Package: tipse (via r-universe)

May 12, 2026

Title Tipping Point Analysis for Survival Endpoints

Version 2.0

Description Implements tipping point sensitivity analysis for time-to-event endpoints under different missing data scenarios, as described in Oodally et al. (2025) <doi:10.48550/arXiv.2506.19988>. Supports both model-based and model-free imputation, multiple imputation workflows, plausibility assessment and visualizations. Enables robust assessment for regulatory and exploratory analyses.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 3.5)

LazyData true

Imports MASS, ggplot2, survival, dplyr, stats, utils, knitr, purrr, rmarkdown

VignetteBuilder knitr

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Ajmal Oodally [cre, aut] (ORCID: <https://orcid.org/0000-0001-8143-5011>), Craig Wang [aut] (ORCID: <https://orcid.org/0000-0003-1804-2463>), Zheng Li [ctb] (ORCID: <https://orcid.org/0000-0002-7655-0290>)

Maintainer Ajmal Oodally <ajmal.oodally@novartis.com>

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

Date/Publication 2026-05-12 15:17:06 UTC

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

RemoteRef HEAD

RemoteSha 03e674094463fd42fe20fd740f1c9852e420fa6d

Contents

`assess_plausibility` *Assess Clinical Plausibility of Imputation Results*

Description

This function facilitates the evaluation of clinical plausibility at the tipping point. It provides a text summary comparing event rates, follow-up duration, or hazard ratios between treatment arms depending on the imputation method and arm specified. NOTE: this function only supports imputation in one arm.

Usage

```
assess_plausibility(tipse, verbose = TRUE)
```

Arguments

`tipse`	A `tipse` object returned by one of tipping_point_model_free or tipping_point_model_based.
`verbose`	Logical. If `TRUE`, prints assessment details.

Value

A character string summarizing the key information to facilitate clinical plausibility assessment based on the imputation scenario.

Examples

```
cox1 <- survival::coxph(Surv(AVAL, EVENT) ~ TRT01P, data = codebreak200)
result <- tipping_point_model_free(
  dat = codebreak200,
  reason = "Early dropout",
  impute = "docetaxel",
  cox_fit = cox1,
  method = "percentile sampling"
```

```
)

assess_plausibility(result)
```

`codebreak200` *Patient level data from dummy trial*

Description

Based on re-constructed Kaplan-Meier plot from CodeBreak 200 trial (de Langen et al., 2023)

Usage

```
codebreak200
```

Format

A data frame with 345 rows and 5 columns:

SUBJID Dummy patient ID

TRT01P Treatment arm (Sotorasib or Docetaxel)

AVAL PFS time in days

EVENT Indicator for PFS event

CNSRRS Censoring reason (Early dropout or Other)

MAXAVAL Maximum potential survival time, duration between randomization to data cut-off

Source

De Langen, A.J., Johnson, M.L., Mazieres, J., Dingemans, A.M.C., Mountzios, G., Pless, M., Wolf, J., Schuler, M., Lena, H., Skoulidis, F. and Yoneshima, Y., 2023. Sotorasib versus docetaxel for previously treated non-small-cell lung cancer with KRASG12C mutation: a randomised, open-label, phase 3 trial. The Lancet, 401(10378), pp.733-746.

extenet *Patient level data from dummy trial*

Description

Based on re-constructed Kaplan-Meier plot from ExteNET trial (Martin et al., 2017)

Usage

```
extenet
```

Format

A data frame with 2840 rows and 5 columns:

SUBJID Dummy patient ID

TRT01P Treatment arm (Neratinib or placebo)

AVAL iDFS time in days

EVENT Indicator for iDFS event

CNSRRS Censoring reason (Lost to follow-up or Other)

MAXAVAL Maximum potential survival time, duration between randomization to data cut-off

Source

Martin, M., Holmes, F.A., Ejlertsen, B., Delaloge, S., Moy, B., Iwata, H., von Minckwitz, G., Chia, S.K., Mansi, J., Barrios, C.H. and Gnant, M., 2017. Neratinib after trastuzumab-based adjuvant therapy in HER2-positive breast cancer (ExteNET): 5-year analysis of a randomised, double-blind, placebo-controlled, phase 3 trial. The lancet oncology, 18(12), pp.1688-1700.

impute_landmark *Model-free imputation via landmark sampling*

Description

patients will be assigned deterministically an event time at the time of censoring or extend the censoring time to the potential maximum follow-up of each patient.

Usage

```
impute_landmark(dat, reason, impute, npts, J, seed)
```

Arguments

dat	data.frame containing at least 5 columns: TRT01P (treatment arm as factor), AVAL (survival time), EVENT (event indicator), CNSRRS (censoring reason) and MAXAVAL (maximum potential survival time, duration between randomization to data cut-off)
reason	a string specifying the censoring reasons which require imputation. It must be one of the reasons from variable CNSRRS.
impute	a string specifying the treatment arm(s) which require imputation. It must be one of the arms from variable TRT01P, the first level of TRT01P is considered as the control arm.
npts	number of patients to be imputed
J	numeric indicating number of imputations.
seed	Integer. Random seed for reproducibility.

Details

patients will be assigned deterministically an event time at the time of censoring or extend the censoring time to the potential maximum follow-up of each patient.

Value

a list of data.frame from each imputation with imputed AVAL and EVENT, where original variables are kept as AVAL and EVENT.

Examples

```
impute_landmark(
  dat    = codebreak200,
  reason = "Early dropout",
  impute = "docetaxel",
  npts   = 5,
  J      = 5,
  seed   = 1
)
```

---

impute_model *Model-based imputation from parametric distributions*

Description

Impute data with Weibull or exponential distribution conditional on follow-up time

Usage

```
impute_model(
  dat,
  reason,
  impute,
  imputation_model = c("weibull", "exponential"),
  alpha,
  J,
  seed = NULL
)
```

Arguments

dat	data.frame containing at least 5 columns: TRT01P (treatment arm as factor), AVAL (survival time), EVENT (event indicator), CNSRRS (censoring reason) and MAXAVAL (maximum potential survival time, duration between randomization to data cut-off)
reason	a string specifying the censoring reasons which require imputation. It must be one of the reasons from variable CNSRRS.
impute	a string specifying the treatment arm(s) which require imputation. It must be one of the arms from variable TRT01P, the first level of TRT01P is considered as the control arm.
imputation_model	a string specifying the parametric distribution used for imputation, can be "Weibull" or "exponential".
alpha	hazard inflation (if treatment arm is imputed) or deflation (if control arm is imputed) rate
J	numeric indicating number of imputations.
seed	Integer. Random seed for reproducibility.

Details

First fit model based on the data without dropout. And then impute the survival outcome based on exponential or Weibull distribution for those who dropped out.

Value

a list of data.frame from each imputation with imputed AVAL and EVENT, where original variables are kept as AVALo and EVENTo.

Examples

```
impute_model(
  dat              = codebreak200,
  reason           = "Early dropout",
  impute           = "docetaxel",
  imputation_model  = "weibull",
  alpha            = 0.7,
```

```
  J                = 5,
  seed             = 1
)
```

`impute_percentile` *Model-free imputation via percentile sampling*

Description

randomly sample from the percentile of best or worst patients (ordered by their observed times regardless of event or censoring) who do not require imputation.

Usage

```
impute_percentile(dat, reason, impute, percentile, J, seed = NULL)
```

Arguments

`dat`	data.frame containing at least 5 columns: TRT01P (treatment arm as factor), AVAL (survival time), EVENT (event indicator), CNSRRS (censoring reason) and MAXAVAL (maximum potential survival time, duration between randomization to data cut-off)
`reason`	a string specifying the censoring reasons which require imputation. It must be one of the reasons from variable CNSRRS.
`impute`	a string specifying the treatment arm(s) which require imputation. It must be one of the arms from variable TRT01P, the first level of TRT01P is considered as the control arm.
`percentile`	numeric between 1 and 100, indicating the best (or worst) percentile of subjects to sample from.
`J`	numeric indicating number of imputations.
`seed`	Integer. Random seed for reproducibility.

Details

We define two sets of subjects to sample from depending on the `impute` argument:

1. **Worst percentile of observations from treatment arm** $\forall i \in N \mid \min\{T_i, C_i\} \leq F^{-1}_{\min\{T_i,C_i\}}(\kappa)$. This set includes all indices i where the minimum of T_i (event time) and C_i (censoring time) is **less than or equal to** the κ-th percentile of its distribution.
2. **Best percentile of observations control arm** $\forall i \in N \mid \min\{T_i, C_i\} \geq F^{-1}_{\min\{T_i,C_i\}}(\kappa)$. This set includes all indices i where the minimum of T_i and C_i is **greater than or equal to** the κ-th percentile of its distribution.

where $F(\cdot)$ denotes the cumulative distribution function (CDF) of the observed times and $F^{-1}(\kappa)$ is the inverse CDF (quantile function) at percentile κ.

Value

a list of data.frame from each imputation with imputed AVAL and EVENT, where original variables are kept as AVALo and EVENTo.

Examples

```
impute_percentile(
  dat        = codebreak200,
  reason     = "Early dropout",
  impute     = "docetaxel",
  percentile = 30,
  J          = 5,
  seed       = 1
)
```

`plot.tipse` *Plot Pooled Kaplan–Meier Curves from Tipping Point Analysis*

Description

Visualizes averaged (pooled) Kaplan-Meier survival curves across multiple tipping point parameters, highlighting the tipping point where the upper CL of the hazard ratio crosses 1.

Usage

```
## S3 method for class 'tipse'
plot(x, type = c("Kaplan-Meier", "Tipping Point"), ...)
```

Arguments

`x`	An S3 object of class `"tipse"` returned from tipping_point_model_free or tipping_point_model_based.
`type`	Type of plot, either "Kaplan-Meier" or "Tipping Point".
`...`	Additional arguments to specify title, subtitle, xlab and ylab.

Details

- If `type = Kaplan-Meier`, then the KM curves from multiply imputed datasets were pooled using Rubin's rules after complementary log-log transformation as described in Marshall et al. (2009). it can be of interest to visually assess the scenario that tips the result and the shift it causes to the original KM curve, although there is no objective measure to assess the robustness of the result.
- If `type = Tipping Point`, then the HR estimation across the range of tipping point parameters are plotted.

Value

A ggplot2 object displaying pooled Kaplan–Meier curves.

References

Marshall, A., Altman, D.G., Holder, R.L. et al. Combining estimates of interest in prognostic modelling studies after multiple imputation: current practice and guidelines. BMC Med Res Methodol 9, 57 (2009). https://doi.org/10.1186/1471-2288-9-57

Examples

```
cox1 <- survival::coxph(Surv(AVAL, EVENT) ~ TRT01P, data = codebreak200)
result <- tipping_point_model_based(
  dat = codebreak200,
  reason = "Early dropout",
  impute = "docetaxel",
  imputation_model = "weibull",
  J = 10,
  tipping_range = seq(0.1, 1, by = 0.05),
  cox_fit = cox1
)

plot(result, type = "Kaplan-Meier")
plot(result, type = "Tipping Point")


# Imputation in both arms
result2 <- tipping_point_model_based(
  dat = codebreak200,
  reason = "Early dropout",
  impute = c("docetaxel", "sotorasib"),
  imputation_model = "weibull",
  J = 10,
  tipping_range = list(seq(0.4, 0.7, by = 0.1), seq(0.5, 1.5, by = 0.2)),
  cox_fit = cox1
)

plot(result2, type = "Tipping Point")
```

`pool_results` *Pooling results using Rubin's Rule*

Description

Pooling results from multiple imputations using Rubin's Rule

Usage

```
pool_results(dat, cox.fit, conf.level = 0.95)
```

Arguments

`dat`	a list of data.frames from multiple imputation using one alpha or kappa parameter
`cox.fit`	a coxph object which is used to compute HRs for each imputed datasets
`conf.level`	confidence level for the returned confidence interval, default to be 0.95.

Details

The Rubin's rule is applied to the Cox PH model results across imputed datasets as:

1. *Compute pooled HR:*

$$\bar{HR}_\lambda = \exp\left(\frac{1}{M}\sum_{m=1}^{M}\log(HR_m)\right)$$

2. *Compute pooled variance:*

$$\bar{\sigma}^2_\lambda = \frac{1}{M}\sum_{m=1}^{M}\sigma^2_m + \frac{1+\frac{1}{M}}{M-1}\sum_{m=1}^{M}\left(\log(HR_m) - \overline{\log(HR_\lambda)}\right)^2$$

3. *Compute CI:*

$$\bar{HR}_\lambda \times \exp\left(\pm t_{\alpha/2}\sqrt{\bar{\sigma}^2_\lambda}\right)$$

Value

a data.frame of pooled hazard ratio and confidence interval estimate using Rubin's Rule

Examples

```
cox_fit <- survival::coxph(Surv(AVAL, EVENT) ~ TRT01P, data = codebreak200)
imputed_list <- impute_percentile(
  dat        = codebreak200,
  reason     = "Early dropout",
  impute     = "docetaxel",
  percentile = 30,
  J          = 5,
  seed       = 1
)
pool_results(imputed_list, cox_fit)
```

summary.tipse *Summarize Tipping Point Results (ARD Format)*

Description

Creates a concise, analysis-results dataset (ARD) from a tipping point analysis. Identifies the tipping point parameter where the upper CL of the hazard ratio crosses 1 and summarizes key metrics.

Usage

```
## S3 method for class 'tipse'
summary(object, ...)
```

Arguments

object	A tipse object returned by tipping_point_model_free or tipping_point_model_based.
...	Additional arguments not used.

Value

A data frame summarizing:

- HR - hazard ratio at that tipping point
- CONFINT - 95% CI at tipping point
- METHOD - sampling type used
- ARMIMP - arm imputed
- TIPPT - parameter where upper CL first crosses 1
- TIPUNIT - parameter meaning
- DESC - textual interpretation

Examples

```
# Hazard deflation in the control arm
cox1 <- survival::coxph(Surv(AVAL, EVENT) ~ TRT01P, data = codebreak200)
result1 <- tipping_point_model_based(
  dat = codebreak200,
  reason = "Early dropout",
  impute = "docetaxel",
  imputation_model = "weibull",
  J = 10,
  tipping_range = seq(0.1, 1, by = 0.05),
  cox_fit = cox1
)

summary(result1)

# Imputation in both arms
```

```
result2 <- tipping_point_model_based(
  dat = codebreak200,
  reason = "Early dropout",
  impute = c("docetaxel", "sotorasib"),
  imputation_model = "weibull",
  J = 10,
  tipping_range = list(seq(0.1, 1, by = 0.2), seq(0.5, 1.5, by = 0.2)),
  cox_fit = cox1,
  verbose = TRUE,
  seed = 12345
)

summary(result2)
```

`tipping_point_model_based` *Tipping Point Analysis (Model-Based)*

Description

Performs a model-based tipping point analysis on time-to-event data by repeatedly imputing censored observations under varying assumptions. The model-based framework assumes that censored patients have a multiple of hazard fitted via a parametric survival model compared to the rest of patients in the same arm (Akinson et al, 2019).

Usage

```
tipping_point_model_based(
  dat,
  reason,
  impute,
  imputation_model = "weibull",
  J = 10,
  tipping_range = seq(0.05, 1, by = 0.05),
  cox_fit = NULL,
  verbose = FALSE,
  seed = NULL
)
```

Arguments

`dat`	data.frame containing at least 5 columns: TRT01P (treatment arm as factor), AVAL (survival time), EVENT (event indicator), CNSRRS (censoring reason) and MAXAVAL (maximum potential survival time, duration between randomization to data cut-off)
`reason`	Vector specifying censoring reasons to be imputed.

`impute`	a character vector specifying the arm(s) to impute. Can be one arm or both arms (a length-2 vector). Each value must be one of the arms from variable TRT01P. When both arms are supplied, imputation is applied independently.
`imputation_model`	used to fit model to observed data (should be "Weibull" or "exponential")
`J`	numeric indicating number of imputations.
`tipping_range`	Numeric vector, or when `length(impute) == 2` optionally a named or unnamed list of two numeric vectors (one per arm). When a list is supplied, all combinations of the two vectors are evaluated.
`cox_fit`	A Cox model that will be used to calculate HRs on imputed datasets. In case of inclusion of stratification factors or covariates, conditional HR will be used.
`verbose`	Logical. If `TRUE`, prints progress and analysis details.
`seed`	Integer, default as NULL. Random seed for reproducibility.

Details

The **model-based tipping point analysis** provides a reproducible and intuitive framework for exploring the robustness of treatment effects in time-to-event (survival) endpoints when censoring may differ between study arms.

A parametric survival model is fitted using maximum likelihood. This function applies a hazard deflation on control arm or hazard inflation on treatment arm, and impute survival times based on the parametric model with additional sampling of the parameters from a multivariate normal distribution. This imputation procedure is iterated across a range of tipping point parameters `tipping_range`. For each parameter value:

1. Multiple imputed datasets are generated (`J` replicates), where censored observations in the selected arm are reassigned event times according to the imputation method.
2. A Cox proportional hazards model is fitted to each imputed dataset.
3. Model estimates are pooled using **Rubin's rules** to obtain a combined hazard ratio and confidence interval for that tipping point parameter.

The process yields a series of results showing how the treatment effect changes as increasingly conservative or optimistic assumptions are made about censored observations. The *tipping point* is defined as the smallest value (hazard inflation) or biggest value (hazard deflation) of the sensitivity parameter for which the upper bound of the hazard ratio confidence interval crosses 1 - i.e., where the apparent treatment benefit is lost.

Value

A `tipse` object containing:

original data Input argument from 'data'.

imputation_results A data frame of combined pooled model results across tipping points

original_HR The original hazard ratio.

reason_to_impute Input argument from 'reason'.

arm_to_impute Input argument from 'impute'.

method_to_impute Input argument from 'method'.

imputation_data A list of imputed datasets for each tipping point value.

seed Random seed.

References

Atkinson, A., Kenward, M. G., Clayton, T., & Carpenter, J. R. (2019). Reference-based sensitivity analysis for time-to-event data. Pharmaceutical statistics, 18(6), 645-658.

Examples

```
cox1 <- survival::coxph(Surv(AVAL, EVENT) ~ TRT01P, data = codebreak200)
result <- tipping_point_model_based(
  dat = codebreak200,
  reason = "Early dropout",
  impute = "docetaxel",
  imputation_model = "weibull",
  J = 10,
  tipping_range = seq(0.1, 1, by = 0.05),
  cox_fit = cox1
)
```

---

tipping_point_model_free

Tipping Point Analysis (Model-Free)

Description

Performs a model-free tipping point analysis on time-to-event data by repeatedly imputing censored observations under varying assumptions. The model-free framework assumes that censored patients share similar survival behavior with those from whom they are sampled, without fitting any parametric survival model.

Usage

```
tipping_point_model_free(
  dat,
  reason,
  impute,
  J = 10,
  tipping_range = seq(5, 95, by = 5),
  cox_fit = NULL,
  verbose = FALSE,
  method = c("percentile sampling", "landmark sampling"),
  seed = NULL
)
```

Arguments

dat	data.frame containing at least 5 columns: TRT01P (treatment arm as factor), AVAL (survival time), EVENT (event indicator), CNSRRS (censoring reason) and MAXAVAL (maximum potential survival time, duration between randomization to data cut-off)
reason	Vector specifying censoring reasons to be imputed.
impute	a character vector specifying the arm(s) to impute. Can be one arm or both arms (a length-2 vector). Each value must be one of the arms from variable TRT01P. When both arms are supplied, imputation is applied independently for each arm.
J	numeric indicating number of imputations.
tipping_range	Numeric vector or when `length(impute) == 2` optionally a named or unnamed list of two numeric vectors (one per arm). When a list is supplied, all combinations of the two vectors are evaluated. Percentiles to use when `method = "percentile sampling"`. Number of patients to impute when `method = "landmark sampling"`.
cox_fit	A Cox model that will be used to calculate HRs on imputed datasets. In case of inclusion of stratification factors or covariates, conditional HR will be used.
verbose	Logical. If `TRUE`, prints progress and analysis details.
method	Character. Either `"percentile sampling"` or `"landmark sampling"`.
seed	Integer, default as NULL. Random seed for reproducibility.

Details

The **model-free tipping point analysis** provides a reproducible and intuitive framework for exploring the robustness of treatment effects in time-to-event (survival) endpoints when censoring may differ between study arms.

Two sampling modes are supported:

- `method = "percentile sampling"` - performs re-sampling of event times from the best or worst percentile of observed patients ranked by their event or censoring time. The `tipping_range` specifies the percentiles of the observed data from which event times will be sampled to impute censored patients. For the treatment arm, use the worst percentiles (shortest survival times) from the observed data of both arms. For the control arm, use the best percentiles (longest survival times).
- `method = "landmark sampling"` - imputes a fixed number of censored patients deterministically. The `tipping_range` specifies the number of patients to be imputed. For the treatment arm, it defines the number of patients that will be assumed to have an event at their time of censoring. For the control arm, it defines the number of patients that will be assumed to be event-free at data cut-off, their maximum potential follow-up time.

This function iteratively applies the percentile- or landmark-sampling imputation procedure across a range of tipping point parameters `tipping_range`. For each parameter value:

1. Multiple imputed datasets are generated (`J` replicates), where censored observations in the selected arm are replaced by sampled or reassigned event times according to the imputation method.

2. A Cox proportional hazards model is fitted to each imputed dataset.
3. Model estimates are pooled using **Rubin's rules** to obtain a combined hazard ratio and confidence interval for that tipping point parameter.

The process yields a series of results showing how the treatment effect changes as increasingly conservative or optimistic assumptions are made about censored observations. The *tipping point* is defined as the smallest value of the sensitivity parameter (percentile or number of imputed patients) for which the upper bound of the hazard ratio confidence interval crosses 1 - i.e., where the apparent treatment benefit is lost.

Value

A `tipse` object containing:

original data Input argument from 'data'.

imputation_results A data frame of combined pooled model results across tipping points

original_HR The original hazard ratio.

reason_to_impute Input argument from 'reason'.

arm_to_impute Input argument from 'impute'.

method_to_impute Input argument from 'method'.

imputation_data A list of imputed datasets for each tipping point value.

seed Random seed.

Examples

```
cox1 <- survival::coxph(Surv(AVAL, EVENT) ~ TRT01P, data = codebreak200)
result <- tipping_point_model_free(
  dat = codebreak200,
  reason = "Early dropout",
  impute = "docetaxel",
  J = 10,
  tipping_range = seq(5, 95, by = 5),
  cox_fit = cox1,
  method = "percentile sampling"
)

result2 <- tipping_point_model_free(
  dat = codebreak200,
  reason = "Early dropout",
  impute = "docetaxel",
  J = 10,
  tipping_range = seq(1, 21, by = 2),
  cox_fit = cox1,
  method = "landmark sampling"
)
```

Index