits, n = 50, seed = 1,
            input_format = "semi_wide",
            sex_coding = list(male = "M", female = "F"))

## End(Not run)
```

`growthcleanr_app` *Run the growthcleanr Shiny application*

Description

Launches the interactive growthcleanr Shiny interface in a local web browser. The app runs entirely on the user's own machine: data is read into the local R session, cleaned with `cleangrowth()`, and results are saved to the user's computer. Nothing is uploaded anywhere.

Usage

```
growthcleanr_app(...)
```

Arguments

`...`	Additional arguments passed to `shiny::runApp()`.

Details

The app needs the `shiny`, `DT`, `shinyjs`, and `shinyFiles` packages, which growthcleanr imports, so a standard growthcleanr install is enough to run it.

Value

This function is called for its side effect of launching the app and does not return a value; interrupt R to stop the app.

Examples

```
## Not run:
growthcleanr_app()

## End(Not run)
```

growth_cdc_ext_infants *CDC Growth Reference Table*

Description

Part of default CDC-derived tables

Details

Contains CDC LMS parameters and CSD values (median plus half the distance to +/-2 SD) for weight, height, head circumference, and BMI, by age in days and sex.

growthfile_cdc_ext_infants.csv.gz

Used in function `cleangrowth()`

growth_who_ext *WHO Growth Reference Table*

Description

Part of default WHO-derived tables

Details

Contains WHO LMS parameters and CSD values for weight, height, head circumference, and BMI, by age in days (0 to 5 years) and sex.

growthfile_who.csv.gz

Used in function `cleangrowth()`

longwide *Transform data in growthcleanr format into wide structure for BMI calculation*

Description

longwide transforms data from long to wide format. Ideal for transforming output from growthcleanr::cleangrowth() into a format suitable for growthcleanr::ext_bmiz().

Usage

```
longwide(
  long_df,
  id = "id",
  subjid = "subjid",
  sex = "sex",
  agedays = "agedays",
  param = "param",
  measurement = "measurement",
  gcr_result = "exclude",
  include_all = FALSE,
  inclusion_types = c("Include"),
  extra_cols = NULL,
  keep_unmatched_data = FALSE,
  sex_coding = .gc_default_sex_coding,
  param_coding = NULL
)
```

Arguments

long_df	A data frame (or data.table) to be transformed. Expects columns: id, subjid, sex, agedays, param, measurement, and an exclusion column from cleangrowth() (named "exclude" by default; the cleangrowth() output data.table can be passed directly).
id	name of observation ID column
subjid	name of subject ID column
sex	name of sex descriptor column
agedays	name of age (in days) descriptor column
param	name of parameter column to identify each type of measurement
measurement	name of measurement column containing the actual measurement data
gcr_result	name of column of results from growthcleanr::cleangrowth(). Defaults to "exclude", the name of the exclusion column in cleangrowth() output. If your exclusion column has a different name, pass it explicitly. If the named column is missing and the default "exclude" was used, the error says so and suggests gcr_result = "gcr_result" (the column name 2.x scripts used).

include_all
: Determines whether the function keeps all exclusion codes. If TRUE, all exclusion types are kept and the inclusion_types argument is ignored. Defaults to FALSE.

inclusion_types
: Vector indicating which exclusion codes from the cleaning algorithm should be included in the data, given that include_all is FALSE. For all options, see growthcleanr::cleangrowth(). Defaults to c("Include"). A value that is not a growthcleanr version 3 exclusion code (for example the 2.x code "Exclude-Carried-Forward") matches no rows, so longwide() warns and points to the code mapping in vignette("experienced-start-here").

extra_cols
: Vector of additional columns to include in the output. If a column C1 differs on agedays matched height and weight values, then include separate ht_C1 and wt_C1 columns as well as a match_C1 column that gives booleans indicating where ht_C1 and wt_C1 are the same. If the agedays matched height and weight columns are identical, then only include a single version of C1. Defaults to empty vector (not keeping any additional columns).

keep_unmatched_data
: boolean indicating whether to keep height/weight observations that do not have a matching weight/height on that day

sex_coding
: Named list with elements male and female giving the sex codes used in long_df, exactly as for cleangrowth(). The default recognizes the same codes cleangrowth() does (0/1, m/M, f/F, b/g, male/female, boy/girl; case- and whitespace-insensitive). If you passed a sex_coding to cleangrowth(), pass the same one here. Sex is output as M or F, which ext_bmiz() reads directly. As in cleangrowth(), with the default coding longwide() stops if sex has only 1s and 2s, and gives a prominent warning if every value is 1 across more than one subject; passing sex_coding removes both. A warning is also given if any retained row's sex is missing or not recognized (its sex becomes NA, so ext_bmiz() cannot compute BMI z-scores for it).

param_coding
: Optional named list mapping your parameter labels to growthcleanr's canonical parameters, exactly as for cleangrowth() (e.g. list(WEIGHTKG = "wt", HEIGHTCM = "ht")); pass the same one you gave cleangrowth(). Parameters are read the same way cleangrowth() reads them: imperial values (HEIGHTIN, LENGTHIN, WEIGHTLBS) are converted to cm/kg and LENGTHCM is treated as height, so ht and wt are always metric. Head circumference rows are not used; rows with an unrecognized parameter are dropped with a warning.

Value

Returns a data frame transformed from long to wide. Includes only values flagged with indicated inclusion types. Potentially includes additional columns if arguments are passed to extra_cols. For each subject, heights without corresponding weights for a given age (and vice versa) will be dropped unless keep_unmatched_data is set to TRUE. The subject and age columns keep the names you passed as subjid and agedays (for example patid); the other columns are agey, agem, sex (recoded to M or F), ht_id, wt_id, wt (kg), and ht (cm). If no rows are kept, a data frame with these columns and no rows is returned.

Examples

```
# Run on a small subset of given data
df <- syngrowth[syngrowth$subjid %in% unique(syngrowth$subjid)[1:5], ]
gc_out <- cleangrowth(df)
df$exclude <- gc_out$exclude[match(df$id, gc_out$id)]

# Convert to wide format (default gcr_result = "exclude")
wide_df <- longwide(df)

# Include all inclusion types
wide_df <- longwide(df, include_all = TRUE)

# Specify which exclusion codes to retain (default `exclude` column uses
# the Summary codes; pass `exclude_detail = TRUE` to cleangrowth() and use
# `gcr_result = "exclude_detailed"` for the step-level Detailed codes)
wide_df <- longwide(df, inclusion_types = c("Include", "Exclude-CF"))
```

`permissiveness_presets` *Return preset setting values for each permissiveness level*

Description

Returns a named list of four permissiveness levels (loosest, looser, tighter, tightest), each containing the full set of adult algorithm settings for that level. Default level is "looser".

Usage

```
permissiveness_presets()
```

Value

Named list of four lists, each containing setting values

See Also

Section 1D of `vignette("configuration")` for what each level is intended for and the preset values; `vignette("adult-algorithm")` for what each setting does.

Examples

```
presets <- permissiveness_presets()
names(presets)
presets$looser$overall_wt_max
```

rc-reference-medians *Recentering reference medians*

Description

Contains reference median z-scores (column rcmedian.sd) by age in days, sex, and param, used for the default recentering in the child algorithm (sd.recenter = NA). See vignette("recentering").

rcfile-resmoothed.csv.gz

Used in function `cleangrowth()`

read_anthro *Function to calculate z-scores and csd-scores based on anthro tables.*

Description

Function to calculate z-scores and csd-scores based on anthro tables.

Usage

```
read_anthro(path = "", cdc.only = FALSE)
```

Arguments

`path`	Path to supplied reference anthro data. Defaults to package anthro tables.
`cdc.only`	Whether or not only CDC data should be used. Defaults to false.

Value

A function `f(param, agedays, sex, measurement, csd = FALSE)` that returns, for each measurement, its LMS z-score (default) or, when `csd = TRUE`, its CSD ("standard deviation") score against the selected reference (WHO below age 5 years and CDC from age 5 years, or CDC at all ages when `cdc.only = TRUE`; the 2-5 year WHO/CDC blending used by cleangrowth() is not applied here).

Examples

```
# Return calculating function with all defaults
afunc <- read_anthro()

# Return calculating function while specifying a path and using only CDC data
afunc <- read_anthro(path = system.file("extdata", package = "growthcleanr"),
                     cdc.only = TRUE)
```

recode_sex *Recode binary sex variable for compatibility*

Description

recode_sex recodes a binary sex variable for a given source column in a data frame or data table. Only exact matches of sourcem/sourcef are recoded; any other value (e.g. "M"/"F") becomes NA in targetcol without a warning. ext_bmiz() reads only the column named sex, so to prepare data for it use targetcol = "sex" (with targetm = "M", targetf = "F" if the data might have only one sex, since ext_bmiz() accepts numeric sex only when both 1 and 2 are present). Not needed between cleangrowth() and ext_bmiz(): longwide() already outputs sex as M or F.

Usage

```
recode_sex(
  input_data,
  sourcecol = "sex",
  sourcem = "0",
  sourcef = "1",
  targetcol = "sex_recoded",
  targetm = 1L,
  targetf = 2L
)
```

Arguments

input_data	a data frame or data table to be transformed. Expects a source column containing a binary sex variable.
sourcecol	name of sex descriptor column. Defaults to "sex"
sourcem	variable indicating "male" sex in input data. Defaults to "0"
sourcef	variable indicating "female" sex in input data. Defaults to "1"
targetcol	desired name of recoded sex descriptor column. Defaults to "sex_recoded"
targetm	value to use for "male" sex in targetcol. Defaults to 1
targetf	value to use for "female" sex in targetcol. Defaults to 2

Value

Returns a data table with recoded sex variables.

Examples

```
# Run on given data
df <- as.data.frame(syngrowth)

# Run with all defaults
df_r <- recode_sex(df)
```

```
# Specify different targets
df_rt <- recode_sex(df, targetcol = "sexr", targetm = "Male", targetf = "Female")

# Specify different inputs
df_ri <- recode_sex(df_rt, sourcecol = "sexr", sourcem = "Male", sourcef = "Female")
```

`run_gc_cli` *Run growthcleanr from the command line (CSV in, CSV out)*

Description

Testable core behind the `inst/scripts/gcdriver.R` command-line driver. Reads a CSV, runs `cleangrowth` on it (data-frame style), and writes the result to a CSV. Only the options in `opts` are forwarded, so `cleangrowth()`'s own defaults apply to everything else.

Usage

```
run_gc_cli(infile, outfile, opts = list())
```

Arguments

`infile`	Path to the input CSV. Read with `data.table::fread`. Must use the canonical column names (`subjid`, `param`, `agedays`, `sex`, `measurement`, optional `id`) unless remapped via the column-name options in `opts`.
`outfile`	Path to write the cleaned result to, via `data.table::fwrite`. Overwritten if it exists.
`opts`	Named list of arguments forwarded verbatim to `cleangrowth()` (already coerced to the right types). Typically built from command-line flags by the driver; may be supplied directly in tests or from other R code. An empty list runs `cleangrowth()` with all defaults.

Value

The output path, invisibly.

See Also

`cleangrowth`; the command-line driver `inst/scripts/gcdriver.R`.

Examples

```
## Not run:
run_gc_cli("visits.csv", "cleaned.csv",
           opts = list(adult_cutpoint = 18, quietly = TRUE))

## End(Not run)
```

simple_bmi *Compute BMI using standard formula*

Description

simple_bmi Computes BMI using standard formula. Assumes input compatible with output from longwide().

Usage

```
simple_bmi(wide_df, wtcol = "wt", htcol = "ht")
```

Arguments

wide_df	A data frame or data table containing heights and weights in wide format, e.g., after transformation with longwide()
wtcol	name of the weight value column (kg), default 'wt'
htcol	name of the height value column (cm), default 'ht'

Value

Returns a data table with the added column "bmi"

Examples

```
# Simple usage
# Run on a small subset of given data
df <- as.data.frame(syngrowth)
df <- df[df$subjid %in% unique(df[, "subjid"])[1:2], ]
# cleangrowth() returns the data with an 'exclude' column
clean_df <- cleangrowth(df)
# Convert to wide format
wide_df <- longwide(clean_df)
wide_df_with_bmi <- simple_bmi(wide_df)

# Specifying different column names; note that quotes are used
colnames(wide_df)[colnames(wide_df) %in% c("wt", "ht")] <-
  c("weight", "height")
wide_df_with_bmi <- simple_bmi(wide_df, wtcol = "weight", htcol = "height")
```

syngrowth *syngrowth*

Description

A synthetic set of 77,721 measurements from 2,697 subjects generated using Synthea, with measurement errors for testing with growthcleanr. Contains both pediatric and adult data.

Usage

```
syngrowth
```

Format

A data frame with six variables: `id`, `subjid`, `sex`, `agedays`, `param`, and `measurement`

Details

Example electronic health record (heightcm, weightkg) data.

tanner_ht_vel *Tanner Growth Velocity Table*

Description

Part of default reference tables

Details

Contains height velocities from Tanner and Davies (1985)

tanner_ht_vel.csv.gz

Used in function `cleangrowth()`

References

Tanner JM, Davies PS. Clinical longitudinal standards for height and height velocity for North American children. J Pediatr. 1985 Sep;107(3):317-329. doi:10.1016/s00223476(85)805011.

who_hc_maxvel *WHO Maximum Head Circumference Velocity for (3σ)*

Description

Part of default WHO-derived tables

Details

Contains three standard deviations for the World Health Organization values of maximum head circumference velocities.

who_hc_maxvel_3sd_infants.csv.gz

Used in function `cleangrowth()`

References

World Health Organization. WHO Child Growth Standards: Growth velocity based on weight, length and head circumference: Methods and development. Geneva: World Health Organization; 2009. ISBN 9789241547635.

who_hc_vel_3sd *WHO Head Circumference Velocity for (3σ)*

Description

Part of default WHO-derived tables

Details

Contains three standard deviations for the World Health Organization values of head circumference velocities.

who_hc_vel_3sd_infants.csv.gz

Used in function `cleangrowth()`

References

World Health Organization. WHO Child Growth Standards: Growth velocity based on weight, length and head circumference: Methods and development. Geneva: World Health Organization; 2009. ISBN 9789241547635.

who_ht_maxvel *WHO Maximum Height Velocity for (3σ)*

Description

Part of default WHO-derived tables

Details

Contains three standard deviations for the World Health Organization values of maximum height velocities.

who_ht_maxvel_3sd.csv.gz

Used in function `cleangrowth()`

References

World Health Organization. WHO Child Growth Standards: Growth velocity based on weight, length and head circumference: Methods and development. Geneva: World Health Organization; 2009. ISBN 9789241547635.

who_ht_vel_3sd *WHO Height Velocity for (3σ)*

Description

Part of default WHO-derived tables

Details

Contains three standard deviations for the World Health Organization values of height velocities.

who_ht_vel_3sd.csv.gz

Used in function `cleangrowth()`

References

World Health Organization. WHO Child Growth Standards: Growth velocity based on weight, length and head circumference: Methods and development. Geneva: World Health Organization; 2009. ISBN 9789241547635.

Index

∗ **datasets**
    syngrowth, 43

CDCref_d, 5
child_z3_reference, 6
cleangrowth, *3*, 6, *22, 23*, *30–33*, *41*
cleangrowth(), *34*
cleangrowth_checkpoint, 22, *24*
cleangrowth_checkpoint_combine, *23*, 23

ewma, 25
ext_bmiz, 26

fent_foraga, 29
fenton2025_ms_lookup_smoothed, 28

gc_cli_path, 29
gc_partial_run, 30
gc_preload_refs, *16*, 31
gc_runtrial, *6*, 32
growth_cdc_ext_infants, 35
growth_who_ext, 35
growthcleanr *(*growthcleanr-package*)*, 3
growthcleanr-package, 3
growthcleanr_app, *3*, 34

longwide, 36

permissiveness_presets, 38

rc-reference-medians, 39
read_anthro, 39
recode_sex, 40
run_gc_cli, *30*, 41

shiny::runApp(), *34*
simple_bmi, 42
syngrowth, 43

tanner_ht_vel, 43

who_hc_maxvel, 44
who_hc_vel_3sd, 44
who_ht_maxvel, 45
who_ht_vel_3sd, 45