

Package ‘pregnancy’

May 9, 2026

Title Calculate and Track Dates and Medications During Pregnancy

Version 0.2.1

Description Provides functionality for calculating pregnancy-related dates and tracking medications during pregnancy and fertility treatment. Calculates due dates from various starting points including last menstrual period and IVF (In Vitro Fertilisation) transfer dates, determines pregnancy progress on any given date, and identifies when specific pregnancy weeks are reached. Includes medication tracking capabilities for individuals undergoing fertility treatment or during pregnancy, allowing users to monitor remaining doses and quantities needed over specified time periods. Designed for those tracking their own pregnancies or supporting partners through the process, making use of options to personalise output messages. For details on due date calculations, see <https://www.acog.org/clinical/clinical-guidance/committee-opinion/articles/2017/05/methods-for-estimating-the-due-date>.

License MIT + file LICENSE

URL <https://ellakaye.github.io/pregnancy/>,
<https://github.com/EllaKaye/pregnancy>

BugReports <https://github.com/EllaKaye/pregnancy/issues>

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2.9000

Imports anytime, cli, dplyr, lubridate, rlang

Suggests covr, knitr, rmarkdown, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Depends R (>= 3.5)

VignetteBuilder knitr

NeedsCompilation no

Author Ella Kaye [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0002-7300-3718>)

Maintainer Ella Kaye <ella.kaye@gmail.com>

Repository CRAN

Date/Publication 2026-01-13 09:10:02 UTC

Contents

calculate_due_date	2
calculate_test_date	4
date_when	5
due_date-option	7
how_far	8
medications	9
medications-option	10
medications_remaining	11
person	13

Index **15**

calculate_due_date	<i>Calculate estimated due date and birth period</i>
--------------------	--

Description

Calculates the estimated due date based on a start date and type. The function supports different start date types including last menstrual period (LMP), conception date, and embryo transfer dates. It also provides an estimated birth period, which spans from 37 weeks (birth period start) to 42 weeks (birth period end).

Usage

```
calculate_due_date(
  start_date,
  start_type = c("LMP", "conception", "transfer_day_3", "transfer_day_5",
    "transfer_day_6"),
  cycle = 28
)
```

Arguments

start_date	Date or character string representing a date, e.g. "YYYY-MM-DD". The starting reference date. The interpretation of this date depends on the start_type parameter.
start_type	character. One of: <ul style="list-style-type: none"> • "LMP": Last Menstrual Period date (default) • "conception": Date of conception • "transfer_day_3": Date of day 3 embryo transfer

- "transfer_day_5": Date of day 5 embryo transfer
 - "transfer_day_6": Date of day 6 embryo transfer
- cycle numeric. Length of menstrual cycle in days. Only used when start_type = "LMP". Must be between 20 and 44 days. Defaults to 28 days.

Details

The due date is calculated as follows:

- For LMP: Ovulation is estimated as `start_date + cycle - 14` days, then 266 days are added
- For conception: 266 days are added to the conception date
- For embryo transfers: The appropriate number of days are subtracted to get to conception date (3, 5, or 6 days), then 266 days are added

The birth period start date is 21 days before the due date (37 weeks pregnant), and the birth period end date is 14 days after the due date (42 weeks pregnant).

If `start_date` is a character string, the conversion to a `Date` is handled by `anytime::anydate()`.

Value

Returns a `Date` object representing the estimated due date invisibly. Also prints informative messages showing:

- The estimated due date
- When the birth period begins (37 weeks)
- When the birth period ends (42 weeks)

See Also

- [date_when\(\)](#) for finding dates at specific weeks of pregnancy
- [how_far\(\)](#) for calculating current progress in pregnancy

Examples

```
# Calculate due date from last menstrual period
calculate_due_date("2025-01-31")

# Calculate from conception date
calculate_due_date("2025-02-14", start_type = "conception")

# Calculate from day 5 embryo transfer
calculate_due_date(as.Date("2025-02-19"), start_type = "transfer_day_5")

# Calculate with non-standard cycle length
calculate_due_date("2025-01-31", cycle = 35)
```

calculate_test_date *Calculate pregnancy test date*

Description

Calculates the recommended date for taking a pregnancy test based on a start date and type. The function supports both urine and blood tests, with blood tests typically being viable 2 days earlier than urine tests.

Usage

```
calculate_test_date(
  start_date,
  start_type = c("LMP", "conception", "transfer_day_3", "transfer_day_5",
    "transfer_day_6"),
  cycle = 28,
  test_type = c("urine", "blood")
)
```

Arguments

start_date	Date or character string representing a date, e.g. "YYYY-MM-DD". The starting reference date. The interpretation of this date depends on the start_type parameter.
start_type	character. One of: <ul style="list-style-type: none"> • "LMP": Last Menstrual Period date (default) • "conception": Date of conception • "transfer_day_3": Date of day 3 embryo transfer • "transfer_day_5": Date of day 5 embryo transfer • "transfer_day_6": Date of day 6 embryo transfer
cycle	numeric. Length of menstrual cycle in days. Only used when start_type = "LMP". Must be between 20 and 44 days. Defaults to 28 days.
test_type	character. One of: <ul style="list-style-type: none"> • "urine": Home pregnancy test (default) • "blood": Blood test at clinic

Details

The test date is calculated as follows:

1. First, the ovulation date is calculated (see [calculate_due_date\(\)](#) for details)
2. For urine tests: 14 days are added to the ovulation date
3. For blood tests: 12 days are added to the ovulation date

Blood tests can typically detect pregnancy earlier than urine tests due to their greater sensitivity in detecting hCG hormone levels.

If start_date is a character string, the conversion to a Date is handled by anytime::anydate().

Value

Returns a Date object invisibly representing the recommended test date. Also prints informative messages showing:

- The recommended date for a urine test
- The recommended date for a blood test

See Also

- [calculate_due_date\(\)](#) for calculating the estimated due date

Examples

```
# Calculate test date from last menstrual period
calculate_test_date("2025-01-31")
```

```
# Calculate for blood test from conception date
calculate_test_date(
  start_date = "5023-02-14",
  start_type = "conception",
  test_type = "blood"
)
```

```
# Calculate from day 5 embryo transfer
calculate_test_date(
  as.Date("2025-02-19"),
  start_type = "transfer_day_5"
)
```

```
# Calculate with non-standard cycle length
calculate_test_date("2025-01-31", cycle = 35)
```

date_when

Calculate and display date of specific pregnancy week

Description

Calculate and display date of specific pregnancy week

Usage

```
date_when(weeks, due_date = NULL, person = NULL, today = Sys.Date())
```

Arguments

weeks	Numeric value indicating the number of weeks of pregnancy to calculate the date for.
due_date	Date or character string representing a date, e.g. "YYYY-MM-DD". The expected due date. If NULL, will try to use the "pregnancy.due_date" option. Required if option not set.
person	The person who is pregnant, to determine the grammar for the output message. Can be: <ul style="list-style-type: none"> • "I", "1", "1st", "first", or numeric 1 for first person • "you", "2", "2nd", "second", or numeric 2 for second person • Any other name for third person • NULL: will try to use the "pregnancy.person" option. Defaults to "You" if the option is set.
today	Date or character string representing a date, e.g. "YYYY-MM-DD". Represents the reference date for calculations. Default is Sys.Date(). This parameter exists primarily for testing and documentation purposes and it is unlikely to make sense for the user to need or want to change it from the default.

Details

The function calculates when someone will be/was a specific number of weeks pregnant based on their due date. It handles past, present and future dates appropriately in its messaging. The due date can be provided directly or set globally using options("pregnancy.due_date"). Similarly, the person being referenced can be provided directly or set globally using options("pregnancy.person").

If date_when or today is a character string, the conversion to a Date is handled by `anytime::anydate()`.

Value

Invisibly returns a Date object of when the specified week of pregnancy occurs/occurred/will occur.

Prints messages to the console showing:

- When the specified week of pregnancy occurs/occurred/will occur
- How far in the past/future that date is from today (unless that date is the current date)

See Also

[calculate_due_date\(\)](#) for calculating the due date [set_due_date\(\)](#) for setting the due date as a global option [how_far\(\)](#) for calculating current pregnancy progress

Examples

```
# Set a due date
date_when(20, due_date = "2025-12-01")
date_when(33, due_date = as.Date("2025-12-01"), person = "Sarah")
```

due_date-option	<i>Set or get the pregnancy.due_date option</i>
-----------------	---

Description

Functions to get and set the default due date used throughout the package. This affects calculations in various functions that determine pregnancy progress and timing. Settings persist for the current R session only, unless added to .Rprofile. `set_due_date()` sets the "pregnancy.due_date" option and `get_due_date()` retrieves it.

Usage

```
set_due_date(due_date)
```

```
get_due_date()
```

Arguments

`due_date` A Date or character string representing a date, e.g. "YYYY-MM-DD", specifying the estimated due date, or NULL to unset the option.

Value

Both functions invisibly return the current due date setting:

- `get_due_date()` returns the current setting (a Date object) or NULL if not set
- `set_due_date()` returns the due date value that was set

See Also

- [calculate_due_date\(\)](#) to calculate a due date based on other dates
- [how_far\(\)](#) and other functions that use the due date for calculations

Examples

```
# Store original setting (without messages)
original_due_date <- getOption("pregnancy.due_date")

# Check current setting
get_due_date()

# Set due date and check again
set_due_date("2025-09-15")
get_due_date()

# Restore original setting (without messages)
options(pregnancy.due_date = original_due_date)
```

how_far	<i>Calculate pregnancy progress and time remaining</i>
---------	--

Description

Calculates and displays how far along a pregnancy is on a specific date, including weeks pregnant, days remaining until due date, and overall progress percentage.

Usage

```
how_far(on_date = Sys.Date(), due_date = NULL, person = NULL)
```

Arguments

on_date	Date or character string representing a date, e.g. "YYYY-MM-DD". The date for which to calculate pregnancy progress. Defaults to current system date.
due_date	Date or character string representing a date, e.g. "YYYY-MM-DD". The expected due date. If NULL, will try to use the "pregnancy.due_date" option. Required if option not set.
person	The person who is pregnant, to determine the grammar for the output message. Can be: <ul style="list-style-type: none"> • "I", "1", "1st", "first", or numeric 1 for first person • "you", "2", "2nd", "second", or numeric 2 for second person • Any other name for third person • NULL: will try to use the "pregnancy.person" option. Defaults to "You" if the option is set.

Details

The function assumes a standard pregnancy length of 280 days (40 weeks) when calculating progress. It handles past, present, and future dates appropriately by adjusting message grammar. If the calculation shows more than 42 weeks of pregnancy, a different message is displayed noting this unusual duration.

The function uses the `cli` package for formatted message output and supports proper pluralization of weeks/days in messages.

If `on_date` or `due_date` are character strings, the conversion to a `Date` is handled by `anytime::anydate()`.

Value

Invisibly returns the number of days along in the pregnancy. Prints a formatted message to the console with pregnancy progress information.

Global Options

- `pregnancy.due_date`: Date object setting default due date
- `pregnancy.person`: Character string setting default person

See Also

[set_due_date\(\)](#), [set_person\(\)](#)

Examples

```
# Current progress with explicit due date
# Note that output will depend on date the function is run
how_far(due_date = "2025-12-01")

# Progress on a specific date
how_far(on_date = "2025-11-01", due_date = "2025-12-01")

# With custom person
how_far(on_date = "2025-11-01", due_date = "2025-12-01", person = "Sarah")

# Set global options
date_opt <- getOption("pregnancy.due_date") # save current option
set_due_date("2025-12-01")
how_far()
options(pregnancy.due_date = date_opt) # return original option
```

medications

An example medications table

Description

A data frame with example medications that might be used during fertility treatment/first trimester. It is an example of a data frame that might be used as the `meds` argument to [medications_remaining\(\)](#).

Usage

```
medications
```

```
medications_simple
```

Format

`medications` is a data frame with 14 rows and 5 columns. `medications_simple` is a data frame with 4 rows and 5 columns.

medication Name of the medication

format Format of medication

quantity Number taken per day

start_date Date to start taking the medication

stop_date Final date on which the medication is taken. See details.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 4 rows and 5 columns.

Details

Note that the same medication (prednisolone in this example) has several rows, first because the quantity taken per day changes, then because it needs to be taken on non-consecutive days.

See Also

[medications_remaining\(\)](#)

Examples

```
medications
medications_simple
```

medications-option *Set or get the pregnancy.medications option*

Description

Functions to get and set the default medications data frame used in the [medications_remaining\(\)](#) function. Settings persist for the current R session only, unless added to .Rprofile. [set_medications\(\)](#) sets the "pregnancy.medications" option and [get_medications\(\)](#) retrieves it.

Usage

```
set_medications(meds)
```

```
get_medications()
```

Arguments

meds	Data frame containing medication schedule. Must have the following columns: <ul style="list-style-type: none"> • medication (character/factor): Name of the medication • format (character/factor): Format of the medication (e.g., pill, injection) • quantity (numeric): Number of units to take per day • start_date (Date or character string representing a date, e.g. "YYYY-MM-DD"): Date to start taking the medication • stop_date (Date or character string representing a date, e.g. "YYYY-MM-DD"): Final date on which the medication is taken <p>If NULL, will try to use the "pregnancy.medications" option. Required if option not set.</p>
------	--

Value

Both functions invisibly return the current medications setting:

- [get_medications\(\)](#) returns the current setting (a data frame) or NULL if not set
- [set_medications\(\)](#) returns the medications data frame that was set

See Also

- [medications_remaining\(\)](#), [medications](#)

Examples

```
# Store original setting (without messages)
original_medications <- getOption("pregnancy.medications")

# Set the option
set_medications(pregnancy::medications)

# Get the option
get_medications()

# Restore original setting (without messages)
options(pregnancy.medications = original_medications)
```

medications_remaining *Calculate remaining medications to be taken*

Description

Calculates and displays how many medications remain to be taken as of a specific date, based on a schedule of medications with start and stop dates. Results can be grouped either by medication name or by format (e.g., tablet, injection).

Usage

```
medications_remaining(  
  meds = NULL,  
  group = c("medication", "format"),  
  on_date = Sys.Date(),  
  until_date = NULL  
)
```

Arguments

meds	Data frame containing medication schedule. Must have the following columns: <ul style="list-style-type: none">• medication (character/factor): Name of the medication• format (character/factor): Format of the medication (e.g., pill, injection)• quantity (numeric): Number of units to take per day• start_date (Date or character string representing a date, e.g. "YYYY-MM-DD"): Date to start taking the medication• stop_date (Date or character string representing a date, e.g. "YYYY-MM-DD"): Final date on which the medication is taken
------	--

	If NULL, will try to use the "pregnancy.medications" option. Required if option not set.
group	Character string specifying how to group the results. One of: <ul style="list-style-type: none"> • "medication": Group by medication name (default) • "format": Group by medication format
on_date	Date or character string representing a date, e.g. "YYYY-MM-DD", specifying the date from which to calculate remaining medications. Defaults to current system date
until_date	Date or character string representing a date, e.g. "YYYY-MM-DD", specifying cut-off date for remaining medications. If NULL, defaults to the latest stop_date in medications.

Details

If on_date, until_date start_date or stop_date is a character vector, the conversion to a Date is handled by `anytime::anydate()`.

If any start_date is NA in any row, that row will **not** be counted in the remaining quantities. If any stop_date is NA, it throws an error.

Value

Returns a data frame containing remaining quantities, grouped as specified. Assumes that the function is being called first thing in the day, i.e. before any of on_date's medications have been taken. The data frame has two columns:

- Either 'medication' or 'format' depending on grouping
- quantity: Total number of units remaining

Only medications with remaining quantities > 0 are included.

If no medications remain, a message is printed to the console indicating this, and a data frame with 0 rows is returned invisibly.

Global Options

- `pregnancy.medications`: Data frame setting default medication schedule

See Also

- [set_medications\(\)](#) for setting default medication schedule
- [get_medications\(\)](#) for retrieving current medication schedule
- [medications](#) for an example medications data frame

Examples

```

# Define medications table
#' # Create example medication schedule
meds <- data.frame(
  medication = c("progynova", "prednisolone", "clexane"),
  format = c("tablet", "tablet", "injection"),
  quantity = c(3, 2, 1),
  start_date = c("2025-04-21", "2025-04-26", "2025-05-08"),
  stop_date = c("2025-04-30", "2025-05-07", "2025-09-05")
)

# Calculate remaining medications
medications_remaining(meds, on_date = "2025-04-21")

medications_remaining(meds, group = "format", on_date = "2025-04-21")

# Calculate medications for a specified period
medications_remaining(
  meds = meds,
  on_date = "2025-04-23",
  until_date = "2025-04-30"
)

# Set and use global medications option
#' Store original medications setting (without message)
original_medications <- getOption("pregnancy.medications")
set_medications(pregnancy::medications)
medications_remaining(on_date = as.Date("2025-05-01"))

# Restore original medications setting (without message)
options(pregnancy.medications = original_medications)

```

 person

Set or get the pregnancy.person option for pregnancy-related messages

Description

Functions to get and set the default person used in messages throughout the package. This affects the grammar and pronouns used in various function outputs. Settings persist for the current R session only, unless added to .Rprofile. `set_person()` sets the "pregnancy.person" option and `get_person()` retrieves it.

Usage

```
set_person(person)
```

```
get_person()
```

Arguments

- person The person who is pregnant, to determine the grammar for the output message. Can be:
- "I", "1", "1st", "first", or numeric 1 for first person
 - "you", "2", "2nd", "second", or numeric 2 for second person
 - Any other name for third person
 - NULL: will try to use the "pregnancy.person" option. Defaults to "You" if the option is set.

Value

Both functions invisibly return the current person setting:

- `get_person()` returns the current setting (a character string) or NULL if not set
- `set_person()` returns the person value that was set

See Also

[how_far\(\)](#) and other functions that use the person setting for message formatting

Examples

```
# Store original setting (without messages)
original_person <- getOption("pregnancy.person")

# Check current setting
get_person()

# Set to first person (using string)
set_person("I")
get_person()

# Set to second person (using number)
set_person(2)
get_person()

# Set to a specific name
set_person("Sarah")
get_person()

# Restore original setting (without messages)
options(pregnancy.person = original_person)
```

Index

* datasets

- medications, 9

- calculate_due_date, 2
- calculate_due_date(), 4–7
- calculate_test_date, 4

- date_when, 5
- date_when(), 3
- due_date-option, 7

- get_due_date (due_date-option), 7
- get_medications (medications-option), 10
- get_medications(), 10, 12
- get_person (person), 13

- how_far, 8
- how_far(), 3, 6, 7, 14

- medications, 9, 11, 12
- medications-option, 10
- medications_remaining, 11
- medications_remaining(), 9–11
- medications_simple (medications), 9

- person, 13

- set_due_date (due_date-option), 7
- set_due_date(), 6, 9
- set_medications (medications-option), 10
- set_medications(), 10, 12
- set_person (person), 13
- set_person(), 9