Package: menstrualcycleR (via r-universe)

November 8, 2026

Title Phase-Aligned Cycle Time Scaling (PACTS) and Menstrual Cycle Analysis

Version 1.1.1

Description Implements Phase-Aligned Cycle Time Scaling (PACTS) for menstrual cycle data, placing observations from cycles of differing length onto a common continuous timeline that accounts for variability in ovulation timing. Provides cycle time variables anchored on either menses onset or ovulation, summaries of how ovulation was determined in each cycle, and plotting functions for cycle-aligned outcomes. Methods are described in Nagpal et al. (2025) <doi:10.1016/j.psyneuen.2025.107584>.

URL https://menstrualcycler.clearlabresearch.com/, https://github.com/eisenlohrmoullab/menstrualcycleR

BugReports https://github.com/eisenlohrmoullab/menstrualcycleR/issues

License GPL (>= 3)

Encoding UTF-8

Imports dplyr, ggplot2 (>= 3.4.0), magrittr, purrr, rlang, tidyr, zoo, viridis, glue, lubridate, shiny

NeedsCompilation no

Depends R (>= 3.5.0)

Suggests knitr, rmarkdown, testthat (>= 3.0.0), marginaleffects, mgcv, shinyjs, writexl

Config/testthat/edition 3

VignetteBuilder knitr

LazyData true

Config/roxygen2/version 8.1.0

Author Anisha Nagpal [aut] (ORCID: <https://orcid.org/0000-0001-7111-9322>), Tory Eisenlohr-Moul [aut, cre] (ORCID: <https://orcid.org/0000-0002-9231-3105>)

Maintainer Tory Eisenlohr-Moul <temo@uchicago.edu>

Repository https://cran.r-universe.dev

Date/Publication 2026-11-08 19:51:47 UTC

RemoteUrl https://github.com/cran/menstrualcycleR

RemoteRef HEAD

RemoteSha dcbe01f9836a189bf6ad07e1d79d1d9f91130003

Contents

cycle_plot 2
cycle_plot_individual 4
cycledata 5
cycledata_check 6
cycledata_special 7
launch_app 11
pacts_scaling 11
summary_ovulation 17

Index **20**

`cycle_plot` *Plot Symptom Data Across the Menstrual Cycle*

Description

This function creates a visual representation of symptom data across the menstrual cycle. It calculates mean and deviation values for the symptom by participant, allows centering the visualization on menses or ovulation, and provides flexibility in choosing the y-axis scale for the plot.

Usage

```
cycle_plot(
  data,
  symptom,
  centering = "menses",
  include_impute = TRUE,
  y_scale = "person-centered_roll",
  rollingavg = 5,
  align_val = "center",
  se = FALSE
)
```

Arguments

data	A dataframe containing the input data. Must include columns for scaled cycle day variables and the symptom variable. Scaled cycle day variables are added to your dataframe after applying the pacts_scaling() function. (See examples below)
symptom	A string specifying the symptom variable to analyze.
centering	A string indicating the centering phase of the cycle ("menses" or "ovulation"). Default is "menses".
include_impute	A boolean indicating whether to use imputed cycle time values based on NC cycle length norms. Default is TRUE.
y_scale	A string specifying the y-axis scale. Options are "person-centered", "person-centered_roll", or "means". Default is "person-centered_roll".
rollingavg	A numeric indicating how many days of a rolling average to use, the default is 5
align_val	From the zoo package, using rollapply: alignment of rolling avg. Can be "center", "left", "right"
se	A boolean indicating whether to include a standard error ribbon. Default is False

Value

A list containing the following elements: 1. A dataframe: The input dataframe augmented with person-centered mean (.m) and deviation (.d) values for the symptom variable. 2. A summary dataframe: Contains the summarized y-axis variable values for each 5% increment of the cycle percentage. 3. A ggplot object: A plot visualizing the summarized data across the menstrual cycle.

Examples

```
# Example usage:
# Load the example dataset
cycle_df = cycledata

data_with_scaling <- pacts_scaling(
  cycle_df,
  id = id,
  date = daterated,
  menses = menses,
  ovtoday = ovtoday,
  lower_cyclength_bound = 21,
  upper_cyclength_bound = 35
)

cycle_plot_data <- cycle_plot(
  data_with_scaling,
  "symptom",
  centering = "menses",
  include_impute = TRUE,
  y_scale = "person-centered"
)
cycle_plot_data$plot
```

`cycle_plot_individual` *Generate Cycle-Specific Plots and Summary Data for a Given ID*

Description

This function creates cycle-specific plots and summary statistics for a specified individual (`id`), storing both in a named list for easy access.

Usage

```
cycle_plot_individual(
  data,
  id,
  symptoms,
  centering = "menses",
  y_scale = "person-centered",
  include_impute = TRUE,
  rollingavg = 5
)
```

Arguments

`data`	A dataframe containing menstrual cycle data, including `id` and `cyclenum` columns.
`id`	Numeric id number for the specific individual for whom cycle plots should be generated.
`symptoms`	A vector of strings specifying the symptom variable to analyze that exist in `data`.
`centering`	A string indicating the centering phase of the cycle ("menses" or "ovulation").
`y_scale`	A string specifying the y-axis scale ("person-centered", "person-centered_roll", "raw", or "roll").
`include_impute`	A boolean indicating whether to use imputed cycle time values.
`rollingavg`	A numeric indicating how many days of a rolling average to use, the default is 5

Value

A named list, one element per entry in `symptoms`. Each of those is itself a named list, one element per cycle (named `"Cycle_1"`, `"Cycle_2"`, ...), each containing:

- `$plot`: The cycle-specific ggplot object with the ID displayed
- `$summary`: The corresponding summary data

So for a single symptom `"symptom"` and its first cycle, access the plot as `result[["symptom"]][["Cycle_1"]]$plot`.

Examples

```
cycle_df = cycledata

data_with_scaling <- pacts_scaling(
  cycle_df,
  id = id,
  date = daterated,
  menses = menses,
  ovtoday = ovtoday,
  lower_cyclength_bound = 21,
  upper_cyclength_bound = 35
)

result <- cycle_plot_individual(
  data_with_scaling,
  id = unique(data_with_scaling$id)[1],
  symptoms = "symptom"
)

# the first cycle's plot and summary for "symptom":
result[["symptom"]][["Cycle_1"]]$plot
result[["symptom"]][["Cycle_1"]]$summary
```

---

cycledata *Example longitudinal daily dataset with cycle measures*

Description

This dataset contains daily symptom and cycle information used to test and demonstrate the package.

Usage

```
cycledata
```

Format

A data frame with 619 rows and 5 variables:

id Participant ID

daterated Date of symptom observation

symptom Daily rating of symptom outcome of interest

menses 1 indicating first day of menstruation, menses onset

ovtoday 1 indicating estimated day of ovulation in study, based on biomarkers of ovulation.

Source

Simulated or anonymized data

`cycledata_check` *Analyze Non-Missing Symptom Data by Cycle Phase*

Description

This function provides detailed information about the availability of non-missing symptom data, stratified by cycle phases (luteal and follicular) and overall availability. It also generates visualizations to aid in data quality assessment.

Usage

```
cycledata_check(data, symptom_columns)
```

Arguments

`data`	A data frame containing the input data. • Must include columns for scaled cycle day variables and the symptom or dependent variables of interest.
`symptom_columns`	A character vector of column names representing the dependent variables of interest. For example, `c("symptom1", "symptom2", "symptom3")`, where each name corresponds to a column in your data set.

Value

A list containing three components:

- `by_id`: A data frame summarizing non-missing values for each ID and symptom, including:
  1. Total number of non-missing values.
  2. Number of non-missing values during the luteal phase.
  3. Number of non-missing values during the follicular phase.
- `overall`: A data frame summarizing non-missing values for each symptom, including:
  1. Total number of non-missing values.
  2. Number of non-missing values during the luteal phase.
  3. Number of non-missing values during the follicular phase.
- `data_symptom_plots`: Visualizations showing the availability of non-missing data for each symptom, stratified by ID.

Examples

```
# Load example dataset
cycle_df = cycledata

data_with_scaling <- pacts_scaling(
  cycle_df,
  id = id,
```

```
  date = daterated,
  menses = menses,
  ovtoday = ovtoday,
  lower_cyclength_bound = 21,
  upper_cyclength_bound = 35
)

# Analyze symptom data availability
data_available_info <- cycledata_check(
  data_with_scaling,
  symptom_columns = c("symptom")
)

# View results
print(data_available_info$by_id)
print(data_available_info$overall)
print(data_available_info$data_symptom_plots$symptom)
```

`cycledata_special` *Practice dataset of special cases: one person per setting*

Description

A second, deliberately awkward practice dataset. Where cycledata is an ordinary daily diary, every person here exists to exercise exactly one situation that `pacts_scaling()` has a setting for, or one situation no setting can rescue. The `case` column names which. Use it to see what a setting actually changes before switching it on for your own data, and to recognise a person whose columns come back empty for a reason.

Usage

```
cycledata_special
```

Format

A data frame with 998 rows and 6 variables:

id Participant ID, 1 to 15. One person per case.

daterated Date of observation

symptom Daily rating, 0 to 10, with a perimenstrual peak. `NA` on roughly 8 percent of days, at random, so that coverage figures have to be counted over days that were actually rated.

menses `1` on the first day of menses, `0` otherwise

ovtoday `1` on a biomarker-confirmed day of ovulation, `0` otherwise

case Which special case this person is, in words. **Reads `NA` on rows the package itself adds** – calendar-filled gaps and imputed anchors – which is deliberate: it is the cheapest way to see which rows in your output nobody rated. Carry it across those rows with `dplyr::group_by(id) |> tidyr::fill(case, .direction = "downup")`.

Details

`cycledata` cannot do this job: all 25 of its people have every cycle in range, none was observed before their first recorded menses onset, none has a right-censored luteal phase, and none has a calendar gap. A dataset in which nothing unusual happens cannot show a setting working, and cannot catch a claim about one that is wrong.

The fifteen cases

Figures are from a run at the default settings unless a setting is named. "Days covered" counts rows with a non-missing value, out of all rows for that person after the package fills the calendar.

1. **Ordinary** Three complete cycles, ovulation confirmed, every phase in range. The baseline: without it you cannot tell a setting that recovered nothing from a dataset that is broken.
2. **Eight rated days before the first recorded onset** With `impute_leading_ovulation = TRUE`, `cyclic_time_impute` gains 15 days – but 7 of those rows are ones the package fabricated, because the imputed ovulation lands a week before this diary begins. Gains more rows than data, which is the caution that matters for this setting.
3. **Thirty rated days before the first onset** The same setting gains 15 of those 30 days and fabricates nothing. **The 15 is a placement, not a cap**, which this person is here to show: `leading_ovulation_luteal_days = 25` scales 25 days, `= 40` scales 40 and fabricates 10 rows before the diary begins, and the phase-length caps never gate any of it (a 25-day imputed luteal phase scales with `luteal_phase_max_days` left at its default of 18). So this setting has no upper bound of any kind – not the cycle-length bounds, which need a cycle length there is none of, and not the phase caps. What limits it is the number you pass.
4. **A confirmed ovulation among the leading days** Nothing is imputed (the setting declines when a confirmed ovulation is already there), and those days scale anyway, in the confirmed-only columns, with no setting switched on. The six days before that ovulation have no onset on their left, so no follicular fraction exists for them and they stay empty.
5. **A confirmed ovulation at the end, no closing onset** With `impute_next_menses = TRUE`, the cycle closes 14 days after that ovulation: 14 days are gained in `cyclic_time` as well as `cyclic_time_impute`, and 5 rows are fabricated to reach the imputed onset.
6. **An 18-day cycle, ovulation not confirmed** Shorter than `lower_cyclength_bound`, so no ovulation is imputed and the cycle has no cycle time at all – while the 28-day cycles either side of it do.
7. **A 42-day cycle, ovulation not confirmed** Longer than `upper_cyclength_bound`, same result. Widening the two bounds to `c(18, 43)` imputes an ovulation for this cycle and person 6's, and recovers 58 days between them.
8. **Confirmed ovulation 22 days before the next onset** The luteal phase runs past `luteal_phase_max_days`, inside a 35-day cycle that is still within the cycle-length bounds. `cyclic_time` covers 14 days; `cyclic_time_impute` covers 36, recovering the luteal phase through the phase-cap fallback and flagging 21 of those rows in `cyclic_time_impute_extended_phase`. `luteal_phase_max_days = 25` returns all 22 days to `cyclic_time` itself.
9. **Confirmed ovulation 26 days after the onset** The same, on the follicular side of `follicular_phase_max_days`: 10 days in `cyclic_time`, 36 in `cyclic_time_impute`, 26 flagged. `follicular_phase_max_days = 30` returns all 26.

10. **Confirmed ovulation 5 days before the next onset** Under `luteal_phase_min_days`, and this is the asymmetry: the fallback ignores the phase *ceilings* but still applies the *floors*, so unlike people 8 and 9 these days are empty in every column. Only lowering `luteal_phase_min_days` recovers them.
11. **An eleven-day hole in the diary** Eleven days with no rows at all, swallowing neither an onset nor an ovulation. They come back in the output with no rating on them, which is why a coverage figure counted over output rows overstates what you have.
12. **Ovulation confirmed on no day** Every cycle in range, so every cycle takes the imputed path: `cyclic_time` is empty for this person and `cyclic_time_impute` covers 86 days. The confirmed-versus-imputed contrast, in one person.
13. **No onset and no confirmed ovulation** No anchor of any kind. Nothing scales, under any setting. The case that reads as a bug.
14. **A confirmed ovulation but no onset ever recorded** The menses-to-menses counter never starts, so there is no cycle length – and that one day still scales, pinned at `cyclic_time = 1`, because the ovulation anchor is set independently of any cycle. With `impute_next_menses = TRUE` a closing onset is imputed and the whole luteal phase scales, 15 days, in the confirmed columns too.
15. **Both opt-in rules at once, at opposite ends** Ten rated days before the first onset *and* a confirmed ovulation at the end with no closing onset. The only person who exercises both rules together, and therefore the only one where the order they are applied in could matter. It does not: 15 days at the front, 14 at the back, 29 together, with rows fabricated at both ends.

Overlaps, and where the order of operations matters

`pacts_scaling()` applies `impute_next_menses` **first** and `impute_leading_ovulation` **second**, so the leading rule can see a table that already contains an imputed onset. Four things follow, all measured on this dataset rather than reasoned from the code:

- **The two rules do not interact.** They act on opposite ends of a diary, and on person 15, who has both, the days gained together are exactly the sum of the days each gains alone. The order is therefore not something a caller has to think about.
- **An imputed onset never becomes a leading-day anchor**, and this holds structurally rather than by luck. `impute_next_menses` only ever imputes an onset forward from a confirmed ovulation, so whenever that imputed onset is a person's *first* – person 14, who has no observed onset at all – there is by construction a confirmed ovulation before it, which is precisely the condition on which `impute_leading_ovulation` declines. No combination of the two settings imputes an ovulation before an imputed onset.
- `leading_ovulation_luteal_days` **is ungated in both directions**, which person 3 demonstrates: nothing bounds how many leading days it will scale except the number you pass. Treat a value above the default as a methods decision to record, not a recovery setting to turn up.
- **Persons 8 and 9 depend on a bound they do not demonstrate.** The phase-cap fallback that recovers them is itself gated on the cycle falling inside `[lower_cyclength_bound, upper_cyclength_bound]`, and both their cycles are 35 days – the default upper bound exactly. At `upper_cyclength_bound = 34` their recovered days drop from 72 to 24 and the extended-phase flag from 47 to 0, so they stop demonstrating the phase caps at all. The CLEAR lab's own `[20, 43]` window keeps them working; a narrower one silently changes what they show.

Three overlaps are deliberate and not conflicts. Persons 5 and 14 both exercise `impute_next_menses`, but on 5 the imputed onset closes a cycle that already had an opening onset, while on 14 it creates the only onset that person has. Person 12 is what the whole dataset looks like under `ovtoday = NULL`, which is worth having in one person as well as one call. And person 10 exists to contradict persons 8 and 9: the phase-length floors and ceilings are not symmetric, and only the ceilings have a fallback.

Note

This dataset, its documentation and its automatic checks were drafted with Claude Code and reviewed by the package authors, who are responsible for their content. See the README for the package's full statement on AI tool use.

Source

Generated, not real and not derived from any real record: the anchors are written out by hand and the symptom values come from a fixed random seed. The script that builds it, with the arithmetic for every case asserted as it runs, is `data-raw/cycledata_special.R` in the package sources; `tests/testthat/test_cycledata_special.R` checks that each case still behaves as its label says, so a change to a default that stops a person demonstrating their setting fails a test instead of passing silently.

See Also

cycledata for the ordinary practice dataset, `pacts_scaling()` for the settings each person exercises.

Examples

```
# What each person is, and how much of them scales at the defaults
scaled <- pacts_scaling(
  cycledata_special,
  id = id, date = daterated, menses = menses, ovtoday = ovtoday
)

# Two settings, and the days each one gains
leading <- pacts_scaling(
  cycledata_special,
  id = id, date = daterated, menses = menses, ovtoday = ovtoday,
  impute_leading_ovulation = TRUE
)

# People 2, 3 and 15 gain 15 days each; nobody else changes
gained <- function(x) tapply(!is.na(x$cyclic_time_impute), x$id, sum)
gained(leading) - gained(scaled)

# And the rows the package added, which carry no rating
sum(is.na(leading$symptom)) - sum(is.na(scaled$symptom))
```

launch_app *Launch the Menstrual Cycle Shiny App*

Description

This function launches an interactive Shiny application designed to help users upload and process their menstrual cycle data. The app provides tools to apply Phase-Aligned Cycle Time Scaling (PACTS), generate scaled cycleday variables, and visualize results in a browser interface.

Usage

```
launch_app()
```

Details

Users can upload a `.csv` file, process their data using built-in PACTS functionality, and explore cycle-aligned visualizations to support analysis and interpretation.

Requires three packages that are not installed automatically with menstrualcycleR, because they are needed only for this app and not for `pacts_scaling()` or any other exported function. **shinyjs** and **writexl** are suggested dependencies and install from CRAN; **writexl** backs the app's download buttons. **cpass** is not on CRAN and installs from GitHub with `remotes::install_github("lasy/cpass")`; it powers the app's optional CPASS tab only. `launch_app()` checks for both and reports which are missing rather than failing part-way through.

Value

Called for its side effect of launching the Shiny application. Returns the result of `shiny::runApp()` invisibly; the call blocks until the app is closed. Throws an error, without launching, if any required package is missing or if the app directory cannot be found.

pacts_scaling *Phase-Aligned Cycle Time Scaling (PACTS)*

Description

This function uses the Phase-Aligned Cycle Time Scaling (PACTS) method, as described in Nagpal et al. (2025), to compute continuous, biologically meaningful cycle time variables in a long-format dataset, where each row represents a unique observation date for a specific naturally-cycling individual.

Usage

```
pacts_scaling(
  data,
  id,
  date,
  menses,
  ovtoday = NULL,
  lower_cyclength_bound = 21,
  upper_cyclength_bound = 35,
  impute_next_menses = FALSE,
  next_menses_luteal_days = 14,
  next_menses_max_window = 20,
  impute_leading_ovulation = FALSE,
  leading_ovulation_luteal_days = 15,
  luteal_phase_min_days = 7,
  luteal_phase_max_days = 18,
  follicular_phase_min_days = 8,
  follicular_phase_max_days = 25
)
```

Arguments

`data`	A data frame containing the required input variables.
`id`	A unique identifier for each naturally-cycling individual in the dataset.
`date`	A date column corresponding to the day of observation. Ensure all id-date combinations are unique. You may wish to reclassify post-midnight survey entries to the previous day to maintain alignment with sleep or daily tracking data.
`menses`	A binary column (`0`/`1`) indicating the *first* day of menses onset, where `1` marks onset. All subsequent bleeding days should be set to `0`. Periovulatory spotting should also be excluded.
`ovtoday`	A binary column (`0`/`1`) indicating the day of estimated ovulation. Pass the bare column name. If no ovulation biomarker was collected, pass `ovtoday = NULL`: the column is created as all-`NA` for you, every cycle takes the imputed-ovulation path, and a message says so. Omitting the argument entirely is an error rather than a silent fallback, so that forgetting it when you do have ovulation data cannot quietly produce an all-imputed analysis. Passing a column name that is not in `data` is also an error. Equivalently you may supply your own column of all `0`s or `NA`s – use a column of all `0`s or `NA`s in that case. Accepted ovulation determination methods include: • **Urinary LH surge tests**: Code `ovtoday == 1` on the day *after* the first positive test (LH +1). Specify the brand and threshold (e.g., 40 mIU/ml). • **Basal body temperature (BBT)**: Code `ovtoday == 1` on the day after the BBT nadir (BBT +1). Specify measurement method. • **Daily hormone assays**: See Nagpal et al. (2025) for guidance.
`lower_cyclength_bound`	Numeric lower bound of cycle lengths eligible for ovulation IMPUTATION. Default is 21. Despite the historical argument name, this does not bound which

cycles are scaled: a cycle with a confirmed ovulation is scaled whatever its length, gated by its phase lengths instead (see "Internal phase-length caps" below). **This differs from Nagpal et al. (2025) Section 2.1.1**, which states that cycle time "is computed only for 21-35 day cycles"; that sentence describes the imputation gate, and the package applies the carve-out named in the section's next sentence ("If ovulation is confirmed, users may modify this range") automatically rather than on request. `summary_ovulation()` reports how many cycles fall outside 21-35 days. To exclude out-of-range cycles from an analysis, filter on `mcyclength_complete` after scaling.

`upper_cyclength_bound`
: Numeric upper bound of cycle lengths eligible for ovulation imputation. Default is 35. See `lower_cyclength_bound` for why this does not decide which cycles are scaled.

`impute_next_menses`
: Logical; default `FALSE`. When `TRUE`, opt in to imputing a *next-menses onset* from a biomarker-confirmed ovulation that has no recorded closing menses, so that cycle becomes scalable instead of being dropped for a missing anchor. This is the mirror image of the built-in ovulation imputation (which imputes ovulation *backward* from an observed menses at menses minus 15): here a menses onset is imputed *forward* from a confirmed ovulation, at `ovulation` + `next_menses_luteal_days`. Leaving `FALSE` keeps all previous behavior byte-for-byte identical. Only the general rule is applied; any study-specific gating (for example, blocking imputation across treatment phases or documented off-study breaks, or trust-ordered de-duplication of anchors) is the caller's responsibility to apply on top.

`next_menses_luteal_days`
: Numeric; days after a confirmed ovulation at which to place the imputed next-menses onset when `impute_next_menses = TRUE`. Default 14 (the population-average luteal length; ovulation + 14 is the last follicular day, i.e. the "LH+15" convention).

`next_menses_max_window`
: Not currently used – kept for argument-signature stability, has no effect on the result. See `?impute_next_menses_onsets`, the `max_window` entry, for why: through 0.1.6 this bounded the search for a closing menses, which was a real bug (a genuine luteal phase just past the window got a fabricated onset that overwrote actually-observed data). As of 0.1.7 a real closing menses at any distance always prevents imputation. Passing this argument explicitly (any value) triggers a warning, since it would otherwise be a silent no-op for a caller relying on the old behavior.

`impute_leading_ovulation`
: Logical; default `FALSE`. When `TRUE`, opt in to imputing an ovulation for the days a participant was observed BEFORE their first recorded menses onset (the left-censored start of participation): ovulation is placed `leading_ovulation_luteal_days` days before that first onset, so those leading days scale as the end of a luteal phase in `cyclic_time_impute` / `cyclic_time_imp_ov` (never in the confirmed-only columns). Marked in the new column `ovtoday_leading_impute`; a blank row is added when the imputed day precedes the first observed row. Nothing is imputed when a confirmed ovulation already lies before the first onset. Leaving

FALSE keeps all previous behavior identical. See ?impute_leading_ovulation_anchors.

leading_ovulation_luteal_days
: Numeric; days before the first menses onset at which the leading ovulation is placed when impute_leading_ovulation = TRUE. Default 15 (the package's backward-count convention).

luteal_phase_min_days, luteal_phase_max_days
: Numeric bounds (days) on how long a **confirmed** ovulation's luteal phase (ovulation to next menses) may be for cyclic_lut/cyclic_time/luteal_length to scale it. Defaults 7 and 18, from Bull et al. (2019) norms in 21-35 day cycles. Independent of lower_cyclength_bound/upper_cyclength_bound, which bound the *whole cycle*, not this one phase – widen these directly if your study population has longer or shorter luteal phases than the default norms assume. See the "Internal phase-length caps" section below.

follicular_phase_min_days, follicular_phase_max_days
: Numeric bounds (days) on how long a **confirmed** ovulation's follicular phase (menses to ovulation) may be for cyclic_fol/cyclic_time to scale it. Defaults 8 and 25, from Bull et al. (2019) norms in 21-35 day cycles. Same independence from lower_cyclength_bound/upper_cyclength_bound as the luteal pair above. See the "Internal phase-length caps" section below.

Details

The PACTS method aligns observations across cycles by centering time either on menses onset or ovulation. This allows researchers to model menstrual cycle dynamics as continuous functions of time. Continuous cycle measurement increases statistical precision and power relative to categorical phase coding (Nagpal et al., 2025, Section 1.1). The function requires identification of menses onset (menses) and the estimated day of ovulation (ovtoday), which may be determined via an ovulation biomarker (preferred) or imputed based on typical luteal phase length when unavailable.

When ovulation is not directly assessed, the function imputes ovulation as 15 days prior to the next menses onset (i.e., the last day of the follicular phase), based on the population-average luteal phase length. Imputed ovulation days are recorded in a new binary column, ovtoday_impute.

Reporting how often ovulation was confirmed using an ovulation biomarker versus imputed is important for transparency and scientific rigor. Whenever possible, researchers should use ovulation biomarkers such as LH tests or basal body temperature (BBT). Neither pinpoints the day of ovulation, which would require ultrasound; both indicate that ovulation occurred within 24 to 36 hours (Nagpal et al., 2025, Section 2.1.2). This function supports both confirmed and imputed ovulation, allowing analyses to flexibly account for variable data availability across participants and cycles.

For further guidance on ovulation identification and justification of the -15 day imputation approach, see:

- Nagpal et al. (2025). *Studying the Menstrual Cycle as a Continuous Variable: Implementing Phase-Aligned Cycle Time Scaling (PACTS) with the* menstrualcycleR *package*. *Psychoneuroendocrinology*, 107584. https://doi.org/10.1016/j.psyneuen.2025.107584
- Schmalenberger et al. (2021). *How to study the menstrual cycle: Practical tools and recommendations*. *Psychoneuroendocrinology, 123*, 104895. https://doi.org/10.1016/j.psyneuen.2020.104895

Value

The original data frame with the following additional columns. **Use the four** `cyclic_time*` **columns for analysis** – they are the variables reported in Nagpal et al. (2025) and the ones this package actively maintains. The four `scaled_cycleday*` columns are an earlier, legacy implementation of the same idea, kept only for backward compatibility with existing analysis code; they are numerically similar but *not* identical to their `cyclic_time*` counterparts (different coverage, small value differences even on rows both cover) and should not be treated as interchangeable with them.

- `cyclic_time`: The primary, menses-anchored cycle time variable, centered on menses onset (`menses == 1 -> 0`) and spanning -1 (start of luteal phase) to +1 (ovulation). Only covers cycles with biomarker-confirmed ovulation whose luteal and follicular phases both fall within `[luteal_phase_min_days, luteal_phase_max_days]` / `[follicular_phase_min_days, follicular_phase_max_days]` – see "Internal phase-length caps" below.
- `cyclic_time_impute`: Same as `cyclic_time`, but with broader coverage: falls back to imputed ovulation (`ovtoday_impute`) for cycles without confirmed ovulation, and then to the phase-cap fallback described below for phases that exceed the internal caps. Flagged by `cyclic_time_impute_extended_phase` (below) wherever that last fallback tier supplied the value.
- `cyclic_time_ov`: The ovulation-anchored counterpart of `cyclic_time` – centered on confirmed ovulation (`ovtoday == 1 -> 0`), spanning -1 (start of follicular phase) to +1 (end of luteal phase). Same confirmed-ovulation-only, phase-cap-gated coverage as `cyclic_time`.
- `cyclic_time_imp_ov`: The ovulation-anchored counterpart of `cyclic_time_impute` – same imputed-ovulation and phase-cap fallbacks, flagged by `cyclic_time_imp_ov_extended_phase`.
- `ovtoday_impute`: A binary column indicating imputed ovulation days (value `1`) for cycles without confirmed ovulation, estimated as 15 days before menses onset. Report the confirmed-versus-imputed rate for transparency.
- `cyclic_time_impute_extended_phase`: Binary column (`0`/`1`) marking rows where `cyclic_time_impute` was filled by the phase-cap fallback described below (value `1`), versus the normal confirmed or imputed-ovulation paths (`0`). Report this rate alongside the imputed-ovulation rate for full transparency about coverage.
- `cyclic_time_imp_ov_extended_phase`: The same flag as `cyclic_time_impute_extended_phase`, for the ovulation-centered `cyclic_time_imp_ov` column.
- `mcyclength`: The length, in days, of the menses-to-menses cycle this row belongs to. **On** `cycle_incomplete == 1` **rows this is not a cycle length** – it is the number of days observed so far in a still-open trailing cycle (see `cycle_incomplete` below). Use `mcyclength_complete` instead if you want to threshold cycle length directly.
- `mcyclength_complete`: Same as `mcyclength`, but `NA` whenever `cycle_incomplete == 1`. **Prefer this column for any length restriction** – e.g. `mcyclength_complete >= 21 \& mcyclength_complete <= 35` – since an incomplete cycle's `NA` here can never satisfy a numeric comparison, so it is excluded automatically without a separate `cycle_incomplete == 0` clause.
- `m2mcount`: A forward day count starting from each menses onset (day of onset = 0).
- `cyclenum`: Which complete menses-to-menses cycle, in sequence, this row belongs to, per `id`.
- `cycle_incomplete`: Binary (`0`/`1`); `1` marks a still-open trailing cycle with no closing menses yet observed (unless `impute_next_menses = TRUE` closes it – see below). On these rows `mcyclength` is days-observed-so-far, not a cycle length – `mcyclength_complete` is `NA` instead.

- `luteal_length`: The confirmed-ovulation luteal phase length in days, only populated when it falls within `[luteal_phase_min_days, luteal_phase_max_days]` (`NA` otherwise, including for imputed-ovulation or incomplete cycles).
- `menses_impute`: Present only when `impute_next_menses = TRUE`. A binary column marking menses onsets that were *imputed forward* from a confirmed ovulation (value `1`) versus observed onsets (`0`). Report the imputed-versus-observed onset rate for transparency, just as you would for imputed ovulation.
- `scaled_cycleday`, `scaled_cycleday_impute`, `scaled_cycleday_ov`, `scaled_cycleday_imp_ov`: Legacy variables – see note above. Same conceptual meaning as their `cyclic_time*` counterparts (confirmed-only / imputed-inclusive, x menses-anchored / ovulation-anchored) but computed independently; do not mix the two families within one analysis.

Internal phase-length caps, and how they differ from the cycle-length bounds

a confirmed ovulation's luteal phase (ovulation to next menses) is only scaled by `cyclic_lut`/`cyclic_time` when it falls within `[luteal_phase_min_days, luteal_phase_max_days]` (default 7-18 days), and its follicular phase (menses to ovulation) only within `[follicular_phase_min_days, follicular_phase_max_days]` (default 8-25 days) – both from Bull et al. (2019) norms in 21-35 day cycles. **As of this version these four bounds are caller-adjustable, independently of** `lower_cyclength_bound`/`upper_cyclength_bound`, which bound the whole cycle, not either phase individually. In every version through 0.1.6 they were hardcoded (not arguments at all), so a cycle could fall entirely inside the caller's accepted overall length (e.g. the whole `[20,43]`-day CLEAR lab standard) and still have one phase silently uncovered by `cyclic_time`, because that phase alone exceeded the fixed 18 or 25 day limit with no way to widen it. A caller whose study population has genuinely longer or shorter phases than the Bull et al. norms should now widen these four arguments directly, rather than relying on the fallback described next. `cyclic_time_impute` additionally falls back, for exactly the days a capped phase would otherwise leave with no coverage at all (a confirmed ovulation already suppresses the ordinary imputed-ovulation fallback – see `ovtoday_impute`; and a still-open trailing cycle, `cycle_incomplete == 1`, never gets this fallback either, matching every other imputed column), to the same phase fraction computed *without* the phase-length ceiling (the floor still applies), gated only on the overall cycle falling within the caller's own `[lower_cyclength_bound, upper_cyclength_bound]`. That fallback exists for whatever still runs past even a caller-widened phase bound, trading phase-length plausibility for coverage in the imputed column only – it never touches `cyclic_time` itself. Those filled rows (excluding the ovulation-anchor day itself, whose value is always pinned by a separate, pre-existing override) are flagged in `cyclic_time_impute_extended_phase`. `cyclic_time_imp_ov`, the ovulation-centered sibling of `cyclic_time_impute`, receives the identical fallback, flagged in `cyclic_time_imp_ov_extended_phase`. **This fallback only ever widens coverage, in one direction:** the four bound arguments narrow `cyclic_time`/`cyclic_time_ov` directly, but the fallback ignores `luteal_phase_max_days`/`follicular_phase_max_days` entirely (it checks only the floor and the overall cycle-length bound) – so narrowing either `*_max_days` argument below its default to exclude implausibly long phases from analysis does *not* exclude them from `cyclic_time_impute`/`cyclic_time_imp_ov`, which recover them anyway through this fallback. A caller who wants that stricter behavior in the imputed columns too should additionally filter out rows where `cyclic_time_impute_extended_phase == 1` (or the `_imp_ov` equivalent) after calling this function; there is no argument that disables the fallback itself.

History. These caps existed as hardcoded literals in every released version of this package (v0.1.0 onward; git history traces the specific thresholds back further, to before `cyclic_time` existed at all) until they became adjustable arguments in this version. The published Nagpal et al. (2025) paper

and its supplement do not mention either cap. The package's own "Getting Started" vignette now discloses both caps (Section 3, "Cycle Length Inclusion Criteria"). Both caps, and the fixed-vs-adjustable distinction (as it stood before this version), were stated at https://menstrualcycler.clearlabresearch.com/pacts-explainer.html ("Inclusion defaults") – an automated documentation pass describing the code's actual behavior at the time, not a deliberate methods decision recorded anywhere else. See NEWS.md for the full history.

Examples

```
# Load the example dataset
cycle_df = cycledata

data_with_scaling <- pacts_scaling(
  cycle_df,
  id = id,
  date = daterated,
  menses = menses,
  ovtoday = ovtoday,
  lower_cyclength_bound = 21,
  upper_cyclength_bound = 35
)


# View the result
print(data_with_scaling)
```

---

summary_ovulation *Summarize Ovulation Confirmation and Imputation*

Description

This function provides a summary of how ovulation was identified across the dataset – either through direct confirmation using an ovulation biomarker or via imputation based on menstrual cycle timing.

Usage

```
summary_ovulation(data)
```

Arguments

data	A dataframe containing the output of pacts_scaling() – specifically id, cyclenum, ovtoday, ovtoday_impute, and mcyclength_complete. (See examples below)

Details

Specifically, it counts the number of cycles in which ovulation was:

- **Confirmed** using an ovulation biomarker (i.e., a 1 in the `ovtoday` column), such as urinary LH surge tests or basal body temperature (BBT).
- **Imputed** using the backward-count method, which estimates ovulation as 15 days before the subsequent menses onset (`ovtoday_impute == 1`), based on the typical length of the luteal phase.

The output includes both:

- Overall summary counts across the entire dataset.
- Per-individual summaries to support participant-level quality checks and reporting.

Report this summary in publications: it states how often ovulation timing was measured versus estimated. Ovulation-biomarker confirmation is preferred where available. Note that a positive LH-surge test or a basal body temperature (BBT) nadir does not pinpoint the day of ovulation, which would require ultrasound; it indicates that ovulation occurred within 24 to 36 hours (Nagpal et al., 2025, Section 2.1.2), and the width of that window depends on the detection method.

When ovulation biomarkers are unavailable, the -15 day backward count is recommended over a cycle-midpoint assumption, because the luteal phase varies less in length than the follicular phase. Against 33 hormone-confirmed cycles it was off by a mean absolute 0.97 days (SD 0.88; Nagpal et al., 2025, Section 3.2).

For further guidance on ovulation identification and the rationale for the -15 day imputation method, see:

- Nagpal et al. (2025). *Studying the Menstrual Cycle as a Continuous Variable: Implementing Phase-Aligned Cycle Time Scaling (PACTS) with the* `menstrualcycleR` *package*. *Psychoneuroendocrinology*, 107584. https://doi.org/10.1016/j.psyneuen.2025.107584
- Schmalenberger et al. (2021). *How to study the menstrual cycle: Practical tools and recommendations*. *Psychoneuroendocrinology, 123*, 104895. https://doi.org/10.1016/j.psyneuen.2020.104895

Value

A list with two data frames:

- `ovstatus_total`: A dataset-level summary showing:
  1. The number of cycles with confirmed ovulation (`ovtoday == 1`).
  2. The number of cycles with imputed ovulation (`ovtoday_impute == 1`).
- `ovstatus_id`: A participant-level summary showing, for each unique ID:
  1. The number of cycles whose length falls outside 21-35 days. Descriptive only: a confirmed-ovulation cycle outside that range is still scaled, gated by its phase lengths instead (see `lower_cyclength_bound` in `?pacts_scaling`). Still-open trailing cycles, whose length is not yet knowable, are not counted.
  2. The number of cycles with confirmed ovulation.
  3. The number of cycles with imputed ovulation.

Examples

```
cycle_df = cycledata

data_with_scaling <- pacts_scaling(
  cycle_df,
  id = id,
  date = daterated,
  menses = menses,
  ovtoday = ovtoday,
  lower_cyclength_bound = 21,
  upper_cyclength_bound = 35
)

ov_summary = summary_ovulation(data_with_scaling)
print(ov_summary)
ov_summary$ovstatus_total
ov_summary$ovstatus_id
```

Index

∗ **biomarkers**
summary_ovulation, 17
∗ **covariate**
pacts_scaling, 11
∗ **cycle**
cycledata_check, 6
pacts_scaling, 11
summary_ovulation, 17
∗ **datasets**
cycledata, 5
cycledata_special, 7
∗ **data**
cycledata_check, 6
∗ **imputation**
summary_ovulation, 17
∗ **menstrual**
cycledata_check, 6
pacts_scaling, 11
summary_ovulation, 17
∗ **ovulation**
pacts_scaling, 11
summary_ovulation, 17
∗ **phase**
cycledata_check, 6
pacts_scaling, 11
∗ **quality**
cycledata_check, 6
∗ **scaling**
pacts_scaling, 11
∗ **symptoms**
cycledata_check, 6
∗ **time-varying**
pacts_scaling, 11

cycle_plot, 2
cycle_plot_individual, 4
cycledata, 5, 7, *10*
cycledata_check, 6
cycledata_special, 7

launch_app, 11

pacts_scaling, 11
pacts_scaling(), *10*

summary_ovulation, 17