

Package ‘insectecol’

September 27, 2026

Type Package

Title Insect Ecology Data Analysis Toolkit

Version 1.0.1

Description A collection of analytical tools for insect ecology research, currently covering age-stage, two-sex life table analysis and dose-response bioassays. The life table module supports fast batch processing of multi-group datasets, validates raw 'csv' data, computes cohort size, mean fecundity, age-stage survival rates, age-specific survival, age-specific fecundity, life expectancy, and derived population parameters (net reproductive rate, intrinsic and finite rates of increase, mean generation time), simultaneously generates age-stage survival curves for all groups, and exports all tabular results and plots to 'Excel' in a single run. The bioassay module estimates lethal concentrations by the traditional and the weighted (improved) linear regression methods and by probit analysis, with Abbott correction, 95% confidence intervals and chi-square goodness-of-fit tests; the lethal proportion can be set freely (e.g., 25%, 50%, 70% or 90%), so any LC value such as the LC25, LC70 or LC90 can be computed, not only the LC50. The regression plots and tables are exported to 'Excel'. Planned extensions include more insect ecology indicators, such as median lethal temperature/time (LT50) and thermal constants (effective accumulated temperature).

License MIT + file LICENSE

Encoding UTF-8

Imports dplyr, ggplot2 (>= 3.5.0), grid, magrittr, openxlsx, ragg, scales, sysfonts, readr, showtext, tidyr, utils

Config/roxygen2/version 8.1.0

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

URL <https://github.com/SeaGhost-0/insectecol>

BugReports <https://github.com/SeaGhost-0/insectecol/issues>

Depends R (>= 3.5)

NeedsCompilation no
Author Wangyao Li [aut, cre, cph]
Maintainer Wangyao Li <1561941342@qq.com>
Repository CRAN
Date/Publication 2026-09-27 16:10:07 UTC

Contents

build_life_table	2
check_life_table	4
check_path_type	5
default_stage_names	6
get_stage_names	6
lc50_analyze	7
lc50_calculate	9
lifeTable_analyze	10
lifeTable_calculate	12
lifeTable_calculate_all	14
plot_lc50	15
plot_sxj	17
read_lc50	18
read_life_table	20
save_lc50	21
save_lc50_auto	22
save_lc50_plot	23
save_lc50_plot_auto	24
save_results	25

Index	27
--------------	-----------

build_life_table	<i>Build a Life Table Object from User-Supplied Columns</i>
------------------	---

Description

Assembles a life_table object (the same structure returned by [read_life_table](#)) from individual column vectors already loaded into the R session, e.g. after data <- read.csv("XXX.csv"). This is the entry point for analysing data that do not come from a package-conform csv file.

Usage

```
build_life_table(  
  stages,  
  adult_days,  
  sex,  
  oviposition = NULL,
```

```

    stage_names = NULL,
    file_name = "life_table",
    check = TRUE
  )

```

Arguments

stages	Stage-duration columns, one column per immature stage: a data frame, a matrix or a list of equal-length vectors (column j = days spent in immature stage j; blank/NA for stages not reached). If it is a named data frame/list, the names are used as stage names.
adult_days	Numeric vector; adult survival days (NA for individuals that died before the adult stage).
sex	Character/factor vector; F, M or N (died before adult).
oviposition	Optional; daily oviposition records, one column per day: a data frame, a matrix (rows = individuals in the same order as sex) or a single vector (one column). NULL if the reproduction-related parameters should not be computed (see fecundity in lifeTable_calculate_all).
stage_names	Character vector of stage names (length = number of columns of stages). NULL (default) uses the names of stages if it has non-empty names, otherwise default_stage_names .
file_name	Character; data set name (default plot title, base name of the exported xlsx).
check	Logical; validate the data with check_life_table (default TRUE).

Value

A life_table object, ready for all calc_*, plot_sxj and save_results functions.

Examples

```

## The raw example data shipped with the package
f <- system.file("extdata", "Example.csv", package = "insectecol")
## ^^ change to the actual package name
d <- read.csv(f)

## 1) Standard build: column-range subset of stage columns + adult days
##   + sex + oviposition columns (positional indexing is robust to
##   the space-containing headers like "1st instar")
lt1 <- build_life_table(d[2:8], adult_days = d$Adult, sex = d$gender,
  oviposition = d[, 11:17], file_name = "Example")
names(lt1)      # components of the life_table object
head(lt1$df)    # wide table: ID + stages + Adult + gender + oviposition

## 2) Survival analysis only: omit oviposition entirely. Legal since
##   the data checker skips the oviposition check when the table ends
##   at the sex column (use fecundity = FALSE in the analysis).
lt2 <- build_life_table(d[2:8], adult_days = d$Adult, sex = d$gender)

## 3) Named list: the list names become the stage names
lt3 <- build_life_table(list(Egg = d[[2]], "1st instar" = d[[3]],

```

```

      "2nd instar" = d[[4]], "3rd instar" = d[[5]],
      "4th instar" = d[[6]], Prepupa = d[[7]],
      Pupa = d[[8]]),
  adult_days = d$Adult, sex = d$gender,
  oviposition = d[, 11:17])

## 4) Friendly stage names via stage_names: exactly one per IMMATURE
##     stage. The adult labels "Female" and "Male" are appended
##     automatically and must NOT be included.
lt4 <- build_life_table(d[2:8], adult_days = d$Adult, sex = d$gender,
  oviposition = d[, 11:17],
  stage_names = c("Egg", "L1", "L2", "L3", "L4",
    "Prepupa", "Pupa"))

## 5) A common mistake, handled gracefully: stage_names wrongly
##     including the adult labels. The extra two entries are dropped
##     with a warning (only a WARNING - the build still succeeds).
lt5 <- build_life_table(d[2:8], adult_days = d$Adult, sex = d$gender,
  oviposition = d[, 11:17],
  stage_names = c("Egg", "L1", "L2", "L3", "L4",
    "Prepupa", "Pupa",
    "Female", "Male"))

## 6) Skip the consistency check (e.g. oviposition columns
##     deliberately shorter than the adult life span)
lt6 <- build_life_table(d[2:8], adult_days = d$Adult, sex = d$gender,
  oviposition = d[, 11:17], check = FALSE)

```

check_life_table

Validate a Life Table Data Set

Description

Checks the age data (illegal characters, negative values, illegal gaps) and verifies for every female that the number of her oviposition records matches her recorded survival days. If a problem is found, the function stops and reports the exact row and column positions of all offending cells.

Usage

```
check_life_table(lt)
```

Arguments

`lt` A `life_table` object returned by `read_life_table`.

Details

Two kinds of problems are detected:

- **age data errors:** cells in the stage-duration and survival-day columns that contain illegal characters or negative values, or illegal gaps within a sequence of values;
- **oviposition errors:** for every female row, the number of non-empty daily fecundity cells must equal the adult survival days recorded in column $n - 1$.

The check runs automatically inside `read_life_table` unless `check = FALSE` is used there.

Value

`invisible(TRUE)` if no error is found; otherwise the function stops with a detailed error message.

See Also

`read_life_table`

check_path_type	<i>Check the Type of an Input Path</i>
-----------------	--

Description

Checks whether the input path points to a folder or to a file. Before the check, the path is cleaned automatically: backslashes are converted to forward slashes, invisible characters that are often copied along with paths from Windows dialogs (U+202A) are removed, and a redundant trailing "/" is dropped.

Usage

`check_path_type(path)`

Arguments

path	Character string; the path to check. Anything that is not a single, non-missing, non-empty character string is treated as invalid.
------	--

Details

The function never stops: it always returns one of the four type labels below, so callers can branch directly on the result. The cleaned path is used for the existence checks, which makes the function robust against paths copied out of the Windows Explorer address bar. It is used internally by the reading functions of the package to support both folder input (batch mode) and single-file input.

Value

A character string, one of

"folder"	the path exists and is a folder
"csv file"	the path exists, is a file and has the extension csv
"other file"	the path exists and is a file with another extension
"invalid path"	the path does not exist or is not a valid path string

See Also

[read_lc50](#)

Examples

```
check_path_type(tempdir())
check_path_type(system.file("extdata", "bioassay.csv", package = "insectecol"))
```

default_stage_names	<i>Default Developmental Stage Names</i>
---------------------	--

Description

Generates the default stage names used throughout the package: "Egg" followed by "1st instar", "2nd instar", ... one entry per immature stage. The adult labels "Female"/"Male" are appended automatically by [get_stage_names](#) and need not be included here.

Usage

```
default_stage_names(k)
```

Arguments

k Integer; number of immature stages (≥ 1).

Value

A character vector of length k.

Examples

```
default_stage_names(4) # "Egg", "1st instar", "2nd instar", "3rd instar"
```

get_stage_names	<i>Extract the Developmental Stage Names from the Header</i>
-----------------	--

Description

Extracts the names of the developmental stages from the header (first row) of a life table csv file. The header is truncated at the sex column, the ID column is dropped, and the two adult labels before the sex column are replaced by "Female" and "Male".

Usage

```
get_stage_names(lt)
```

Arguments

`lt` A `life_table` object returned by [read_life_table](#).

Details

Stage names are taken from the header exactly as they are spelled in the csv file, so English headers produce English stage names and Chinese headers produce Chinese stage names. Only the two adult labels are always converted to "Female" and "Male". Internally the sex column is located by matching the header entry gender and the ID column by the header entry ID (both are fixed parts of the csv template).

Value

A character vector of stage names, e.g. `c("Egg", "Larva", "Pupa", "Female", "Male")`; used as column names by [calc_sxj](#) and as legend labels by [plot_sxj](#).

See Also

[read_life_table](#), [calc_sxj](#), [plot_sxj](#)

Examples

```
lt <- read_life_table(system.file("extdata", "Example.csv",
                                package = "insectecol"))
get_stage_names(lt)
```

lc50_analyze

Analyse Bioassay Data for LC Estimation (Main Function)

Description

Non-interactive, fully parameter-driven entry point for the LC (lethal concentration) analysis. It (1) assembles the standardised data list from a data frame, a named list of data frames or three parallel vectors, (2) computes the LC estimates with the selected method(s) via [lc50_calculate](#) and (3) optionally builds the regression plot(s) in the style of [plot_lc50](#). Nothing is written to disk and no dialog is opened; export is handled separately by [save_lc50](#) / [save_lc50_plot](#).

Usage

```
lc50_analyze(
  d = NULL,
  concentration = NULL,
  tested = NULL,
  dead = NULL,
  name = "bioassay",
  lc = 0.5,
  method = "traditional",
```

```

plot = FALSE,
plot_method = NULL,
font = "TNM",
unit = NULL,
shape = c("sigmoid", "linear"),
ci = TRUE,
ci_level = 0.95,
error_bar = TRUE,
move_thres = 0.5,
lc_ci = TRUE,
lc_p = TRUE,
lc_lab_gap = 0.35,
lc_lab_gap_right = 0.1,
lc_lab_dy = 0.1,
lc_lab_lh = 1.05
)

```

Arguments

d	Optional; the bioassay data: a data frame with the columns Concentration, Tested and Dead (headers are matched loosely, as in read_lc50 , so a header like "Concentration (mg/L)" works), a named list of such data frames (e.g. the return value of read_lc50), or NULL to build the data from the three vectors below.
concentration, tested, dead	Numeric vectors; the concentration, the number of insects tested and the number of dead insects, one entry per concentration group (replicates = repeated values). Used only when d is NULL.
name	Character; the data set name used in the results and the saved plot file names when d is a single data frame or the vectors are used (default "bioassay"); ignored for a named list input.
lc	Numeric; the lethal proportion (default 0.5 = LC50, e.g. 0.9 = LC90), passed to lc50_calculate .
method	Character; one or several of "traditional", "improved", "probit" or "all", passed to lc50_calculate .
plot	Logical; whether to build the regression plot(s) (default FALSE). The ggplot objects are only returned - not printed, not saved.
plot_method	Character; which of the computed methods to plot (default NULL = the first method that succeeded). Ignored when plot = FALSE.
font, unit, shape, ci, ci_level, error_bar, move_thres, lc_ci, lc_p, lc_lab_gap, lc_lab_gap_right, lc_lab_dy, lc_lab_lh	Plot settings, passed to the internal plot engine exactly as in plot_lc50 (unit = NULL means "mg/L", unit = "" shows no unit).

Value

A list with elements data (the standardised data list, one data frame per data set), results (the list returned by [lc50_calculate](#): results, summary_df, lc) and plot (a named list of ggplot objects

when plot = TRUE, otherwise NULL).

See Also

[read_lc50](#), [lc50_calculate](#), [plot_lc50](#), [save_lc50](#)

Examples

```
## way 1: data frame straight from the package example csv
f <- system.file("extdata", "bioassay.csv", package = "insectecol")
out1 <- lc50_analyze(read_lc50(f), method = "probit")
out1$results$summary_df

## way 2: three parallel vectors, no csv involved; all three methods
conc <- c(0, 1.5, 3, 6, 12, 24)
n    <- c(120, 60, 60, 60, 60, 60)
dead <- c(7, 9, 18, 32, 48, 57)
out2 <- lc50_analyze(concentration = conc, tested = n, dead = dead,
                     name = "trial1", method = "all")
out2$results$summary_df

## way 3: LC90, improved regression, plot on the linear axis
out3 <- lc50_analyze(concentration = conc, tested = n, dead = dead,
                     name = "trial1", lc = 0.9, method = "improved",
                     plot = TRUE, plot_method = "improved",
                     shape = "linear")
out3$plot$trial1      # ggplot object: print(), customise or export
```

lc50_calculate	<i>Batch Calculation of LC Values</i>
----------------	---------------------------------------

Description

Computes the LC estimates for every data set read by [read_lc50](#), using the selected estimation method(s), and returns both the detailed per-file results and a summary data frame.

Usage

```
lc50_calculate(lcd, lc = 0.5, method = "traditional")
```

Arguments

lcd	The named list returned by read_lc50 .
lc	Numeric; the lethal proportion for which the concentration is estimated. The default 0.5 gives the LC50, 0.9 the LC90.
method	Character string or vector; the estimation method(s) to use: one or several of "traditional" (traditional linear regression, the default), "improved" (improved linear regression) and "probit" (probit analysis), case-insensitive, or "all" for all three in a single call. The full English method names are accepted as well.

Details

The selected method(s) are applied to every data set; the default is the traditional linear regression. The summary data frame gets one row per file and method, and the progress log reports the status of every method for every file. A file that fails (e.g. because too few valid concentrations remain after the Abbott correction) does not interrupt the batch: the error message is recorded in the Equation column of the summary data frame instead.

Value

A list with elements	
results	nested list: one element per file, each holding one element per selected method with its result list (or the error message)
summary_df	data frame with one row per file and method: estimate, 95 goodness-of-fit
lc	the lethal proportion used

References

Finney, D. J. (1971) *Probit Analysis*, 3rd edition. Cambridge University Press, Cambridge.

See Also

[read_lc50](#), [lc50_traditional](#), [lc50_improved](#), [lc50_probit](#), [plot_lc50](#), [save_lc50](#)

Examples

```
f <- system.file("extdata", "bioassay.csv", package = "insectecol")
res <- lc50_calculate(read_lc50(f), lc = 0.7)      # LC70
res$summary_df
lc50_calculate(read_lc50(f), method = "all")$summary_df  # all methods
```

lifeTable_analyze	Analyse a Life Table from User-Supplied Columns (Main Function)
-------------------	---

Description

Non-interactive, fully parameter-driven entry point for the age-stage, two-sex life table analysis. It (1) builds a life_table object from column vectors of an already loaded data frame (e.g. after data <- read.csv("XXX.csv"), or accepts a ready life_table object), (2) computes the life table parameters and (3) optionally draws the age-stage survival curves with customisable title, axis titles and legend labels. Nothing is written to disk; export is handled separately by [save_results](#).

Usage

```
lifeTable_analyze(
  lt = NULL,
  stages = NULL,
  adult_days = NULL,
  sex = NULL,
  oviposition = NULL,
  stage_names = NULL,
  file_name = "life_table",
  check = TRUE,
  fecundity = TRUE,
  plot = FALSE,
  title = NULL,
  x_title = "Age(days)",
  y_title = "Age-Stage Survival Rate(Sxj)",
  legend_labels = NULL,
  dpi = 300
)
```

Arguments

<code>lt</code>	Optional; an existing <code>life_table</code> object (from read_life_table or build_life_table). If NULL (default), the object is built from <code>stages</code> , <code>adult_days</code> , <code>sex</code> and <code>oviposition</code> .
<code>stages</code> , <code>adult_days</code> , <code>sex</code> , <code>oviposition</code> , <code>stage_names</code> , <code>file_name</code> , <code>check</code>	Passed to build_life_table (ignored when <code>lt</code> is supplied).
<code>fecundity</code>	Logical; whether to compute the reproduction-related parameters (F, F _{xj} , m _x , R ₀ , r, lambda, T). FALSE skips them entirely - oviposition is then not required at all and may be left NULL.
<code>plot</code>	Logical; whether to draw the age-stage survival curves (default FALSE). The returned ggplot object can be printed, customised further or passed to save_results .
<code>title</code>	Character; plot title. NULL = <code>file_name</code> .
<code>x_title</code> , <code>y_title</code>	Character; axis titles. Defaults "Age(days)" and "Age-Stage Survival Rate(Sxj)".
<code>legend_labels</code>	Character vector; legend labels, one per stage (immature stages + Female + Male), e.g. <code>c("Egg", "1st instar", "Pupa", "Female", "Male")</code> . NULL (default) = the stage names of the data (Egg, 1st instar, 2nd instar, ..., Female, Male).
<code>dpi</code>	Numeric; resolution used for scaling the text of the plot (default 300).

Value

A list with components `lt` (the `life_table` object), `results` (the list returned by [lifeTable_calculate_all](#)) and `plot` (the ggplot object when `plot` = TRUE, otherwise NULL).

See Also

[build_life_table](#), [lifeTable_calculate_all](#), [plot_sxj](#), [save_results](#)

Examples

```
## The example raw data shipped with the package (the same layout as
## the csv template: ID + immature stage columns + Adult + gender +
## one column per oviposition day of the females)
f <- system.file("extdata", "Example.csv", package = "insectecol")
## ^^ change "lifeTable" to the actual package name
d <- read.csv(f)
names(d) # with check.names = TRUE (default) the names become
# ID, Egg, X1st.instar, X2nd.instar, ..., Prepupa, Pupa,
# Adult, gender, ...

## --- way 1: pass a column-range subset of the data frame
## (positional indexing: works regardless of how the names were mangled)
out1 <- lifeTable_analyze(stages = d[2:8], adult_days = d$Adult,
                        sex = d$gender, oviposition = d[, 11:17],
                        file_name = "Example - way 1")
out1$results$N # number of individuals
out1$results$Summary # all life table parameters

## --- way 2: pass a named list of single columns
## (the list names become the stage names in plots and results)
out2 <- lifeTable_analyze(stages = list(Egg = d[[2]], "1st instar" = d[[3]],
                                       "2nd instar" = d[[4]], "3rd instar" = d[[5]],
                                       "4th instar" = d[[6]], Prepupa = d[[7]],
                                       Pupa = d[[8]]),
                        adult_days = d$Adult, sex = d$gender,
                        fecundity = FALSE) # survival analysis only,
                                       # oviposition not supplied
out2$results$N

## --- way 3: select the stage columns by their original names
## (re-read with check.names = FALSE to keep "1st instar", "2nd instar", ...)
d3 <- read.csv(f, check.names = FALSE)
out3 <- lifeTable_analyze(stages = d3[, c("Egg", "1st instar", "2nd instar",
                                       "3rd instar", "4th instar",
                                       "Prepupa", "Pupa")],
                        adult_days = d3$Adult, sex = d3$gender,
                        oviposition = d3[, 11:17],
                        stage_names = c("Egg", "L1", "L2", "L3", "L4",
                                       "Prepupa", "Pupa"),
                        plot = TRUE,
                        legend_labels = c("Egg", "L1", "L2", "L3", "L4",
                                       "Prepupa", "Pupa",
                                       "Female", "Male"))
out3$plot # print or further customise the ggplot object
```

Description

Runs the complete workflow (reading, validation, calculation, plotting and exporting) for every csv file in a folder, or for a single csv file. Each data set gets its own Excel workbook; in addition an `all.xlsx` with the summary of all files is created. Files that fail (e.g. because of data errors) are skipped and reported at the end without interrupting the remaining files.

Usage

```
lifeTable_calculate(  
  path,  
  output_path = NULL,  
  plot = TRUE,  
  keep_tiff = FALSE,  
  dpi = 300  
)
```

Arguments

<code>path</code>	Character; the data path: a folder (all csv files inside are analysed) or a single csv file. The type of the path is determined by <code>check_path_type()</code> .
<code>output_path</code>	Character; the export folder. Defaults to the parent folder of the csv file (single-file mode) or the data folder itself (folder mode).
<code>plot</code>	Logical; whether the age-stage survival curves are generated and embedded into the workbooks (default TRUE).
<code>keep_tiff</code>	Logical; whether to keep the standalone tiff files (default FALSE).
<code>dpi</code>	Numeric; resolution of the exported images (default 300).

Value

A summary data frame with one row per successfully analysed file (population parameters as columns); the attribute `error_files` contains the names of the files that failed.

See Also

[read_life_table](#), [lifeTable_calculate_all](#), [plot_sxj](#), [save_results](#)

Examples

```
f <- system.file("extdata", "Example.csv", package = "insectecol")  
lifeTable_calculate(f, output_path = file.path(tempdir(), "insectecol-demo"))
```

```
lifeTable_calculate_all
```

Calculate All Parameters of One Life Table

Description

Convenience wrapper that computes all life table parameters of one data set in a single call: cohort size, mean fecundity, the age-stage survival rates, the age-specific rates and the derived population parameters (R_0 , r , λ , T).

Usage

```
lifeTable_calculate_all(lt, fecundity = TRUE)
```

Arguments

<code>lt</code>	A <code>life_table</code> object returned by read_life_table .
<code>fecundity</code>	Logical; whether to compute the reproduction-related parameters (F , F_{xj} , m_x , R_0 , r , λ , T). <code>FALSE</code> skips them entirely; no oviposition data are then required.

Details

The intermediate results are passed on internally, so nothing is computed twice: s_{xj} first, then l_x , F_{xj} and m_x , then r and R_0 , and finally $\lambda = \exp(r)$ and $T = \log(R_0) / r$.

Value

A named list with elements N , F , s_{xj} , l_x , f_{xj} , m_x , e_x , R_0 , r , λ and T . With `fecundity = FALSE` (or when no oviposition data are supplied), f_{xj} and m_x are `NULL` and F , R_0 , r , λ , T are `NA_real_`.

See Also

The individual `calc_*` functions; [lifeTable_calculate](#) for the batch workflow.

Examples

```
f <- system.file("extdata", "Example.csv", package = "insectecol")
lt <- read_life_table(f)
results <- lifeTable_calculate_all(lt)
results$R0
lifeTable_calculate_all(lt, fecundity = FALSE)$N
```

plot_lc50

*LC50 Regression Plots***Description**

Plots every data set: observed points, the fitted curve of the computed method with its pointwise confidence band, and dashed reference lines marking the LC estimate, which is itself marked by a circle where it lies on the fitted curve. By default the concentration axis is on a log10 scale, which gives the classical symmetric S-shaped curve; shape = "linear" restores the original linear axis.

Usage

```
plot_lc50(
  results,
  save_path = NULL,
  font = "TNM",
  width = 7,
  height = 6,
  dpi = 300,
  unit = NULL,
  shape = c("sigmoid", "linear"),
  ci = TRUE,
  ci_level = 0.95,
  error_bar = TRUE,
  move_thres = 0.5,
  method = NULL,
  lc_ci = TRUE,
  lc_p = TRUE,
  lc_lab_gap = 0.35,
  lc_lab_gap_right = 0.1,
  lc_lab_dy = 0.1,
  lc_lab_lh = 1.05
)
```

Arguments

results	Result list of lc50_calculate .
save_path	Folder for the png files; NULL (default) displays the plots only.
font	Font family (default "TNM").
width, height	Figure size in inches (default 7 x 6).
dpi	Resolution of the saved files (default 300); at any dpi the figures keep the physical size they have at 300 dpi.
unit	Unit of the concentration (e.g. "mg/L"), used in the LC label and the x-axis title. NULL (the default) is treated as "mg/L"; pass "" to show no unit at all.
shape	"sigmoid" (default): log10 concentration axis, the symmetric S-shaped dose-response curve. "linear": the original linear concentration axis.

ci	Logical (default TRUE): draw the pointwise confidence band of the fitted curve.
ci_level	Confidence level of the curve band and of the replicate error bars (default 0.95).
error_bar	Logical (default TRUE): replicate rows of the same concentration are pooled to a single point, the Abbott-corrected $\text{sum(Dead)} / \text{sum(Tested)}$ (equal to the replicate mean when the replicate groups are of equal size), with a Wilson score interval at ci_level as the error bar, clipped to [0, 1]. FALSE draws every raw row as a plain point (the previous behaviour).
move_thres	Numeric (default 0.5). A dashed reference line that does not land on a regular tick normally gets an extra tick whose value is labelled next to the axis like a regular tick. If the LC position is at most move_thres regular tick spacings away from the nearest tick (measured on the display axis, i.e. log10 concentrations for shape = "sigmoid"), that label would overlap the neighbouring tick label, so the value is drawn inside the panel instead: the concentration just above the x axis to the right of the vertical dashed line, the mortality just right of the y axis above the horizontal dashed line (each flips to the other side of its dashed line when it would not fit). 0 disables the move; with evenly spaced ticks 0.5 moves every value that is not midway between two ticks.
method	Character scalar, which methods to plot: a subset of c("traditional", "improved", "probit"), or "all" (default) for every method present in the results object.
lc_ci	Logical (default TRUE): show the 95 interval of the LC estimate as a second line of the LC reference label, e.g. (0.98-1.55) below LC50 = 1.23 mg/L. FALSE omits the line.
lc_p	Logical (default TRUE): append the chi-square goodness-of-fit result (chi-square statistic and P value) as an additional line of the LC reference label. FALSE omits the line.
lc_lab_gap	Clearance between the vertical reference line and the near edge of the LC label when the label sits LEFT of the line, in text widths of the label itself (default 0.35). Larger pushes the label further away from the line; smaller moves it towards it.
lc_lab_gap_right	The same clearance when the label sits RIGHT of the line (default 0.1, smaller than lc_lab_gap because the label then hangs below the crossing, where a smaller gap keeps it closer to the reference line).
lc_lab_dy	Clearance between the LC label block and the LC crossing, in y-axis units (default 0.1): the distance from the crossing to the edge of the block that faces it. The block is placed in the diagonal quadrant the fitted curve never enters (above the crossing when the label sits left of the vertical reference line, below it when the label sits right), anchored by that facing edge, so adding lines or changing lc_lab_lh grows the block away from the crossing and never onto the dashed reference line. Larger moves the whole block further from it.
lc_lab_lh	Line spacing of the LC label in multiples of its font size (1 = single spacing, default 1.05). The lines are spaced evenly whichever of them lc_ci / lc_p switches on.

Details

Replicates of the same concentration are pooled and drawn as the Abbott-corrected pooled mortality with Wilson score intervals. The LC reference label is centred around the crossing of the two dashed reference lines. Because the sigmoid only ever passes through the lower-left and upper-right quadrants around that crossing, the label is placed in one of the two free ones: above the crossing when it sits left of the vertical reference line, below it when it sits right, at a clearance of `lc_lab_dy`.

Value

Named list of ggplot objects (invisibly).

See Also

[save_lc50](#), [save_lc50_plot](#)

Examples

```
f <- system.file("extdata", "bioassay.csv", package = "insectecol")
res <- lc50_calculate(read_lc50(f))
plots <- plot_lc50(res, save_path = tempdir())
plots <- plot_lc50(res, shape = "linear", save_path = tempdir()) # original axis
plots <- plot_lc50(res, ci = FALSE, error_bar = FALSE,
                  save_path = tempdir()) # bare version
```

plot_sxj

Age-Stage Survival Rate Curves

Description

Draws the age-stage survival rate $s(x,j)$ of every developmental stage (including the female and male adults) against age in days, in the style of the classical TWOSEX-MSChart plots.

Usage

```
plot_sxj(
  lt,
  sxj = NULL,
  title = NULL,
  x_title = "Age(days)",
  y_title = "Age-Stage Survival Rate(Sxj)",
  legend_labels = NULL,
  dpi = 300
)
```

Arguments

lt	A life_table object returned by read_life_table .
sxj	Optional; the result of calc_sxj . Supplying it avoids recomputing the age-stage survival rates.
title	Character; plot title. Defaults to the name of the csv file.
x_title	Character; x axis title (default "Age(days)").
y_title	Character; y axis title (default "Age-Stage Survival Rate(Sxj)").
legend_labels	Character vector; legend labels, one per stage (immature stages plus Female and Male), e.g. c("Egg", "1st instar", "Pupa", "Female", "Male"). NULL (default) uses the stage names of the data.
dpi	Numeric; resolution (default 300). Only influences the scaling of the graphical elements (title, axis labels, legend), so that the plot looks identical at 300 and 600 dpi.

Details

To keep the figure clean, each stage is drawn only over the age window in which it actually occurs (extended by two days on both sides). Immature stages are drawn as coloured dots connected by a thin line; females are grey and males black, both with diamond points. By default all text of the figure is in English; title, axis titles and legend labels can be customised.

The text sizes are calibrated for being drawn while showtext is active at its default internal dpi (96); [save_results](#) takes care of this when exporting. If you save the plot yourself, switch showtext on around the [ggsave](#) call, otherwise the text comes out about 300/96 times too large.

Value

A ggplot object that can be customised further or saved with [ggsave](#).

See Also

[calc_sxj](#), [save_results](#)

Examples

```
f <- system.file("extdata", "Example.csv", package = "insectecol")
p <- plot_sxj(read_life_table(f))
```

read_lc50

Read Bioassay Data for LC Estimation

Description

Reads the raw csv files of a dose-response bioassay (one file per insecticide, population or similar) and returns a list of standardised data frames that the LC functions of the package work with.

Usage

```
read_lc50(path = NULL)
```

Arguments

path	Character string; the data path: a folder (all csv files inside are read) or a single csv file. If NULL (the default), a folder selection dialog is opened.
------	---

Details

Each csv file must contain one row per concentration with three required columns: the concentration, the number of insects tested and the number of dead insects. The column names are matched loosely against the fixed keywords of the csv template, so headers with additional text such as units (e.g. a concentration header with "(mg/L)" appended) are recognised as well. Rows with a concentration of zero are treated as the control group and are used for the Abbott correction during the analysis.

The data are standardised and validated while reading: rows with non-numeric or missing entries are dropped, the number of tested insects must be positive, the number of dead insects must lie between zero and the number of tested insects, at least one concentration greater than zero must be present, and the rows are sorted by increasing concentration. The file encoding is detected automatically (UTF-8 with BOM and GBK are tried), so files written by both English and Chinese versions of Excel can be read.

Value

A named list with one data frame per csv file; the list elements are named after the files (without extension) and each data frame has the columns Concentration, Tested and Dead.

References

Abbott, W. S. (1925) A method of computing the effectiveness of an insecticide. *Journal of Economic Entomology* 18(2), 265-267.

See Also

[lc50_calculate](#) for the analysis workflow, [check_path_type](#) for the path handling.

Examples

```
f <- system.file("extdata", "bioassay.csv", package = "insectecol")
lcd <- read_lc50(f)
lcd$bioassay
if (interactive()) lcd <- read_lc50() # interactive folder dialog
```

read_life_table	<i>Read an Age-Stage, Two-Sex Life Table from a csv File</i>
-----------------	--

Description

Reads a raw csv file containing the individual daily records of an age-stage, two-sex life table experiment and returns a `life_table` object, the central data structure that all other functions of the package work with.

Usage

```
read_life_table(path, check = TRUE)
```

Arguments

<code>path</code>	Character string; path to the csv file.
<code>check</code>	Logical; if TRUE (the default) the data are validated by check_life_table immediately after reading. Set to FALSE to skip validation.

Details

The csv file contains one row per individual. If the sex column is located at column n , the layout must be:

- column 1: individual ID (header ID),
- columns 2 to $n - 2$: number of days spent in each developmental stage (egg, larva, ..., pupa),
- column $n - 1$: adult survival days,
- column n : sex, coded F (female), M (male) or N (died before the adult stage),
- columns $n + 1$ onward: daily oviposition of the females (one column per day; cells after the death of a female stay empty).

The first line of the file must contain the stage names as column headers, including the ID column (ID) and the sex column (gender). The sex column is located automatically by scanning the first data row for the markers F/M/N. The file encoding is detected automatically, so UTF-8 and GBK files are both supported.

Value

A `life_table` object; a named list with components

<code>data</code>	data frame with the raw csv content
<code>file_name</code>	file name without extension
<code>n</code>	column index of the sex column
<code>n_1</code>	$n + 1$, the first column of the oviposition data
<code>n_2</code>	$n - 2$, the column index of the pupal stage
<code>header</code>	the first row of the file (stage names), as character
<code>encoding</code>	the detected file encoding
<code>path</code>	the full path of the csv file

References

Chi, H. and Liu, H. (1985) Two new methods for study of insect population ecology. *Bull. Inst. Zool. Acad. Sin* 24(2), 225-240.

Chi, H. (1988) Life-table analysis incorporating both sexes and variable development rates among individuals. *Environmental Entomology* 17(1), 26-34.

See Also

[check_life_table](#) for data validation, [lifeTable_calculate](#) for the complete analysis workflow.

Examples

```
f <- system.file("extdata", "Example.csv", package = "insectecol")
lt <- read_life_table(f)
head(lt$data[, 1:5])
```

save_lc50	<i>Export LC Results to Excel</i>
-----------	-----------------------------------

Description

Writes the results of [lc50_calculate](#) into a multi-sheet Excel workbook: a summary sheet with the LC estimates, confidence intervals and regression parameters of all files and methods, plus one detail sheet per file with the preprocessed data and the parameters of the successfully computed methods.

Usage

```
save_lc50(results, output_dir = NULL, filename = "LC50_results.xlsx")
```

Arguments

results	The result list returned by lc50_calculate .
output_dir	Character string; the output folder. If NULL (the default), a folder selection dialog is opened.
filename	Character string; the name of the output file (default "LC50_results.xlsx").

Details

The detail sheet of each file contains the preprocessed data (concentration, tested and dead insects, raw and corrected mortality, log concentration and probit) followed by a parameter block of every method that succeeded for that file. Files for which no method succeeded get an empty sheet.

Value

The full path of the exported xlsx file (invisibly).

See Also

[lc50_calculate](#), [plot_lc50](#)

Examples

```
f <- system.file("extdata", "bioassay.csv", package = "insectecol")
save_lc50(lc50_calculate(read_lc50(f)), output_dir = tempdir())
```

save_lc50_auto	<i>Save the LC Results of Every Data File (One xlsx per csv)</i>
----------------	--

Description

Reads the csv file(s) at path, computes the LC values with [lc50_calculate](#) and writes one Excel workbook per csv file with [save_lc50](#) - named after the data file and placed next to the raw data.

Usage

```
save_lc50_auto(path = NULL, lc = 0.5, method = "traditional", suffix = NULL)
```

Arguments

path	Character string; a csv file or a folder with csv files, classified with check_path_type . NULL (the default) opens a folder selection dialog.
lc	Numeric; the lethal proportion, passed on to lc50_calculate (default 0.5 = LC50).
method	Character string; the estimation method, passed on to lc50_calculate (default "traditional").
suffix	Optional character string appended to the output file names, e.g. "_v2"; the default NULL adds nothing.

Details

Each csv file is processed completely on its own (read, calculate, export) in its own loop pass, so the outputs of different files can never mix. LB_48.csv produces LB_48.xlsx in the same folder; for a folder every csv file inside is processed the same way. A file that cannot be read is skipped with a message (a single-file input that cannot be read is an error); a file whose computation fails still gets its workbook with the error recorded in the summary sheet. Non-default settings are appended to the names so repeated runs do not overwrite each other: lc = 0.9 gives LB_48_LC90.xlsx, method = "probit" gives LB_48_probit.xlsx.

Value

The paths of the written xlsx files, invisibly.

See Also

[save_lc50_plot_auto](#) for the matching figure export, [save_lc50](#) for a custom output location

Examples

```
f <- system.file("extdata", "bioassay.csv", package = "insectecol")
tmp <- file.path(tempdir(), "bioassay.csv")
file.copy(f, tmp, overwrite = TRUE)
save_lc50_auto(tmp) # -> <tempdir>/bioassay.xlsx
```

save_lc50_plot	<i>Save LC50 Plots</i>
----------------	------------------------

Description

Saves one plot or a list of plots from [plot_lc50](#), like `ggsave(path, plot, device = "tiff", width = 12, height = 8, dpi = 300, units = "cm", bg = "white")` but with the dpi handling of `plot_lc50` applied. The same plot object can be written at any dpi without being re-created.

Usage

```
save_lc50_plot(
  plot,
  path = NULL,
  device = "tiff",
  width = 12,
  height = 8,
  dpi = 300,
  units = "cm",
  bg = "white",
  ...
)
```

Arguments

<code>plot</code>	A ggplot or a (named) list of ggplots.
<code>path</code>	Output file (single plot) or folder (list of plots, or a path without extension); NULL (default) opens a folder selection dialog.
<code>device, width, height, units, bg</code>	Passed on to <code>ggsave</code> (defaults "tiff", 12, 8, "cm", "white").
<code>dpi</code>	Resolution of the written file (default 300).
<code>...</code>	Further arguments passed on to <code>ggsave</code> .

Value

Path(s) of the written file(s), invisibly.

See Also

[plot_lc50](#), [save_lc50](#)

Examples

```
f <- system.file("extdata", "bioassay.csv", package = "insectecol")
plots <- plot_lc50(lc50_calculate(read_lc50(f)))
save_lc50_plot(plots$bioassay, file.path(tempdir(), "LC50_demo.tiff"))
```

save_lc50_plot_auto	<i>Save the LC Figure of Every Data File (One image per csv)</i>
---------------------	--

Description

Reads the csv file(s) at path, computes the LC values, draws the regression plot of [plot_lc50](#) for every file and saves it with [save_lc50_plot](#) - named after the data file and placed next to the raw data.

Usage

```
save_lc50_plot_auto(
  path = NULL,
  lc = 0.5,
  method = "traditional",
  suffix = NULL,
  device = "tiff",
  dpi = 600,
  width = 12,
  height = 8,
  units = "cm",
  bg = "white",
  font = "TNM",
  unit = NULL,
  shape = c("sigmoid", "linear"),
  ci = TRUE,
  ci_level = 0.95,
  error_bar = TRUE,
  move_thres = 0.5,
  lc_ci = TRUE,
  lc_p = TRUE,
  lc_lab_gap = 0.35,
  lc_lab_gap_right = 0.1,
  lc_lab_dy = 0.1,
  lc_lab_lh = 1.05,
  preview = FALSE
)
```

Arguments

path, lc, method, suffix
 Same as [save_lc50_auto](#).

device, width, height, dpi, units, bg
 Figure settings, passed on to [save_lc50_plot](#) (defaults "tiff", 12 x 8 cm, 600 dpi, white background).

font, unit, shape, ci, ci_level, error_bar, move_thres, lc_ci, lc_p,
 lc_lab_gap, lc_lab_gap_right, lc_lab_dy, lc_lab_lh
 Plot settings, passed on to [plot_lc50](#) unchanged.

preview
 Logical (default FALSE); also print every figure on the screen.

Details

The pipeline and the file-naming rules are the same as in [save_lc50_auto](#), plus `_linear` for `shape = "linear"`: `LB_48.csv` gives `LB_48.tiff`, `lc = 0.9`, `method = "probit"` gives `LB_48_LC90_probit.tiff`. The two functions are fully independent: each reads and computes the data on its own, so they can be called alone or in any order. A file whose computation fails gets no figure. With `method = "all"` the figure shows the first method that succeeded.

Value

Invisibly a list with elements `plots` (named list of the ggplot objects, e.g. for `print()` or `save_lc50_plot(plots, ...)`) and `files` (paths of the written images).

See Also

[save_lc50_auto](#), [plot_lc50](#), [save_lc50_plot](#)

Examples

```
f <- system.file("extdata", "bioassay.csv", package = "insectecol")
tmp <- file.path(tempdir(), "bioassay.csv")
file.copy(f, tmp, overwrite = TRUE)
save_lc50_plot_auto(tmp)           # -> <tempdir>/bioassay.tiff
```

 save_results

Save the Results of One Life Table Analysis

Description

Writes all results of one data set into a multi-sheet Excel workbook (`<file name>_out.xlsx`): the population parameters, the age-stage survival rates, the age-specific rates and, optionally, the survival curve plot.

Usage

```
save_results(
  lt,
  results,
  output_path = getwd(),
  plot = NULL,
  keep_tiff = FALSE,
  dpi = 300
)
```

Arguments

<code>lt</code>	A <code>life_table</code> object returned by read_life_table .
<code>results</code>	The result list returned by lifeTable_calculate_all .
<code>output_path</code>	Character; folder the workbook is written to. Defaults to the current working directory.
<code>plot</code>	A <code>ggplot</code> object (usually from plot_sxj); if <code>NULL</code> (default) no image is exported.
<code>keep_tiff</code>	Logical; whether to keep the standalone tiff file next to the workbook in addition to the copy embedded in it. Default <code>FALSE</code> , i.e. the tiff is deleted after being embedded.
<code>dpi</code>	Numeric; resolution of the exported image (default 300).

Details

The tiff is written through the internal `lt_ggsave()`, which enables `showtext` for the export device and pins `showtext`'s internal `dpi` to the value the text sizes of [plot_sxj](#) are calibrated for. The exported figure therefore looks the same in every R session, no matter what `showtext` settings are left over in the session. If the reproduction-related parameters were skipped (`fecundity = FALSE` in [lifeTable_calculate_all](#)), the corresponding values in the Summary sheet are NA and the sheets "Female fecundity (F_xj)" and "Age-specific fecundity (m_x)" are omitted.

Value

The path of the exported `xlsx` file (invisibly).

See Also

[lifeTable_calculate](#), [plot_sxj](#)

Examples

```
f <- system.file("extdata", "Example.csv", package = "insectecol")
lt <- read_life_table(f)
results <- lifeTable_calculate_all(lt)
save_results(lt, results, tempdir())
```

Index

build_life_table, [2](#), [11](#)

calc_sxj, [7](#), [18](#)

check_life_table, [3](#), [4](#), [20](#), [21](#)

check_path_type, [5](#), [19](#), [22](#)

default_stage_names, [3](#), [6](#)

get_stage_names, [6](#), [6](#)

ggsave, [18](#)

lc50_analyze, [7](#)

lc50_calculate, [7–9](#), [9](#), [15](#), [19](#), [21](#), [22](#)

lc50_improved, [10](#)

lc50_probit, [10](#)

lc50_traditional, [10](#)

lifeTable_analyze, [10](#)

lifeTable_calculate, [12](#), [14](#), [21](#), [26](#)

lifeTable_calculate_all, [3](#), [11](#), [13](#), [14](#), [26](#)

plot_lc50, [7–10](#), [15](#), [22–25](#)

plot_sxj, [7](#), [11](#), [13](#), [17](#), [26](#)

read_lc50, [6](#), [8–10](#), [18](#)

read_life_table, [2](#), [4](#), [5](#), [7](#), [11](#), [13](#), [14](#), [18](#),
[20](#), [26](#)

save_lc50, [7](#), [9](#), [10](#), [17](#), [21](#), [22–24](#)

save_lc50_auto, [22](#), [25](#)

save_lc50_plot, [7](#), [17](#), [23](#), [24](#), [25](#)

save_lc50_plot_auto, [23](#), [24](#)

save_results, [10](#), [11](#), [13](#), [18](#), [25](#)